Hindawi
Mathematical Problems in Engineering
Volume 2019, Article ID 6478506, 15 pages
https://doi.org/10.1155/2019/6478506

Research Article

Adaptive PD Control Based on RBF Neural Network for a Wire-Driven Parallel Robot and Prototype Experiments

Yuqi Wang[1,2] **Qi Lin**[1] **Xiaoguang Wang**[1] **and Fangui Zhou**[1]

[1]*Xiamen University, 361005 Xiamen, Fujian, China*
[2]*Nanjing Institute of Technology, 211167 Nanjing, Jiangsu, China*

Correspondence should be addressed to Qi Lin; qilin@xmu.edu.cn

Received 7 August 2018; Accepted 25 December 2018; Published 12 February 2019

Academic Editor: Akhil Garg

An adaptive PD control scheme is proposed for the support system of a wire-driven parallel robot (WDPR) used in a wind tunnel test. The control scheme combines a PD control and an adaptive control based on a radial basis function (RBF) neural network. The PD control is used to track the trajectory of the end effector of the WDPR. The experimental environment, the external disturbances, and other factors result in uncertainties of some parameters for the WDPR; therefore, the RBF neural network control method is used to approximate the parameters. An adaptive control algorithm is developed to reduce the approximation error and improve the robustness and control precision of the WDPR. It is demonstrated that the closed-loop system is stable based on the Lyapunov stability theory. The simulation results show that the proposed control scheme results in a good performance of the WDPR. The experimental results of the prototype experiments show that the WDPR operates on the desired trajectory; the proposed control method is correct and effective, and the experimental error is small and meets the requirements.

1. Introduction

Neural networks are widely used in engineering, especially in the field of industrial automation [1]. The application is also very popular for robot control. A neural network is based on the biological neural networks of the human brain from the perspective of information processing and represents a simple model that forms different networks according to different connection modes. Why have neural networks attracted broad attention in the field of robot control? The reason is that this approach meets the basic requirements of the control theory and control systems engineering and helps to solve relevant problems [2]. Neural networks fully approximate complex nonlinear mapping relationship, possess the characteristics of learning and adapting to the dynamic characteristics of uncertain systems, and exhibit strong robustness and fault tolerance [3]. As a result, a large number of researchers have applied neural networks for the control of parallel robots in nonlinear systems.

Dinh et al. [4] proposed a dynamic neural network (DNN) based on a robust observer for uncertain nonlinear systems. The observer structure consists of a DNN, a dynamic filter, and a sliding-mode feedback term. Simulations and experiments using a two-link robot manipulator were performed to demonstrate the effectiveness of the proposed method in comparison with several other state estimation methods. Wang et al. [5] used a recursive neural network to optimize the structure of complex robots. Melingui et al. [6] proposed two subcontrollers for a class of bionic continuum robots termed the "compact bionic handling arm" (CBHA). One controller is encapsulated in the other and both are implemented in real time to control the CBHA's end-effector position. The first subcontroller controls the CBHA's kinematics based on a distal supervised learning scheme. The second subcontroller controls the CBHA's kinetics based on an adaptive neural control. The experimental results of using a CBHA robot demonstrated the accurate tracking of the CBHA's end-effector position. Sabahi et al. [7] proposed a new indirect type-2 fuzzy neural network predictive (T2FNNP) controller for a class of nonlinear input-delay systems in the presence of unknown disturbances and uncertainties. The proposed T2FNNP controller was applied to a nonlinear inverted pendulum and single-link

robot manipulator systems with input time-varying delay; the performance was compared with that of a type-1 fuzzy sliding predictive (T1FSP) controller. The simulation results showed the efficiency of the proposed T2FNNP controller. Wen et al. [8] proposed an improved radial basis functional (RBF) neural network by means of a hybrid force/position control for robotic manipulation; good stability and transient performance of the system were achieved. Zhao et al. [9, 10] proposed an adaptive output-feedback control for a class of nonsmooth nonlinear systems and an adaptive fuzzy hierarchical sliding-mode control method for a class of multi-input multi-output unknown nonlinear time-delay systems with input saturation. Wang et al. [11, 12] presented an adaptive neural control approach for nonstrict-feedback nonlinear systems in the presence of unmodeled dynamics, unknown control directions, and input dead-zone nonlinearity and proposed a robust adaptive tracking control for nonaffine stochastic nonlinear switching systems.

At present, most of the literature on the use of neural networks for robot control is focused on rigid robots (arms are rigid, such as link robots) and papers on multi-DOF wire-driven parallel robots (WDPRs) used in wind tunnel tests are rarely seen. In this study, we propose a new adaptive PD control scheme based on an RBF neural network for a WDPR with 6 DOF and driven by 8 wires; the nature and task requirements are based on the WDPR described in reference [13]. Compared with other types of neural networks, RBF neural networks avoid the issue of local minimization and slow training and satisfy the requirements of real-time control. In the RBF neural network compensation control strategy, it is important to ensure the stability of the closed-loop control system. At the same time, in the designed control system, the RBF neural network can approximate the unknown function [14]. One advantage of the proposed control method is that the RBF neural network and adaptive PD control are integrated into the WDPR, and other advantages of the proposed control method are verified by the prototype experiment.

In this study, the main purpose is to design a new tracking control scheme for a 6-DOF 8-WDPR used in a wind tunnel test. The main contributions of this study are the following: (1) the PD control is used to track the trajectory of the end effector; (2) the RBF neural network approximation is used to compensate for the uncertainties of the system, such as external disturbances; (3) the adaptive control is used to achieve a high trajectory tracking performance of the WDPR; (4) the simulation results and the prototype experimental results show that the RBF neural network results in good control performance of the WDPR and meets the requirements of the wind tunnel test.

The rest of the paper is organized as follows. Section 2 presents the problem. Section 3 describes the RBF network design of the WDPR. The control design and the stability analysis of the WDPR are described in Section 4. The simulation results are presented in Section 5. The experimental results of the prototype are provided in Section 6. The conclusions are given in Section 7.

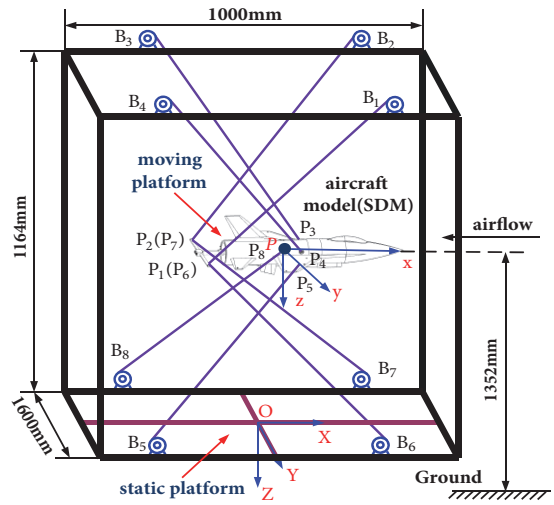


Figure 1: Structural diagram of the WDPR.

2. Problem Description

2.1. Description of WDPR. A WDPR is a specially designed support system and a parallel device for wind tunnel tests. The robot converts the motion and force of the driver to the motion and force of the end-effector in parallel using the wire as the medium. The system has the advantages of a simple structure, high rigidity, large workspace, low inertia, fast movement, easy assembly and disassembly, and reconfiguration [15, 16]. It is a complex multiple-input and multiple-output nonlinear system. The structural diagram of the WDPR is shown in Figure 1. Figure 2 shows the WDPR in the wind tunnel. In Figure 2, the end effector is a standard dynamics model (SDM). The end effector (aircraft model) is located on the moving platform and it is driven by eight wires with six DOF.

In this study, the WDPR is composed of a control system, a driving system, an executive system, a measurement system, and other components. The driving system represents the power source to guarantee the movement of the end effector and is used as the input for the motion control. The dynamic analysis and modeling of the end effector are the basis of the motion control and the key to ensuring the system performance. The two dynamic equations are shown as follows:

$$\begin{aligned} \mathbf{M}_0\ddot{\boldsymbol{\theta}}_m + \mathbf{D}_0\dot{\boldsymbol{\theta}}_m + \boldsymbol{\tau}_l &= \boldsymbol{\tau} \\ \mathbf{M}(\mathbf{X})\ddot{\mathbf{X}} + \mathbf{N}\left(\mathbf{X}, \dot{\mathbf{X}}\right) - \mathbf{w}_g - \mathbf{w}_e &= -\mathbf{J}^{\mathrm{T}}\mathbf{T} \end{aligned} \tag{1}$$

where $\mathbf{M}_0$ is equivalent to the inertia matrix of the driver, $\mathbf{D}_0$ is equivalent to the viscous friction coefficient matrix of the driver, $\boldsymbol{\theta}_m$ is the angle of the motor, $\boldsymbol{\tau}_l$ is the load torque produced by wire tension, $\boldsymbol{\tau}_l = \mu\mathbf{T}$, μ is the transmission coefficient of the ball screw, $\mathbf{T}$ is the wire tension vector, $\boldsymbol{\tau}$ is the output torque vector of the driver, $\mathbf{M}(\mathbf{X})$ is the inertial matrix of the end effector, $\mathbf{X} = (X_P, Y_P, Z_P, \phi, \theta, \psi)^{\mathrm{T}}$ is the pose of the end effector, $\dot{\mathbf{X}}$ is the pose velocity of the end effector, the relationship between the angle of the motor $\boldsymbol{\theta}_m$

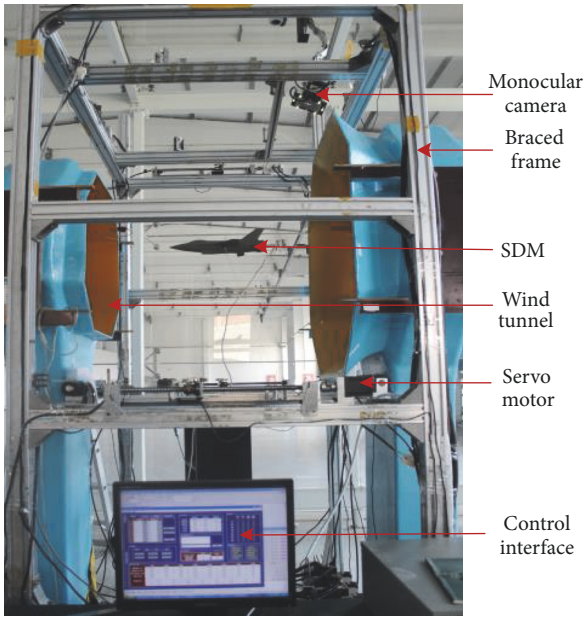


Figure 2: WDPR in wind tunnel.

and the pose velocity $\dot{\mathbf{X}}$ is $\dot{\boldsymbol{\theta}}_m = (1/\mu)\mathbf{JG}\dot{\mathbf{X}}$, $\mathbf{G}$ is a transition matrix based on the attitude angle, $\mathbf{J}$ is the Jacobi matrix, $\mathbf{N}(\mathbf{X}, \dot{\mathbf{X}})$ is a nonlinear Coriolis centrifugal force matrix, $\mathbf{w}_g$ is the gravity vector of the end effector, $\mathbf{w}_g = (0, 0, mg, 0, 0, 0)^{\mathrm{T}}$, and $\mathbf{w}_e$ is subject to the external dynamic loads at the end effector.

By combining the dynamic equation of the driving system with the dynamic equation of the model, the dynamic equation of the entire system is as follows [13]:

$$\begin{aligned}&\left(\mathbf{M}(\mathbf{X}) - \left(\frac{1}{\mu^2}\right)\cdot \mathbf{J}^{\mathrm{T}}\mathbf{M}_0\mathbf{JG}\right)\ddot{\mathbf{X}}\\ &\quad - \left(\frac{1}{\mu^2}\right)\left(\mathbf{J}^{\mathrm{T}}\mathbf{M}_0\dot{\mathbf{J}}\mathbf{G} + \mathbf{J}^{\mathrm{T}}\mathbf{M}_0\mathbf{J}\dot{\mathbf{G}} + \mathbf{J}^{\mathrm{T}}\mathbf{D}_0\mathbf{JG}\right)\dot{\mathbf{X}}\\ &= -\left(\frac{1}{\mu}\right)\cdot \mathbf{J}^{\mathrm{T}}\boldsymbol{\tau} + \mathbf{w}_g + \mathbf{w}_e - \mathbf{N}\left(\mathbf{X}, \dot{\mathbf{X}}\right)\end{aligned} \tag{2}$$

where $\mathbf{M}(\mathbf{X}) = \begin{bmatrix} (m\mathbf{I})_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{A}_{G3\times3}\mathbf{H} \end{bmatrix}$, $\mathbf{G} = \begin{bmatrix} \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{H} \end{bmatrix}$. The meanings and derivations of the parameters $\mathbf{A}_{G3\times3}$ and $\mathbf{H}$ are detailed in reference [13], because the derivation of (2) and its parameters are relatively complex.

Equation (2) is multiplied by $-\mu$ on both sides and (3) is obtained.

$$\begin{aligned}&-\mu\left(\mathbf{M}(\mathbf{X}) - \left(\frac{1}{\mu^2}\right)\cdot \mathbf{J}^{\mathrm{T}}\mathbf{M}_0\mathbf{JG}\right)\ddot{\mathbf{X}}\\ &\quad + \left(\frac{1}{\mu}\right)\left(\mathbf{J}^{\mathrm{T}}\mathbf{M}_0\dot{\mathbf{J}}\mathbf{G} + \mathbf{J}^{\mathrm{T}}\mathbf{M}_0\mathbf{J}\dot{\mathbf{G}} + \mathbf{J}^{\mathrm{T}}\mathbf{D}_0\mathbf{JG}\right)\dot{\mathbf{X}}\\ &= \mathbf{J}^{\mathrm{T}}\boldsymbol{\tau} - \mu\mathbf{w}_g - \mu\mathbf{w}_e + \mu\mathbf{N}\left(\mathbf{X}, \dot{\mathbf{X}}\right)\end{aligned} \tag{3}$$

Let $\mathbf{A} = -\mu(\mathbf{M}(\mathbf{X}) - (1/\mu^2)\cdot \mathbf{J}^{\mathrm{T}}\mathbf{M}_0\mathbf{JG})$, $\mathbf{B} = (1/\mu)(\mathbf{J}^{\mathrm{T}}\mathbf{M}_0\dot{\mathbf{J}}\mathbf{G} + \mathbf{J}^{\mathrm{T}}\mathbf{M}_0\mathbf{J}\dot{\mathbf{G}} + \mathbf{J}^{\mathrm{T}}\mathbf{D}_0\mathbf{JG})$, $\mathbf{C} = -\mu\mathbf{N}(\mathbf{X}, \dot{\mathbf{X}})$, $\mathbf{D} = -\mu\mathbf{w}_g - \mu\mathbf{w}_e$.

Equation (3) is simplified as

$$\mathbf{A}\ddot{\mathbf{X}} + \mathbf{B}\dot{\mathbf{X}} + \mathbf{C} = \mathbf{J}^{\mathrm{T}}\boldsymbol{\tau} + \mathbf{D} \tag{4}$$

2.2. Problem Formulation of the WDPR. In practical engineering applications, there are many uncertainties, such as the experimental environment, the external disturbances, and other factors. These factors cannot be taken into account when the dynamic model of the system is developed. If those factors are taken into account, the dynamic equation will become too complex to be solved. Thus, the dynamic equations for an established system are generally incomplete.

Because it is impossible to obtain a complete dynamic equation of a complex nonlinear system in practical engineering applications, the nominal model of the WDPR is defined as follows:

$$\mathbf{A}_0\ddot{\mathbf{X}} + \mathbf{B}_0\dot{\mathbf{X}} + \mathbf{C}_0 = \mathbf{J}^{\mathrm{T}}\boldsymbol{\tau} + \mathbf{D} \tag{5}$$

where $\mathbf{A}_0$, $\mathbf{B}_0$, and $\mathbf{C}_0$ are known parameters, $\mathbf{A}_0$ and $\mathbf{B}_0$ represent matrices with 6 rows and 6 columns, and $\mathbf{C}_0$ represents a matrix of 6 rows and 1 column.

Let $\Delta\mathbf{A} = \mathbf{A}_0 - \mathbf{A}$, $\Delta\mathbf{B} = \mathbf{B}_0 - \mathbf{B}$, and $\Delta\mathbf{C} = \mathbf{C}_0 - \mathbf{C}$; then, combine (4) with (5) to be simplified as

$$\mathbf{J}^{\mathrm{T}}\boldsymbol{\tau} + \mathbf{D} = \left(\mathbf{A}_0 - \Delta\mathbf{A}\right)\ddot{\mathbf{X}} + \left(\mathbf{B}_0 - \Delta\mathbf{B}\right)\dot{\mathbf{X}} + \left(\mathbf{C}_0 - \Delta\mathbf{C}\right) \tag{6}$$

Equation (6) is rearranged and (7) is obtained:

$$\mathbf{A}_0\ddot{\mathbf{X}} + \mathbf{B}_0\dot{\mathbf{X}} + \mathbf{C}_0 = \Delta\mathbf{A}\ddot{\mathbf{X}} + \Delta\mathbf{B}\dot{\mathbf{X}} + \Delta\mathbf{C} + \mathbf{J}^{\mathrm{T}}\boldsymbol{\tau} + \mathbf{D} \tag{7}$$

Due to the error between the whole dynamic model and the nominal model, the unknown disturbance term $\mathbf{D} + \Delta\mathbf{A}\ddot{\mathbf{X}} + \Delta\mathbf{B}\dot{\mathbf{X}} + \Delta\mathbf{C}$ is obtained. Let $\boldsymbol{\varsigma}(\mathbf{X}, \dot{\mathbf{X}}, \ddot{\mathbf{X}}) = \mathbf{D} + \Delta\mathbf{A}\ddot{\mathbf{X}} + \Delta\mathbf{B}\dot{\mathbf{X}} + \Delta\mathbf{C}$, then $\mathbf{A}_0\ddot{\mathbf{X}} + \mathbf{B}_0\dot{\mathbf{X}} + \mathbf{C}_0 = \mathbf{J}^{\mathrm{T}}\boldsymbol{\tau} + \boldsymbol{\varsigma}(\mathbf{X}, \dot{\mathbf{X}}, \ddot{\mathbf{X}})$.

In the torque control algorithm, the nominal model is used; if $\varsigma(\mathbf{X}, \dot{\mathbf{X}}, \ddot{\mathbf{X}})$ is known, the control can be designed as follows:

$$\begin{aligned}\boldsymbol{\tau} = &\left(\mathbf{J}^{\mathrm{T}}\right)^{+}\\ &\cdot\left[\mathbf{A}_0\left(\ddot{\mathbf{X}}_d - \mathbf{K}_d\dot{\mathbf{e}} - \mathbf{K}_p\mathbf{e}\right) + \mathbf{B}_0\dot{\mathbf{X}} + \mathbf{C}_0 - \varsigma\left(\mathbf{X}, \dot{\mathbf{X}}, \ddot{\mathbf{X}}\right)\right]\end{aligned} \tag{8}$$

where $\mathbf{K}_P$ and $\mathbf{K}_d$ are the proportional gain and differential gain, respectively; they are positive definite matrices.

Equation (8) is substituted into (4) and we obtain

$$\begin{aligned}&\mathbf{A}\ddot{\mathbf{X}} + \mathbf{B}\dot{\mathbf{X}} + \mathbf{C} = \mathbf{J}^{\mathrm{T}} \times \left(\mathbf{J}^{\mathrm{T}}\right)^{+}\\ &\cdot\left[\mathbf{A}_0\left(\ddot{\mathbf{X}}_d - \mathbf{K}_d\dot{\mathbf{e}} - \mathbf{K}_p\mathbf{e}\right) + \mathbf{B}_0\dot{\mathbf{X}} + \mathbf{C}_0 - \varsigma\left(\mathbf{X}, \dot{\mathbf{X}}, \ddot{\mathbf{X}}\right)\right]\\ &+ \mathbf{D}\end{aligned} \tag{9}$$

where $\mathbf{J}^{\mathrm{T}} \times (\mathbf{J}^{\mathrm{T}})^{+} = \mathbf{I}_{6\times6}$ and $\mathbf{J}$ is a matrix with 8 rows and 6 columns.

Equation (9) is rearranged and the error system equation (10) that can be obtained by simplification is as follows:

$$\ddot{\mathbf{e}} + \mathbf{K}_d\dot{\mathbf{e}} + \mathbf{K}_p\mathbf{e} = 0 \tag{10}$$

In (10), $\mathbf{e} = \mathbf{X} - \mathbf{X}_d$ is the pose error, $\dot{\mathbf{e}} = \dot{\mathbf{X}} - \dot{\mathbf{X}}_d$ is the pose velocity error, and $\ddot{\mathbf{e}} = \ddot{\mathbf{X}} - \ddot{\mathbf{X}}_d$ is the pose acceleration error. $\mathbf{X}_d$ is the theoretical pose of the end effector.

The purpose of this study is to design a stable robust controller based on a nominal model. In practical engineering applications, $\varsigma(\mathbf{X}, \dot{\mathbf{X}}, \ddot{\mathbf{X}})$ is usually unknown; therefore, it is necessary to estimate $\varsigma(\mathbf{X}, \dot{\mathbf{X}}, \ddot{\mathbf{X}})$ and compensate for it.

3. RBF Neural Network Design of the WDPR

An inaccurate robot model can reduce the tracking performance of the robot. Therefore, neural networks are widely used in robot control due to their learning linearity and nonlinear mapping [17–20]. In order to compensate for the unknown item $\boldsymbol{\varsigma}(\mathbf{X}, \dot{\mathbf{X}}, \ddot{\mathbf{X}})$, an RBF neural network is used to approximate the unknown item and further improve the control accuracy of the WDPR.

The RBF neural network with compensated control is a network structure with three layers based on the practical application and the input of the WDPR, as shown in

$$\mathbf{z}(\bullet) = \boldsymbol{\Xi}^{\mathrm{T}} \boldsymbol{\Theta}(\mathbf{x}) \tag{11}$$

where $\mathbf{x}$ is the input vector comprised of the pose error and the velocity error; $\mathbf{z}(\bullet)$ is the output of the RBF neural network; $\boldsymbol{\Theta}$ is the output of the Gaussian basis function, which is a column vector with 5 rows and 1 column; $\boldsymbol{\Xi}$ is the weight of the RBF neural network and is a matrix with 5 rows and 6 columns, as shown below.

$$\begin{aligned}
\boldsymbol{\Xi} &= \begin{bmatrix} \Xi_1 & \Xi_2 & \Xi_3 & \Xi_4 & \Xi_5 \end{bmatrix}^{\mathrm{T}} \\
&= \begin{bmatrix} \Xi_{11} & \Xi_{12} & \Xi_{13} & \Xi_{14} & \Xi_{15} \\ \Xi_{21} & \Xi_{22} & \Xi_{23} & \Xi_{24} & \Xi_{25} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \Xi_{61} & \Xi_{62} & \Xi_{63} & \Xi_{64} & \Xi_{65} \end{bmatrix}^{\mathrm{T}}, \\
\mathbf{x} &= \begin{bmatrix} x_1; x_2; x_3; x_4; x_5; x_6; x_7; x_8; x_9; x_{10}; x_{11}; x_{12} \end{bmatrix} \\
&= \begin{bmatrix} e_1; e_2; e_3; e_4; e_5; e_6; de_1; de_2; de_3; de_4; de_5; de_6 \end{bmatrix}, \\
\boldsymbol{\Theta} &= \begin{bmatrix} \Theta_1, \Theta_2, \cdots \Theta_5 \end{bmatrix}^{\mathrm{T}}. \\
\Theta_j &= \exp\left(-\frac{\left\| \mathbf{x} - c_{ij} \right\|^2}{2 b_j^{\,2}} \right) \quad (i = 1 \cdots 12,\ j = 1 \cdots 5)
\end{aligned} \tag{12}$$

where c_{ij} is the coordinate vector of the center point of the Gaussian basis function of the j-th neuron in the hidden layer; $\mathbf{c} = [c_{ij}]$ is the matrix with 12 rows and 5 columns; b_j is the width of the Gaussian basis function of the j-th neuron in the hidden layer.

The RBF neural network structure for the control of the WDPR is shown in Figure 3 where the input $\mathbf{x}$ of the RBF neural network structure is 12 and the output $\mathbf{z}(\bullet)$ is a column vector with 6 rows and 1 column.

Assumption 1. Let $\boldsymbol{\varsigma}(\mathbf{X}, \dot{\mathbf{X}}, \ddot{\mathbf{X}}) = \mathbf{z}(\bullet)$, $\hat{\mathbf{z}}^{\iota}(\bullet) = \hat{\boldsymbol{\Xi}}^{\iota\mathrm{T}} \boldsymbol{\Theta}(\mathbf{x})$ is the ideal output of the RBF neural network, and $\hat{\boldsymbol{\Xi}}^{\iota\mathrm{T}}$ is the ideal weight. Suppose that a very small positive number $\Lambda = 0.001$ is given so that $\mathbf{z}(\bullet) - \hat{\mathbf{z}}^{\iota}(\bullet)$ satisfies the following condition:

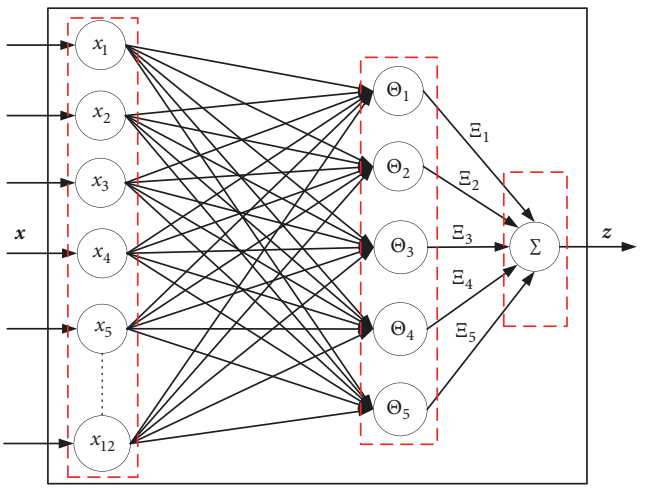


Figure 3: RBF neural network structure for the control of the WDPR.

$$\max \left\| \mathbf{z}(\bullet) - \hat{\mathbf{z}}^{\iota}(\bullet) \right\| \le \Lambda \tag{13}$$

Assumption 2. Let the approximation error $\Gamma = \mathbf{z}(\bullet) - \hat{\mathbf{z}}^{\iota}(\bullet)$; it is assumed that the approximation error Γ is bounded; that is,

$$\Gamma_0 = \sup \left\| \mathbf{z}(\bullet) - \hat{\mathbf{z}}^{\iota}(\bullet) \right\| \tag{14}$$

When Γ_0 is bounded, the unknown interference term $\boldsymbol{\varsigma}(\mathbf{X}, \dot{\mathbf{X}}, \ddot{\mathbf{X}}) = \mathbf{z}(\bullet)$ and the approximation compensation term $\hat{\mathbf{z}}^{\iota}(\bullet)$ of the RBF neural network tend to be consistent and when both are equal, the approximation error is zero. At that moment, the control system operates under optimal conditions.

4. Control Design and Stability Analysis

4.1. Control Design. The schematic diagram of the control system of the WDPR is shown in Figure 4.

Considering the experimental environment, the external disturbances, and other factors, the adaptive PD control based on the RBF neural network is as shown in

$$\boldsymbol{\tau} = \left(\mathbf{J}^{\mathrm{T}} \right)^{+} \left[\mathbf{A}_0 \left(\ddot{\mathbf{X}}_d - \mathbf{K}_d \dot{\mathbf{e}} - \mathbf{K}_p \mathbf{e} \right) + \mathbf{B}_0 \dot{\mathbf{X}} + \mathbf{C}_0 - \hat{\mathbf{z}}(\bullet) \right] \tag{15}$$

where $\boldsymbol{\tau} = \boldsymbol{\tau}_1 + \boldsymbol{\tau}_2$, $\boldsymbol{\tau}_1$ is the PD control, $\boldsymbol{\tau}_2$ is an adaptive control based on the RBF neural network, and $\hat{\mathbf{z}}(\bullet)$ is the estimated value of the ideal output $\hat{\mathbf{z}}^{\iota}(\bullet)$ of the RBF neural network.

The proposed control defined by (4) is used and the error closed-loop system equation (16) is obtained:

$$\ddot{\mathbf{e}} + \mathbf{K}_d \dot{\mathbf{e}} + \mathbf{K}_p \mathbf{e} = \mathbf{A}_0^{-1} \left(\mathbf{z}(\bullet) - \hat{\mathbf{z}}(\bullet) \right) \tag{16}$$

where $\mathbf{e} = \begin{bmatrix} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \end{bmatrix}^{\mathrm{T}}$, $\dot{\mathbf{e}} = \begin{bmatrix} de_1 & de_2 & de_3 & de_4 & de_5 & de_6 \end{bmatrix}^{\mathrm{T}}$, $e_i (i = 1 \cdots 6)$ is the

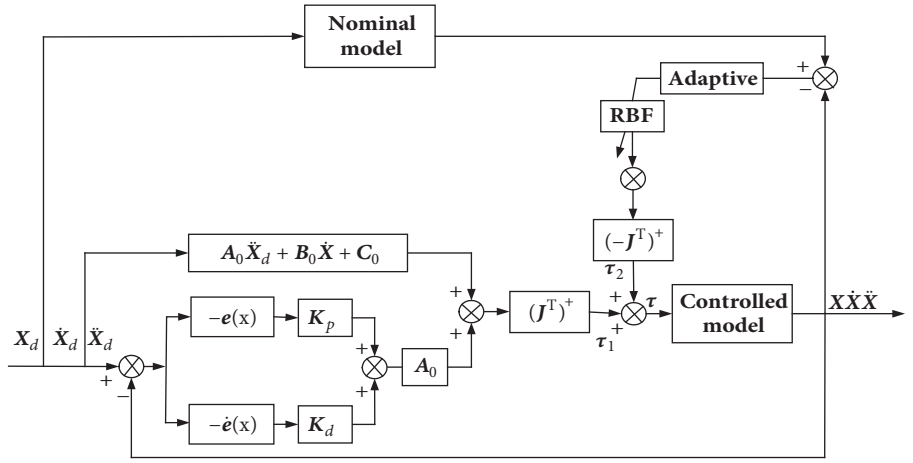


Figure 4: Schematic diagram of the control system.

subtraction of the theoretical pose $\mathbf{X}_d$ and the actual pose $\mathbf{X}$, respectively, and $de_i(i = 1 \cdots 6)$ is the subtraction of the theoretical pose velocity $\dot{\mathbf{X}}_d$ and the actual pose velocity $\dot{\mathbf{X}}$, respectively. Let $\mathbf{x} = \begin{bmatrix}\mathbf{e} & \dot{\mathbf{e}}\end{bmatrix}^{\mathrm{T}}$, then (16) can be transformed into

$$\dot{\mathbf{x}} = \begin{bmatrix}\dot{\mathbf{e}} & \ddot{\mathbf{e}}\end{bmatrix}^{\mathrm{T}} = \mathbf{K}_1\mathbf{x} + \mathbf{K}_2\left(\mathbf{z}(\bullet) - \widehat{\mathbf{z}}(\bullet)\right) \tag{17}$$

where $\mathbf{K}_1 = \begin{bmatrix}\mathbf{0}_{6\times6} & \mathbf{I}_{6\times6} \\ -\mathbf{K}_p & -\mathbf{K}_d\end{bmatrix}$ and $\mathbf{K}_2 = \begin{bmatrix}0 \\ \mathbf{A}_0^{-1}\end{bmatrix}$.

Because of $\mathbf{z}(\bullet) - \widehat{\mathbf{z}}(\bullet) = \Gamma + \widehat{\mathbf{z}}^{\prime}(\bullet) - \widehat{\mathbf{z}}(\bullet)$, let $\widehat{\mathbf{z}}^{\prime}(\bullet) - \widehat{\mathbf{z}}(\bullet) = \widehat{\Xi}^{\mathrm{T}}\boldsymbol{\Theta}(\mathbf{x})$; (17) is transformed into the following form:

$$\dot{\mathbf{x}} = \mathbf{K}_1\mathbf{x} + \mathbf{K}_2\left(\Gamma + \widehat{\Xi}^{\mathrm{T}}\boldsymbol{\Theta}(\mathbf{x})\right) \tag{18}$$

where $\widehat{\Xi}^{\mathrm{T}} = \widehat{\Xi}^{\prime\mathrm{T}} - \widehat{\Xi}^{\mathrm{T}}$ and $\widehat{\Xi}^{\mathrm{T}}$ is bounded. $\widehat{\Xi}^{\mathrm{T}}$ is the estimated weight.

4.2. Stability Analysis. (1) The stability of the above-described closed-loop system is analyzed. The Lyapunov function is

$$\mathbf{V} = 2\mathbf{x}^{\mathrm{T}}\Pi\mathbf{x} + 2tr\left(\widehat{\Xi}^{\mathrm{T}}\widehat{\Xi}\right) + 2\boldsymbol{\Theta}(\mathbf{x})^{\mathrm{T}}\boldsymbol{\Theta}(\mathbf{x}) \tag{19}$$

where Π is the symmetric positive definite matrix that satisfies

$$\Pi\mathbf{K}_1 + \mathbf{K}_1{}^{\mathrm{T}}\Pi = -\boldsymbol{\Upsilon} \tag{20}$$

$\boldsymbol{\Upsilon} \geq \mathbf{0}$, $\mathbf{V} > \mathbf{0}$; therefore, the function $\mathbf{V}$ is a positive definite matrix.

(2) The derivative of the Lyapunov function $\mathbf{V}$ can be obtained:

$$\begin{aligned}\dot{\mathbf{V}} &= 2\dot{\boldsymbol{\Theta}}(\mathbf{x})^{\mathrm{T}}\boldsymbol{\Theta}(\mathbf{x}) + 2\boldsymbol{\Theta}(\mathbf{x})^{\mathrm{T}}\dot{\boldsymbol{\Theta}}(\mathbf{x}) + 2tr\left(\dot{\widehat{\Xi}}^{\mathrm{T}}\widehat{\Xi}\right) \\ &\quad + 2tr\left(\widehat{\Xi}^{\mathrm{T}}\dot{\widehat{\Xi}}\right) + 2\dot{\mathbf{x}}^{\mathrm{T}}\Pi\mathbf{x} + 2\mathbf{x}^{\mathrm{T}}\Pi\dot{\mathbf{x}} \\ &= 4tr\left(\dot{\widehat{\Xi}}^{\mathrm{T}}\widehat{\Xi}\right) + 4\boldsymbol{\Theta}(\mathbf{x})^{\mathrm{T}}\dot{\boldsymbol{\Theta}}(\mathbf{x}) + 2\dot{\mathbf{x}}^{\mathrm{T}}\Pi\mathbf{x} + 2\mathbf{x}^{\mathrm{T}}\Pi\dot{\mathbf{x}}\end{aligned} \tag{21}$$

where $\dot{\mathbf{x}}$ is used in (21) and we obtain

$$\begin{aligned}\dot{\mathbf{V}} &= 2\left(\mathbf{x}^{\mathrm{T}}\mathbf{K}_1{}^{\mathrm{T}} + \left(\Gamma + \widehat{\Xi}^{\mathrm{T}}\boldsymbol{\Theta}(\mathbf{x})\right)^{\mathrm{T}}\mathbf{K}_2{}^{\mathrm{T}}\right)\Pi\mathbf{x} \\ &\quad + 2\mathbf{x}^{\mathrm{T}}\Pi\left(\mathbf{K}_1\mathbf{x} + \mathbf{K}_2\left(\Gamma + \widehat{\Xi}^{\mathrm{T}}\boldsymbol{\Theta}(\mathbf{x})\right)\right) + 4tr\left(\dot{\widehat{\Xi}}^{\mathrm{T}}\widehat{\Xi}\right) \\ &\quad + 4\boldsymbol{\Theta}(\mathbf{x})^{\mathrm{T}}\dot{\boldsymbol{\Theta}}(\mathbf{x})\end{aligned} \tag{22}$$

Equation (22) is simplified and we obtain

$$\begin{aligned}\dot{\mathbf{V}} &= 2\mathbf{x}^{\mathrm{T}}\Pi\mathbf{K}_1\mathbf{x} + 2\mathbf{x}^{\mathrm{T}}\Pi\mathbf{K}_2\Gamma + 2\mathbf{x}^{\mathrm{T}}\Pi\mathbf{K}_2\widehat{\Xi}^{\mathrm{T}}\boldsymbol{\Theta}(\mathbf{x}) \\ &\quad + 2\mathbf{x}^{\mathrm{T}}\mathbf{K}_1{}^{\mathrm{T}}\Pi\mathbf{x} + 2\Gamma^{\mathrm{T}}\mathbf{K}_2{}^{\mathrm{T}}\Pi\mathbf{x} + 2\boldsymbol{\Theta}(\mathbf{x})^{\mathrm{T}}\widehat{\Xi}\mathbf{K}_2{}^{\mathrm{T}}\Pi\mathbf{x} \\ &\quad + 4tr\left(\dot{\widehat{\Xi}}^{\mathrm{T}}\widehat{\Xi}\right) + 4\boldsymbol{\Theta}(\mathbf{x})^{\mathrm{T}}\dot{\boldsymbol{\Theta}}(\mathbf{x})\end{aligned} \tag{23}$$

$$\begin{aligned}\dot{\mathbf{V}} &= 2\mathbf{x}^{\mathrm{T}}\left(\Pi\mathbf{K}_1 + \mathbf{K}_1{}^{\mathrm{T}}\Pi\right)\mathbf{x} + \left(2\mathbf{x}^{\mathrm{T}}\Pi\mathbf{K}_2\Gamma\right. \\ &\quad \left. + 2\mathbf{x}^{\mathrm{T}}\Pi\mathbf{K}_2\widehat{\Xi}^{\mathrm{T}}\boldsymbol{\Theta}(\mathbf{x}) + 2\Gamma^{\mathrm{T}}\mathbf{K}_2{}^{\mathrm{T}}\Pi\mathbf{x} + 2\boldsymbol{\Theta}(\mathbf{x})^{\mathrm{T}}\widehat{\Xi}\mathbf{K}_2{}^{\mathrm{T}}\Pi\mathbf{x}\right) \\ &\quad + 4tr\left(\dot{\widehat{\Xi}}^{\mathrm{T}}\widehat{\Xi}\right) + 4\boldsymbol{\Theta}(\mathbf{x})^{\mathrm{T}}\dot{\boldsymbol{\Theta}}(\mathbf{x})\end{aligned} \tag{24}$$

Because $2\mathbf{x}^{\mathrm{T}}\Pi\mathbf{K}_2\Gamma = 2\Gamma^{\mathrm{T}}\mathbf{K}_2{}^{\mathrm{T}}\Pi\mathbf{x}$, $2\mathbf{x}^{\mathrm{T}}\Pi\mathbf{K}_2\widehat{\Xi}^{\mathrm{T}}\boldsymbol{\Theta}(\mathbf{x}) = 2\boldsymbol{\Theta}(\mathbf{x})^{\mathrm{T}}\widehat{\Xi}\mathbf{K}_2{}^{\mathrm{T}}\Pi\mathbf{x}$, (24) is simplified as

$$\begin{aligned}\dot{\mathbf{V}} &= 4tr\left(\mathbf{K}_2{}^{\mathrm{T}}\Pi\mathbf{x}\boldsymbol{\Theta}(\mathbf{x})^{\mathrm{T}}\widehat{\Xi} + \dot{\widehat{\Xi}}^{\mathrm{T}}\widehat{\Xi}\right) - 2\mathbf{x}^{\mathrm{T}}\boldsymbol{\Upsilon}\mathbf{x} + 4\Gamma^{\mathrm{T}}\mathbf{K}_2{}^{\mathrm{T}}\Pi\mathbf{x} \\ &\quad + 4\boldsymbol{\Theta}(\mathbf{x})^{\mathrm{T}}\dot{\boldsymbol{\Theta}}(\mathbf{x})\end{aligned} \tag{25}$$

(3) Adaptive control technology is usually used to deal with dynamic uncertainty [21–23]; therefore, in order to respond to the dynamic characteristics of the WDPR motion process and disturbances and ensure the asymptotic stability of the system, the adaptive control algorithm is designed as follows [24]:

$$\dot{\widehat{\Xi}}^{\mathrm{T}} = -\mathbf{K}_2{}^{\mathrm{T}}\Pi\mathbf{x}\boldsymbol{\Theta}(\mathbf{x})^{\mathrm{T}} \tag{26}$$

$$\dot{\widehat{\Xi}} = -\boldsymbol{\Theta}(\mathbf{x})\,\mathbf{x}^{\mathrm{T}}\Pi\mathbf{K}_2 \tag{27}$$

Equation (27) is substituted into (25) and we obtain

$$\dot{\mathbf{V}} = -2\mathbf{x}^{\mathrm{T}}\boldsymbol{\Upsilon}\mathbf{x} + 4\Gamma^{\mathrm{T}}\mathbf{K}_2{}^{\mathrm{T}}\Pi\mathbf{x} + 4\boldsymbol{\Theta}(\mathbf{x})^{\mathrm{T}}\dot{\boldsymbol{\Theta}}(\mathbf{x}) \tag{28}$$

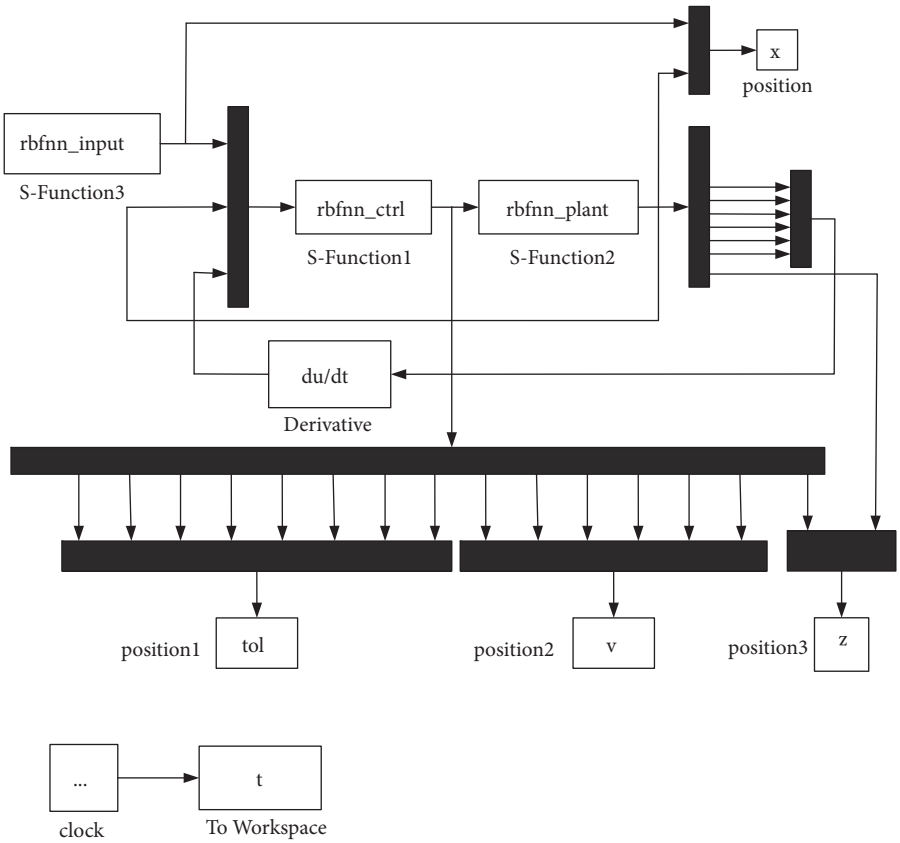


Figure 5: The program diagram of the RBF neural network simulation.

To ensure that the closed-loop system is stable, first of all, $\dot{\mathbf{V}} \leq \mathbf{0}$ needs to be proven. According to the known conditions, $\|\Gamma\| \leq \|\Gamma_0\|$, $\boldsymbol{\Theta}(\mathbf{x})^{\mathrm{T}}\dot{\boldsymbol{\Theta}}(\mathbf{x}) > \mathbf{0}$, $\|\mathbf{K}_2\| = \|\mathbf{A}_0{}^{-1}\|$, $\dot{\mathbf{V}}$ can be obtained:

$$\begin{aligned}\dot{\mathbf{V}} \leq{}& -2\lambda_{\min}(\boldsymbol{\Upsilon})\|\mathbf{x}\|^2 + 4\boldsymbol{\Theta}(\mathbf{x})^{\mathrm{T}}\dot{\boldsymbol{\Theta}}(\mathbf{x}) + 4\|\Gamma_0\|\left\|\mathbf{A}_0{}^{-1}\right\| \\ &\cdot \lambda_{\max}(\Pi)\|\mathbf{x}\| = -2\left(\lambda_{\min}(\boldsymbol{\Upsilon})\|\mathbf{x}\|^2 - 2\boldsymbol{\Theta}(\mathbf{x})^{\mathrm{T}}\dot{\boldsymbol{\Theta}}(\mathbf{x})\right. \\ &\left. - 2\|\Gamma_0\|\left\|\mathbf{A}_0{}^{-1}\right\|\lambda_{\max}(\Pi)\|\mathbf{x}\|\right)\end{aligned} \tag{29}$$

In (29), $\lambda_{\min}(\boldsymbol{\Upsilon})$ is the minimum eigenvalue of the matrix $\boldsymbol{\Upsilon}$ and $\lambda_{\max}(\Pi)$ is the maximum eigenvalue of the matrix Π.

In order to ensure that $\dot{\mathbf{V}} \leq \mathbf{0}$, the conditions of (30), (31), and (32) must be satisfied:

$$\begin{aligned}\|\mathbf{x}\|\lambda_{\min}(\boldsymbol{\Upsilon})\|\mathbf{x}\| \geq{}& 2\|\Gamma_0\|\left\|\mathbf{A}_0{}^{-1}\right\|\lambda_{\max}(\Pi)\|\mathbf{x}\| \\ &+ 2\boldsymbol{\Theta}(\mathbf{x})^{\mathrm{T}}\dot{\boldsymbol{\Theta}}(\mathbf{x})\end{aligned} \tag{30}$$

$$\begin{aligned}\|\mathbf{x}\|\lambda_{\min}(\boldsymbol{\Upsilon}) \geq{}& 2\|\Gamma_0\|\left\|\mathbf{A}_0{}^{-1}\right\|\lambda_{\max}(\Pi) \\ &+ \frac{2\boldsymbol{\Theta}(\mathbf{x})^{\mathrm{T}}\dot{\boldsymbol{\Theta}}(\mathbf{x})}{\|\mathbf{x}\|}\end{aligned} \tag{31}$$

$$\begin{aligned}\|\mathbf{x}\| \geq{}& \frac{2\|\Gamma_0\|\left\|\mathbf{A}_0{}^{-1}\right\|\lambda_{\max}(\Pi)}{\lambda_{\min}(\boldsymbol{\Upsilon})} \\ &+ \frac{2\boldsymbol{\Theta}(\mathbf{x})^{\mathrm{T}}\dot{\boldsymbol{\Theta}}(\mathbf{x})}{\lambda_{\min}(\boldsymbol{\Upsilon})\|\mathbf{x}\|}\end{aligned} \tag{32}$$

That is, it is possible to improve the convergence effect by either increasing the eigenvalue of the matrix $\boldsymbol{\Upsilon}$, decreasing the eigenvalue of the matrix Π, or decreasing Γ_0. When $\dot{\mathbf{V}} \leq \mathbf{0}$, the closed-loop system is stable.

5. Simulation Experiments

In order to verify whether the proposed control method satisfies the requirements for practical applications, it is combined with the dynamic equation of the WDPR and a MATLAB/Simulink simulation is conducted. The 6-DOF WDPR is used as the simulation object. Based on the theory from [24], better convergence results can be obtained when $\mathbf{K}_p$ and $\mathbf{K}_d$ have the following values:

$$\begin{aligned}\mathbf{K}_p &= \mathrm{diag}\left(\begin{bmatrix}3000 & 11000 & 12000 & 95 & 80 & 180\end{bmatrix}\right), \\ \mathbf{K}_d &= \mathrm{diag}\left(\begin{bmatrix}150 & 1900 & 2100 & 25 & 22 & 40\end{bmatrix}\right).\end{aligned} \tag{33}$$

Due to the experimental environment, the external disturbances, and other factors, the external disturbances of the system are set to δ_i [24]:

$$\delta_i = \delta_{d1} + \delta_{d2}\|e\| + \delta_{d3}\|\dot{e}\| \tag{34}$$

In (31), $\delta_{d1} = 1$, $\delta_{d2} = 2$, $\delta_{d3} = 3$, $i = 1, 2 \cdots 6$.

$$\boldsymbol{\delta} = \begin{bmatrix}\delta_1 & \delta_2 & \delta_3 + mg & \delta_4 & \delta_5 & \delta_6\end{bmatrix} \tag{35}$$

The simulation program of the RBF neural network is established, as shown in Figure 5; the parallel robot has six DOF and the theoretical input value is set as follows:

Table 1: Position of P_i and B_i ($i = 1 \ldots 8$).

Symbol	Coordinate point (mm)	Symbol	Coordinate point (mm)
P_1	$(-30,19,-19)^T$	B_1	$(472,815,-129)^T$
P_2	$(-30,-19,-19)^T$	B_2	$(515,-772,-129)^T$
P_3	$(17,-53,-5)^T$	B_3	$(-472,-815,-129)^T$
P_4	$(17,53,-5)^T$	B_4	$(-515,772,-129)^T$
P_5	$(17,53,5)^T$	B_5	$(-472,815,-90)^T$
P_6	$(-30,19,-19)^T$	B_6	$(515,772,-90)^T$
P_7	$(-30,-19,-19)^T$	B_7	$(472,-815,-90)^T$
P_8	$(17,-53,5)^T$	B_8	$(-515,-772,-90)^T$

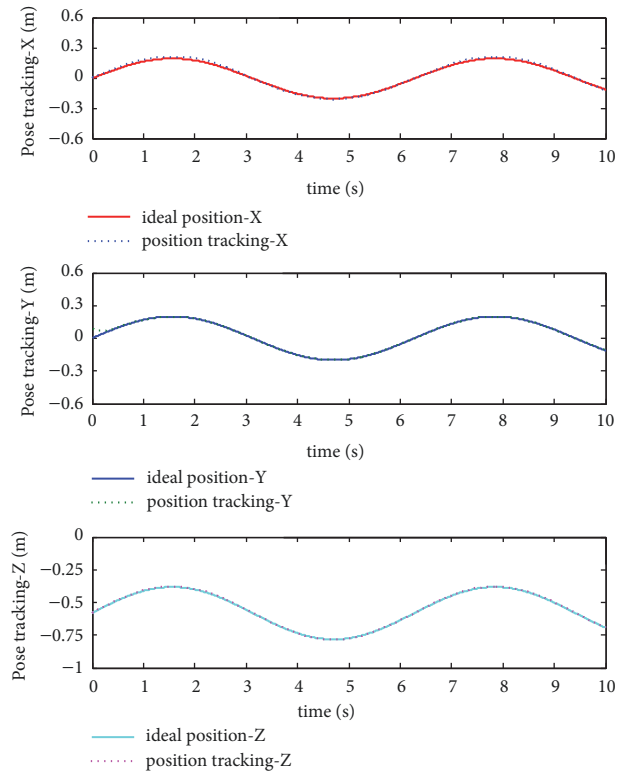


Figure 6: Position tracking for X, Y, and Z.

$$\mathbf{X}_d = \left[0.2 \times \sin(t) \quad 0.2 \times \sin(t) \quad 0.2 \times \sin(t) - 0.582 \quad \left(\frac{pi}{6}\right) \times \sin(t) \quad \left(\frac{pi}{6}\right) \times \sin(t) \quad \left(\frac{pi}{6}\right) \times \sin(t)\right]^T \tag{36}$$

Due to the nature of the task and the required results, the relative error is used, because it reflects the measurement errors appropriately. The range of the relative error of the pose and pose velocity is from -5% to 5%; the range of the relative error of the pose compensation is from -5% to 5%.

In Figure 1, the position of the connection point P_i in the aircraft model (the end effector) is in a moving coordinate system and the position of the hinge point B_i in the pulley is in a static coordinate system; the coordinates are shown in Table 1. In the moving coordinate system in Figure 1, P_1 (P_6) and P_2 (P_7), P_3 and P_4, and P_5 and P_8 are symmetrical on the x-axis.

The simulation results of the proposed control method for the WDPR are shown in Figures 6–18.

It can be seen from Figures 6 and 7 that the tracking curves of the theoretical pose and the actual pose are well

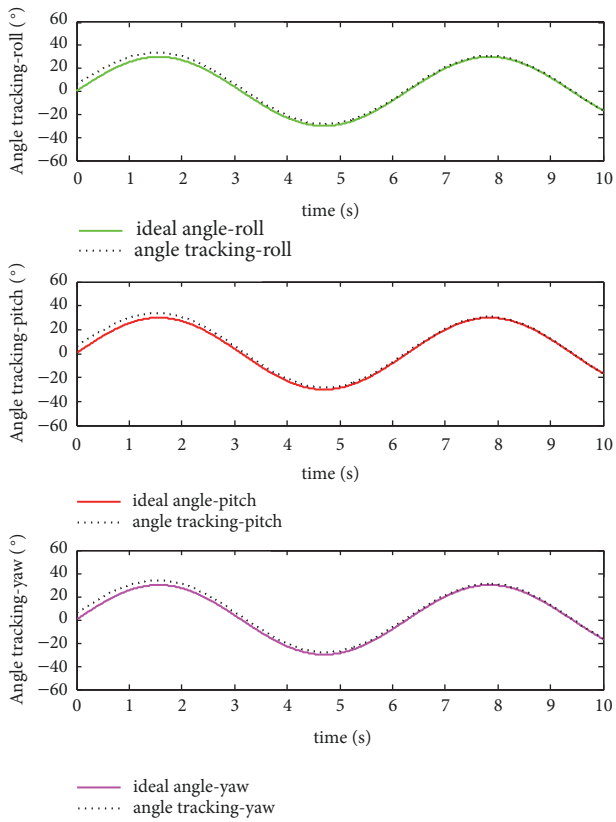


Figure 7: Angle tracking for roll, pitch, and yaw.

aligned. In Figure 8, the relative error results show that the designed control is correct and meets the experimental requirements. The reason for the errors in the simulation results may be due to some unknown factors.

Figures 9–11 show the results of the motor control input, the pose compensation, and the relative error, respectively. It can be seen from Figure 9 that the initial control input is slightly larger, which is probably due to the initial start of the motor; however, it does not exceed the rated torque of the motor (640 N.mm) and the control input of the 8 motors tends to stabilize after 2.8 s. In Figure 10, it can be seen that the pose uncertainty ($vi(i = 1 \cdots 6)$) and the approximation compensation ($vni(i = 1 \cdots 6)$) tend to be consistent after 0.02s. The relative error between the pose uncertainty and the approximation compensation is in the range of -0.025 to 0.025 within 10s (Figure 11). These results indicate that the compensation is effective. Figure 12 shows that the tension changes for the 8 wires are relatively small, which minimizes the risk of broken wires. It can be seen from Figure 13 that the RBF neural network has a good approximation performance. The simulation results are correct and represent the actual conditions.

Therefore, the simulation results show that the RBF neural network effectively approximates the error function and the tracking error of the WDPR is greatly improved; in addition, the proposed RBF neural network adaptive PD control is feasible and reliable and meets the basic requirements of the WDPR.

6. Prototype Experiments

In order to verify the correctness and effectiveness of the designed neural network adaptive PD control method, an experimental verification is performed using the WDPR prototype platform (Figure 14(a)).

The experimental platform consists of software and hardware. The software is programmed using C++ in Visual Studio. The hardware includes the PC, servo system, motion controller, mechanical structure, and other components. The PC serves as the main controller; the type of servo system is SGM7J-04AFC6S with a power of 400 W. The mechanical structure consists of the frame, the ball screw, the end effector, the wire-driven system, etc. The frame is constructed of aluminum profiles and its specific size is shown in Figure 1. An IMAC-HX integrated motion controller (Turbo PMAC) is used. The output of the controller is an analog signal and the DC regulated power supply of DRP-240-24 is used to supply the controller, as shown in Figure 14(b).

In order to carry out the neural network adaptive PD control experiment, the servo motor is set to torque mode and the hardware and software environment are configured and connected. The dynamic control software is designed to complete the control experiment of the WDPR prototype platform. The control software includes the main control program, trajectory planning, inverse kinematics model, dynamic model, and other modules; a screenshot of the control software is shown in Figure 15.

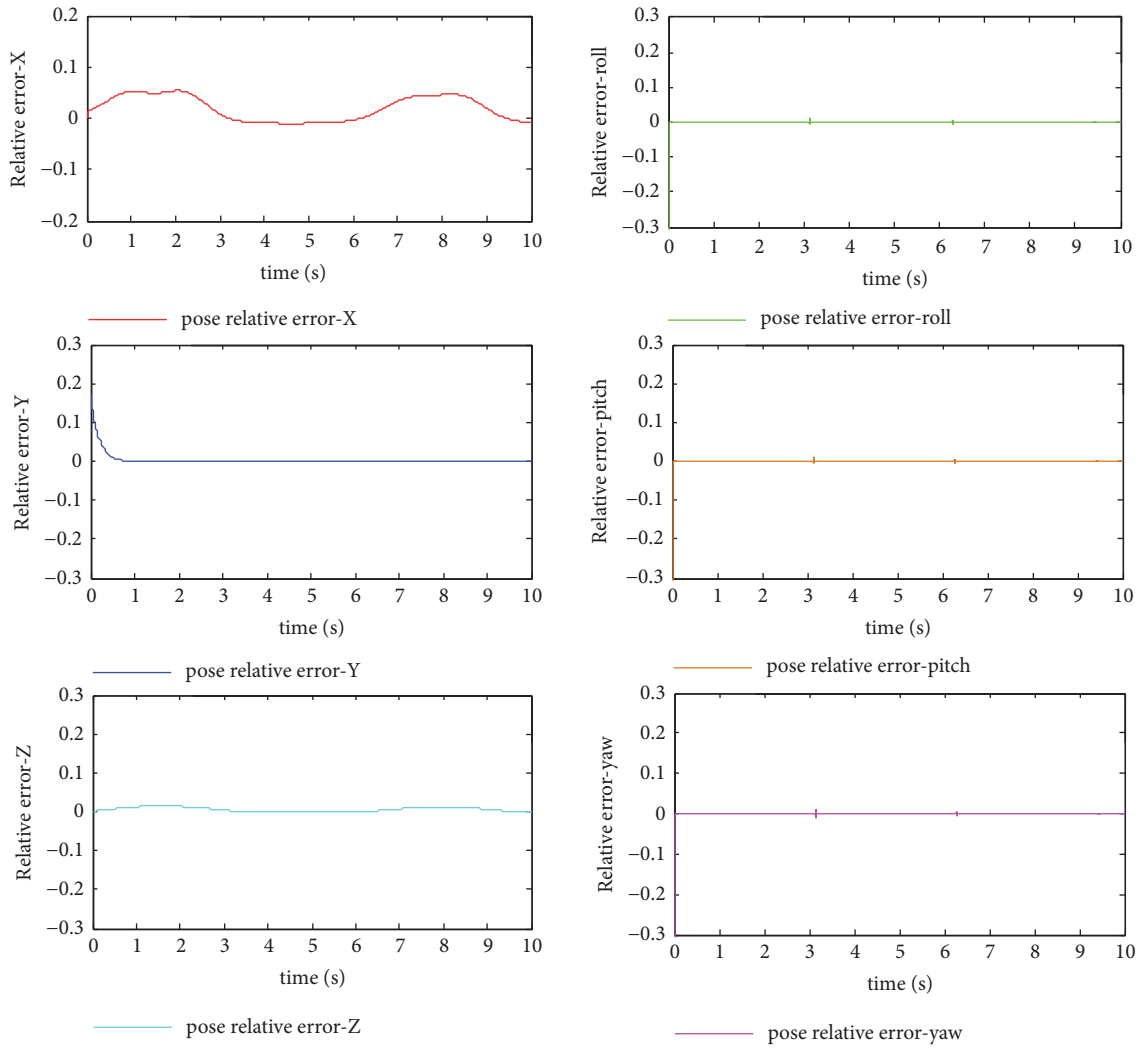


Figure 8: Relative error (pose).

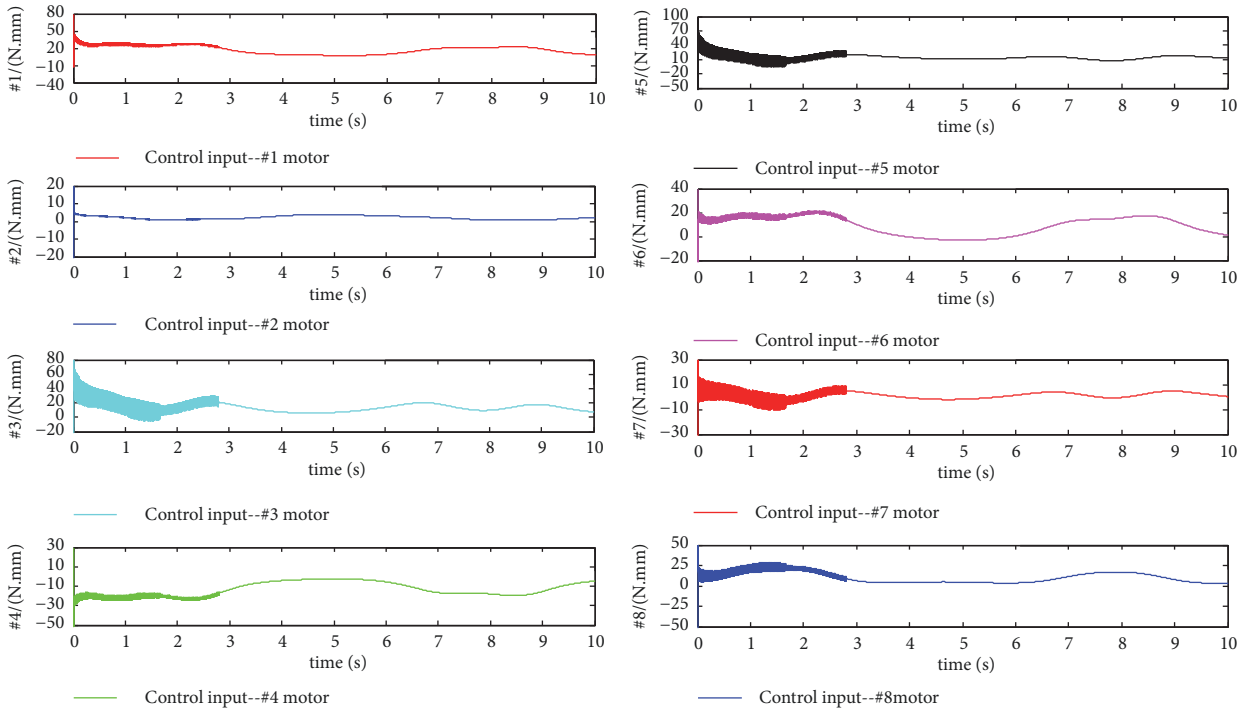
Since the dynamic control software involves many matrix operations for modeling and control, the Eigen library is applied to handle the operation of the matrix vector. The dynamic control software invokes Microsoft Foundation Classes (MFC) to design the visual interface of the program. The software interface of the WDPR prototype experimental platform designed by using the class library MFC is shown in Figure 16.

The Jacobi matrix $\mathbf{J}$, the pseudoinverse $(\mathbf{J}^{\mathrm{T}})^{+}$of the transpose of the Jacobi matrix, and the neural network compensation operation involve many matrix operations and the calculation speed is too slow for real-time online operation. Therefore, the pseudoinverse $(\mathbf{J}^{\mathrm{T}})^{+}$of the transpose of the Jacobi matrix is calculated offline by MATLAB simulation and the compensation of the neural network is sampled and invoked in real time to improve the control efficiency of the WDPR prototype platform.

Because the WDPR prototype is a complex nonlinear system, the feasibility and control performance of the designed neural network adaptive PD control are verified by using a one-DOF motion and the desired trajectory of the end effector is set as $\mathbf{X}_d = [0.2 \times \sin(t) \;\; 0 \;\; -0.582 \;\; 0 \;\; 0 \;\; 0]^{\mathrm{T}}$.

The experimental results of the WDPR prototype are shown in Figures 17–23.

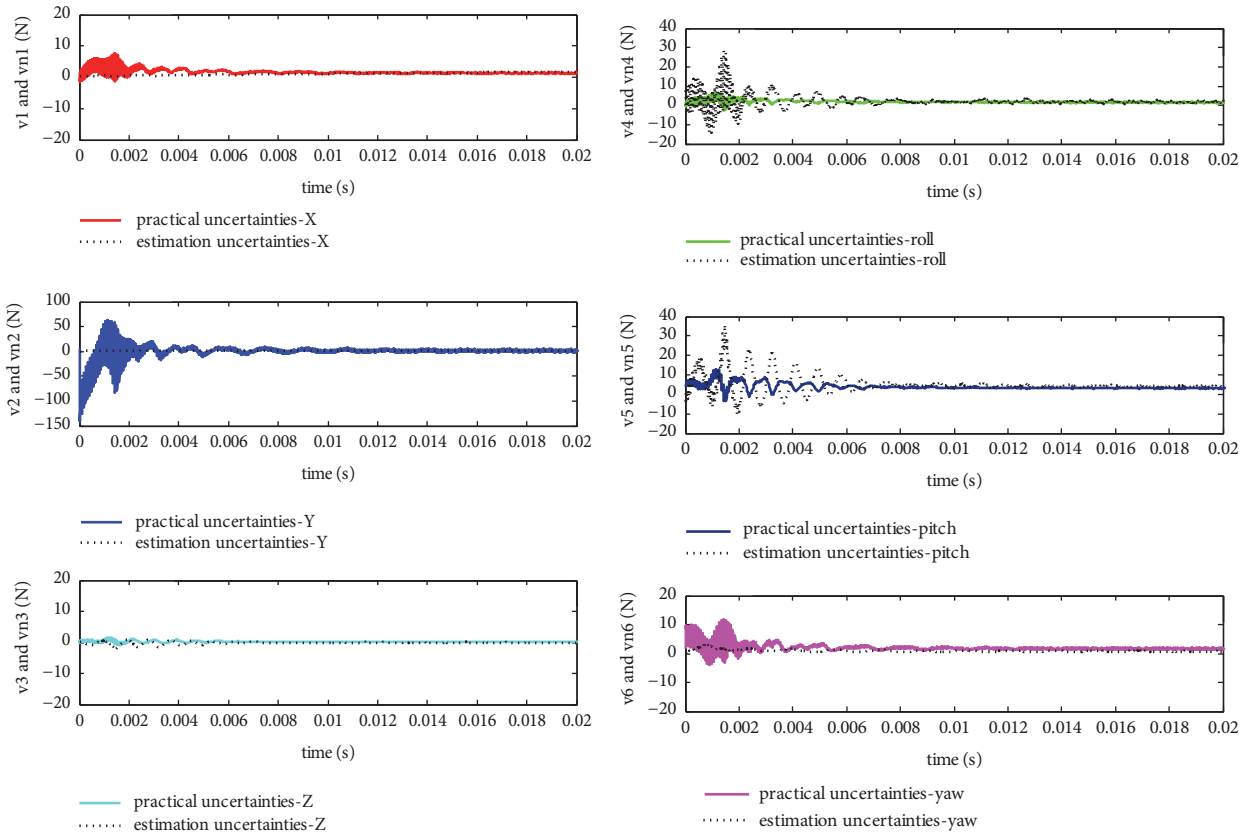
The experimental results (Figures 17–23) show that when the proposed control method based on the RBF neural network compensation is used for the dynamic tracking control of the WDPR, the actual pose of the end effector is smooth and continuous, the absolute error of the pose is small (±3mm), and the relative error is also small (±1%). We use the #1 motor as an example; the results show that the trend of the actual angular velocity of the motor is correct, the absolute error is ±0.1 rad/s, and the relative error is ±1.5%; the actual output torque of the motor (±0.08N.m) is also within the range of the rated torque of the #1 motor. Therefore, it can be concluded that the proposed RBF neural network adaptive PD control method ensures that the end effector of the WDPR operates on the basis of the desired trajectory. The errors meet the experimental requirements and the tracking performance is good.



(a) Control input of motors 1-4

(b) Control input of motors 5-8

Figure 9: Torque input of motors.



(a) Compensation of X, Y, and Z

(b) Compensation of roll, pitch, and yaw

Figure 10: Pose uncertainty and approximation compensation.

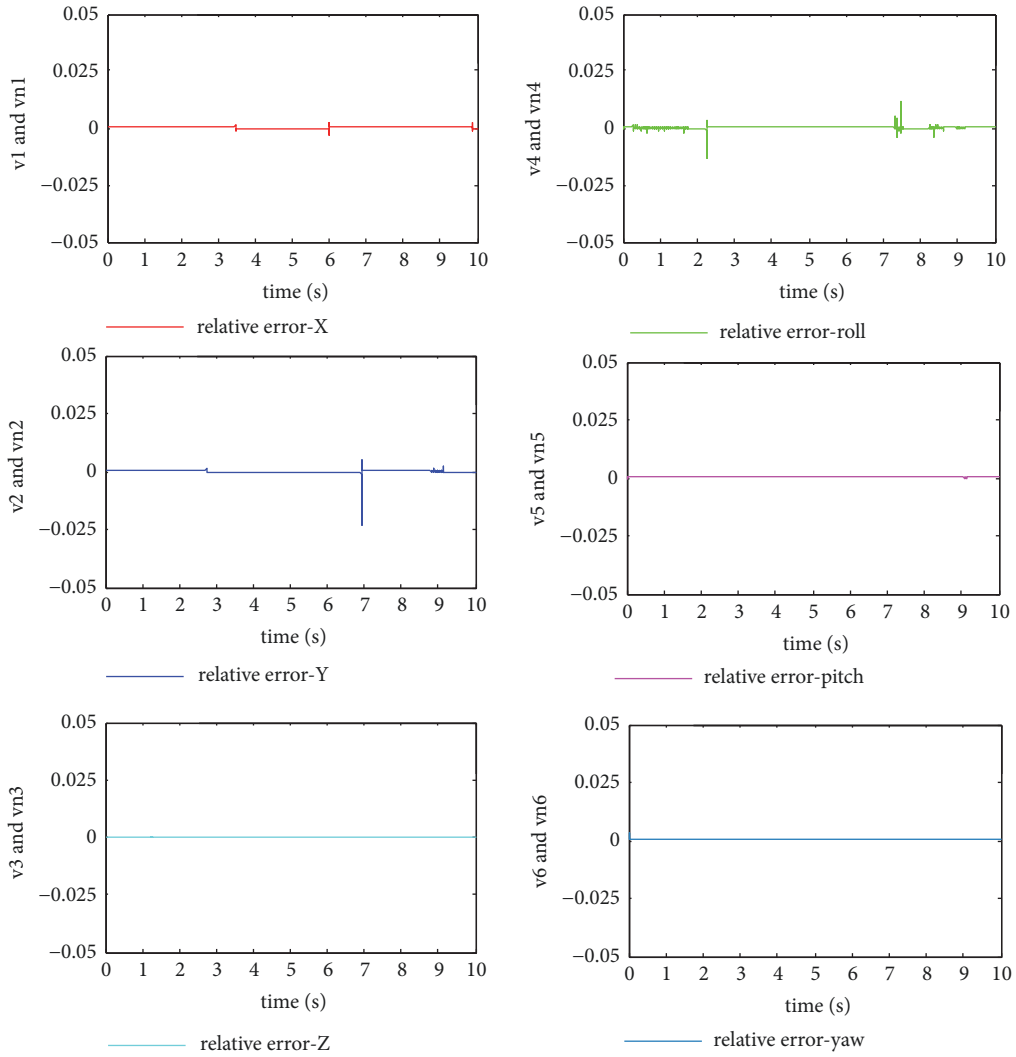


Figure 11: Relative error (pose compensation).

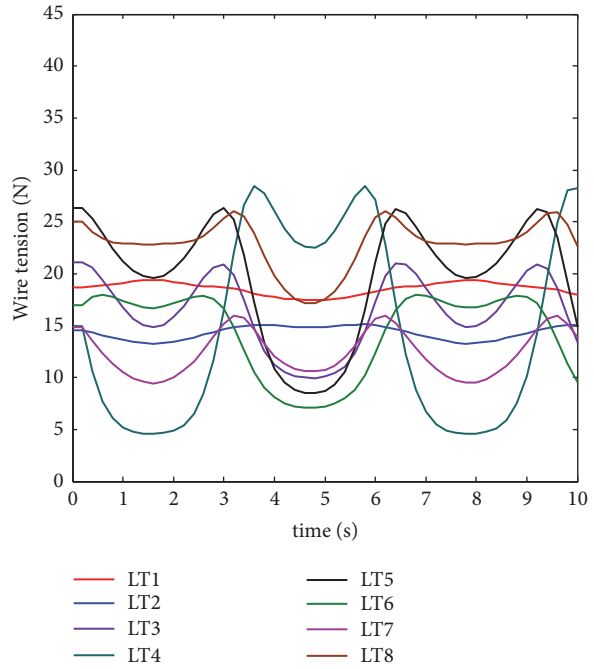


Figure 12: Wire tension.

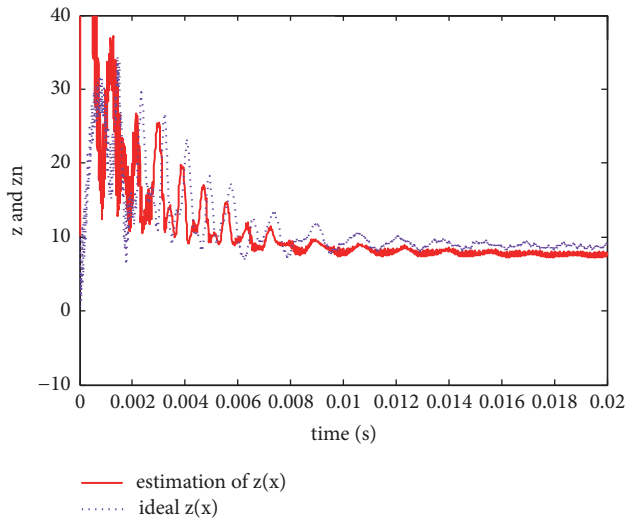


Figure 13: RBF neural network approximation of $\|\mathbf{z}(\bullet)\|$.

(a) WDPR-8 prototype platform (b) IMAC controller

Figure 14: WDPR-8 prototype platform and IMAC controller.

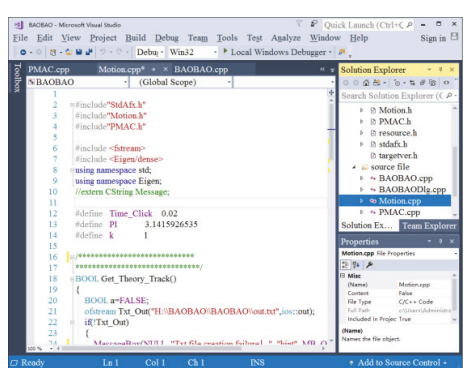


Figure 15: Control software.

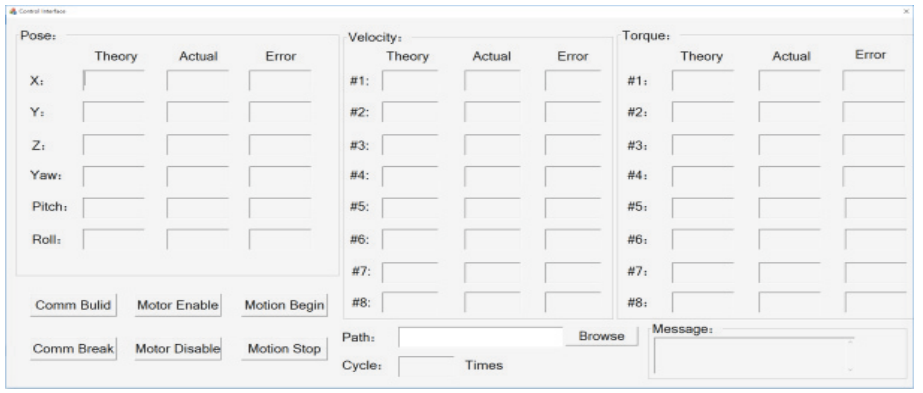


Figure 16: Software interface of experimental platform.

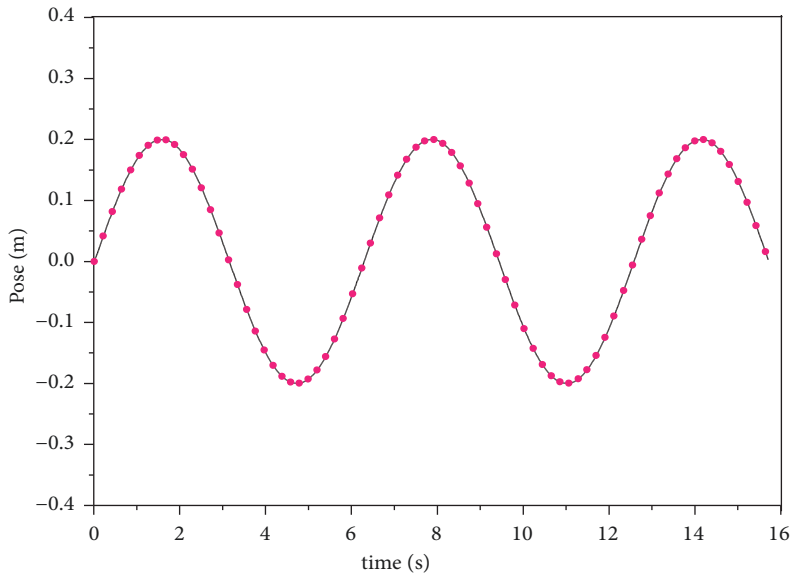


Figure 17: Actual pose of the end effector.

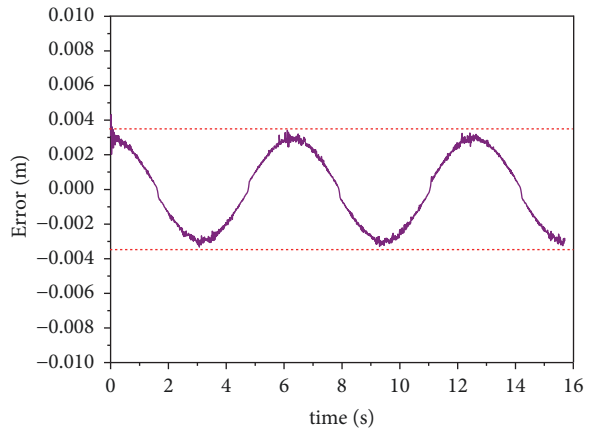


Figure 18: Absolute error of the pose.

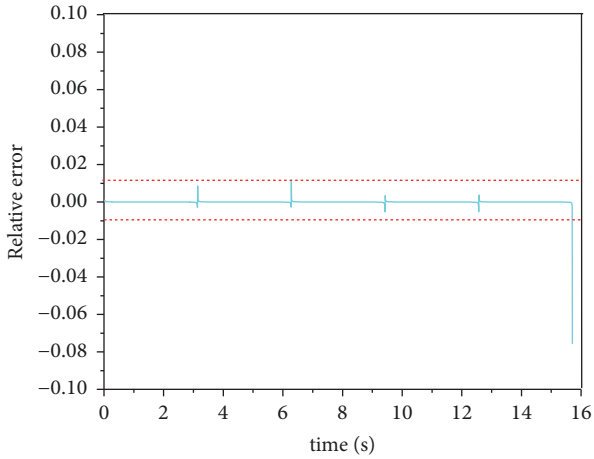


Figure 19: Relative error of the pose.

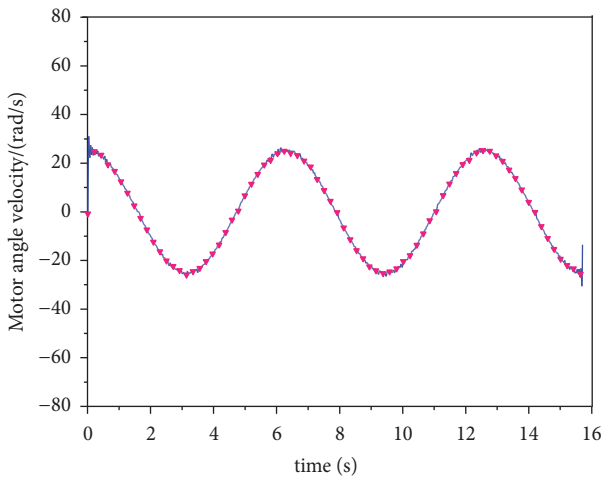


Figure 20: Actual angular velocity of #1 motor.

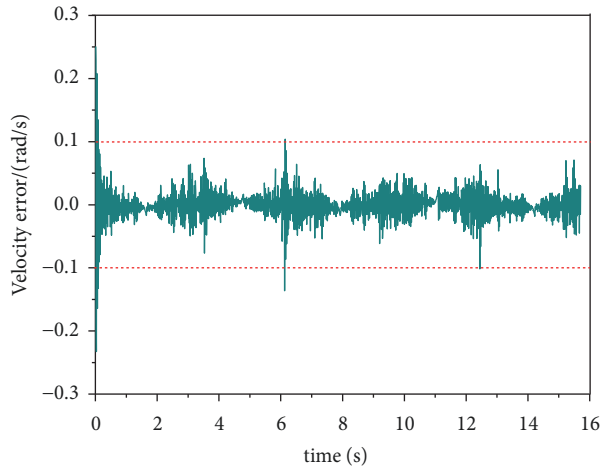


Figure 21: Absolute error of #1 motor (angular velocity).

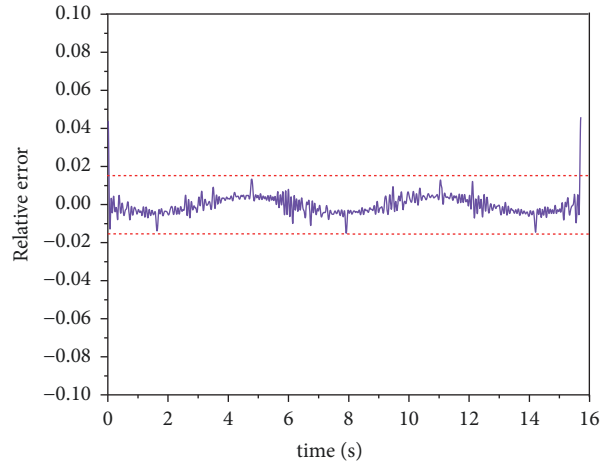


Figure 22: Relative error of #1 motor (angular velocity).

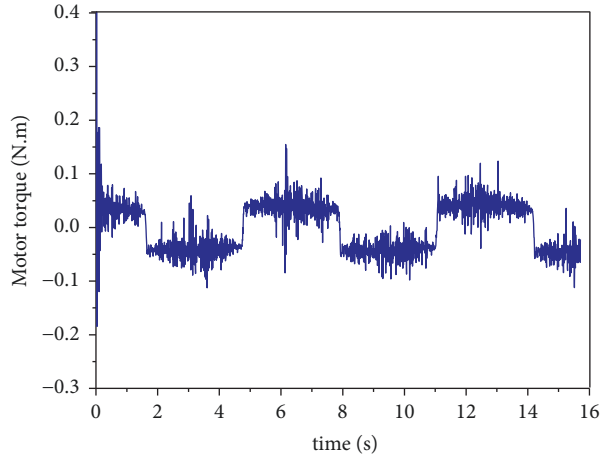


Figure 23: Output torque of #1 motor.

7. Conclusions

In this study, a new tracking control method is proposed for a WDPR and prototype experiments are conducted. The proposed control method is a torque control scheme based on the known dynamic model of the WDPR. The adaptive PD control method is based on RBF neural network compensation and the trajectory tracking performance of the WDPR is improved by an approximation of the unknown parameters. The approach is based on the Lyapunov stability theory and an adaptive control method is used to improve the convergence of the RBF input; the stability of the closed-loop system of the WDPR is demonstrated. The simulation results and the experimental results show that it is feasible to use an RBF neural network for the WDPR and fully proves the validity of the proposed control method. The research results provide guidance for the application of RBF neural networks to control a WDPR prototype for wind tunnel tests. In order to ensure the applicability of the WDPR to engineering practice, in future research, we plan to upgrade the WDPR system and enable the WDPR to achieve complex multi-degree-of-freedom movement to meet the requirements of wind tunnel test indicators.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grants nos. 11072207, 11472234, and 11702232) and the Fundamental Research Funds for the Central Universities (20720180071).

References

[1] A. K. Salima, G. S. Dwarakish, K. V. Liju, T. Justin, D. Gayathri, and R. Rajeesh, "Weekly prediction of tides using neural networks," in *proceedings of the 8th International Conference on Asian and Pacific Coasts (APAC 2015)*, vol. 116, pp. 678–682, 2015.

[2] T. Fukuda and T. Shibata, "Theory and applications of neural networks for industrial control systems," *IEEE Transactions on Industrial Electronics*, vol. 39, no. 6, pp. 472–489, 1992.

[3] Z. Defeng, *Neural Network Application Design of MATLAB*, China Machine Press, Beijing, China, 2nd edition, 2011.

[4] H. T. Dinh, R. Kamalapurkar, S. Bhasin, and W. E. Dixon, "Dynamic neural network-based robust observers for uncertain nonlinear systems," *Neural Networks*, vol. 60, pp. 44–52, 2014.

[5] Y. Wang, L. Cheng, Z.-G. Hou, J. Yu, and M. Tan, "Optimal formation of multirobot systems based on a recurrent neural network," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 2, pp. 322–333, 2016.

[6] A. Melingui, O. Lakhal, B. Daachi, J. B. Mbede, and R. Merzouki, "Adaptive neural network control of a compact bionic handling arm," *IEEE/ASME Transactions on Mechatronics*, vol. 20, no. 6, pp. 2862–2875, 2015.

[7] K. Sabahi, S. Ghaemi, J. Liu, and M. A. Badamchizadeh, "Indirect predictive type-2 fuzzy neural network controller for a class of nonlinear input - delay systems," *ISA Transactions®*, vol. 71, pp. 185–195, 2017.

[8] S. Wen, J. Yuan, and J. Zhu, "Radial basis functional link network and hamilton jacobi issacs for force/position control in robotic manipulation," *Mathematical Problems in Engineering*, vol. 2012, Article ID 568796, 10 pages, 2012.

[9] X. Zhao, X. Wang, G. Zong, and H. Li, "Fuzzy-approximation-based adaptive output-feedback control for uncertain non-smooth nonlinear systems," *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 6, pp. 3847–3859, 2018.

[10] X. Zhao, H. Yang, W. Xia, and X. Wang, "Adaptive fuzzy hierarchical sliding-mode control for a class of MIMO nonlinear time-delay systems with input saturation," *IEEE Transactions on Fuzzy Systems*, vol. 25, no. 5, pp. 1062–1077, 2017.

[11] H. Wang, H. R. Karimi, P. X. Liu, and H. Yang, "Adaptive neural control of nonlinear systems with unknown control directions and input dead-zone," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 48, no. 11, pp. 1897–1907, 2018.

[12] H. Wang, P. X. Liu, and B. Niu, "Robust fuzzy adaptive tracking control for nonaffine stochastic nonlinear switching systems," *IEEE Transactions on Cybernetics*, vol. 48, no. 8, pp. 2462–2471, 2018.

[13] Y. SuiLu, *Key Theoretical Issue Analysis and Applied Research on Wind Tunnel Tests of WDPSS-8, [Ph.D. thesis]*, Xiamen University, China, 2013.

[14] "Summary of RBF Neural Networks," Baidu.com, 2012, https://wenku.baidu.com/view/fdfe3e3e0b4c2e3f572763ec.html.

[15] S. Behzadipour and A. Khajepour, "Design of reduced DOF parallel cable-based robots," *Mechanism and Machine Theory*, vol. 39, no. 10, pp. 1051–1065, 2004.

[16] C. B. Pham, S. H. Yeo, G. Yang, M. S. Kurbanhusen, and I.-M. Chen, "Force-closure workspace analysis of cable-driven parallel mechanisms," *Mechanism and Machine Theory*, vol. 41, no. 1, pp. 53–69, 2006.

[17] J. B. Butcher, D. Verstraeten, B. Schrauwen, C. R. Day, and P. W. Haycock, "Reservoir computing and extreme learning machines for non-linear time-series data analysis," *Neural Networks*, vol. 38, pp. 76–89, 2013.

[18] M. Chowdhury, A. Alouani, and F. Hossain, "How much does inclusion of non-linearity and multi-point pattern recognition improve the spatial mapping of complex patterns of groundwater contamination?" *Nonlinear Processes in Geophysics*, vol. 16, no. 2, pp. 313–317, 2009.

[19] W. Si, X. Dong, and F. Yang, "Adaptive neural control for stochastic pure-feedback non-linear time-delay systems with output constraint and asymmetric input saturation," *IET Control Theory & Applications*, vol. 11, no. 14, pp. 2288–2298, 2017.

[20] M. Bhuyan, K. K. Sarma, and N. E. Mastorakis, "Nonlinear mobile link adaptation using modified FLNN and channel sounder arrangement," *IEEE Access*, vol. 5, pp. 10390–10402, 2017.

[21] W. Huanqing, S. Peng, L. Hongyi, and Q. Zhou, "Adaptive neural tracking control for a class of nonlinear systems with dynamic uncertainties," *IEEE Transactions on Cybernetics*, vol. 47, no. 10, pp. 3075–3087, 2017.

[22] B. S. Park, S. J. Yoo, J. B. Park, and Y. H. Choi, "Adaptive neural sliding mode control of nonholonomic wheeled mobile robots with model uncertainty," *IEEE Transactions on Control Systems Technology*, vol. 17, no. 1, pp. 207–214, 2009.

[23] M. Ghaemi, S. K. Hosseini-Sani, and M. H. Khooban, "Direct adaptive general type-2 fuzzy control for a class of uncertain non-linear systems," *IET Science, Measurement & Technology*, vol. 8, no. 6, pp. 518–527, 2014.

[24] L. Jinkun, *RBF Neural Network Control for Mechanical Systems Design, Analysis and MATLAB Simulation*, Tsinghua University Press, Beijing, China, 2014.