

Article

Compliant Contact Force Control for Aerial Manipulator of Adaptive Neural Network-Based Robust Control

Qian Fang  and Pengjun Mao *

School of Mechanical and Electrical Engineering, Henan University of Science and Technology, Luoyang 471003, China; fangqian@haust.edu.cn

* Correspondence: 9905286@haust.edu.cn

Abstract: Aerial manipulators expand the application scenarios of manipulators into the air. To complete various operations, the contact force between the aerial manipulator and the target must be precisely controlled. In this study, we first established the mathematical models of the multirotor and the manipulator separately. Their mutual influence is regarded as each other's disturbance, and the overall linkage mechanism is established through analysis. Then, a robust sliding mode control strategy is developed for accurate trajectory tracking. The controller is derived from Lyapunov theory, which can ensure the stability of the closed-loop system. To compensate for the effect of system uncertainty, an adaptive radial basis function neural network is devised to approximate the part of the controller containing the model information. In addition, an impedance controller is designed to convert force control into position control to make the manipulator contact with the target compliantly. Finally, the simulation and experimental results indicate that the proposed method can guarantee the accuracy of the contact force and has good robustness.

Keywords: aerial manipulator; sliding mode control; RBF neural network; impedance control



Citation: Fang, Q.; Mao, P. Compliant Contact Force Control for Aerial Manipulator of Adaptive Neural Network-Based Robust Control. *Sensors* **2024**, *24*, 2556. <https://doi.org/10.3390/s24082556>

Academic Editor: Fanglai Zhu

Received: 5 February 2024

Revised: 10 April 2024

Accepted: 13 April 2024

Published: 16 April 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In recent years, robot manipulators have been widely used in many fields, such as manufacturing [1], assembly [2], grinding [3], and drilling [4]. However, these robotic arms are almost always mounted on fixed bases, significantly limiting their working space. In order to expand the operating space, installing the manipulator on a non-fixed platform for operation has become a hot topic in current research. As a combination of aerial platform and manipulator, the aerial manipulator has better maneuverability in three-dimensional space and flexibility to operate. It has consequential value in multiple domains like national defense, rescue, and scientific research and has recently attracted more and more attention from researchers.

The Unmanned Aerial Vehicle (UAV) platform and the manipulator are both highly nonlinear systems, and their combination makes the entire system more complex. In addition, external interference and internal uncertainty make it more difficult to control the system accurately. Some researchers treat the UAV and the manipulator as an entirety, establish a unified model, and design a set of controllers to control all system states directly [5–7]. However, the system developed by this method has high dimensions. For embedded systems such as aerial manipulators, their computing resources are limited, making it hard to realize real-time calculation of the system state. Especially when the load of the manipulator changes or contact force occurs with the external environment, the accuracy of the coupling model established offline is incredibly reduced, which will seriously affect the controller's performance. Therefore, many researchers regard the UAV platform and the manipulator as two separate subsystems and the interaction between them as mutual disturbances, which dramatically reduces the dimensionality of the system model. By designing controllers for each subsystem, the controller's performance can be

effectively improved [8–12]. These studies use multi-rotors as motion platforms to drive manipulators to complete operating tasks. However, the body coordinate system of the multi-rotor changes constantly when it is moving, which causes the coordinate system of the manipulator to change with the body coordinate system constantly. Continuous coordinate transformation will seriously affect the accuracy of control. Therefore, a more stable contact method and robust controller are needed to achieve high-precision contact force operations.

In this paper, we propose an adaptive robust control strategy based on impedance control for the aerial manipulator to exert the contact force accurately. The main contributions can be summarized as follows:

1. By discussing the characteristics of aerial operations, a new operating mode for aerial manipulators is designed, in which the UAV platform only provides a fixed fulcrum in the air, and the manipulator works as the fixed manipulator.
2. A nonsingular global fast terminal sliding mode control (NGFTSMC) method is proposed, which can quickly converge the state error in a limited time to achieve accurate trajectory tracking of UAVs and robotic arms. Furthermore, to make the system resistant to the model's uncertainty, RBFNN is used to estimate the part of the controller that contains the model information.
3. A double-layer control structure is designed to achieve compliant contact between the manipulator and the target and reduce the impact of contact force on the stability of the aerial manipulation system. The outer layer impedance control converts the expected contact force into the desired position trajectory, and the inner layer position control realizes the trajectory tracking of the desired position.

The rest of this article is organized as follows. Section 2 provides an overview of related work. Section 3 describes the dynamics model of the UAV platform and the manipulator, respectively. Section 4 designs robust sliding mode controllers based on adaptive RBFNN for both subsystems and an impedance controller for the compliant contact of the aerial manipulator. The simulation and experimental results and analysis are shown in Sections 5 and 6. Finally, Section 7 gives the conclusion.

2. Related Work

The fixed base in the air dramatically expands the operating range of the manipulator, but the key to achieving an aerial fixed base is precise trajectory tracking. The multirotor is an underactuated, strongly coupled nonlinear system, and its sensitivity to internal uncertainty and external disturbances is essential to its stability. Moreover, the manipulator's movement and the target's reaction force also seriously affect the stability of the multirotor. Designing effectively robust controllers to fix these problems poses a significant challenge to researchers.

Many studies have shown that the sliding mode controller is one of the most effectively robust control techniques. Almakhles DJ designed a double-loop control structure and proposed a controller with integral sliding mode and backstepping sliding mode methods to ensure the position trajectory tracking capability under uncertainties [13]. Labbadi M improved proportional integral differential sliding mode control with the super-twist algorithm, providing good robustness against the time-varying external disturbances, solving the chattering issue, and avoiding discontinuousness of input signals [14]. Wang X proposed a sensor-based Incremental Nonlinear Dynamic Inversion Sliding Mode Control driven by Sliding Mode Disturbance Observers, which can simultaneously reduce the model dependency of the controller and the uncertainties in the closed-loop system [15]. However, the traditional sliding mode control method cannot guarantee the rapid convergence of trajectory tracking within a limited time, which is an enormous disadvantage for real-time systems. Therefore, terminal sliding mode control methods were proposed to solve this problem. Labbadi M designed a double-loop control strategy, where the outer layer utilizes a robust adaptive back-stepping controller to control the position, and the inner layer uses a controller that combined back-stepping and fast terminal sliding modes to

control the attitude [16]. Nekoukar V suggested a new robust flight control system consisting of an adaptive fuzzy terminal sliding mode controller and two proportional-derivative controllers to guarantee flight stability and efficient tracking of pre-defined paths [17]. Tripathi VK designed a finite-time super-twisting sliding mode control scheme to assure finite-time convergence of tracking error with chattering attenuation and developed a high-order sliding mode observer to determine unknown bounded lumped disturbances acting on the quadrotor [18]. In [19], he also proposed an adaptive fast terminal sliding-mode controller with a power ratio proportional reaching law, which can track the position and altitude of a quadrotor with parametric uncertainties and external disturbances. Razmjooei H presented a time-varying chattering-free disturbance observer-based position tracking control law of serial robotic manipulators to track a reference signal in a finite time and employ a positive-increasing function associated with the control/observer objectives to improve the control performance [20].

The design of the above controllers depends on the system model. Nevertheless, the parameters of the system model often change during actual operation, which will seriously deteriorate the performance of the controller. The Radial Basis Function Neural Network (RBFNN) is a special type of neural network structure that can design an adaptive rate based on the Lyapunov stability theorem. It has strong self-learning and nonlinear mapping abilities and can approximate any nonlinear function. Therefore, many controllers are designed based on RBFNN to eliminate their dependence on the system model and compensate for uncertain signals to obtain better trajectory tracking effects. Luo H developed a robust double-loop control scheme for quadrotor speed and attitude tracking, where the outer loop was designed with a coupling controller to ensure velocity tracking based on the stability of altitude tracking control and the inner loop to control the attitude by utilizing the RBFNN to compensate for model disturbance uncertainty [21]. Tao M proposed a singularity-free terminal sliding mode control scheme improved by RBFNN and Extended State Observer (ESO) that does not require prior knowledge about unknown nonlinear uncertainties and external disturbances for quadrotor UAVs [22]. Zhang Q proposed a terminal sliding mode attitude controller based on the RBFNN uncertainty compensator and optimized it via a particle swarm intelligence algorithm with two fitness functions, solving the non-unique actuator action caused by over-actuation of the tilt-rotor quadcopter [23]. Huang S proposed a non-singular fast terminal sliding mode controller for trajectory tracking control of the quadrotor, with a disturbance observer to estimate the external interference and a neural network approximator to develop an online estimate of the model uncertainty [24].

The manipulator is the operational part of the aerial manipulation system. Environmental constraints restrict the movement of the manipulator and generate contact forces. To reduce the impact of the target's reaction force on the entire aerial manipulation system, the contact between the manipulator and the target should be compliant. Compliance control has been widely used in robotics, such as welding [25], polishing [26], and human-computer interaction [27]. Traditional compliance control methods include force/position hybrid control [28] and impedance control [29]. In actual environments, the contact between the manipulator and the target has problems of low absolute accuracy and lack of real pose information, which makes force/position hybrid control hard to achieve. Nonetheless, the impedance control theory applies the manipulator's motion trajectory and contact force to a dynamic framework, so the contact force can be controlled by following the position trajectory. Therefore, the manipulator also needs to design a high-precision trajectory-tracking controller.

The relevant theories for designing trajectory-tracking controllers for quadrotors are also widely used in manipulators. Tran DT proposed a non-singular fast terminal sliding mode controller for manipulator position tracking and used adaptive RBFNN to approximate and eliminate uncertainties [30]. Zhang W utilized the fractional-order method in the proposed non-singular fast terminal sliding mode controller to improve the tracking performance of the controller and designed RBFNN to approximate the unknown

nonlinear function of the system to accomplish model-free control [31]. Jie W proposed a fast fractional-order terminal sliding mode controller with a new perturbation estimator that applied the data-driven method RBFNN to compensate for the estimation error in the conventional sliding perturbation observer to improve the tracking accuracy and reduce the chattering [32]. Kim SJ proposed an adaptive robust RBFNN non-singular terminal sliding mode controller to reduce swinging in the snake robot's head where the RBFNN compensates for interference and an adaptive robust term to make up for the shortcomings of neural network control to eliminate system chattering [33].

This study designed a new contact force control strategy based on the aerial manipulation system composed of a multirotor UAV and a multi-joint manipulator. First, to provide a fixed aerial platform for the manipulator, a non-singular global fast terminal sliding mode controller based on RBFNN was designed for accurate trajectory tracking. Then, a double-loop control scheme was designed for the compliant contact between the manipulator and the target. The outer loop used a position-based impedance controller to generate the desired trajectory, and the inner loop proposed a trajectory-tracking control algorithm based on the design concept of the UAV platform controller.

3. System Modeling

The aerial manipulation system consists of a multirotor UAV and an n-link manipulator. The system structure is shown in Figure 1.

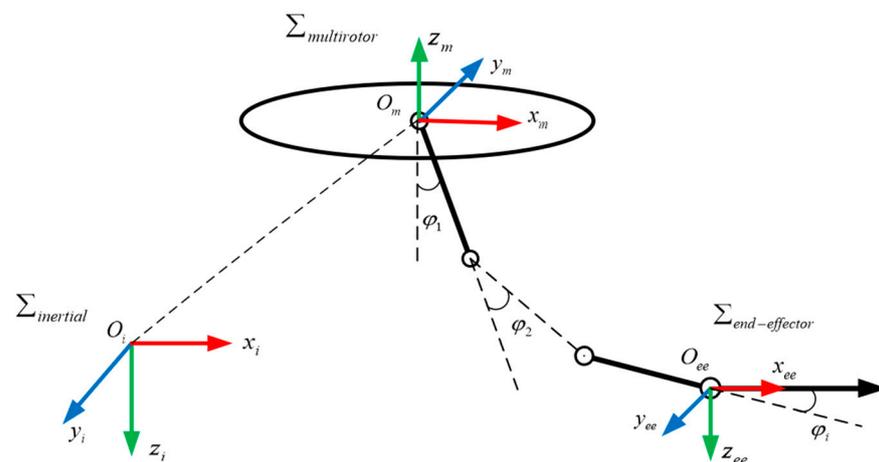


Figure 1. Aerial manipulation system structure diagram.

In Figure 1, $\Sigma_{inertial}$ represents the inertial coordinate system with the origin at O_i , $\Sigma_{multirotor}$ represents the multirotor body coordinate system with the origin at O_m , $\Sigma_{end-effector}$ represents the end-effector coordinate system with the origin at O_{ee} , and φ_i is the angle of rotation of the i -th link manipulator.

To simplify the system structure, we assume:

- (1) The body of the multirotor and manipulator is rigid, and the structure is symmetrical.
- (2) The masses of the multirotor and manipulator are evenly distributed, and the centers of mass coincide with the geometric centers.

Combining the multirotor and manipulator increases the control dimension of the entire system. For embedded systems such as aerial manipulators, real-time high-dimensional matrix transformation calculation is hard to achieve. Therefore, this study regards the multirotor and the manipulator as two independent subsystems. The multirotor subsystem is only used as a fixed aerial platform, while the manipulator subsystem is only used as the operating tool. The forces between them are regarded as mutual disturbances. Based on this setting, we built the models of two subsystems separately.

3.1. Multirotor Platform Modeling

Assuming that the multirotor is rigid, its dynamic equation can be described using the Newton–Euler formula as follows:

$$\begin{cases} m\ddot{\mathbf{P}} + mg\mathbf{e}_3 = \mathbf{u}_P + \boldsymbol{\tau}_P \\ \mathbb{J}\ddot{\boldsymbol{\Phi}} + \mathbb{C}(\boldsymbol{\Phi}, \dot{\boldsymbol{\Phi}})\dot{\boldsymbol{\Phi}} = \mathbf{u}_\Phi + \boldsymbol{\tau}_\Phi \end{cases} \quad (1)$$

where $\mathbf{P} = [x, y, z]^T$ and $\boldsymbol{\Phi} = [\phi, \theta, \psi]^T$ represent the position and Euler angle of the multirotor in the inertial coordinate system, respectively; m is the mass of the entire aerial manipulator; g is the acceleration of gravity; $\mathbf{e}_3 = [0, 0, 1]^T$ is the unit vector along the $O_i z_i$ axis in the inertial coordinate system; \mathbb{J} and \mathbb{C} represent the inertia and coriolis matrix in the inertial coordinate system; \mathbf{u}_P and $\boldsymbol{\tau}_P$ represent the thrust force and disturbance generated by the multirotor in the translational direction; and \mathbf{u}_Φ and $\boldsymbol{\tau}_\Phi$ represent the torque and disturbance received in the rotational direction.

The rotation matrix \mathbf{R}_m^i from the multirotor body coordinate system $\Sigma_{\text{multirotor}}$ to the inertial coordinate system Σ_{inertial} can be expressed as

$$\mathbf{R}_m^i = \begin{bmatrix} \cos\theta\cos\psi & \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi \\ \cos\theta\sin\psi & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi \\ -\sin\theta & \sin\phi\cos\theta & \cos\phi\cos\theta \end{bmatrix} \quad (2)$$

The movement process of aerial manipulators aiming at accomplishing precise contact operations is usually performed within a restricted range. This type of operation requires high stability of the multirotor platform but low requirements for its maneuverability. Therefore, when the multirotor UAV runs at a low speed and small angle, Equation (1) can be rewritten through simplification and decoupling as

$$\begin{cases} \ddot{x} = (\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi)\frac{u_1}{m} + d_x \\ \ddot{y} = (\cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi)\frac{u_1}{m} + d_y \\ \ddot{z} = \cos\phi\cos\theta\frac{u_1}{m} - g + d_z \end{cases} \quad (3)$$

$$\begin{cases} \ddot{\phi} = \left(\frac{I_{yy} - I_{zz}}{I_{xx}}\right)\dot{\theta}\dot{\psi} - \frac{J}{I_{xx}}\dot{\theta}\Omega + \frac{u_2}{I_{xx}} + d_\phi \\ \ddot{\theta} = \left(\frac{I_{zz} - I_{xx}}{I_{yy}}\right)\dot{\phi}\dot{\psi} - \frac{J}{I_{yy}}\dot{\phi}\Omega + \frac{u_3}{I_{yy}} + d_\theta \\ \ddot{\psi} = \left(\frac{I_{xx} - I_{yy}}{I_{zz}}\right)\dot{\phi}\dot{\theta} + \frac{u_4}{I_{zz}} + d_\psi \end{cases} \quad (4)$$

where I_i ($i = xx, yy, zz$) represents the inertia matrix of the aerial manipulator; J_m represents the rotational inertia of the multirotor motor; Ω represents the total remaining speed of the multirotor motor; u_i ($i = 1, 2, 3, 4$) is the control input of the multirotor; and d_i ($i = x, y, z, \phi, \theta, \psi$) represents the lumped disturbance caused by the internal uncertainty, the external environment, and the action of the manipulator.

3.2. Manipulator Modeling

According to Newton–Euler's theorem, the Cartesian dynamics equation of an n-DOF (Degree of Freedom) rigid robot manipulator system can be expressed as

$$\mathcal{B}(\boldsymbol{\varphi})\ddot{\boldsymbol{\varphi}} + \mathcal{H}(\boldsymbol{\varphi}, \dot{\boldsymbol{\varphi}})\dot{\boldsymbol{\varphi}} + \mathcal{V}(\boldsymbol{\varphi}) + \mathcal{T}_c + \mathcal{T}_d = \boldsymbol{\tau} \quad (5)$$

where $\boldsymbol{\varphi} = [\varphi_1 \ \varphi_2 \ \cdots \ \varphi_i]^T$ is the joint angular vector of the i -link manipulator, $\dot{\boldsymbol{\varphi}}, \ddot{\boldsymbol{\varphi}}$ represent the joint angular velocity vector and acceleration vector of the manipulator, $\mathcal{B}(\boldsymbol{\varphi})$ is the symmetric positive definite inertia matrix, $\mathcal{H}(\boldsymbol{\varphi}, \dot{\boldsymbol{\varphi}})$ is the centrifugal force and coriolis force matrix, $\mathcal{V}(\boldsymbol{\varphi})$ is the gravity matrix, $\boldsymbol{\tau}$ is the joint torque control input vector, \mathcal{T}_c is the torque vector of the interaction between the manipulator and the target, and \mathcal{T}_d is the lumped disturbance of the manipulator.

Let η be the position vector of the end effector in the task space. Joint space can be mapped to task space through forward kinematics as follows:

$$\eta = \text{Map}(\varphi) \quad (6)$$

where $\text{Map}(\varphi)$ is the mapping relationship from the joint space to the task space. Therefore, the relationship between the velocity in joint space and the velocity in task space can be described as

$$\dot{\eta} = \mathcal{J}(\varphi)\dot{\varphi} \quad (7)$$

where the Jacobian matrix $\mathcal{J}(\varphi)$ represents the relationship between the virtual end effector speed and the virtual joint speed. By differentiating Equation (7), the acceleration term of the end effector is determined as

$$\ddot{\eta} = \mathcal{J}(\varphi)\ddot{\varphi} + \dot{\mathcal{J}}(\varphi)\dot{\varphi} \quad (8)$$

Then, the dynamics of the manipulator in the task space can be expressed as

$$\begin{aligned} \mathcal{B}_\eta(\varphi)\ddot{\eta} + \mathcal{H}_\eta(\varphi, \dot{\varphi})\dot{\eta} + \mathcal{V}_\eta(\varphi) + \mathcal{F}_c + \mathcal{F}_d &= \mathcal{F}_\eta \\ \mathcal{B}_\eta(\varphi) &= \mathcal{J}^{-T}(\varphi)\mathcal{B}(\varphi)\mathcal{J}^{-1}(\varphi) \\ \mathcal{H}_\eta(\varphi, \dot{\varphi}) &= \mathcal{J}^{-T}(\varphi)\left(\mathcal{H}(\varphi, \dot{\varphi}) - \mathcal{B}(\varphi)\mathcal{J}^{-1}(\varphi)\dot{\mathcal{J}}(\varphi)\right)\mathcal{J}^{-1}(\varphi) \\ \mathcal{V}_\eta(\varphi) &= \mathcal{J}^{-T}(\varphi)\mathcal{V}(\varphi) \\ \mathcal{F}_\eta &= \mathcal{J}^{-T}(\varphi)\mathcal{T} \\ \mathcal{F}_c &= \mathcal{J}^{-T}(\varphi)\mathcal{T}_c \\ \mathcal{F}_d &= \mathcal{J}^{-T}(\varphi)\mathcal{T}_d \end{aligned} \quad (9)$$

3.3. Aerial Platform Attitude Compensation

The contact force between the manipulator and the target is generated by the torque of the manipulator motor, and the corresponding counter-torque acts on the aerial platform and is offset by the thrust generated by the multirotor. Since the lift of the propeller is the only source of power for the aerial manipulator in the air, it can be considered that the contact force is generated by the thrust of the multirotor. Therefore, in order for the manipulator to exert a stable contact force on the target, the multirotor platform should always stay hovering. According to Equation (3), disregarding the influence of disturbance, the equilibrium relationship of contact force exerted in the hovering state can be obtained as

$$\begin{cases} (\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi)u_1 - \mathcal{F}_{c-x} = 0 \\ (\cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi)u_1 - \mathcal{F}_{c-y} = 0 \\ \cos\phi\cos\theta u_1 - mg - \mathcal{F}_{c-z} = 0 \end{cases} \quad (10)$$

In this study, the manipulator generates contact force with the vertical target in a point contact mode, so $\mathcal{F}_{c-y} = 0$ and $\mathcal{F}_{c-z} = 0$. We can set the yaw angle $\psi = 0$; then according to Equation (10), the contact force \mathcal{F}_{c-x} can be calculated as

$$\mathcal{F}_{c-x} = mg \tan \theta \quad (11)$$

It can be seen that \mathcal{F}_{c-x} is determined by the gravity mg and pitch angle θ of the quadrotor. Since the gravity of the aerial manipulator is constant, the value of \mathcal{F}_{c-x} corresponds to the pitch angle θ one-to-one. Therefore, when the desired contact force is $\mathcal{F}_{desired}$, the desired pitch angle $\theta_{desired}$ of the multirotor can be calculated as

$$\theta_{desired} = \arctan\left(\frac{\mathcal{F}_{desired}}{mg}\right) \quad (12)$$

As the aerial platform for the manipulator, changes in the attitude of the multirotor will change the relationship between the manipulator coordinate system and the inertial

coordinate system. To prevent the kinematic models of the two subsystems from interfering with each other, the changes in the attitude of the multirotor should be compensated by the corresponding angle of the manipulator joint rotation so that the multirotor can be regarded as a fixed platform in the air. The above analysis is illustrated in Figure 2.

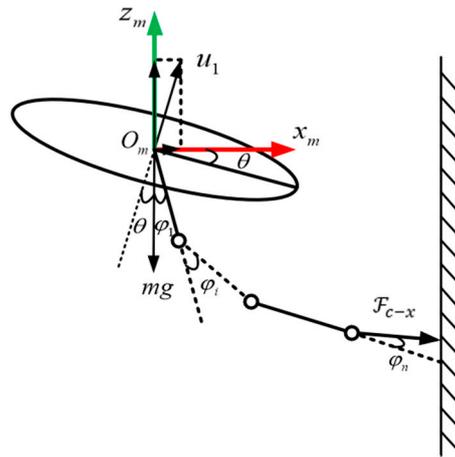


Figure 2. Analysis of contact between aerial manipulator and target.

4. Controller Design and Stability Analysis

In this part, we first designed a robust trajectory-tracking controller for the multirotor and used RBFNN to eliminate the controller's dependence on the system model. Then, to achieve the compliance contact of the manipulator, an impedance position controller was devised to obtain the manipulator's desired trajectory. Finally, bringing in the multirotor controller's design theories, the manipulator's trajectory-tracking controller was developed.

4.1. Multirotor Platform Controller Design

Taking the height controller as an example, the height error is defined as

$$e = z - z_d \quad (13)$$

where z_d is the desired height. We can design the nonsingular fast terminal sliding mode surface as

$$s = \dot{e} + \alpha_1 e + \beta_1 |e|^{\mu_1/v_1} \text{sgn}(e) \quad (14)$$

where $\alpha_1 > 0$, $\beta_1 > 0$, μ_1 and v_1 are positive odd numbers and satisfy $\mu_1 < v_1$. $\text{sgn}(e)$ is the switching function of the error.

$$\text{sgn}(e) = \begin{cases} 1, & e > 0 \\ 0, & e = 0 \\ -1, & e < 0 \end{cases} \quad (15)$$

Taking the time derivative of Equation (14),

$$\dot{s} = \ddot{e} + \alpha_1 \dot{e} + \beta_1 \frac{\mu_1}{v_1} |e|^{(\mu_1-v_1)/v_1} \dot{e} \text{sgn}(e) \quad (16)$$

Equation (11) can be rewritten as follows by substituting Equation (3):

$$\dot{s} = \frac{\cos\phi\cos\theta}{m} u_1 - g + \ddot{z}_d - \ddot{z}_d + \alpha_1 \dot{e} + \beta_1 \frac{\mu_1}{v_1} |e|^{(\mu_1-v_1)/v_1} \dot{e} \text{sgn}(e) \quad (17)$$

In order to promote the system state to reach the sliding surface during the entire approximation process quickly, the fast arrival law is proposed as

$$\dot{s} = -\alpha_2 |s|^{\mu_2} \text{sgn}(s) - \beta_2 |s|^{v_2} \text{sgn}(s) \tag{18}$$

where $\alpha_2 > 0, \beta_2 > 0, 0 < \mu_2 < 1$ and $v_2 > 1$. According to Equations (17) and (18), the controller u_1 is designed as

$$u_1 = \frac{m}{\cos\phi\cos\theta} (\ddot{z}_d + g - \alpha_1 \dot{e} - \beta_1 \frac{\mu_1}{v_1} |e|^{(\mu_1-v_1)/v_1} \dot{e} \text{sgn}(e) - \alpha_2 |s|^{\mu_2} \text{sgn}(s) - \beta_2 |s|^{v_2} \text{sgn}(s) - D_z \text{sgn}(s)) \tag{19}$$

where $D_z > |d_z|$ is the robust term of the controller; its function is to overcome the influence of external disturbance on the system trajectory and guide it to move to the sliding mode surface. We can define the Lyapunov function as

$$V_1 = \frac{1}{2} s^2 \tag{20}$$

Taking the time derivative of Equation (20) and substituting Equation (17)

$$\dot{V}_1 = s\dot{s} = s \left(\frac{\cos\phi\cos\theta}{m} u_1 - g + d_z - \ddot{z}_d + \alpha_1 \dot{e} + \beta_1 \frac{\mu_1}{v_1} |e|^{(\mu_1-v_1)/v_1} \dot{e} \text{sgn}(e) \right) \tag{21}$$

Substituting Equation (19) into Equation (21), \dot{V}_1 can be deduced as

$$\begin{aligned} \dot{V}_1 &= -\alpha_2 |s|^{\mu_2+1} - \beta_2 |s|^{v_2+1} - s D_z \text{sgn}(s) + d_z s \\ &= -\alpha_2 |s|^{\mu_2+1} - \beta_2 |s|^{v_2+1} - (D_z |s| - d_z s) \end{aligned} \tag{22}$$

When $D_z \geq |d_z|, \dot{V}_1 \leq 0$, the system converges stably. According to the sliding mode control theory, the equivalent part of the designed controller is

$$u_{eq} = \frac{m}{\cos\phi\cos\theta} \left(\ddot{z}_d + g - \alpha_1 \dot{e} - \beta_1 \frac{\mu_1}{v_1} |e|^{(\mu_1-v_1)/v_1} \dot{e} \text{sgn}(e) \right) \tag{23}$$

The equivalent controller is related to the system model. In order to eliminate the impact of model uncertainty on the controller, we introduce the RBF neural network to estimate the equivalent controller. We can define the input of the RBF neural network as $\mathbf{X} = (e_z \dot{e}_z z_d \dot{z}_d \ddot{z}_d)^T$, and the equivalent control in Equation (23) is the ideal RBF neural network output, which can be expressed as

$$u_{eq} = \mathbf{W}^T \mathbf{h}(\mathbf{X}) + \varepsilon \tag{24}$$

where \mathbf{W} is the ideal weight of the neural network, ε is a small positive real number that represents the approximation error of the neural network to the nonlinear uncertain function, and $\mathbf{h}(\mathbf{X})$ is the Gaussian function that nonlinear mapping of the neural network, which can be expressed as

$$h_i(\mathbf{X}) = \exp\left(-\frac{\|\mathbf{X} - \mathbf{c}_j\|^2}{b_j^2}\right), j = 1, 2, \dots, \aleph \tag{25}$$

where \mathbf{c}_j and b_j are the center value and width of the Gaussian function, and \aleph is the number of neurons in the hidden layer of the network. We can let \hat{u}_{eq} be the approximation output of the RBF neural network to the equivalent controller u_{eq} :

$$\hat{u}_{eq} = \hat{\mathbf{W}}^T \mathbf{h}(\mathbf{X}) \tag{26}$$

nipulator and the target will significantly impact the stability of the entire system, making the contact force uncontrollable. In order to make the contact between the manipulator and the target compliant and the contact force change smoothly, this paper proposes a double-loop control structure. The outer loop is position-based impedance control, which converts the desired contact force into the desired position trajectory. The inner loop is a trajectory-tracking controller that accurately tracks the position trajectory generated by the outer loop.

4.2.1. Outer Loop Position-Based Impedance Control

The contact procedure of the manipulator and the target is divided into two stages. The first stage involves the manipulator approaching the target, which is free space control. The desired position is the target position, where the flying robotic arm comes into contact with the environment. The second stage is manipulator contact with the target, which is restricted space control, and the desired position is generated by the desired force. To reduce contact impact on the system, an additional impedance control loop is added beside the position control. Based on the impedance relationship model, impedance control has the advantage of having force and position in the same framework, and the relationship between them can be adjusted by changing the impedance parameters. Commonly, the mathematical model of the impedance relationship can be expressed in terms of a second-order differential equation.

$$\mathcal{M}\ddot{E} + \mathcal{B}\dot{E} + \mathcal{K}E = \mathcal{F}_t \quad (33)$$

where \mathcal{M} , \mathcal{B} , \mathcal{K} are the required inertia, damping and stiffness matrices, respectively, and \mathcal{F}_t is the contact force that is exerted by the manipulator on the target measured by the sensor. E is the error between the planned reference trajectory η_r and the expected trajectory η_d calculated based on the contact force, which can be obtained from the Laplace transform of Equation (33) as

$$E(s) = \frac{1}{\mathcal{M}s^2 + \mathcal{B}s + \mathcal{K}} \mathcal{F}_t(s) \quad (34)$$

During the two stages of the contact process, the expected trajectory of the inner position control loop is

$$\eta_d = \begin{cases} \eta_t & (\text{free space}) \\ \eta_r - E & (\text{contact space}) \end{cases} \quad (35)$$

The reference trajectory η_r is determined by the position and parameters of the manipulator, the impedance controller parameters, and the desired contact force \mathcal{F}_Ξ . Equation (33) can be rewritten as

$$\mathcal{M}(\ddot{\eta}_r - \ddot{\eta}) + \mathcal{B}(\dot{\eta}_r - \dot{\eta}) + \mathcal{K}(\eta_r - \eta) = \mathcal{F}_\Xi \quad (36)$$

The contact force is usually determined by the stiffness and damping parameters of the target, and it can be described by the following second-order nonlinear function.

$$\mathcal{F}_\Xi = \mathcal{K}_t(\eta - \eta_t) + \mathcal{B}_t\dot{\eta} \quad (37)$$

where \mathcal{K}_t and \mathcal{B}_t are the target's diagonal positive definite stiffness matrices and damping matrices, respectively. Taking the contact force in a single direction as an example, Equations (36) and (37) can be rewritten as

$$\begin{cases} m(\ddot{\delta}_r - \ddot{\delta}) + \ell(\dot{\delta}_r - \dot{\delta}) + \mathcal{k}(\delta_r - \delta) = \mathcal{f}_\Xi \\ \mathcal{f}_\Xi = \mathcal{k}_t(\delta - \delta_t) + \mathcal{b}_t\dot{\delta} \end{cases} \quad (38)$$

Equation (38) can be expressed as follows by computing Laplace transforms:

$$\begin{cases} (ms^2 + \ell s + \mathcal{k})(\delta_r(s) - \delta(s)) = \mathcal{f}_\Xi(s) \\ \mathcal{f}_\Xi(s) = (\mathcal{b}_t s + \mathcal{k}_t)\delta(s) - \frac{\mathcal{k}_t}{s}\delta_t \end{cases} \quad (39)$$

According to Equation (39), the planned reference position trajectory can be derived as

$$\delta_r(s) = \frac{ms^2 + \ell s + \ell + \ell_t s + \ell_t}{(ms^2 + \ell s + \ell)(\ell_t s + \ell_t)} f_{\Xi}(s) + \frac{\ell_t \delta_t}{s(\ell_t s + \ell_t)} \quad (40)$$

4.2.2. Inner Loop Trajectory Tracking Control

In position-based impedance control, the force tracking performance depends on the accuracy of the inner loop position control. Here, we use the same robust control strategy as the multirotor platform and define the position error of the manipulator as

$$\boldsymbol{\vartheta} = \boldsymbol{\eta}_d - \boldsymbol{\eta} \quad (41)$$

We can design the nonsingular fast terminal sliding mode surface as

$$\boldsymbol{r} = \dot{\boldsymbol{\vartheta}} + \lambda_1 \boldsymbol{\vartheta} + \boldsymbol{\rho}_1 \text{diag}\left(|\boldsymbol{\vartheta}|^{\zeta_1/\varsigma_1}\right) \text{sgn}(\boldsymbol{\vartheta}) \quad (42)$$

where λ_1 and $\boldsymbol{\rho}_1$ are positive definite diagonal matrices, and ζ_1 and ς_1 are positive odd numbers and satisfy $\zeta_1 < \varsigma_1$. Taking the time derivative of Equation (42) results in

$$\begin{aligned} \mathcal{B}_{\eta} \dot{\boldsymbol{r}} &= \mathcal{B}_{\eta} \left(\ddot{\boldsymbol{\vartheta}} + \lambda_1 \dot{\boldsymbol{\vartheta}} + \frac{\zeta_1}{\varsigma_1} \boldsymbol{\rho}_1 \text{diag}\left(|\boldsymbol{\vartheta}|^{(\zeta_1-\varsigma_1)/\varsigma_1}\right) \text{diag}\left(\dot{\boldsymbol{\vartheta}}\right) \text{sgn}(\boldsymbol{\vartheta}) \right) \\ &= \mathcal{B}_{\eta} \left(\ddot{\boldsymbol{\eta}}_d - \ddot{\boldsymbol{\eta}} + \lambda_1 \dot{\boldsymbol{\vartheta}} + \frac{\zeta_1}{\varsigma_1} \boldsymbol{\rho}_1 \text{diag}\left(|\boldsymbol{\vartheta}|^{(\zeta_1-\varsigma_1)/\varsigma_1}\right) \text{diag}\left(\dot{\boldsymbol{\vartheta}}\right) \text{sgn}(\boldsymbol{\vartheta}) \right) \\ &= \mathcal{B}_{\eta} \left(\ddot{\boldsymbol{\eta}}_d + \lambda_1 \dot{\boldsymbol{\vartheta}} + \frac{\zeta_1}{\varsigma_1} \boldsymbol{\rho}_1 \text{diag}\left(|\boldsymbol{\vartheta}|^{(\zeta_1-\varsigma_1)/\varsigma_1}\right) \text{diag}\left(\dot{\boldsymbol{\vartheta}}\right) \text{sgn}(\boldsymbol{\vartheta}) \right) - \mathcal{B}_{\eta} \ddot{\boldsymbol{\eta}} \\ &= \mathcal{B}_{\eta} \left(\ddot{\boldsymbol{\eta}}_d + \lambda_1 \dot{\boldsymbol{\vartheta}} + \frac{\zeta_1}{\varsigma_1} \boldsymbol{\rho}_1 \text{diag}\left(|\boldsymbol{\vartheta}|^{(\zeta_1-\varsigma_1)/\varsigma_1}\right) \text{diag}\left(\dot{\boldsymbol{\vartheta}}\right) \text{sgn}(\boldsymbol{\vartheta}) \right) + \mathcal{H}_{\eta} \dot{\boldsymbol{\eta}} + \mathcal{V}_{\eta} + \mathcal{F}_c + \mathcal{F}_d - \mathcal{F}_{\eta} \\ &= \mathcal{B}_{\eta} \left(\ddot{\boldsymbol{\eta}}_d - \lambda_1 \dot{\boldsymbol{\vartheta}} - \frac{\zeta_1}{\varsigma_1} \boldsymbol{\rho}_1 \text{diag}\left(|\boldsymbol{\vartheta}|^{(\zeta_1-\varsigma_1)/\varsigma_1}\right) \text{diag}\left(\dot{\boldsymbol{\vartheta}}\right) \text{sgn}(\boldsymbol{\vartheta}) \right) + \mathcal{H}_{\eta} \left(\dot{\boldsymbol{\eta}}_d - \dot{\boldsymbol{\vartheta}} \right) + \mathcal{V}_{\eta} + \mathcal{F}_c + \mathcal{F}_d - \mathcal{F}_{\eta} \\ &= \mathcal{B}_{\eta} \left(\ddot{\boldsymbol{\eta}}_d - \lambda_1 \dot{\boldsymbol{\vartheta}} - \frac{\zeta_1}{\varsigma_1} \boldsymbol{\rho}_1 \text{diag}\left(|\boldsymbol{\vartheta}|^{(\zeta_1-\varsigma_1)/\varsigma_1}\right) \text{diag}\left(\dot{\boldsymbol{\vartheta}}\right) \text{sgn}(\boldsymbol{\vartheta}) \right) \\ &\quad + \mathcal{H}_{\eta} \left(\dot{\boldsymbol{\eta}}_d - \lambda_1 \dot{\boldsymbol{\vartheta}} - \boldsymbol{\rho}_1 \text{diag}\left(|\boldsymbol{\vartheta}|^{\zeta_1/\varsigma_1}\right) \text{sgn}(\boldsymbol{\vartheta}) \right) + \mathcal{V}_{\eta} + \mathcal{F}_c + \mathcal{F}_d - \mathcal{F}_{\eta} - \mathcal{H}_{\eta} \boldsymbol{r} \end{aligned} \quad (43)$$

Let

$$\Delta(\boldsymbol{\eta}_r) = \mathcal{B}_{\eta} \ddot{\boldsymbol{\eta}}_r + \mathcal{H}_{\eta} \dot{\boldsymbol{\eta}}_r + \mathcal{V}_{\eta} + \mathcal{F}_c \quad (44)$$

where $\dot{\boldsymbol{\eta}}_r = \dot{\boldsymbol{\eta}}_d - \lambda_1 \dot{\boldsymbol{\vartheta}} - \boldsymbol{\rho}_1 \text{diag}\left(|\boldsymbol{\vartheta}|^{\zeta_1/\varsigma_1}\right) \text{sgn}(\boldsymbol{\vartheta})$, and Equation (43) can be rewritten as

$$\mathcal{B}_{\eta} \dot{\boldsymbol{r}} = -\mathcal{H}_{\eta} \boldsymbol{r} + \Delta + \mathcal{F}_d - \mathcal{F}_{\eta} \quad (45)$$

Since $\Delta(\boldsymbol{\eta})$ contains all model information, the RBF neural network can be used to approximate $\Delta(\boldsymbol{\eta})$ to design a robust controller that does not require any model information. We can define the input of RBF neural network as $\boldsymbol{X} = (\boldsymbol{\vartheta} \ \dot{\boldsymbol{\vartheta}} \ \boldsymbol{\eta}_d \ \dot{\boldsymbol{\eta}}_d \ \ddot{\boldsymbol{\eta}}_d)^T$; the approximate output of the RBF neural network to the equivalent controller Δ is

$$\hat{\Delta} = \hat{\boldsymbol{W}}^T \boldsymbol{h}(\boldsymbol{X}) \quad (46)$$

The ideal RBF neural network equivalent control output in Equation (45) is defined as

$$\Delta = \boldsymbol{W}^T \boldsymbol{h}(\boldsymbol{X}) + \boldsymbol{\varepsilon} \quad (47)$$

The estimation error of the neural network is

$$\tilde{\Delta} = \Delta - \hat{\Delta} = \boldsymbol{W}^T \boldsymbol{h}(\boldsymbol{X}) + \boldsymbol{\varepsilon} - \hat{\boldsymbol{W}}^T \boldsymbol{h}(\boldsymbol{X}) = \tilde{\boldsymbol{W}}^T \boldsymbol{h}(\boldsymbol{X}) + \boldsymbol{\varepsilon} \quad (48)$$

We can define the nonsingular global fast terminal sliding mode controller as

$$\mathcal{F}_\eta = \hat{\Delta} + \lambda_2 \text{diag}(|r|^{\zeta_2}) \text{sgn}(r) + \rho_2 \text{diag}(|r|^{\zeta_2}) \text{sgn}(r) + (D_\eta + Y_\eta) \text{sgn}(r) \quad (49)$$

where λ_2 and ρ_2 are positive definite diagonal matrices, $0 < \zeta_2 < 1$ is the ratio of two odd integers, and $\zeta_2 > 0$. By substituting Equation (49), Equation (45) can be rewritten as

$$\begin{aligned} \mathcal{B}_\eta \dot{r} &= -\mathcal{H}_\eta r + \Delta + \mathcal{F}_d - \left(\hat{\Delta} + \lambda_2 \text{diag}(|r|^{\zeta_2}) \text{sgn}(r) + \rho_2 \text{diag}(|r|^{\zeta_2}) \text{sgn}(r) + (D_\eta + Y_\eta) \text{sgn}(r) \right) \\ &= -\mathcal{H}_\eta r + \tilde{W}^T h + \varepsilon + \mathcal{F}_d \\ &\quad - \left(\lambda_2 \text{diag}(|r|^{\zeta_2}) \text{sgn}(r) + \rho_2 \text{diag}(|r|^{\zeta_2}) \text{sgn}(r) + (D_\eta + Y_\eta) \text{sgn}(r) \right) \end{aligned} \quad (50)$$

We can define the Lyapunov function as

$$L = \frac{1}{2} r^T \mathcal{B}_\eta r + \frac{1}{2} \text{tr} \left(\tilde{W}^T \Gamma^{-1} \tilde{W} \right) \quad (51)$$

Equation (51) can be rewritten as follows by taking the time derivative:

$$\begin{aligned} \dot{L} &= r^T \mathcal{B}_\eta \dot{r} + \frac{1}{2} r^T \dot{\mathcal{B}}_\eta r + \text{tr} \left(\tilde{W}^T \Gamma^{-1} \dot{\tilde{W}} \right) \\ &= r^T \left(-\mathcal{H}_\eta r + \tilde{W}^T h + \varepsilon + \mathcal{F}_d - \lambda_2 \text{diag}(|r|^{\zeta_2}) \text{sgn}(r) - \rho_2 \text{diag}(|r|^{\zeta_2}) \text{sgn}(r) - (D_\eta + Y_\eta) \text{sgn}(r) \right) + \frac{1}{2} r^T \dot{\mathcal{B}}_\eta r + \text{tr} \left(\tilde{W}^T \Gamma^{-1} \dot{\tilde{W}} \right) \\ &= -\sum_{i=1}^n \lambda_{2i} |r_i|^{\zeta_2+1} - \sum_{i=1}^n \rho_{2i} |r_i|^{\zeta_2+1} + \frac{1}{2} r^T (\dot{\mathcal{B}}_\eta - 2\mathcal{H}_\eta) r - \text{tr} \tilde{W}^T (\Gamma^{-1} \dot{\tilde{W}} - h r^T) \\ &\quad + r^T (\varepsilon + \mathcal{F}_d) - |r|^T (D_\eta + Y_\eta) \end{aligned} \quad (52)$$

The manipulator dynamic model parameters are skewed symmetrically, so

$$r^T (\dot{\mathcal{B}}_\eta - 2\mathcal{H}_\eta) r = 0 \quad (53)$$

We can define RBF neural network adaptive law as

$$\dot{\tilde{W}} = \Gamma h r^T \quad (54)$$

Equation (52) can be written as follows by substituting Equations (53) and (54):

$$\dot{L} = -\sum_{i=1}^n \lambda_{2i} |r_i|^{\zeta_2+1} - \sum_{i=1}^n \rho_{2i} |r_i|^{\zeta_2+1} + r^T (\varepsilon + \mathcal{F}_d) - |r|^T (D_\eta + Y_\eta) \quad (55)$$

When $D_\eta \geq |\mathcal{F}_d|$ and $Y_\eta \geq |\varepsilon|$, $\dot{L} \leq 0$, the closed-loop system is stable. Based on the above analysis, the structure of the manipulator control system is shown in Figure 4.

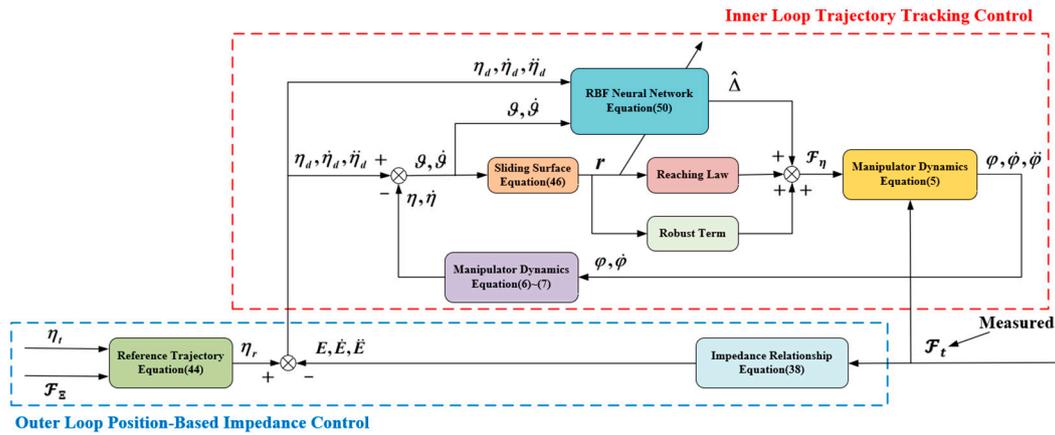


Figure 4. The structure of the manipulator control system.

5. Simulations

In this section, some simulations are performed on the aerial manipulator to verify the effectiveness of the proposed control strategy. These simulations for the quadrotor and manipulator were performed separately in MATLAB/Simulink R2022a software. Since many studies have been completed on ideal conditions, this study mainly works on the situation with random interference.

5.1. Simulation Setting

The aerial manipulator used in the simulation is a quadrotor equipped with a planar 3DOF manipulator. The parameters of the aerial manipulator are $m = 2.32$ kg, $g = 9.8$ m·s⁻², $[I_{xx} \ I_{yy} \ I_{zz}] = [0.032 \ 0.032 \ 0.048]$ kg·m², $l = 0.55$ m, $[m_1 \ m_2 \ m_3] = [0.26 \ 0.26 \ 0.18]$ kg, $[l_1 \ l_2 \ l_3] = [0.25 \ 0.25 \ 0.4]$ m. The gains of each subsystem controller of the aerial manipulator are shown in Table 1.

Table 1. The gains of each subsystem controller of the aerial manipulator.

Subsystem	Parameters	Values
Quadrotor	$\alpha_1, \beta_1, \mu_1, \nu_1$	200, 20, 3, 5
	$\alpha_2, \beta_2, \mu_2, \nu_2$	600, 200, 3, 5
	c	$[-2 \ -1 \ 0 \ 1 \ 2]$
	b	5
	W	$[0 \ 0 \ 0 \ 0 \ 0]$
	ζ	0.4
Manipulator	m, ℓ, k	1, 500, 1000
	k_t, ℓ_t	10,000, 10
	$\lambda_1, \rho_1, \zeta_1, \varsigma_1$	$100I, 20I, 3, 5$
	$\lambda_2, \rho_2, \zeta_2, \varsigma_2$	$20I, 20I, 3, 5$
	Γ	0.01I

In many papers, NGFTSMC without neural networks has been proven to have sound control effects for multi-rotors and manipulators, so the simulation in this study only compares NGFTSMC with and without neural networks. The selection of all parameters was obtained through multiple simulations and can improve the system's performance. To test the robustness of the controller, the system disturbance was set as a random disturbance within $\pm 5\%$ of the nominal model of the system.

5.2. Simulation Results and Analysis

5.2.1. Quadrotor Platform Subsystem

The initial position and Euler angle of the quadrotor subsystem are $[0 \ 0 \ 0]$ m and $[0 \ 0 \ 0]$ rad, and the desired position and Euler angle are set to $[2 \ 2 \ 4]$ m and

$[0 \ 0 \ 0]$ rad. In 0 to 5 s, the quadrotor flies to the destination and remains hovering after 5 s. The manipulator moves to the target in 5 to 10 s and gradually forms a contact force 5 N with the target in 10 to 15 s. In 15 to 20 s, the contact force remains unchanged. Since there is force contact between the manipulator and the target from 10 to 20 s, the desired pitch angle of the quadrotor can be calculated by Equation (12) and the desired contact force.

Figure 5 shows the tracking error of the quadrotor platform in the X-Y-Z axis. The system states controlled by both controllers can track the desired trajectory, but the performance on the X-Y axis is slightly different from that on the Z-axis. In Figure 5a,b, the error of the system state controlled by NGFTSMC based on RBFNN is about 70% that of regular NGFTSMC. Nevertheless, there is a particular time point that at 5 s, the system state error of NGFTSMC control based on RBFNN has a significant overshoot. The overshoot value on the X-axis is 5.7×10^{-4} , and the overshoot value on the Y-axis is 5.8×10^{-4} . At the same time, the error controlled by regular NGFTSMC has no obvious abnormality change at 5 s. In Figure 5c, the performance of both controllers on the Z-axis is more stable. The error of the system state controlled by RBFNN-based NGFTSMC is 30% that of regular NGFTSMC, and there is no overshoot in both performances.

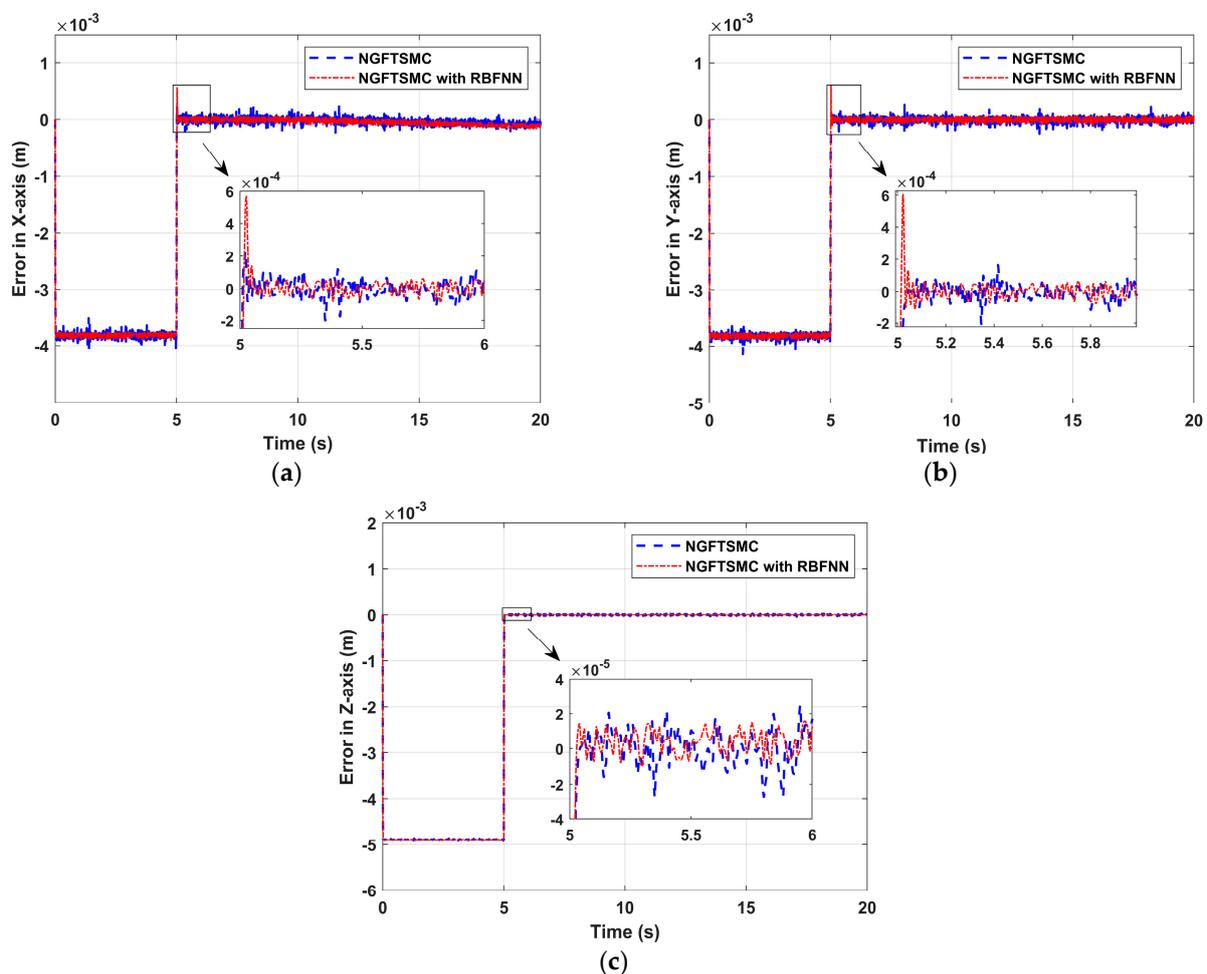


Figure 5. The position tracking performance of the quadrotor platform: (a) Error in X-axis; (b) Error in Y-axis; (c) Error in Z-axis.

Figure 6 depicts the variation of the system Euler angle tracking error over time. As shown in Figure 6a,b, the overshoot of the state error controlled by NGFTSMC based on RBFNN is about 25% that of the regular NGFTSMC. Moreover, corresponding to Figure 5a,b, the state error controlled by the proposed NGFTSMC also has a remarkable overshoot at 5 s. This is because the roll angle and pitch angle are coupled with the motion system in the

Y-axis and X-axis, respectively, so their error shift patterns should be similar to the error trajectories in the Y-axis and X-axis, respectively. The yaw angle is not coupled with other states, so its control accuracy is better than the others. As shown in Figure 6c, the yaw error of the NGFTSMC-controlled system based on RBFNN is 0.3×10^{-5} , which is only about 20% of the regular NGFTSMC system.

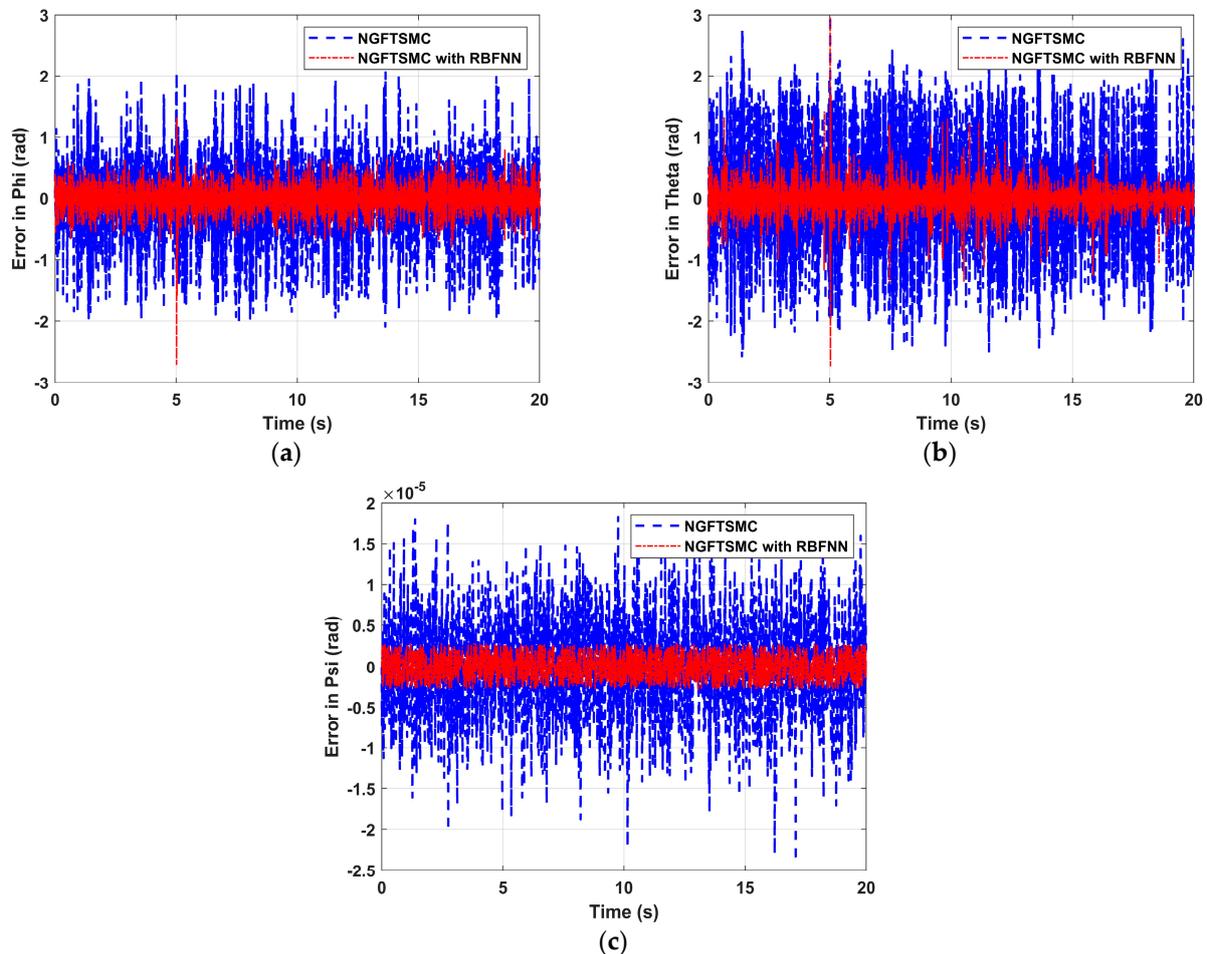


Figure 6. The attitude tracking performance of the quadrotor platform: (a) Error in Phi; (b) Error in Theta; (c) Error in Psi.

Figure 7 depicts the variation in the output torque of the quadrotor platform over time. It can be seen that the torques controlled by NGFTSMC based on RBFNN are smaller than those of the regular NGFTSMC, which means the rotation speed of the motor is slower. Decreasing rotational speed means the motor is more accessible to implement and mechanical loss is less, which is crucial for a mechanical system.

5.2.2. Manipulator Subsystem

The initial state of the manipulator subsystem is $[0 \ 0 \ 0]$ rad. According to the target position and inverse kinematics, the desired state of the manipulator can be calculated as $[0.46 \ 0.79 \ 0.31]$ rad. First, the manipulator moves from the initial posture to the desired posture in 0 to 5 s. Then, the manipulator gradually forms a contact force 5 N with the target in 5 to 10 s. Finally, the contact force is maintained for 10 to 15 s.

Figure 8 illustrates the tracking error results of the manipulator joints 1, 2, and 3 moving along the desired trajectory. It can be seen in the figure that the errors controlled by the NGFTSMC based on RBFNN are more miniature than those of the regular NGFTSMC, with a general error of about 50% and a peak error of less than 30%. Meanwhile, the manipulator is connected in a series, and the error of the joint closer to the manipulator

base is minor. According to the analysis of system operation, this is consistent with the situation of the actual system.

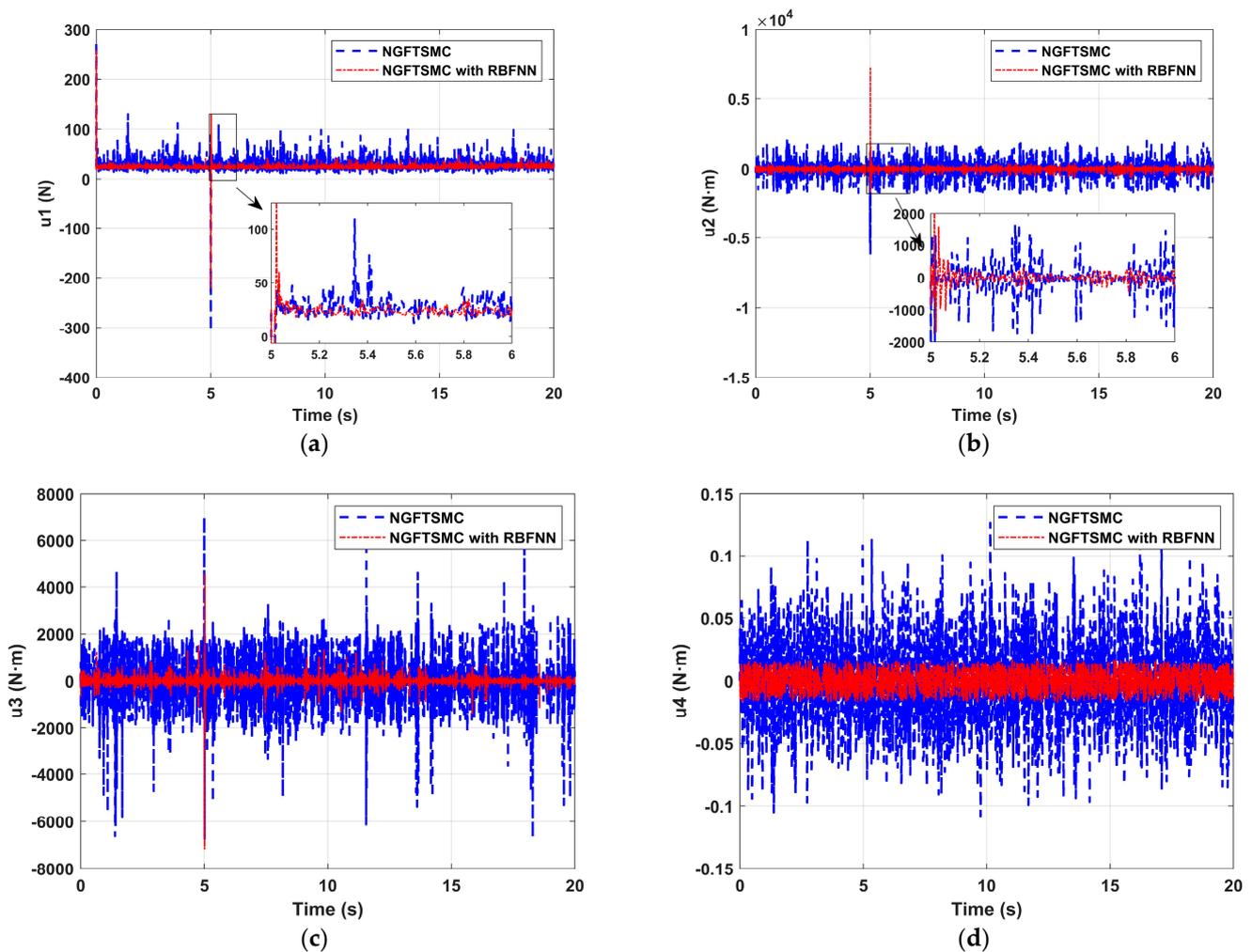


Figure 7. The control input performance of the quadrotor platform: (a) u_1 ; (b) u_2 ; (c) u_3 ; (d) u_4 .

Figure 9 depicts the variation in the position error of the manipulator with time. In Figure 9a, the state error of the approaching phase is more considerable than that of the contact phase, and the error of GFTSMC based on RBFNN is about 30% that of regular GFTSMC. The contact force only exists on the X-axis, so after contacting the target, the smaller error in the X-axis means more accurate force tracking. In Figure 9b, the error gradually increases from 0 to 5 s, while the error is relatively stable within 5 to 15 s, and the error of NGFTSMC based on RBFNN is about 50% that of regular GFTSMC. Since the manipulator has no contact force in the Z-axis and is not constrained, it can be seen that the state error in the Z-axis is significantly greater than the state error in the X-axis.

Figure 10 portrays the contact force error between the manipulator and the target over time. It can be seen that during the contact process of 5 to 15 s, the force-tracking capability of the NGFTSMC based on RBFNN is significantly better than that of the regular NGFTSMC, and the error is about 0.02 N, which is consistent with the performance of the position error in the X-axis. Therefore, the proposed control scheme has better tracking performance and is more effective than regular NGFTSMC.

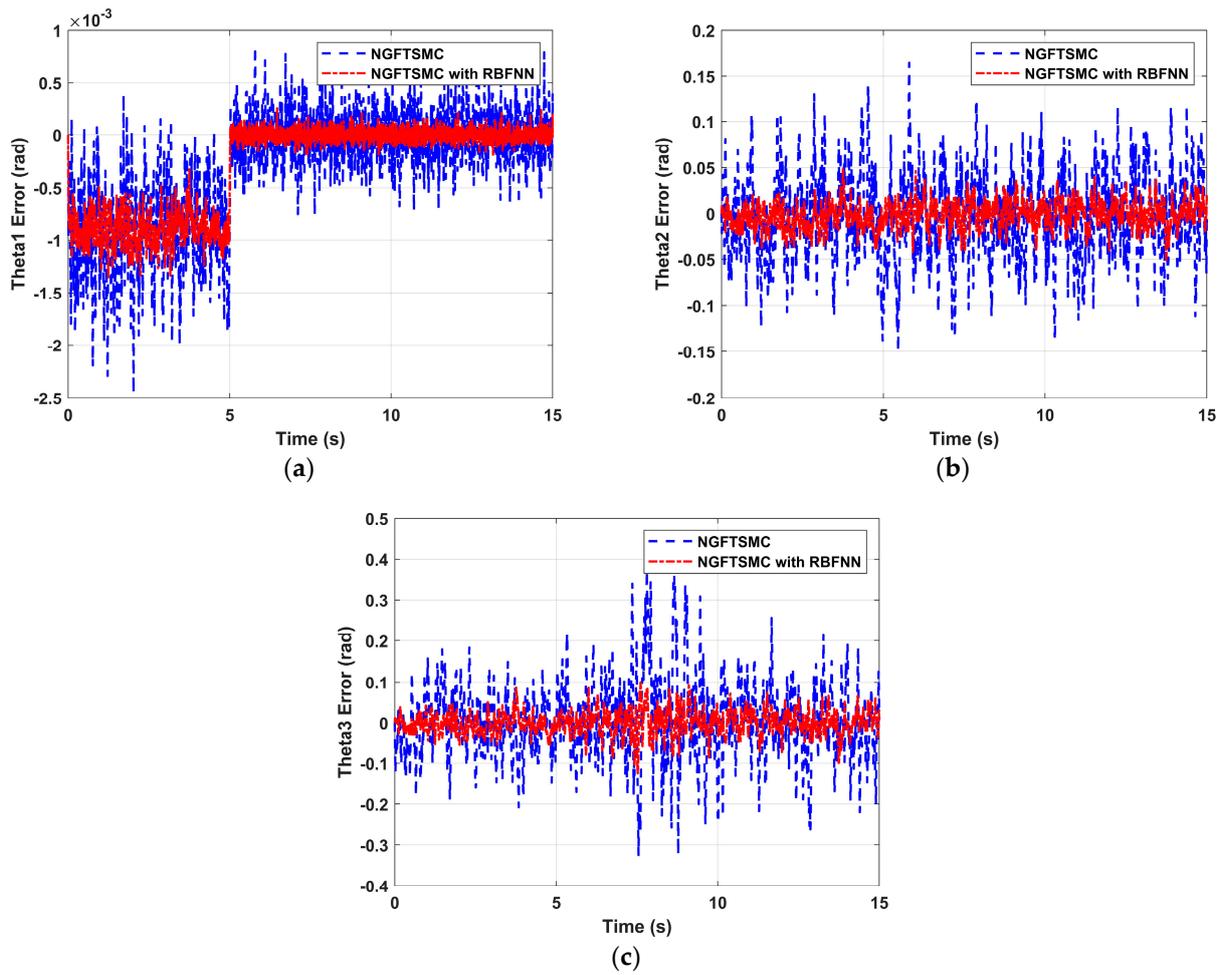


Figure 8. The trajectory tracking performance of the manipulator: (a) The tracking error of joint 1; (b) The tracking error of joint 2; (c) The tracking error of joint 3.

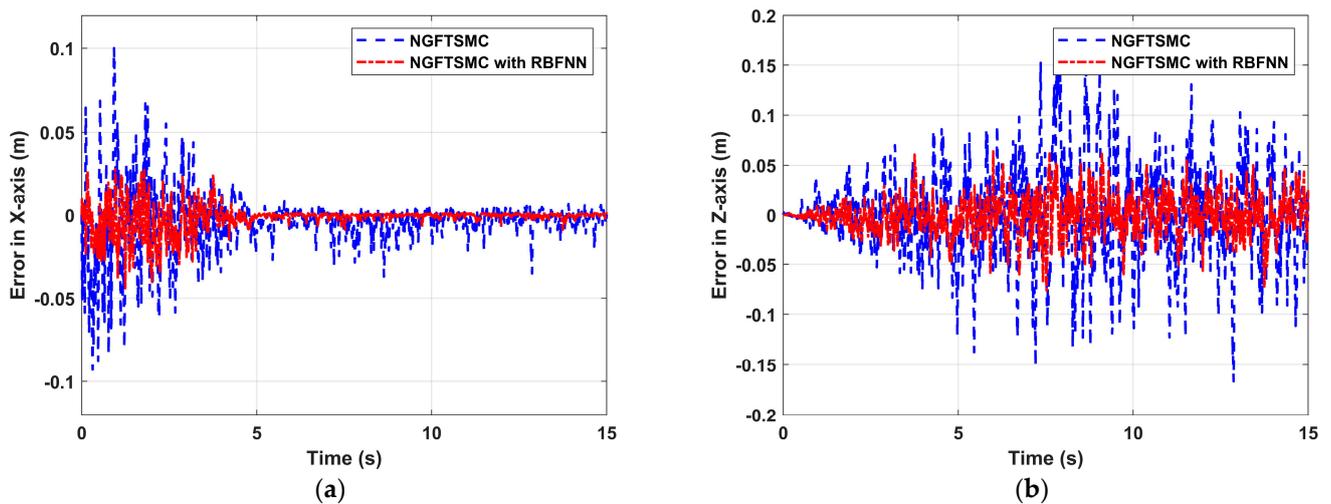


Figure 9. The trajectory tracking performance of the end of manipulator: (a) Error in X-axis; (b) Error in Z-axis.

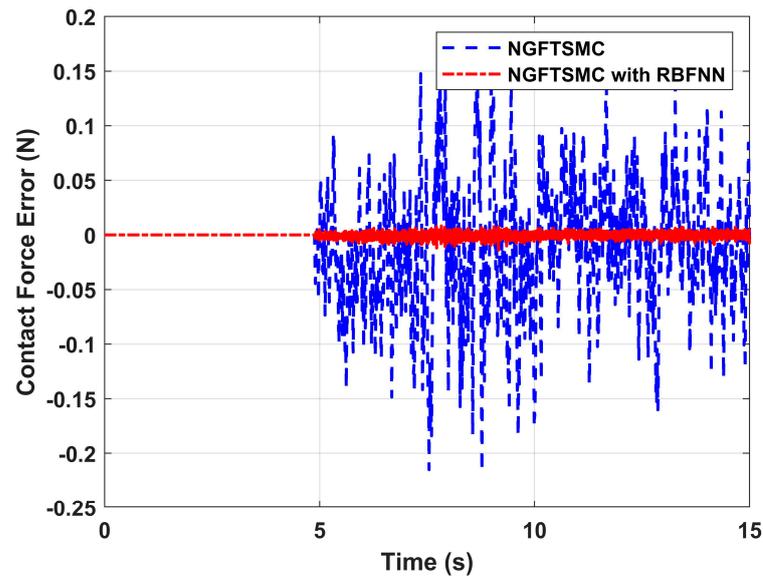


Figure 10. The contact force error between the manipulator and the target.

6. Experiments

In this section, some experiments are accomplished to verify the effectiveness of the proposed controller in exerting precise contact force on a target using the aerial manipulator. The quadrotor base of the aerial manipulator used in the experiment is a 680 mm diameter quadrotor equipped with four 15-inch propellers and a Pixhawk autopilot for low-level driver control. The manipulator is a planar three-link arm installed at the bottom of the quadrotor base. The control algorithms for the quadrotor and manipulator are executed on a Raspberry Pi 4b onboard computer. In addition, a motion-capture system and a pressure-detection system are used to measure the position, attitude, and contact force of the aerial manipulation system precisely. The photograph of the contact force experiment is pictured in Figure 11.



Figure 11. The photograph of the contact force experiment.

Figure 12 exhibits the variation in Euler angles of the quadrotor platform over time in the experiment. It can be noticed that both controllers can quickly converge the system state

to the desired trajectory. However, the state vector fluctuations of the system controlled by NGFTSMC based on RBFNN are minor compared to regular NGFTSMC, where the roll angle and pitch angle are 30%, and the yaw angle is 10%. The influence of external disturbances is not obvious in the laboratory conditions, which would be the reason that the controller performs very well. In addition, there were notable oscillations in roll and pitch at the 5th second. This is because the manipulator generates a large counter-torque at the moment of operation, which dramatically impacts the stability of the quadrotor platform. However, the rapid convergence of the controller allowed the quadrotor platform to quickly restore balance without affecting the operation of the manipulator. Moreover, weight convergence can be observed in Figure 13.

Figure 14 illustrates the contact force between the aerial manipulator and the target measured by the pressure detection system, and Figure 15 depicts the error between the measured and expected contact forces. It can be concluded that in the control system of NGFTSMC based on RBFNN, the contact force error can be controlled within 0.25 N, which is 30% of that of the regular NGFTSMC-controlled system. The results indicate that the proposed control scheme is effective in achieving precise contact force between the aerial manipulator and the target. Through analysis, it can be understood that the stronger the anti-interference ability of the quadrotor platform, the higher the accuracy of the contact force between the aerial manipulator and the target.

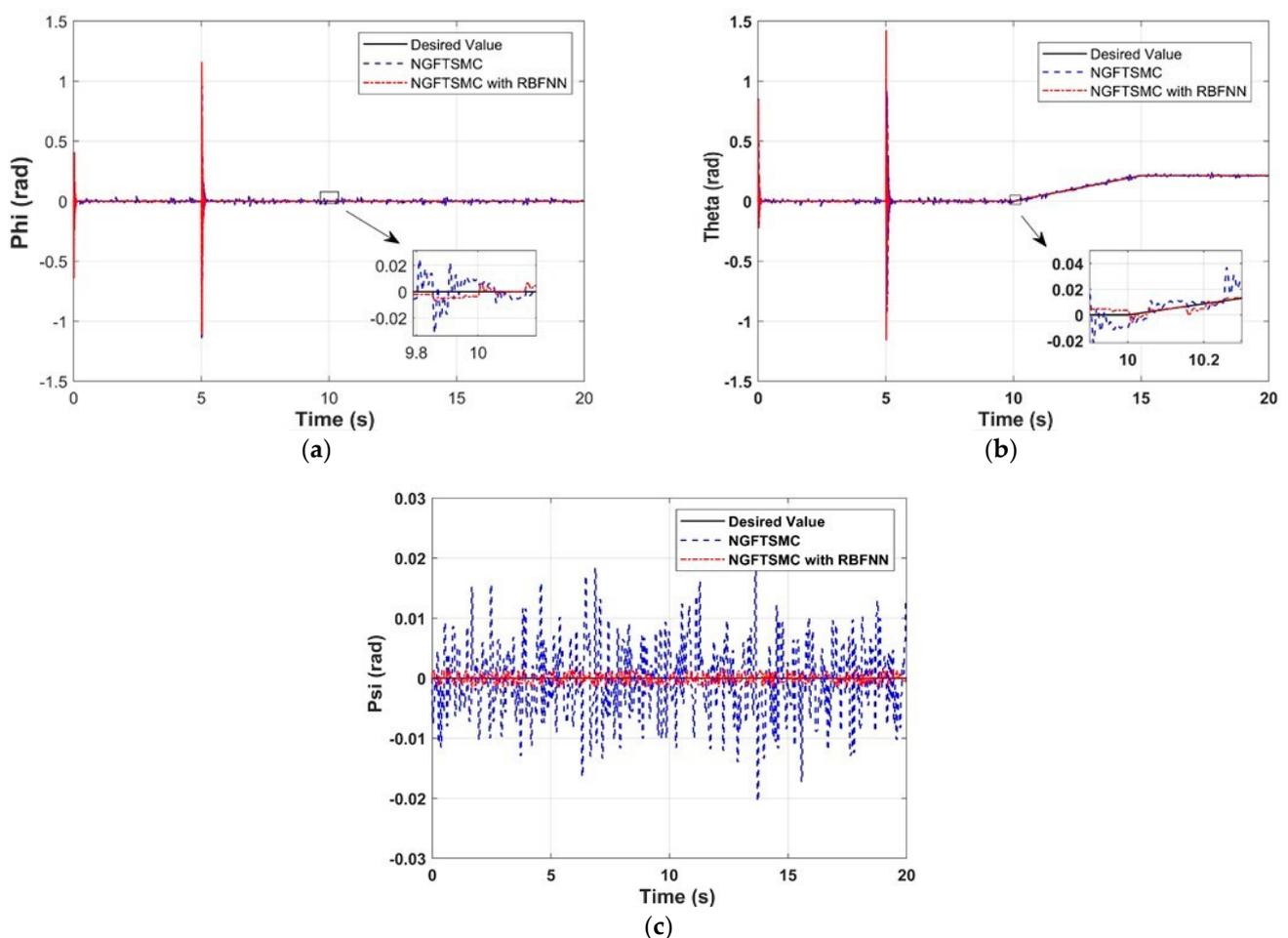


Figure 12. Euler angles of the quadrotor platform in the experiment: (a) The experimental results in Phi; (b) The experimental results in Theta; (c) The experimental results in Psi.

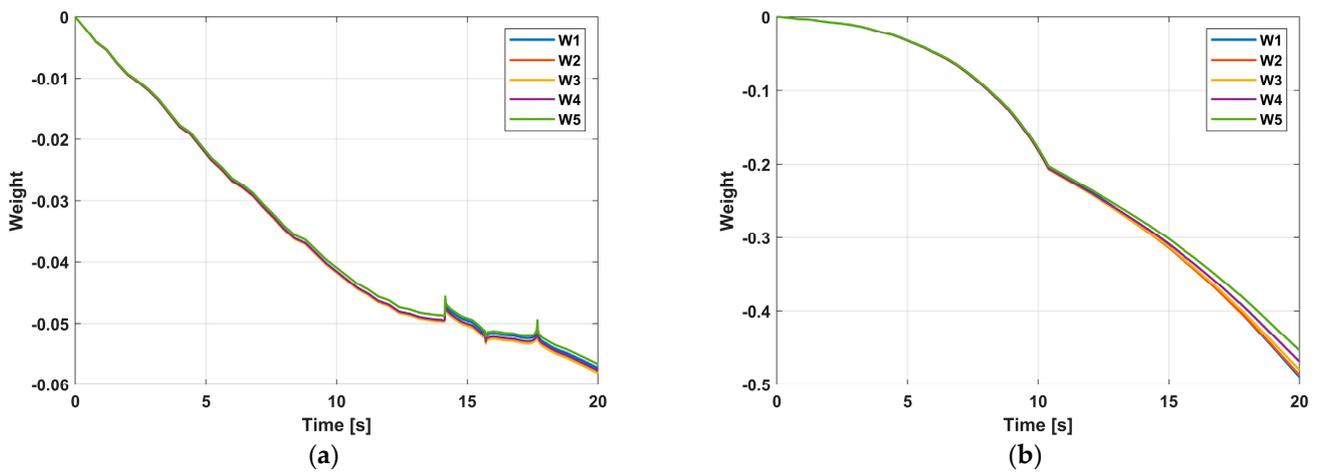


Figure 13. Weights of the RBFNN: (a) Quadrotor; (b) Manipulator.

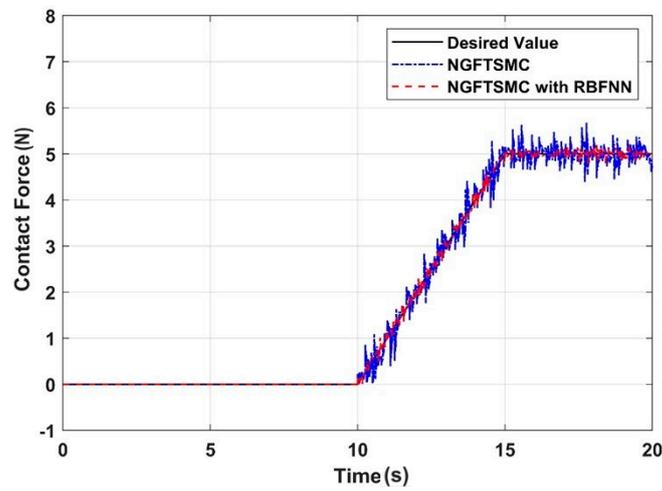


Figure 14. The contact force between the aerial manipulator and the target.

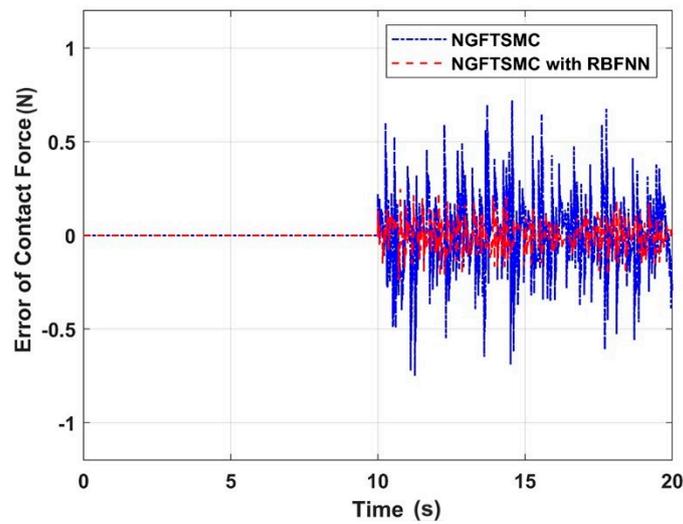


Figure 15. The error of contact force between the aerial manipulator and the target.

7. Conclusions

This study analyzes the characteristics of aerial contact operations and proposes a new control strategy for the aerial manipulation system. The multirotor and manipulator are controlled separately as two independent subsystems of the aerial manipulator. The multirotor subsystem exclusively serves as an aerial platform for the manipulator, providing a stable base in the air. Meanwhile, the manipulator subsystem merely performs contact force operations as an instrument. The interactions between both subsystems are regarded as mutual disturbances. First, each subsystem is modeled separately, and the mutual effects are analyzed and expressed using kinematic methods. Then, a robust sliding mode controller is developed for the aerial platform subsystem, and an adaptive RBF neural network controller is designed to eliminate the dependence on the system model through online learning. In addition, a double-loop impedance position controller is devised for the manipulator to execute compliant control of the contact force between the manipulator and the target. The simulation results indicate that the proposed control strategy has good trajectory-tracking capabilities and can accurately control the contact force. Finally, the experiment verified the effectiveness of the proposed control method, showing that the contact force error between the aerial manipulator and the target can be controlled within 0.25 N, which is 30% of the error of the comparative control scheme.

In this work, the impedance parameters of the target are known, but in unknown environments, the impedance parameters of the target cannot be obtained directly. Future work will focus on using the learning method to generate the desired position only by the desired force and minimize the position error without knowing the environment or the impedance parameters.

Author Contributions: Methodology, Q.F. and P.M.; software, Q.F.; validation, Q.F. and P.M.; data curation, Q.F.; writing—original draft preparation, Q.F.; writing—review and editing, Q.F. and P.M.; supervision, P.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Major Program for Science and Technology of Luoyang (2101018A).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: This work does not include any studies with human participants.

Data Availability Statement: The data used to support the findings of this study are included within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Lattanzi, L.; Cristalli, C.; Massa, D.; Boria, S.; Lépine, P.; Pellicciari, M. Geometrical calibration of a 6-axis robotic arm for high accuracy manufacturing task. *Int. J. Adv. Manuf. Technol.* **2020**, *111*, 1813–1829. [[CrossRef](#)]
2. Ying, K.; Pourhejazy, P.; Cheng, C.; Wang, C. Cyber-physical assembly system-based optimization for robotic assembly sequence planning. *J. Manuf. Syst.* **2021**, *58*, 452–466. [[CrossRef](#)]
3. Guo, W.; Zhu, Y.; He, X. A robotic grinding motion planning methodology for a novel automatic seam bead grinding robot manipulator. *IEEE Access* **2020**, *8*, 75288–75302. [[CrossRef](#)]
4. Sun, L.; Liang, F.; Fang, L. Design and performance analysis of an industrial robot arm for robotic drilling process. *Ind. Robot.* **2019**, *46*, 7–16. [[CrossRef](#)]
5. Lippiello, V.; Ruggiero, F. Cartesian Impedance Control of a UAV with a Robotic Arm. *IFAC Proc. Vol.* **2012**, *45*, 704–709. [[CrossRef](#)]
6. Thomas, J.; Loianno, G.; Sreenath, K.; Kumar, V. Toward image based visual servoing for aerial grasping and perching. In Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 31 May–7 June 2014; pp. 2113–2118.
7. Forte, F.; Naldi, R.; Macchelli, A.; Marconi, L. On the control of an aerial manipulator interacting with the environment. In Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 31 May–7 June 2014; pp. 4487–4492.
8. Meng, X.; He, Y.; Li, Q.; Gu, F.; Yang, L.; Yan, T.; Han, J. Contact force control of an aerial manipulator in pressing an emergency switch process. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 2107–2113.

9. Korpela, C.; Orsag, M.; Oh, P. Towards valve turning using a dual-arm aerial manipulator. In Proceedings of the 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL, USA, 14–18 September 2014; pp. 3411–3416.
10. Estrada, M.; Mintchev, S.; Christensen, D.; Cutkosky, M.; Floreano, D. Forceful manipulation with micro air vehicles. *Sci. Robot.* **2018**, *3*, eaau6903. [[CrossRef](#)]
11. Liu, Y.; Huang, C. DDPG-based adaptive robust tracking control for aerial manipulators with decoupling approach. *IEEE Trans. Cybern.* **2022**, *52*, 8258–8271. [[CrossRef](#)] [[PubMed](#)]
12. Orsag, M.; Korpela, C.M.; Bogdan, S.; Oh, P.Y. Hybrid adaptive control for aerial manipulation. *J. Intell. Robot. Syst.* **2014**, *73*, 693–707. [[CrossRef](#)]
13. Almakhles, D.J. Robust backstepping sliding mode control for a quadrotor trajectory tracking application. *IEEE Access* **2019**, *8*, 5515–5525. [[CrossRef](#)]
14. Labbadi, M.; Cherkaoui, M. Novel robust super twisting integral sliding mode controller for a quadrotor under external disturbances. *Int. J. Dyn. Control.* **2020**, *8*, 805–815. [[CrossRef](#)]
15. Wang, X.; Sun, S.; Kampen, E.; Chu, Q. Quadrotor fault tolerant incremental sliding mode control driven by sliding mode disturbance observers. *Aerosp. Sci. Technol.* **2019**, *87*, 417–430. [[CrossRef](#)]
16. Labbadi, M.; Cherkaoui, M. Robust adaptive backstepping fast terminal sliding mode controller for uncertain quadrotor UAV. *Aerosp. Sci. Technol.* **2019**, *93*, 105306. [[CrossRef](#)]
17. Nekoukar, V.; Dehkordi, N. Robust path tracking of a quadrotor using adaptive fuzzy terminal sliding mode control. *Control. Eng. Pract.* **2021**, *110*, 104763. [[CrossRef](#)]
18. Tripathi, V.; Kamath, A.; Behera, L.; Verma, N.; Nahavandi, S. Finite-time super twisting sliding mode controller based on higher-order sliding mode observer for real-time trajectory tracking of a quadrotor. *IET Control. Theory Appl.* **2020**, *14*, 2359–2371. [[CrossRef](#)]
19. Tripathi, V.; Kamath, A.; Behera, L.; Verma, N.; Nahavandi, S. An adaptive fast terminal sliding-mode controller with power rate proportional reaching law for quadrotor position and altitude tracking. *IEEE Trans. Syst.* **2022**, *52*, 3612–3625. [[CrossRef](#)]
20. Razmjooei, H.; Shafiei, M.H.; Palli, G.; Arefi, M.M. Non-linear finite-time tracking control of uncertain robotic manipulators using time-varying disturbance observer-based sliding mode method. *J. Intell. Robot. Syst.* **2022**, *36*, 104. [[CrossRef](#)]
21. Luo, H.; Zhang, S. Control of Quadrotor Based on RBF Neural Network Adaptive Fast Terminal Sliding-Mode Strategy. *J. Phys. Conf. Ser.* **2023**, *2437*, 012120. [[CrossRef](#)]
22. Tao, M.; He, X.; Xie, S.; Chen, Q. RBFNN-based singularity-free terminal sliding mode control for uncertain quadrotor UAVs. *Comput. Intell. Neurosci.* **2021**, *2021*, 3576783. [[CrossRef](#)] [[PubMed](#)]
23. Zhang, Q.; Yang, Z.; Jiang, Y.; Li, J.; Xu, H.; Xu, C.; Zeng, H.; Han, J.; Lai, S.; Li, H.; et al. Attitude Control for Quadcopter with Tilting Rotors using RBFNN-based Adaptive Terminal Sliding Mode Controller. In Proceedings of the 2019 IEEE International Conference on Real-time Computing and Robotics (RCAR), Irkutsk, Russia, 4–9 August 2019; pp. 778–785.
24. Huang, S.; Yang, Y. Adaptive neural-network-based nonsingular fast terminal sliding mode control for a quadrotor with dynamic uncertainty. *Drones* **2022**, *6*, 206. [[CrossRef](#)]
25. Gan, Y.; Duan, J.; Chen, M.; Dai, X. Multi-robot trajectory planning and position/force coordination control in complex welding tasks. *Appl. Sci.* **2019**, *9*, 924. [[CrossRef](#)]
26. Lakshminarayanan, S.; Kana, S.; Mohan, D.; Manyar, O.; Then, D.; Campolo, D. An adaptive framework for robotic polishing based on impedance control. *Int. J. Adv. Manuf. Technol.* **2021**, *112*, 401–417. [[CrossRef](#)]
27. Roveda, L.; Testa, A.; Shahid, A.; Braghin, F.; Piga, D. Q-Learning-based model predictive variable impedance control for physical human-robot collaboration. *Artif. Intell.* **2022**, *312*, 103771. [[CrossRef](#)]
28. Raibert, M.; Craig, J. Hybrid Position/Force Control of Manipulators. *J. Dyn. Syst. Meas. Control.* **1981**, *103*, 126–133. [[CrossRef](#)]
29. Hogan, N. Impedance Control—An Approach to Manipulation. I—Theory. II—Implementation. III—Applications. *J. Dyn. Syst. Meas. Control.* **1985**, *107*, 304–313.
30. Tran, D.; Truong, H.; Ahn, K. Adaptive nonsingular fast terminal sliding mode control of robotic manipulator based neural network approach. *Int. J. Precis. Eng. Manuf.* **2021**, *22*, 417–429. [[CrossRef](#)]
31. Zhang, W.; Guo, J.; Yu, Z. Fractional-order nonsingular terminal sliding mode control of uncertain robot neural network. In Proceedings of the 2020 Chinese Control and Decision Conference (CCDC), Hefei, China, 22–24 August 2020; pp. 4584–4589.
32. Jie, W.; Cheol, L.; Jaehyung, K.; Hee, K. Fast Fractional-Order Terminal Sliding Mode Control with RBFNN Based Sliding Perturbation Observer for 7-DOF Robot Manipulator. *IEEE Access* **2021**, *9*, 67117–67128. [[CrossRef](#)]
33. Kim, S.; Suh, J. Adaptive Robust RBF-NN Nonsingular Terminal Sliding Mode Control Scheme for Application to Snake Robot’s Head for Image Stabilization. *Appl. Sci.* **2023**, *13*, 4899. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.