

## Research Article

# ARSS: A Novel Aerial Robot Performs Tree Pruning Tasks

Hao Xu <sup>1,2</sup>, Zhong Yang <sup>1</sup>, Le Chang <sup>1</sup>, Kaiwen Lu <sup>1</sup>, Changliang Xu <sup>1</sup>  
and Qiuyan Zhang<sup>3</sup>

<sup>1</sup>College of Automation Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China

<sup>2</sup>School of Mathematics & Physics, Anhui University of Technology, Anhui 243032, Ma'anshan, China

<sup>3</sup>Guizhou Electric Power Research Institute, Guizhou Power Grid Co. Ltd., Guiyang, Guizhou 550000, China

Correspondence should be addressed to Zhong Yang; yangzhong@nuaa.edu.cn

Received 7 August 2020; Revised 10 September 2020; Accepted 28 September 2020; Published 14 October 2020

Academic Editor: Jia-Bao Liu

Copyright © 2020 Hao Xu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In this article, we present a novel aerial robot with a suspended saw (ARSS) for pruning trees that are close to electric power lines. The Robot's movement process includes free flight and aerial pruning. We first established a dynamic model and designed a controller based on the Active Disturbance Rejection Control (ADRC) on the model. Aiming at the problem of saw swing and residual oscillation during the free flight movement, we adopt the linear state feedback to design a swing angle controller. Finally, we use Matlab/Simulink and CoppeliaSim for simulation, and the simulation results verify the effectiveness and feasibility of the controller.

## 1. Introduction

As Aerial Robots are increasingly deployed in a broad spectrum of real-world, there is a growing demand to broaden their application scenario [1]. The Robot has the advantages of small size, strong maneuverability, and the ability to hover at a fixed point. It has become an excellent aerial work platform, and it has been used in cargo transportation [2], spraying pesticides [3], etc. Tree barrier pruning is a new application of aerial robots. A tree barrier is a tree that grows near the power line. Its branches will grow between the power lines and may cause a short circuit on the power line. These problems are caused due to contact tree branches with wires of power towers. This contact is one of the significant reasons for interrupting power services. The traditional method is that a person carries a pruning tool to climb the tree to cut the branches that will cause a short circuit of the power line. This method is hazardous and quickly causes personal injury. However, using aerial robots to carry pruning tools and fly to the vicinity of the branches to be cut will improve work efficiency and safety.

Moreover, the installation position of the cutting tool on the aerial Robot will cause differences in modeling and

control. To the best of our knowledge, the pruning tool can be installed on the top [4–6], front [7], and bottom of the fuselage [8]. The branches that need to be cut are generally at the crown of the tree. Moving the drone to the top of the tree can improve the pruning efficiency. Therefore, our design is to install the pruning tool under the fuselage through a rope hanging method. This installation method is the mode for a multirotor helicopter with suspended loads [9]. When the robot performs tasks, there are several difficulties to overcome. First, because of the movement of the fuselage and its own inertia, the load will swing in the air. The swing will increase the control difficulty and model uncertainty of the robot. Second, due to the limited space under the fuselage and the problem that the shape of the load does not match the fuselage, there are a lot of restrictions on the choice of load.

To the best of our knowledge, research on the multirotor helicopter with suspended loads has been receiving attention, and many effective control methods have been proposed. In [10], a nonlinear controller based on the backstepping control algorithm is designed so that the drone can still perform a good trajectory tracking when the hanging load swings significantly; Alothman and Gu [11]

uses a differential smoothing method to design a four-rotor UAV controller with suspended load, which realizes the trajectory generation and tracking of this system; Faust et al. [12] denotes an iterative linear-quadratic optimal controller, which realizes the trajectory tracking control of the UAV and suppresses the swing of the hanging load. Palunko et al. [13] adopts a reinforcement learning method to train the helicopter flight control system and realize the trajectory planning of hanging load. In [14], the researchers studied the take-off process of a quadcopter drone with a load. Yang et al. [15] solved the problem of the quadrotor achieving safe take-off under the condition that the hanging load is unknown, and the simulation results confirmed the effectiveness of its algorithm. Complex network theory [16, 17] can plan the tasks of aerial robots.

However, the research results of multirotor drones with hanging systems are remarkable, but many methods still have some limitations. The controller designed in [10] can achieve reasonable trajectory tracking control but does not control the position of the hanging load. The control effect of the controller designed in [11, 12] is significant, but the controller is complex and requires high computing resources for the onboard computer. The controller in [13] can only be used under a load of a specific weight. When changing the load, reinforcement learning is required again, and the portability is not good. The authors of [14, 15] focused on the take-off and landing issues, but did not focus on the entire flight process. References [16, 17] only analyze and prove mission planning but do not involve specific controller design.

In this article, we present the ARSS, an Aerial Robot with Suspended Saws. ARSS can use the hanging saw to cut and clean horizontally growing tree barrier. In order to solve the model uncertainty and disturbance in the execution of aerial pruning tasks, we used the Active Disturbance Rejection Control (ADRC) [18] to design the controller. The main components of ADRC include Tracking-Differentiator (TD) [19], Extended State Observer (ESO) [20, 21], and Nonlinear State Error Feedback (NLSEF) [22], all of which we have covered. We summarize the contributions as follows:

- (1) A novel a novel aerial robot can prune trees well.
- (2) As the position and attitude are controlled by the ADRC method without linearizing the system model, this robot system can achieve better control results when it is far from the equilibrium point, and the system has high robustness.
- (3) In addition, the controller has a simple structure and does not require high calculation speed of the onboard microprocessor.

In the rest of the article, we first discuss the dynamic model of ARSS; next, the controller based on ADRC is introduced including attitude control, position control and control allocation. We also verified the entire Robot under Matlab/Simulink and CoppeliaSim. Finally, we give some conclusions and future work.

## 2. Mathematical Model

**2.1. Coordinates and Conventions.** Overall, two coordinate frames were used. The inertial frame  $E\{O_E, X_E, Y_E, Z_E\}$  is fixed on the ground, and its  $z$  axis points upward. And the body frame  $B\{O_B, X_B, Y_B, Z_B\}$  is fixed on the COG, the  $z$  axis of the frame points upward, as seen in Figure 1. The Suspended Saws system is shown in Figure 2,  $L$  is the length of the rope, and  $l$  is the distance from the motor axis to the COG.  $\phi_L$  is the angle between the projection of the connecting rod in the  $O_b X_b Z_b$  plane and the  $z_b$  axis,  $\theta_L$  is the angle between the connecting rod and the above projection.

**2.2. Assumptions.** For the sake of model simplicity and algorithm efficiency, the following assumptions were made:

- (1) The body structure is rigid and symmetric
- (2) The geometric center and CoG of the Robot are the same
- (3) The rope is rigid, non-stretchable, and of no quality
- (4) The load is always under the body

**2.3. Notation.** Throughout the article, we use right-hand rules for the coordinate system. The attitude angles of the body are  $[\theta, \phi, \psi]^T \in \mathbb{R}^3$  including Pitch angle  $\theta$ , Roll angle  $\phi$ , and Yaw angle  $\psi$ , and  $[p, q, r]^T \in \mathbb{R}^3$  denote the angular velocity in the body frame.  $[x, y, z]^T \in \mathbb{R}^3$  denotes the position in the inertial frame.

The matrix  ${}^B_E R$  stands for the rotation matrix that rotates a vector represented in the inertial frame  $\{E\}$  to the body frame  $\{B\}$ . The rotation matrix is defined as

$${}^B_E R = \begin{bmatrix} c_\theta c_\psi & c_\theta s_\psi & -s_\theta \\ s_\theta s_\phi c_\psi - c_\phi s_\psi & s_\theta s_\phi s_\psi + c_\phi c_\psi & c_\theta s_\phi \\ s_\theta c_\phi c_\psi + s_\phi s_\psi & s_\theta c_\phi s_\psi - s_\phi c_\psi & c_\theta c_\phi \end{bmatrix}, \quad (1)$$

where  ${}^B_E R \in SO(3)$ ,  $SO(3) \triangleq \{R | R^T R = \mathbf{I}_3, \det(R) = 1, R \in \mathbb{R}^{3 \times 3}\}$  with  $s_{(\cdot)}$  representing the sine and  $c_{(\cdot)}$  the cosine.

**2.4. Body Dynamic Model.** For the modeling of the body dynamics, the Newton–Euler formalism was used. The Equilibrium equation is as follows:

$$M [\ddot{x} \ \ddot{y} \ \ddot{z}] = \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix}, \quad (2)$$

$$I_b \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} + \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times I_b \begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} \tau_x \\ \tau_y \\ \tau_z \end{bmatrix},$$

with  $M$  being the mass of the aerial Robot and  $I_b$  the moment of inertia.  $I_b$  is defined as follows:

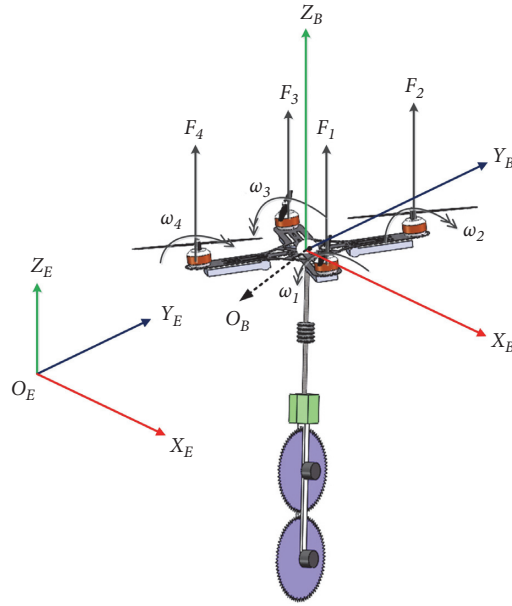


FIGURE 1: The inertial and body frames.

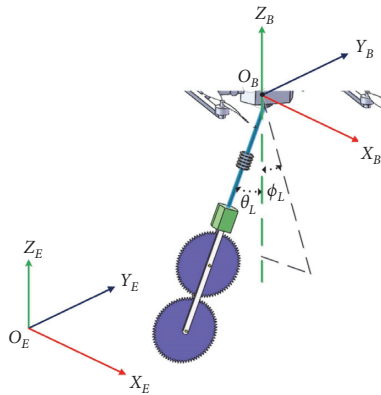


FIGURE 2: The suspended saws system.

$$I_b = \begin{bmatrix} I_{11} & -I_{12} & -I_{13} \\ -I_{12} & I_{22} & -I_{23} \\ -I_{13} & -I_{23} & I_{33} \end{bmatrix}, \quad (3)$$

where  $I_{12}, I_{13}, I_{23}$  are smaller than  $I_{11}, I_{22}, I_{33}$ , respectively. Define

$$\begin{aligned} I_x &= I_{11}, \\ I_y &= I_{22}, \\ I_z &= I_{33}. \end{aligned} \quad (4)$$

The solution to equation (3) is

$$I_b = \text{diag}(I_x, I_y, I_z). \quad (5)$$

According to equation (4), the forces  ${}^B F = [F_x \ F_y \ F_z]^T$  and torques  $\tau^B = [\tau_x \ \tau_y \ \tau_z]^T \in \mathbb{R}^3$  act on the COG of the body. Then the relation between

angular velocity  $[p, q, r]^T$  and attitude angle  $[\theta, \phi, \psi]^T$  can be expressed as

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \mathbf{R}_c \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}, \quad (6)$$

$$\mathbf{R}_c = \begin{bmatrix} 1 & 0 & -\sin \theta \\ 0 & \cos \phi & \sin \phi \cos \theta \\ 0 & -\sin \phi & \cos \phi \cos \theta \end{bmatrix}.$$

Regardless of external disturbances, the aerial Robot is mainly subjected to the gravity in the negative direction of the  $O_e X_e$  axis and the lift  $F_i, i = 1, 2, 3, 4$  generated by the four rotors pointed by the positive direction of the  $O_b Z_b$  axis.  $F_i$  is related to lift coefficient  $C_T$  and rotating speed  $\omega_i$ , which is given by

$$F_i = C_T \omega_i^2. \quad (7)$$

Therefore, let  $F_t$  be the total lift so that

$$F_t = \sum_{i=1}^4 F_i = C_T \sum_{i=1}^4 \omega_i^2. \quad (8)$$

According to equation (1),  ${}^B F = [F_x \ F_y \ F_z]^T$  is obtained as follows:

$$\begin{aligned} \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} &= {}^B_E \mathbf{R}^T \begin{bmatrix} 0 \\ 0 \\ -F_t \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} \\ &= \begin{bmatrix} -(\cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi) F_t \\ -(\cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi) F_t \\ mg - \cos \phi \cos \theta F_t \end{bmatrix}. \end{aligned} \quad (9)$$

Based on assumption (2), the resultant torque is mainly generated by the lift and counter torque at the rotor that does not pass through the center of mass, while gravity does not produce torque. The torque generated by the rotor lift constitutes the moment around the  $O_b X_b$  and  $O_b Y_b$  axis, and the antitorque of the rotor constitutes the moment around the  $O_b Z_b$  axis, given by

$$\begin{aligned} \tau_x &= L(-F_1 + F_2 + F_3 - F_4), \\ \tau_y &= L(F_1 + F_2 - F_3 - F_4), \\ \tau_z &= C_M(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2), \end{aligned} \quad (10)$$

where  $L$  is the distance between the center point of the rotor and the centroid of the Robot.

On the basis of Equations (8) and (10), one has

$$\begin{pmatrix} \tau_x \\ \tau_y \\ \tau_z \end{pmatrix} = \begin{pmatrix} -C_T L & C_T L & C_T L & -C_T L \\ C_T L & C_T L & -C_T L & -C_T L \\ C_M & -C_M & C_M & -C_M \end{pmatrix} \begin{pmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{pmatrix}. \quad (11)$$

By combining Equations (2), (9), and (11), the body dynamic model is expressed as

$$\begin{aligned} \ddot{x} &= -(\cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi) \frac{F_t}{m}, \\ \ddot{y} &= -(\cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi) \frac{F_t}{m}, \\ \ddot{z} &= g - \cos \phi \cos \theta \frac{F_t}{m}, \\ \dot{p} &= \frac{I_y - I_z}{I_x} qr + \frac{1}{I_x} \tau_x, \\ \dot{q} &= \frac{I_z - I_x}{I_y} pr + \frac{1}{I_y} \tau_y, \\ \dot{r} &= \frac{I_x - I_y}{I_z} pq + \frac{1}{I_z} \tau_z. \end{aligned} \quad (12)$$

**2.5. Suspended Saws Model.** For the modeling of the saws dynamics, the Lagrange formalism was used. Its general form is the following:

$$\frac{d}{dt} \left( \frac{\partial \mathcal{L}}{\partial \dot{\Theta}} \right) - \frac{\partial \mathcal{L}}{\partial \Theta} = Q, \quad (13)$$

where  $\mathcal{L}(\Theta, \dot{\Theta}) = k(\Theta, \dot{\Theta}) - u(\Theta)$ ,  $k(\Theta, \dot{\Theta})$  represents the kinetic energy and  $u(\Theta)$  expresses potential energy;  $\Theta$  denotes the generalized coordinate and  $Q$  is generalized forces.

As shown in Figure 1, the equation set from Saws position  $(x_L, y_L, z_L)$  to the body position  $(x, y, z)$  is further obtained as

$$\begin{aligned} x_L &= x + l \cos \theta_L \sin \phi_L, \\ y_L &= y + l \sin \theta_L, \\ z_L &= z - l \cos \theta_L \cos \phi_L, \end{aligned} \quad (14)$$

where  $l$  represents the rod length.

Combining equation (14), kinetic energy  $k(\Theta, \dot{\Theta})$  is expressed as

$$k(\Theta, \dot{\Theta}) = \frac{1}{2} M(\dot{x}^2 + \dot{y}^2 + \dot{z}^2) + \frac{1}{2} m(\dot{x}_L^2 + \dot{y}_L^2 + \dot{z}_L^2). \quad (15)$$

Then potential energy is

$$u(\Theta) = Mgz + mgz_L. \quad (16)$$

Among the saws dynamic model, generalized coordinate  $\Theta = [x, y, z, \theta_L, \phi_L]^T$  and generalized forces  $Q = [F_x, F_y, F_z, 0, 0]^T$ . By combining equations (13)–(17), the ARSS dynamic model is expressed as

$$M(\Theta)\ddot{\Theta} + C(\Theta, \dot{\Theta}) + G(\Theta) = Q, \quad (17)$$

where

$$M(\Theta) = \begin{bmatrix} M+m & 0 & 0 & mlc_{\theta_L}c_{\phi_L} & -mLs_{\theta_L}s_{\phi_L} \\ 0 & M+m & 0 & 0 & mlc_{\theta_L} \\ 0 & 0 & M+m & mlc_{\theta_L}s_{\phi_L} & mLs_{\theta_L}c_{\phi_L} \\ mlc_{\theta_L}c_{\phi_L} & 0 & mlc_{\theta_L}s_{\phi_L} & ml^2c_{\theta_L}^2 & 0 \\ -mLs_{\theta_L}s_{\phi_L} & mlc_{\theta_L} & mLs_{\theta_L}c_{\phi_L} & 0 & ml^2 \end{bmatrix},$$

$$C(\Theta, \dot{\Theta}) = \begin{bmatrix} 0 & 0 & 0 & -ml(c_{\theta_L}s_{\phi_L}\dot{\phi}_L + s_{\theta_L}c_{\phi_L}\dot{\theta}_L) & -ml(s_{\theta_L}c_{\phi_L}\dot{\phi}_L + c_{\theta_L}s_{\phi_L}\dot{\theta}_L) \\ 0 & 0 & 0 & 0 & -mLs_{\theta_L}\dot{\theta}_L \\ 0 & 0 & 0 & ml(c_{\theta_L}c_{\phi_L}\dot{\phi}_L - s_{\theta_L}s_{\phi_L}\dot{\theta}_L) & -ml(s_{\theta_L}s_{\phi_L}\dot{\phi}_L - c_{\theta_L}c_{\phi_L}\dot{\theta}_L) \\ 0 & 0 & 0 & -ml^2c_{\theta_L}s_{\phi_L}\dot{\theta}_L & -ml^2c_{\theta_L}s_{\phi_L}\dot{\phi}_L \\ 0 & 0 & 0 & ml^2c_{\theta_L}s_{\phi_L}\dot{\phi}_L & 0 \end{bmatrix}, \quad (18)$$

$$G(\Theta) = [0 \ 0 \ (M+m)g \ mglc_{\theta_L}s_{\phi_L} \ mgl s_{\theta_L}c_{\phi_L}]^T,$$

$$Q = \begin{bmatrix} {}^B_E R^T [0 \ 0 \ F_t]^T \\ 0 \\ 0 \end{bmatrix} = [F_x \ F_y \ F_z \ 0 \ 0]^T,$$

$$\begin{aligned} \ddot{x} &= \frac{ml}{M+m} (c^3\theta s\phi\dot{\phi}^2 + c\theta s\phi\dot{\theta}^2) + \frac{1}{M} (M+m - m\sin^2\phi\cos^2\theta) \\ &\quad - \frac{m}{M} \sin\theta\cos\theta\sin\phi F_y + \frac{m}{M} \sin\phi\cos\phi\cos^2\theta F_z, \\ \ddot{y} &= \frac{ml}{M+m} (\cos^2\theta\sin\theta\dot{\phi}^2 + \sin\theta\dot{\theta}^2) - \frac{m}{M} \sin\theta\cos\theta\sin\phi F_x \\ &\quad + \left(1 + \frac{m}{M}\cos^2\theta\right) F_y + \frac{m}{M} \sin\theta\cos\theta\cos\phi F, \\ \ddot{z} &= -\frac{ml}{M+m} (\cos^3\theta\cos\phi\dot{\phi}^2 + \cos\theta\cos\phi\dot{\theta}^2) + \frac{m}{M(M+m)} \sin\phi\cos\phi\cos^2\theta F_x \\ &\quad - \frac{m}{M(M+m)} \sin\theta\cos\theta\cos\phi F_y + \frac{1}{M(M+m)} (M+m - m\cos^2\phi\cos^2\theta) F_z - g, \end{aligned} \quad (19)$$

$$\begin{aligned} L\cos\theta_L\ddot{\phi}_L - 2l\sin\theta_L\dot{\theta}_L\dot{\phi}_L + \cos\phi_L\ddot{x} + (\ddot{z} + g)\sin\phi_L &= 0, \\ L\ddot{\theta}_L + l\sin\theta_L\cos\theta_L\dot{\phi}_L^2 + g\cos\phi_L\sin\theta_L - \sin\phi_L\sin\theta_L\ddot{x} + \cos\theta_L\dot{y} + \cos\phi_L\sin\theta_L\ddot{z} &= 0. \end{aligned} \quad (20)$$

### 3. Controller Design

The general aerial Robot only needs to control the position and attitude, and ARSS not only needs to control the position and attitude, but also needs to control the swing angle of suspended saw. The position change of the Robot will cause the saw swing. If the swing angle is not controlled, the saw may swing violently, affecting the robot stability, so it is necessary to control the swing angle of the suspended saw. ADRC has the advantages of low model

dependence and strong anti-disturbance, and it has good applicability to ARSS with complex model and large disturbance.

In this section, we present a baseline controller for the ARSS platform. The controller is based on ADRC including attitude and position control, and uses a linear state feedback to design the swing angle controller. The controller is a double-loop structure, the outer loop is position control, and the inner one is attitude control. Control structure shown in Figure 3.

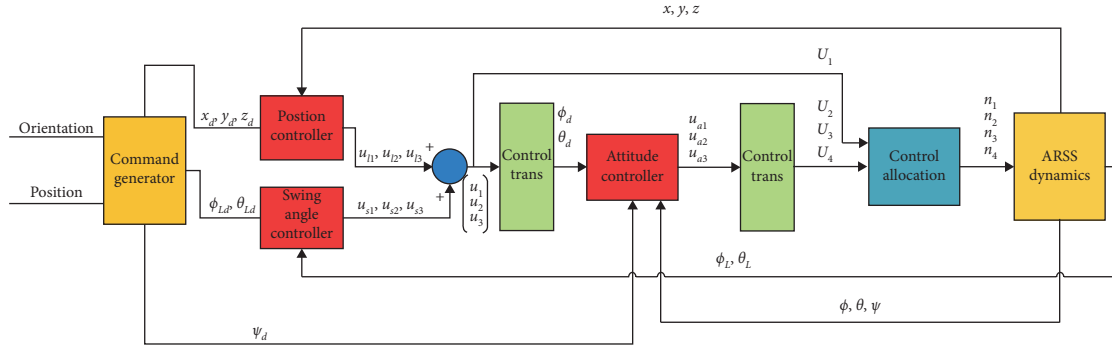


FIGURE 3: The control diagram of ARSS.

**3.1. Attitude Control.** The attitude controller controls the attitude angle  $\alpha$ , so equation (6) needs to be transformed to obtain the expression of Euler angle. Because the expression contains a large number of non-linear elements and coupling terms, a common solution is the small angle approximation method, which limits the attitude angle change in a small range, so that the Euler angle change  $(\dot{\phi}, \dot{\theta}, \dot{\psi})$  is approximately equal to the angular velocity  $(p, q, r)$ . However, such a processing method can only allow the controller to have a good control effect within a small range of attitude angle changes. If the attitude angle changes too much, the performance of the controller will be greatly reduced. The other method is to add a layer of angular velocity loop on the basis of the small angle approximation method to make the attitude controller into a double-loop structure. Although this method has a certain effect, it will make the structure of the controller very complicated and many parameters, it is difficult to adjust the parameters.

ADRC can solve this problem well. ADRC has low dependence on the mathematical model of the controlled object. It can treat the coupling item as an unmodeled part of the system and compensate the coupling item by Extended State Observer (ESO). Therefore, we adopt ADRC to design ARSS attitude controller.

First, the derivative of equation (6) can be obtained:

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \mathbf{R}_c \begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} + \begin{bmatrix} 0 & -\cos \theta & 0 \\ -\sin \phi & -\sin \phi \sin \theta & \cos \phi \cos \theta \\ -\cos \phi & -\sin \phi \cos \theta & -\cos \phi \sin \theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \dot{\theta} \\ \dot{\theta} \dot{\psi} \\ \dot{\phi} \dot{\psi} \end{bmatrix}. \quad (21)$$

Then, combining equations (3)–(5) with equation (21), one has

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = I_b^{-1} \text{diag}(I_y - I_z, I_z - I_x, I_x - I_y) \begin{bmatrix} \text{qr} \\ \text{pr} \\ \text{pq} \end{bmatrix} + I_b^{-1} \begin{bmatrix} \tau_x \\ \tau_y \\ \tau_z \end{bmatrix}. \quad (22)$$

Consequently,

$$\begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = \mathbf{R}_c^{-1} \mathbf{I}_b^{-1} \text{diag}(I_y - I_z, I_z - I_x, I_x - I_y) \begin{bmatrix} \text{qr} \\ \text{pr} \\ \text{pq} \end{bmatrix} + \mathbf{R}_c^{-1} \mathbf{I}_b^{-1} \begin{bmatrix} \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} - \mathbf{R}_c^{-1} \begin{bmatrix} 0 & -\cos \theta & 0 \\ -\sin \phi & -\sin \phi \sin \theta & \cos \phi \cos \theta \\ -\cos \phi & -\sin \phi \cos \theta & -\cos \phi \sin \theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \dot{\theta} \\ \dot{\theta} \dot{\psi} \\ \dot{\phi} \dot{\psi} \end{bmatrix}, \quad (23)$$

where  $\mathbf{R}_c$  is reversible and  $k$  is an integer when  $\theta \neq (2k + 1)\pi/2$  and  $\phi \neq (2k + 1)\pi/2$ .

Equation (23) is very complicated and there are non-linear and coupling terms, it is very difficult to design the controller directly for this model. ADRC can simplify the model and treat the nonlinear and coupling terms as unmodeled parts. Consequently, the equation (23) becomes

$$\begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} s_1(p, q, r, \phi, \theta, \psi, \dot{\phi}, \dot{\theta}, \dot{\psi}) \\ s_2(p, q, r, \phi, \theta, \psi, \dot{\phi}, \dot{\theta}, \dot{\psi}) \\ s_3(p, q, r, \phi, \theta, \psi, \dot{\phi}, \dot{\theta}, \dot{\psi}) \end{bmatrix} + \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}, \quad (24)$$

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \mathbf{R}_c^{-1} \mathbf{I}_b^{-1} \begin{bmatrix} U_2 \\ U_3 \\ U_4 \end{bmatrix},$$

where  $(s_1, s_2, s_3)$  denote coupling and nonlinear part,  $(u_1, u_2, u_3)$  are virtual control and the relationship with input torque is shown in equation (24). From this, the desired control input can be obtained

$$\begin{bmatrix} U_2 \\ U_3 \\ U_4 \end{bmatrix} = \mathbf{I}_b \mathbf{R}_c \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} I_x & 0 & -I_x \sin \theta \\ 0 & I_y \cos \phi & I_y \sin \phi \cos \theta \\ 0 & -I_z \sin \phi & I_z \cos \phi \cos \theta \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}. \quad (25)$$

Here, the math model required by the attitude controller can be obtained, the decoupling of pitch, roll and yaw is

completed, and this multiple input multiple output (MIMO) model is converted into three single-input single-output (SISO) subsystems. From equation (25), these subsystems are second-order linear systems.

Taking the roll angle  $\theta$  for example, we established the ADRC attitude controller to roll motion, and the structure diagram is shown in Figure 4.

When  $z_1 = \theta$ ,  $\dot{z}_1 = \dot{\theta}$ , equation (24) is rewritten as

$$\begin{aligned} \dot{z}_1 &= z_2, \\ \dot{z}_2 &= s_2(p, q, r, \phi, \theta, \psi, \dot{\phi}, \dot{\theta}, \dot{\psi}) + u_2 + w_\theta, \\ \theta &= z_1, \end{aligned} \quad (26)$$

where  $w_\theta$  represents the disturbance to  $\theta$ ;  $s_2(p, q, r, \phi, \theta, \psi, \dot{\phi}, \dot{\theta}, \dot{\psi}) + w_\theta$  represents the sum of internal and external disturbances estimated by the Expanded State Observer (ESO). Equation (26) complies with the second-order form of the ADRC standard and can be designed as a second-order attitude controller. The specific design steps are as follows:

3.1.1. *Design TD.* TD is shown in equation (27):

$$\begin{aligned} v_1(k+1) &= v_1(k) + Tv_2(k), \\ v_2(k+1) &= v_2(k) + Tfh, \\ fh &= fhan(v_1(k) - v(k), v_2(k), r, h), \end{aligned} \quad (27)$$

where  $v(k)$  is the desired pitch angle  $\theta_d$  which is given by the position controller,  $v_1(k)$  is the tracking signal of  $\theta_d$ , and  $v_2(k)$  is the derivative of  $v_1(k)$ .  $T$  is the sampling period of the discrete signal,  $fhan(v_1(k) - v(k), v_2(k), r, h)$  is the nonlinear function, and its expression is as follows:

$$\begin{aligned} d &= rh, \\ d_0 &= dh, \\ y &= v_1(k) - v(k) + hv_2(k), \\ a_0 &= (d^2 + 8r|y|)^{(1/2)}, \\ a &= \begin{cases} v_2(k) + \frac{(a_0 - d)}{2}, & |y| > d_0, \\ v_2(k) + \frac{y}{h}, & |y| \leq d_0, \end{cases} \\ fhan &= \begin{cases} \frac{ra}{d}, & a \leq d_0, \\ -r\text{sgn}(a), & a > d_0, \end{cases} \end{aligned} \quad (28)$$

where  $r$  and  $h$  are adjustable gains.  $r$  is called the fast factor, and a suitably large one can achieve a faster tracking speed;  $h$  is called the filter factor, and the larger the  $h$ , the better the filtering effect. But the increase of  $h$  will bring about an increase in phase delay, generally  $h$  takes 5 ~ 10 T.

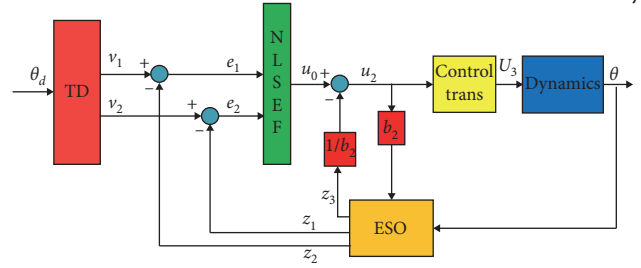


FIGURE 4: Structure diagram in rolling channel.

3.1.2. *Design ESO.* The ESO is as shown in equation (29)

$$\begin{aligned} e &= z_1(k) - y(k), \\ z_1(k+1) &= z_1(k) + T[z_2(k) - \beta_{01}e], \\ z_2(k+1) &= z_2(k) + T[z_3(k) - \beta_{02}\text{fal}(e, \alpha_1, \delta) + bu(k)], \\ z_3(k+1) &= z_3(k) - T\beta_{03}\text{fal}(e, \alpha_2, \delta), \end{aligned} \quad (29)$$

where  $y(k)$  denotes the actual pitch angle of  $\theta$ ;  $z_1$  is the estimated value of  $\theta$ ,  $z_2$  is the estimated value of  $\dot{\theta}$ ,  $z_3$  is the estimated value of total disturbance;  $u$  represents the virtual control  $u_2$  which mentioned in equation (24), and  $b = 1$ .

3.1.3. *Design SEF.* We use the fastest control synthesis function, which is expressed as

$$u_0 = -fhan(e_1, ce_2, r, h), \quad (30)$$

where  $r$  and  $h$  are consistent with the description in equation (29),  $c$  denotes damping coefficient. When  $c \in (0, 1)$ , the system Underdamped; when  $c > 1$ , the system overdamped.

From equation (30), the control output  $u_0$  of SEF can be obtained, then combining equation (31) to compensate for disturbances observed by ESO, we get the total control output  $u_2$  of the rolling channel. Equation (31) is

$$u = u_0 - \left(\frac{z_3}{b}\right), \quad (31)$$

where  $z_3$  and  $b$  are the same as equation (29)

Using the same method, the control outputs  $u_1, u_3$  of the roll, yaw channel attitude controllers can be obtained, respectively. In this case, according to equation (25), the control input  $[U_2, U_3, U_4]^T$  is acquired. The attitude controller is shown in Figure 5.

3.2. *Position Control.* When designing a position controller, we need to control the position of the aerial Robot and suspended saw. The position change will cause the swing of the saw. If the swing angle is not controlled, the saw may swing violently, affecting the stability of the aerial Robot, so it is necessary to control the swing angle of the hanging saw. ADRC has the advantages of low model dependence and strong anti-disturbance, and it has good applicability to ARSS with more complex models and larger disturbances.

Similar to the attitude controller, equation (12) is rewritten as

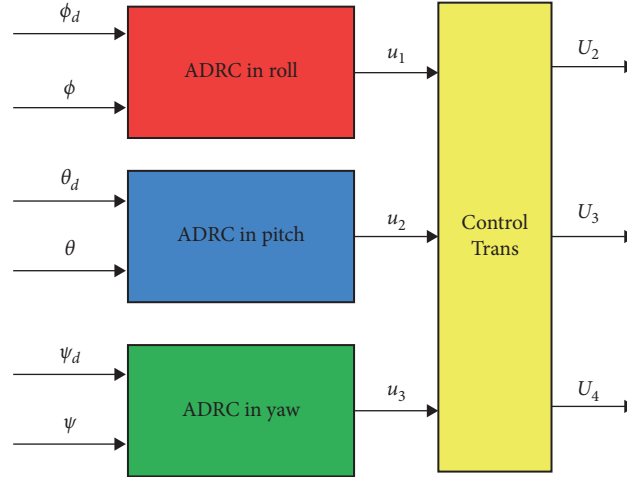


FIGURE 5: Attitude controller structure diagram.

$$(\ddot{x} \ \ddot{y} \ \ddot{z}) = \begin{pmatrix} s_1(\theta, \dot{\theta}, \phi, \dot{\phi}) \\ s_2(\theta, \dot{\theta}, \phi, \dot{\phi}) \\ s_3(\theta, \dot{\theta}, \phi, \dot{\phi}) \end{pmatrix} + \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix}, \quad (32)$$

$$\begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} = T^{-1} \begin{pmatrix} F_x \\ F_y \\ F_z \end{pmatrix}. \quad (33)$$

where

$T$  is defined as follows:

$$T = \begin{bmatrix} \frac{1}{M}(M + m - ms^2\phi_L c^2\theta_L) & -\frac{m}{M}s_{\theta_L}c_{\theta_L}s_{\phi_L} & \frac{m}{M}s_{\phi_L}c_{\phi_L}c^2\theta_L \\ -\frac{m}{M}s_{\theta_L}c_{\theta_L}s_{\phi_L} & 1 + \frac{m}{M}c^2\theta_L & \frac{m}{M}s_{\theta_L}c_{\theta_L}c_{\phi_L} \\ \frac{m(s_{\phi_L}c_{\phi_L}c^2\theta_L)}{M(M+m)} & \frac{m(s_{\theta_L}c_{\theta_L}c_{\phi_L})}{M(M+m)} & \frac{M+m - mc^2\phi_L c^2\theta_L}{M(M+m)} \end{bmatrix}. \quad (34)$$

During the flight of the Aerial Robot, the swing angle of the suspended saw is small. According to the Small-Angle Approximation, the approximate solution is given by

$$\begin{aligned} \sin \phi_L &\approx \phi_L, \\ \cos \phi_L &\approx 1, \\ \sin \theta_L &\approx \theta_L, \\ \cos \theta_L &\approx 1. \end{aligned} \quad (35)$$

Thus, equations (19) and (20) are simplified as

$$\begin{aligned} \ddot{x} + l\ddot{\phi}_L + g\phi_L &= 0, \\ \ddot{y} + l\ddot{\theta}_L + g\theta_L &= 0. \end{aligned} \quad (36)$$

The ARSS is decomposed into the aerial robot part of the standard second-order model and the suspended saw part of the linear model. Position control is changed to control the aerial Robot while ensuring the stability of the linear part, so that the saw swing during flight is gradually reduced.

First, design the position controller of the aerial robot part. Similar to the attitude controller, equation (32) is

converted into three SISO subsystems, and these subsystems are second-order standard.

Taking  $x$  for example. When  $z_1 = x$  and  $\dot{z}_1 = \dot{x}$ , equation (32) is rewritten as

$$\begin{aligned} \dot{z}_1 &= z_2, \\ \dot{z}_2 &= u_1 + w_x, \\ x &= z_1, \end{aligned} \quad (37)$$

where  $w_x$  represents the disturbance to  $x$  and  $u_1$  denotes virtual control. Similar to the attitude controller, the specific design steps are as follows.

3.2.1. Design TD. TD is defined as

$$\begin{aligned} v_1(k+1) &= v_1(k) + T_s v_2(k), \\ v_2(k+1) &= v_2(k) + T_s f h, \\ f h &= \text{fhan}(v_1(k) - v_1(k), v_2(k), r, h), \end{aligned} \quad (38)$$



where  $v(k)$  is the desired position  $x_d$  which is given by the position controller,  $v_1(k)$  is the tracking signal of  $x_d$ , and  $v_2(k)$  is the derivative of  $v_1(k)$ .  $T_s$  is the sampling period of the discrete signal.  $r$  and  $h$  are adjustable gains.  $\text{fhan}(v_1(k) - v(k), v_2(k), r, h)$  is the non-linear function, and its expression is as follows:

$$\begin{aligned} d &= rh, \\ d_0 &= dh, \\ y &= x_1 + hx_2, \\ a_0 &= (d^2 + 8r|y|)^{1/2}, \\ a &= \begin{cases} x_2 + \frac{(a_0 - d)}{2}, & |y| > d_0, \\ x_2 + \frac{y}{h}, & |y| \leq d_0, \end{cases} \quad (39) \\ \text{fhan} &= \begin{cases} -\frac{ra}{d}, & a \leq d_0, \\ -r\text{sgn}(a), & a > d_0. \end{cases} \end{aligned}$$

3.2.2. *Design ESO.* The ESO is shown as

$$\begin{aligned} e &= z_1(k) - y(k), \\ z_1(k+1) &= z_1(k) + T[z_2(k) - \beta_{01}e], \\ z_2(k+1) &= z_2(k) + T[z_3(k) - \beta_{02}\text{fal}(e, \alpha_1, \delta) + bu(k)], \\ z_3(k+1) &= z_3(k) - T\beta_{03}\text{fal}(e, \alpha_2, \delta), \end{aligned} \quad (40)$$

where  $y(k)$  denotes the actual position  $x$ ;  $z_1$  is the estimated value of  $x$ ,  $z_2$  is the estimated value of  $\dot{x}$ , and  $z_3$  is the estimated value of total disturbance  $w_x$ ;  $u$  represents the virtual control  $u_1$  and  $b = 1$ ;  $\beta_{01}$ ,  $\beta_{02}$ ,  $\beta_{03}$ ,  $\alpha_1$  and  $\alpha_2$  are parameters.

$\text{fal}(\ast)$  is a non-linear function, and its expression is shown as

$$\text{fal}(e, \alpha, \delta) = \begin{cases} \frac{e}{\delta^{1-\alpha}}, & |e| \leq \delta, \\ |e|^\alpha \text{sign}(e), & |e| > \delta, \end{cases} \quad (41)$$

where  $\alpha$  and  $\delta$  are parameters, with  $0 < \alpha < 1$  and  $\delta > 0$ .

3.2.3. *Design State Error Feedback (SEF).* The position control does not have strict requirements for fast response, so a linear SEF with simple structure and easy parameter adjustment can be used. We have

$$u_0 = \beta_1 \text{fal}(e_1) + \beta_2 \text{fal}(e_2), \quad (42)$$

where  $\beta_1$  and  $\beta_2$  are parameters.

From equation (40), we compensate the disturbances observed by ESO, we get the total control output  $u_2$  of the channel  $x$ . With the same method, the output and of the other two channels can be obtained, and the required control amount can be obtained according to equation (25). The position controller structure is shown in Figure 6.

3.3. *Suspended Saw Controller.* In the Suspended Saw model,  $\phi_L$  and  $\theta_L$  describe the swing state of the suspended saw.  $\phi_L$  denotes the angle between the projection of the connecting rod in the  $O_b X_b Z_b$  plane and the  $z_b$  axis, the acceleration of the aerial Robot in the  $x$  and  $z$  axes will affect  $\phi_L$ . Similarly, the acceleration in the  $y$ -axis and  $z$ -axis directions will affect  $\theta_L$ . The height and horizontal position are generally not controlled at the same time, so the influence of the acceleration in the  $z$ -axis direction on the swing angle can be ignored. Considering that ARSS performs position control, it also needs to control swing angle, equation (36) is rewritten as

$$\begin{aligned} u'_1 &= u_1 - K_1 \dot{\phi}_L - K_2 \phi_L, \\ u'_2 &= u_2 - K_3 \dot{\theta}_L - K_4 \theta_L. \end{aligned} \quad (43)$$

When designing the position controller, the linear dynamic compensation method of ADRC is used to transform the non-linear model of the aerial Robot into a series model, that is,

$$\begin{aligned} \ddot{x} &= u'_1, \\ \ddot{y} &= u'_2. \end{aligned} \quad (44)$$

Then, combining equations (43) and (44) with equation (36), the system is described by

$$\begin{aligned} l\ddot{\phi}_L - K_1 \dot{\phi}_L + (g - K_2)\phi_L &= 0, \\ l\ddot{\theta}_L - K_3 \dot{\theta}_L + (g - K_4)\theta_L &= 0, \end{aligned} \quad (45)$$

where  $K_i, i = 1, \dots, 4$  are gains.

For this linear closed-loop system, sufficient and necessary conditions for its asymptotic stability are that all the characteristic roots are in the left half of the complex plane; that is, the real parts of all the characteristic roots are all negative. According to the Routh criterion, its characteristic roots are all in the left half of the complex plane and need to satisfy

$$\begin{aligned} l &> 0, \\ -K_1 &> 0, \\ g - K_2 &> 0, \\ -K_3 &> 0, \\ g - K_4 &> 0. \end{aligned} \quad (46)$$

As long as the parameter  $K_i$  satisfies, the swing angle of the hanging saw can be gradually stabilized.

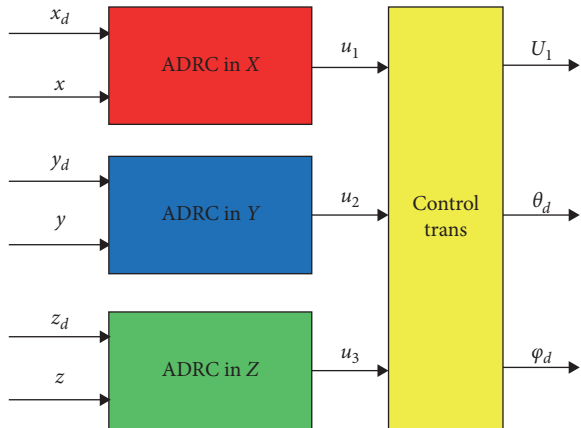


FIGURE 6: Position controller structure diagram.

## 4. Evaluation

In this section, we present our experiment. In each experiment, we simulate a complete tree-barrier removal task. Each task is divided into two stages: free flight and aerial pruning. The controller was tested in simulation, which used CoppeliaSim [23] as a physics environment and to model the Robot. The controller is programmed under MATLAB/Simulink. Table 1 summarizes the numerical values used to simulate, and Table 2 shows the controller parameters. The parameters in Table 2 are adjusted by a parameter self-learning algorithm CARLA-ADRC [24].

All experiments have considered numerical indicators, including integral of squared of tracking error, integral of squared of control signal. In the next section, we will choose one experiment sample to explain.

**4.1. Attitude Control Experiment.** First, verify the attitude controller and design the attitude tracking experiment. For example, the desired signals of roll, pitch, and yaw of  $10^\circ$  are given in the first, fourth, and seventh seconds, respectively. The experiment result is shown in Figure 7.

It can be seen from Figure 7 that the attitude controller can quickly track the desired signal, in which the tracking response time of pitch and roll angle is 1 s, the tracking response time of the yaw angle is 1.3 s, and there is no overshoot in each channel of the attitude angle. It proves that the attitude controller has good speed and stability.

**4.2. Free-Flight Experiments.** When performing the pruning task, we first fly ARSS to a desired position close to the tree. Suppose that in an experiment, the ARSS is controlled to fly to the desired position  $[2, 2, 1.5]^T \in \mathbb{R}^3$  in the inertial frame. At 0 s, give an 1.5 m step signal in the  $z$  axis; at 5 s, give a 2 m step signal in the  $x$ -axis; at 10 s, give a 2 m step signal in the  $y$ -axis, and then keep hovering. The simulation duration is 25 s in total. The curves of position and swing angle during the simulation are shown in Figure 8.

It can be seen from Figure 8(a) that because the swing of the saw cannot be too large, the aerial Robot cannot fly too fast, so the transition process of the TD link arrangement is

TABLE 1: Simulation parameters.

| Parameter | Value                 |
|-----------|-----------------------|
| $M$       | 8.5 kg                |
| $l$       | 0.4 m                 |
| $m$       | 1.0 kg                |
| $L$       | 1.0 m                 |
| $g$       | 9.81 m/s <sup>2</sup> |

TABLE 2: Controller parameters.

| ADRC              | $\phi$ | $\theta$ | $\psi$ | $x$    | $y$    | $z$    |
|-------------------|--------|----------|--------|--------|--------|--------|
| TD- $r$           | 3000   | 3000     | 3000   | 4      | 4      | 4      |
| TD- $h$           | 0.01   | 0.01     | 0.01   | 0.01   | 0.01   | 0.01   |
| ESO- $\beta_{01}$ | 1000   | 1000     | 1000   | 1000   | 1000   | 1000   |
| ESO- $\beta_{02}$ | 19764  | 19764    | 19764  | 19764  | 19764  | 19764  |
| ESO- $\beta_{03}$ | 462902 | 462902   | 462902 | 462902 | 462902 | 462902 |
| ESO- $\alpha_1$   | 0.5    | 0.5      | 0.5    | 0.5    | 0.5    | 0.5    |
| ESO- $\alpha_2$   | 0.25   | 0.25     | 0.25   | 0.25   | 0.25   | 0.25   |
| ESO- $\delta$     | 0.001  | 0.001    | 0.001  | 0.001  | 0.001  | 0.001  |
| ESO- $b$          | 1      | 1        | 1      | 1      | 1      | 1      |
| SEF- $c$          | 0.95   | 1.05     | 0.95   | —      | —      | —      |
| SEF- $r$          | 108.73 | 108.73   | 108.73 | —      | —      | —      |
| SEF- $h$          | 0.05   | 0.05     | 0.05   | —      | —      | —      |
| SEF- $k_1$        | —      | —        | —      | 50     | 50     | 400    |
| SEF- $k_2$        | —      | —        | —      | 41     | 41     | 50     |

relatively smooth. The response curve shows that the position error in the  $x$  and  $y$  axes is always kept within 0.4 m, and there is no overshoot; the error in the  $x$ -axis is kept within 0.05 m, and there is no overshoot. It is verified that the position controller has faster tracking speed and accuracy.

Figure 8(b) shows the tracking response curve of the swing angle. It can be seen that the swing angle is always controlled within  $6^\circ$ . The maximum swing angle appeared in 8 s and 13 s, respectively, reached  $5.7^\circ$ . This is because the deceleration occurs when the ARSS is close to the expected position at this time. After 2 s, the swing angle has dropped to within  $2^\circ$ , and when ARSS hovers, it can gradually stabilize to the equilibrium point and remain stable. This shows that the swing angle controller can well reduce the swing amplitude of the saw and can gradually stabilize after reaching the destination.

**4.3. Aerial Pruning Experiments.** The purpose of ARSS is to perform tree pruning operations. The pruning process is difficult to describe accurately, and the cutting process is not exactly the same for tree bars of different thickness. This experiment will simulate the cutting of a 6 cm diameter tree branch growing laterally to verify the controller. In the early stage, through the actual branch cutting experiment on the ground, the test results were counted and empirical conclusions were drawn: to cut a tree branch with a diameter of 6 cm, a squeezing force of 2 N is required between the saw and the branch, and contact with the tree barrier 2 s, the swing angle reaches  $-11.5$  deg. It can be calculated from this that the body needs to be 20 cm ahead of COG of the saw in the direction of advancement. Considering that the saw

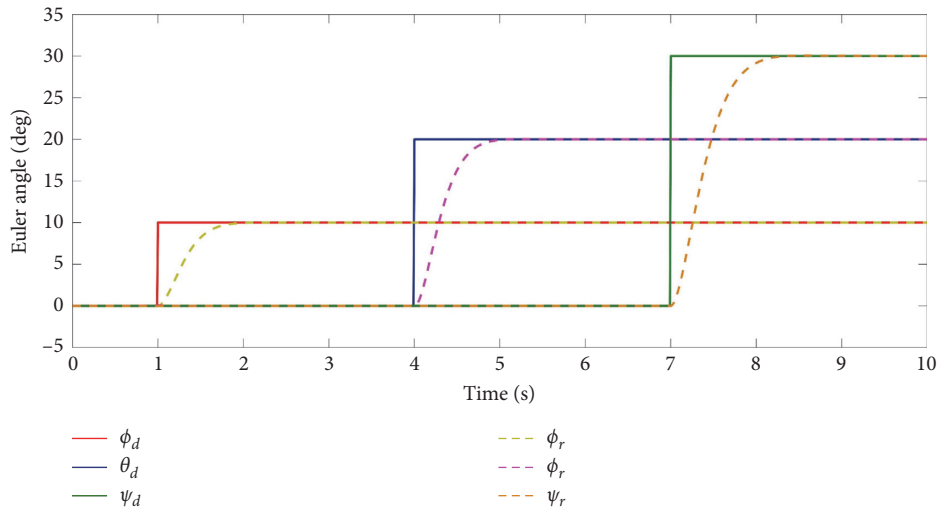
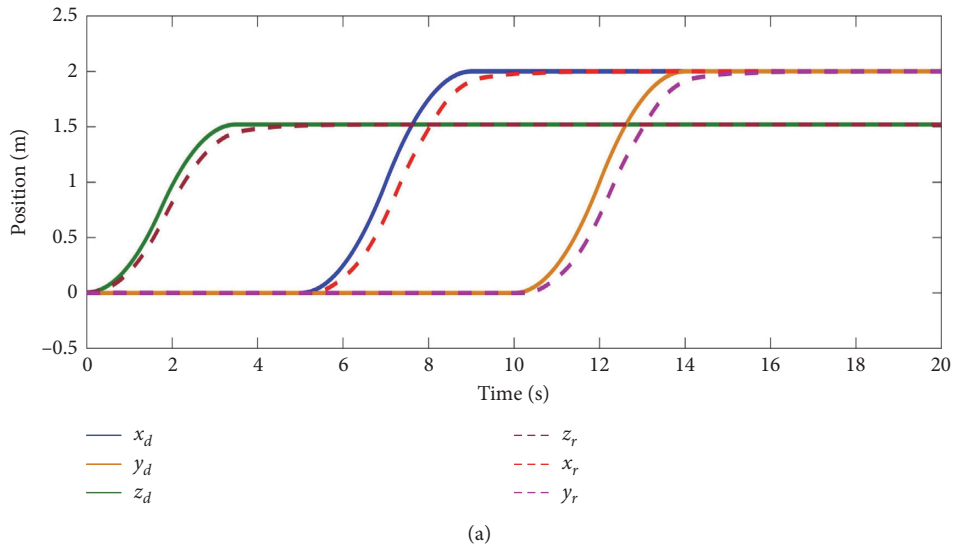
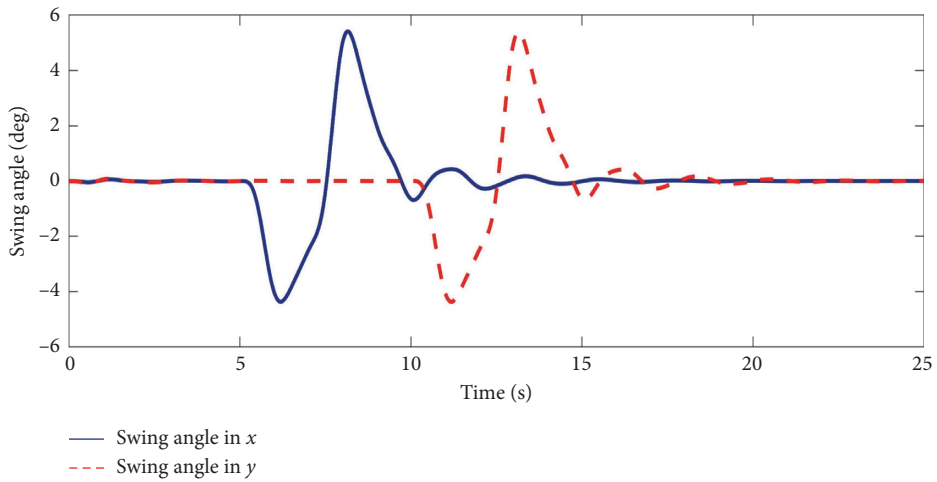


FIGURE 7: Attitude tracking response.



(a)



(b)

FIGURE 8: Position and Swing angle during simulation. (a) Position tracking response. (b) Swing angle tracking response.

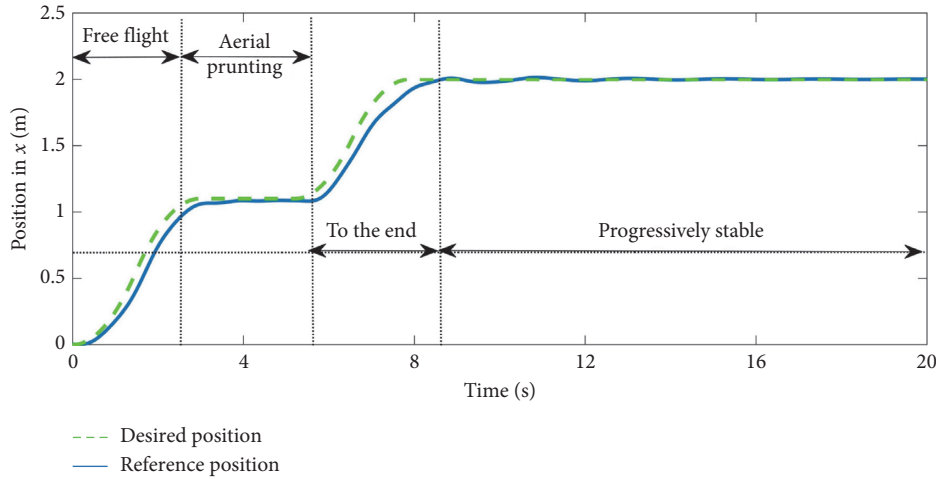


FIGURE 9: Position changes of ARSS during the pruning.

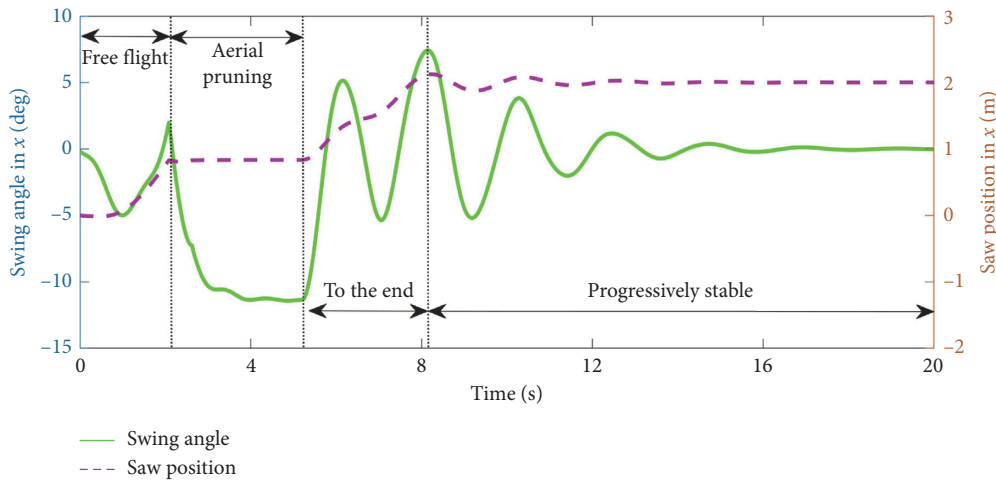


FIGURE 10: Swing angle changes of ARSS during the pruning.

diameter used in this article is 14 cm and the branch diameter is 6 cm, so during the cutting process, the flying platform needs to be 0.1 m in front of the tree branch. In this process, the saw will also receive a cutting force of 50 N.

Based on the above analysis, the cutting process can be approximated as follows: the saw is in continuous contact with the branch for 2 s at a swing angle of  $-11.5^\circ$  and disturbed by a cutting force of 50 N. The experimental process is designed as follows: the initial state of ARSS is hovering, and a tree barrier is placed at a distance of 1 m from the ARSS. On the 0 s, give a step signal of 1.1 m in the  $x$ -axis and set the desired swing angle to  $-11.5^\circ$ . After the aerial Robot reaches the desired position, hover for 2 s and then perform the cutting operation. During this process, a cutting force of 50 N is applied to the saw, and then a step signal of 2 m on the  $x$ -axis is given and the desired swing angle is set to  $0^\circ$ , leaving the Working area. The entire simulation process lasts 20 s. This test mainly tests the position of the  $X$  direction and the performance of the swing angle controller, so only the curve of the  $x$ -axis position and

the swing angle during the simulation process is given, as shown in Figures 9 and 10.

In order to more intuitively explain the simulation results, the simulation process is divided into four stages, as shown in Figure 11.

From Figure 11, ARSS made contact with the tree barrier in 2.2 s, and the swing angle before contact was  $2^\circ$  (in Figure 11(a)). After the contact, due to the tree barrier blocking the saw, the saw stayed at the position of 1 m, while the flying platform continued to fly, reached 1.1 m in the 3.2 s and hovered, the swing angle gradually increased to  $-11.5^\circ$ , and keep the angle until 5.2 s (in Figure 11(b)). After the pruning is completed, due to the loss of the barrier to the saw, the saw quickly swings forward. Under the action of the swing angle controller, it reaches a maximum of  $7^\circ$ , and stays within  $7^\circ$  when going to the destination (in Figure 11(c)), and it can gradually stabilize after reaching the destination (in Figure 11(d)). In terms of position control, the Robot can always quickly track the desired position, and the

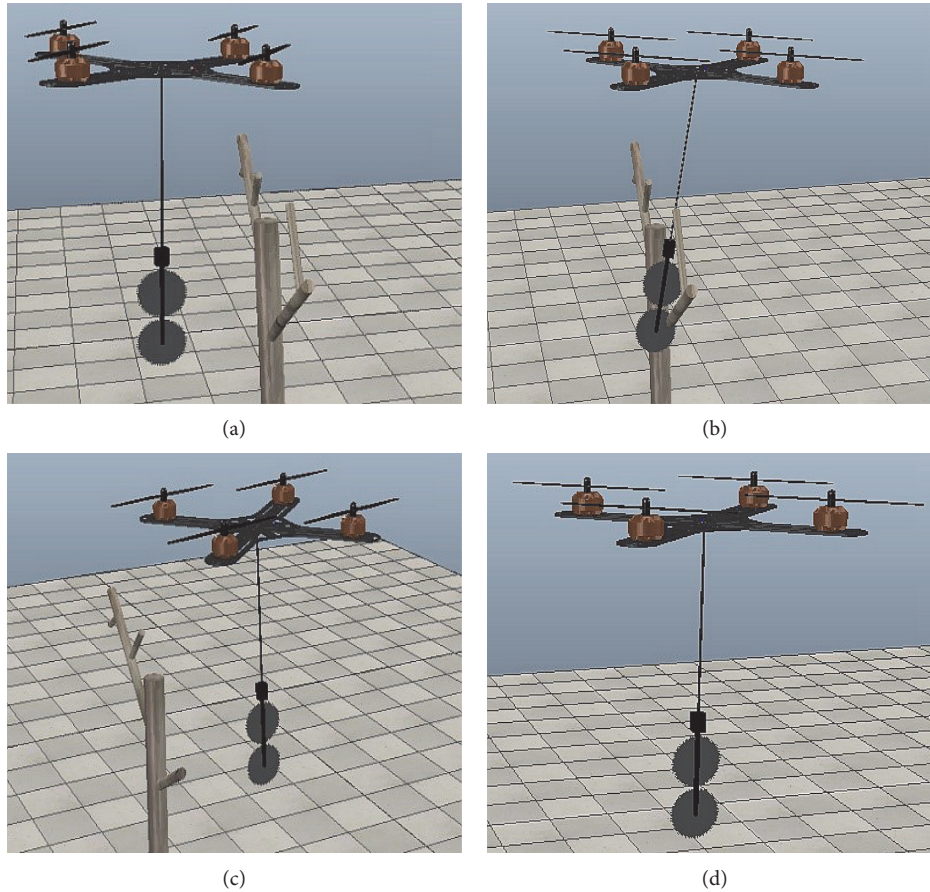


FIGURE 11: Snapshots of ARSS during the pruning.

maximum tracking error is within 0.4 m. After reaching the destination, due to the need to control the swing angle, there is an overshoot of 0.1 m, which can gradually stabilize after a period of time. The experiments demonstrated that the position controller and the swing angle controller can complete the task of aerial pruning well.

## 5. Discussion and Conclusions

Throughout the development of ARSS, a range of issues were encountered on which we are developing a novel aerial platforms. While building the novel robot for the tree pruning has a range of advantages, it also has the following limitations:

- (1) Physical testing still needs to be carried out in future research.
- (2) The control of the swing angle is linearized by a Small-Angle-Approximation to the dynamic model, and the model uncertainty cannot be completely solved.
- (3) The knife and saw will generate force after contact with the tree. The research does not directly control the force, but indirectly realizes the force control through the control of the position, attitude and swing angle.

However, the article introduces how a novel aerial robot can realize flying motion and aerial pruning and designs a position and attitude controller based on the ADRC method, and a swing angle controller via the linear SEF method. The methodology not only realize the fast, accurate, and stable flight movement to the specified position, but also effectively suppress the swing of the hanging load and the residual oscillation after reaching the specified position. The control effect is remarkable, with good speed and robustness.

## Data Availability

The figures, tables, and other data used to support this study are included within the article.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## Acknowledgments

This work was supported by the Guizhou Provincial Science and Technology Projects (Grant no. (2020)2Y044), the Science and Technology Projects of China Southern Power Grid Co., Ltd. (Grant no. 066600KK52170074), the Key Laboratory Projects of Aeronautical Science Foundation of

China (Grant no. 201928052006), and the National Natural Science Foundation of China (Grant no. 61473144).

## References

- [1] R. Mahony, V. Kumar, and P. Corke, "Multirotor aerial vehicles: modeling, estimation, and control of quadrotor," *IEEE Robotics & Automation Magazine*, vol. 19, no. 3, pp. 20–32, 2012.
- [2] X. Liang, Y. Fang, N. Sun, and H. Lin, "A novel energy-coupling-based hierarchical control approach for unmanned quadrotor transportation systems," *IEEE/ASME Transactions on Mechatronics*, vol. 24, no. 1, pp. 248–259, 2019.
- [3] D. Yallappa, M. Veerangouda, D. Maski, V. Palled, and M. Bheemanna, "Development and evaluation of drone mounted sprayer for pesticide applications to crops," in *Proceedings of the GHTC 2017—IEEE Global Humanitarian Technology Conference*, pp. 1–7, San Jose, CA, USA, October 2017.
- [4] J. Molina and S. Hirai, "Pruning tree-branches close to electrical power lines using a skew-gripper and a multirotor helicopter," in *Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM*, pp. 1123–1128, Munich, Germany, July 2017.
- [5] J. Molina and S. Hirai, "Aerial pruning mechanism, initial real environment test," *Robotics and Biomimetics*, vol. 4, p. 15, 2017.
- [6] J. Molina and S. Hirai, "Kinematic analysis of a novel skew-gripper for aerial pruning tasks," in *Proceedings of the 3rd International Conference on Mechatronics and Robotics Engineering*, pp. 134–138, Paris, France, February 2017.
- [7] Q. Zhang, Z. Yang, S. Wang, and Y. Jiang, "Attitude controller design for the aerial trees-pruning robot based on nonsingular fast terminal sliding mode," *International Journal of Advanced Robotic Systems*, vol. 16, no. 4, pp. 1–14, 2019.
- [8] N. Azami, P. Zarafshan, A. M. Kermani, M. Khashehchi, and S. Kouravand, "Design and analysis of an armed-ocrotor to prune trees near the power lines," in *Proceedings of the International Conference of Iranian Aerospace Society*, Tehran, Iran, February 2017.
- [9] K. Klausen, T. I. Fossen, and T. A. Johansen, "Suspended load motion control using multicopters," in *Proceedings of the 2014 22nd Mediterranean Conference on Control and Automation, MED 2014*, pp. 1371–1376, Palermo, Italy, June 2014.
- [10] K. Sreenath, N. Michael, and V. Kumar, "Trajectory generation and control of a quadrotor with a cable-suspended load—a differentially-flat hybrid system," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 4888–4895, Karlsruhe, Germany, May 2013.
- [11] Y. Allothman and D. Gu, "Quadrotor transporting cable-suspended load using iterative linear quadratic regulator (iLQR) optimal control," in *Proceedings of the 2016 8th Computer Science and Electronic Engineering Conference, CEEC 2016*, pp. 168–173, Colchester, UK, September 2016.
- [12] A. Faust, I. Palunko, P. Cruz, R. Fierro, and L. Tapia, "Learning swing-free trajectories for UAVs with a suspended load," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 4902–4909, Karlsruhe, Germany, May 2013.
- [13] I. Palunko, R. Fierro, and P. Cruz, "Trajectory generation for swing-free maneuvers of a quadrotor with suspended payload: a dynamic programming approach," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 2691–2697, Saint Paul, MN, USA, May 2012.
- [14] S. Yang and B. Xian, "Energy-based nonlinear adaptive control design for the quadrotor UAV system with a suspended payload," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 3, pp. 2054–2064, 2020.
- [15] S. Yang, L. Kunqin, and J. Shi, "Design and simulation of the longitudinal autopilot of UAV based on self-adaptive fuzzy PID control," in *Proceedings of the CIS International Conference on Computational Intelligence and Security*, pp. 634–638, Beijing, China, December 2009.
- [16] J.-B. Liu, S. Wang, C. Wang, and S. Hayat, "Further results on computation of topological indices of certain networks," *IET Control Theory & Applications*, vol. 11, no. 13, pp. 2065–2071, 2017.
- [17] J.-B. Liu, J. Cao, A. Alofi, A. AL-Mazrooei, and A. Elaiw, "Applications of Laplacian spectra for  $n$ -prism networks," *Neurocomputing*, vol. 198, pp. 69–73, 2016.
- [18] J. Han, "From PID to active disturbance rejection control," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 3, pp. 900–906, 2009.
- [19] X. Yang and Y. Huang, "Capabilities of extended state observer for estimating uncertainties," in *Proceedings of the American Control Conference*, pp. 3700–3705, St. Louis, MO, USA, June 2009.
- [20] Q. Zheng, L. Q. Gao, and Z. Gao, "On validation of extended state observer through analysis and experimentation," *Journal of Dynamic Systems, Measurement, and Control*, vol. 134, no. 2, pp. 1–6, 2012.
- [21] B.-Z. Guo and Z.-l. Zhao, "On the convergence of an extended state observer for nonlinear systems with uncertainty," *Systems & Control Letters*, vol. 60, no. 6, pp. 420–430, 2011.
- [22] Z. X. Zhou, Y. Y. Qu, J. D. Yang, and D. J. Liu, "Suspension position control of precise planar magnetic bearing," *Kongzhi yu Juece/Control and Decision*, vol. 25, no. 3, pp. 437–444, 2010.
- [23] CoppeliaSim: <https://www.coppeliarobotics.com/>.
- [24] L. Wu, H. Bao, J. L. Du, and C. S. Wang, "A learning algorithm for parameters of automatic disturbances rejection controller," *Zidonghua Xuebao/Acta Automatica Sinica*, vol. 40, no. 3, pp. 556–560, 2014.