

## Article

# Design of Airport Obstacle-Free Zone Monitoring UAV System Based on Computer Vision

Liang Wang , Jianliang Ai, Li Zhang and Zhenlin Xing 

Department of Aeronautics and Astronautics, Fudan University, Shanghai 200433, China; aijl@fudan.edu.cn (J.A.); 18210290016@fudan.edu.cn (L.Z.); 18110290005@fudan.edu.cn (Z.X.)

\* Correspondence: 18110290012@fudan.edu.cn

Received: 24 March 2020; Accepted: 25 April 2020; Published: 27 April 2020



**Abstract:** In recent years, a rising number of incidents between Unmanned Aerial Vehicles (UAVs) and planes have been reported at airports and airfields. A design scheme for an airport obstacle-free zone monitoring UAV system based on computer vision is proposed. The system integrates the functions of identification, tracking, and expelling and is mainly used for low-cost control of balloon airborne objects and small aircrafts. First, a quadcopter dynamic model and 2-Degrees of Freedom (2-DOF) Pan/Tilt/Zoom (PTZ) model are analyzed, and an attitude back-stepping controller based on disturbance compensation is designed. Second, a low and slow small-target self-identification and tracking technology is constructed against a complex environment. Based on the You Only Look Once (YOLO) and Kernel Correlation Filter (KCF) algorithms, an autonomous target recognition and high-speed tracking plan with great robustness and high reliability is designed. Third, a PTZ controller and automatic aiming strategy based on Anti-Windup Proportional Integral Derivative (PID) algorithm is designed, and a simplified, automatic-aiming expelling device, the environmentally friendly gel ball blaster, which features high speed and high accuracy, is built. The feasibility and stability of the project can be verified through prototype experiments.

**Keywords:** quadcopter; computer vision; image processing; target identification; trajectory tracking

## 1. Introduction

The control of air-floating objects and small low-slow aircrafts means monitoring and expelling targeted air-floating object and small low-slow aircraft through technical methods and devices. With the rapid growth of the consumer-grade Unmanned Aerial Vehicle (UAV) market, problems of illegal flying bring about risks to security and privacy. Between 19 and 21 December 2018, hundreds of flights were cancelled at Gatwick Airport near London, UK, following reports of drone sightings close to the runway. The reports caused major disruption, affecting approximately 140,000 passengers and 1000 flights [1]. Accidents caused by drones are more and more frequent, so it is necessary to regulate the use of drones. It is of great significance to deploy anti-UAV and anti-aircraft drift target systems in the airport obstacle-free zone. However, UAV reconnaissance is a challenging task due to their small size and low flying speed. Many technologies such as radar monitoring, audio monitoring, video monitoring, and Radio Frequency (RF) monitoring have the potential to detect and locate drones. Each technique has its advantages and disadvantages.

1. Radar: Radar is mainly used for the measurement and tracking of large aircraft, whereas UAVs usually fly at low altitude and low speed. Aleksander Nowak et al. introduced a method of fast, simultaneous calibration using many mobile Frequency Modulated Continuous Wave (FMCW) radars operating in the network; this is used in the anti-drone scheme [2]. Multerer et al. presented an anti-drone system that consists of a three-dimensional (3D) FMCW Multiple Input

Multiple Output (MIMO) radar and a directional jammer working at 2.4 GHz [3]. Because the radar reflection section of the drone is small and the clutter reflected at low altitude is higher, the monitoring effect of drone is limited.

2. **Audio:** During the flight of the UAV, the motors and propellers will make sound, and the UAV will be detected, classified, and located by a system equipped with acoustic sensors. Lee et al. estimated the Direction of Arrival (DOA) of the drone by using the acoustic signal generated when the drone is flying [4]. In the vicinity of the airport, noise from aircraft engines and vehicles is very loud, so it is difficult to accurately distinguish the audio of UAV from the high ambient noise.
3. **Video:** Video-based detection is essentially a target detection problem in the field of computer vision and pattern recognition. Objects can be detected according to their appearance features, that is, color, contour, geometric shape, or edge [5], and their motion characteristics across successive frames [6]. You Only Look Once (YOLO) is a convolutional neural network that can predict multiple box locations and categories at one time, and can realize end-to-end target detection and recognition. Its greatest advantage is its speed, which makes it suitable for applications in the field of flight target monitoring. Ju et al. proposed a multiscale target detection approach based on YOLO V3 [7]. The YOLO algorithm is also widely used in pedestrian [8], vehicle [9], fire [10], and obstacle [11] identification, and in sports [12], medicine [13] and other monitoring fields. For airport obstacle-free zone monitoring, the combination of a flight platform, YOLO, and a target expelling device is very promising, and can effectively control and expel illegal flying targets. On the other hand, although target detection and tracking based on video images has been well studied, there are still many problems, such as the running rate, which is also a problem in target detection, tracking, and expelling.
4. **RF:** RF interference blocking mainly applies signal interference, acoustic interference, and other technologies to force the target aircraft to lose control of the signal. The method is only applicable to the controlled target aircraft, not to the automatic aircraft, and floating objects in the air can cause great collateral damage. When the RF signal is interrupted, the jammer is usually required to guess at the band on which the signal is running. The typical range from 2.4 to 5.0 MHz means that any technology operating within this range would also be disrupted. In an airport environment, this would be inconvenient and potentially catastrophic. It could cause unintended Electromagnetic Interference (EMI) and affect the performance of air navigation services equipment on the ground and/or onboard aircraft equipment.

Table 1 summarizes the above monitoring techniques. The detection range of the target in the table is obtained from the literature and may vary with the type of target, monitoring environment, hardware parameters, and corresponding algorithm.

**Table 1.** Comparison of monitoring technologies.

Monitoring Technologies	Drone Signature	Localization/Tracking Method	Detection Range	Can it Expel Targets?	Challenges
Radar	Micro-Doppler	Doppler-based tracking delay-based localization	$\leq 3000$ m	NO	Low radar cross-section, low speed and altitude
Audio	Time-frequency feature	DOA-based localization	40–300 m	NO	High ambient noise
Video	Appearance feature motion feature	Motion-based tracking	100–1000 m	NO	Occlusion indistinguishable small object
RF	Communication channel	Received-signal-strength (RSS)/DOA-based localization	$\leq 1000$ m	YES	EMI

Motivated by the above observations, UAV detection and expelling systems should be developed so they do not adversely impact or interfere with safe airport operations, air traffic control, and other air navigation services, or the safe and efficient operation of the National Airspace System (NAS).

The quadcopter UAV features flexible maneuverability and vertical take-off and landing, and is widely used in various fields, making it an excellent air flight platform. However, due to the restrictions of its structure and weight, the limited airborne data processing capacity, and the variety of weather conditions that can be encountered in flight, the design of the integrated identification, tracking, and expelling quadcopter UAV system based on computer vision faces a lot of challenges. Modeling and control, motion planning, perception, and mechanism design are crucial for aerial manipulations [14–16]. There are some challenges for UAVs when performing autonomous vision-based integrating identification, tracking, and expelling. These problems mainly come from the following aspects: (1) Due to the influence of the UAV structure strength, power configuration, flight altitude, ambient temperature and other factors, the control system design is complicated; (2) the airborne minicomputer has a limited data processing capacity; (3) the low-cost single-chip microcomputer is used to realize high-rate visual control and trigger mechanism control; (4) The flight environment of the UAV is complicated and there are many factors that restrict the efficiency. Motivated by these problems, we have systematically studied one UAV's integrated identification, tracking, and expelling functions based on computer vision.

The contributions of this paper are as follows:

1. It has analyzed the influence of parameters such as the quadcopter frame construction, motor, and electronic speed controller, and built a quadcopter dynamic model. Considering the uncertainties of the system model, the external disturbance, and the load variation of the quadrotors, this paper designed a back-stepping attitude controller based on the disturbance compensation. It has analyzed the performance of steering gear parameters and built a 2-Degrees of Freedom (2-DOF) Pan/Tilt/Zoom (PTZ) model.
2. The low-cost single chip microcomputer Arduino (Scarmagno, Italy) was used to communicate with the computer vision module and the underlying flight control system through the serial port was used to realize high-rate visual control and PTZ control.
3. It has proposed an image processing solution based on the airborne low-cost Logitech camera (Romanel, Lausanne, Switzerland), computer Jetson nano (Santa Clara, CA, USA), and based on the YOLO algorithm, Kernel Correlation Filter (KCF) algorithm, OpenCV (Santa Clara, CA, USA) open-source computer vision library, Ubuntu 16.04 (Isle of Man, UK) operating system, and Python3 (Beaverton, OR, USA). This paper deployed the image processing environment to realize accurate and fast airborne image recognition, tracking, and locating. Based on the shooting PTZ controller designed by the Anti-Windup PID algorithm, high-accuracy aiming and fast trajectory correction were realized. Through 3D printing technology, the simulated simple launch device was designed for experimental verification.
4. By combining the ultrasonic sensors (Shenzhen, China), Lidar (Shanghai, China), GPS (Shenzhen, China) and the infrared distance measuring sensors, based on multisensor information fusion technology, this paper has constructed the UAV real-time obstacle avoidance solution centering on fuzzy control. In this process, after setting the input and output, this paper took corresponding measures to simplify the processing, formulate corresponding fuzzy rules, complete the relevant design of a fuzzy controller, and test the real-time obstacle avoidance function.

The rest of the paper is organized as follows. Section 2 gives the system overview and mathematical modeling. In Section 3, the perception and control are described. Experimental results are presented in Section 4. Concluding remarks and future work are discussed in Section 5.

## 2. System Overview and Mathematical Modeling

### 2.1. Overall System Architecture Design

This paper proposes a new identification, tracking and expelling system, which can realize fully automatic, over the horizon and low-cost expulsion of a target object in flight. The design framework is divided into four parts: the underlying flight control system, the target fast recognition and tracking system, the aiming and PTZ control system, and the obstacle avoidance control system. The UAV consists of a quadcopter frame with a 680 mm wheelbase,  $15 \times 7$  carbon fiber propellers (Shanghai, China) and 390 kV 5008 brushless motors (Shanghai, China). It is modeled with Pro/E to obtain the UAV structure model. The PIXHAWK pilot (Berkeley, CA, USA), as the bottom flight controller, runs the self-stabilization algorithm. The Jetson Nano development board is the core image processing unit, while the Arduino development board 1 is the core unit of the aiming design control system, and the Arduino development board 2, single line lidar, and ultrasonic sensor are the core units of the obstacle avoidance control system. They are shown in Figures 1–3.

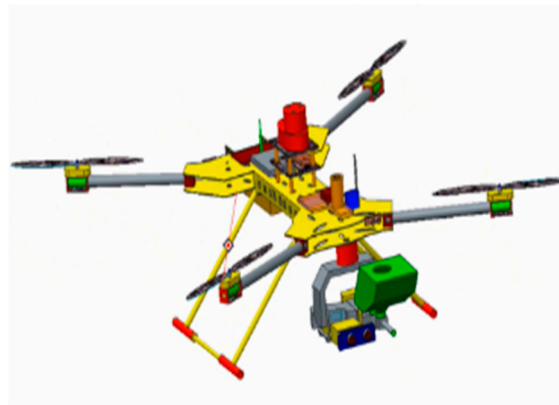


Figure 1. 3D modeling.

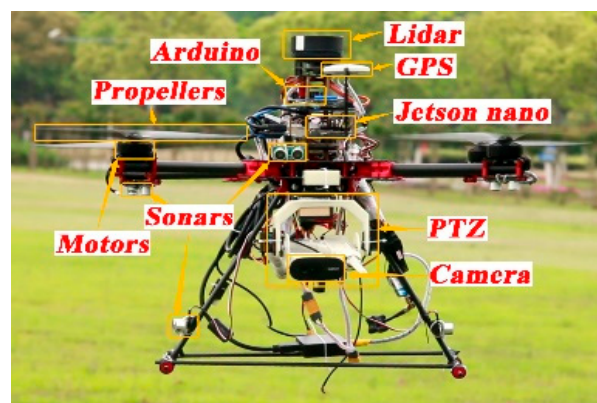


Figure 2. Photo of prototype.



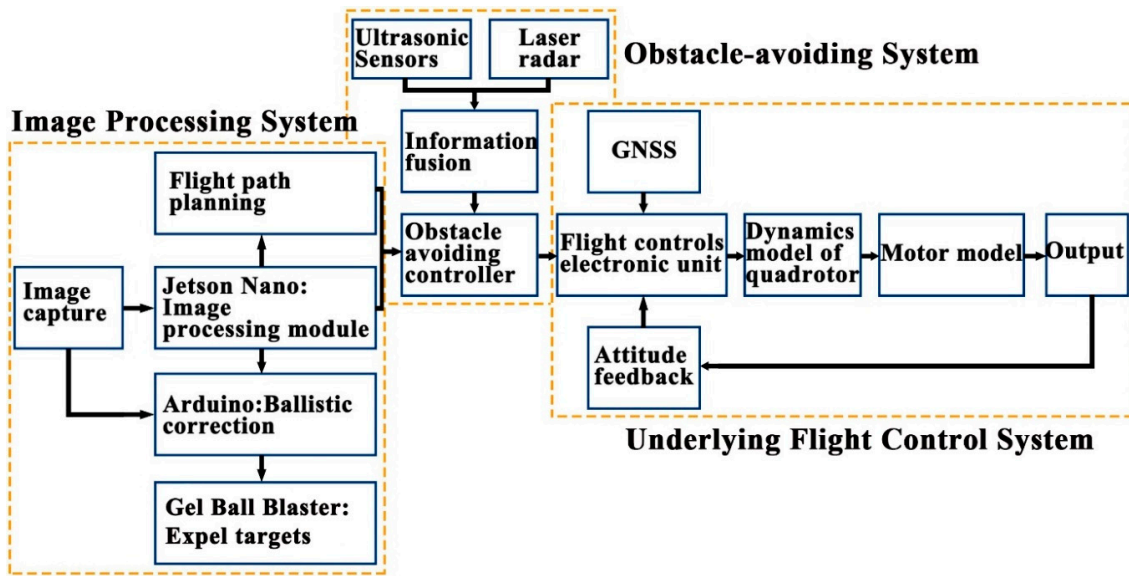


Figure 3. System architecture.

## 2.2. Quadcopter Dynamic Model

An accurate and reliable mathematical model is the foundation for designing a flight platform. The design and study of the flight control system have been carried out conveniently thanks to the dynamic motion simulation platform. By taking the actual flight environment into account, realistic assumptions have been proposed in this section based on the analysis of rotor aerodynamics. Equations of kinematics and dynamics that can be used for accurately describing the flight motion of a UAV (Unmanned Aerial Vehicle) have been derived from the UAV flight principle and the overall force analysis.

First of all, our assumptions are given as follows:

1. Assuming that the earth is a standard sphere, the reference ground is a standard horizontal plane and we can ignore the effect of the Earth's rotation.
2. The quadcopter body is considered to be rigid and we can ignore its elastic deformation.
3. The air drag during the quadcopter flight is proportional to the square of the flight velocity, while the air drag direction is opposite to the velocity direction.

### 2.2.1. Quadcopter Aerodynamic and Torque Analysis

During the quadcopter flight, forces acting on the quadcopter mainly include the lift and drag generated by the rotor, the gravity of the quadcopter, and the frictional drag generated by the quadcopter during flight.

The torque acting on it is primarily composed of aerodynamic torque, torque, and the rolling torque generated by the rotor, as well as the gyroscopic effect torque generated during the rotation of the rotor. The force analysis is shown in Figure 4.

Hence, the external forces received during the quadcopter flight are as follows:

$$F_{SUM} = -k_{tr}\vec{V}^2 + m\vec{g} + \sum_{i=1}^4 (\vec{T}_i + \vec{D}_i), \quad (1)$$

where  $k_{tr}$  represents the frictional drag coefficient between quadcopter and air;  $V$  is its flight speed;  $T_i$  is the lift force generated by the rotor No.  $i$ ; and  $D_i$  is the drag generated by the rotor No.  $i$ .

External torques received are presented as follows:

$$M_{SUM} = (T_2 + T_3 - T_1 - T_4)l_x\vec{x} + (T_1 + T_2 - T_3 - T_4)l_y\vec{y} + \sum_{i=1}^4 (g_{ix}\vec{x} + g_{iy}\vec{y}) + \sum_{i=1}^4 (Q_i\vec{z} + \vec{L}_i + \vec{D}_i h) \quad (2)$$

where  $l_x$  and  $l_y$  are the horizontal distances between the rotor center and the quadcopter centroid, respectively, as shown in Figure 4;  $h$  represents the vertical distance between the rotor center and the quadcopter centroid;  $g_{ix}$  and  $g_{iy}$  are the gyroscopic effect stress torques generated when the rotor No.  $i$  is rotated; and  $J_r$  is the moment of inertia of the motor and the rotor rotating group winding around the motor shaft.

When four motors are applied on the  $x$  and  $y$  axes of the body coordinate system, the total gyroscopic effect torque is presented as follows:

$$Ge_t = \begin{bmatrix} \sum_{i=1}^4 g_{ix} & \sum_{i=1}^4 g_{iy} & \sum_{i=1}^4 g_{iz} \end{bmatrix}^T = \begin{bmatrix} J_r q(\omega_1 + \omega_3 - \omega_2 - \omega_4) \\ -J_r p(\omega_1 + \omega_3 - \omega_2 - \omega_4) \\ 0 \end{bmatrix} \quad (3)$$

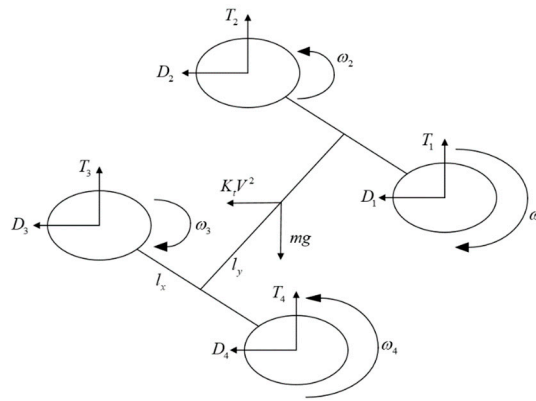


Figure 4. Quadcopter force analysis diagram.

## 2.2.2. Definition of Coordinate System

In order to establish a state equation that can accurately describe quadcopter motion, a reasonable coordinate system should be selected with reference to specific coordinate systems for describing the relevant parameters of quadcopter linear and angular motions. This is not only a critical step in establishing the quadcopter model, but also an essential part of designing and studying the quadcopter flight control system. For better describing the quadcopter motion state, two rectangular coordinate systems are presented as follows:

### 1. Ground coordinate system $C_E$

The ground coordinate system  $C_E : (O_E, x_E, y_E, z_E)$  is defined.  $O_E$  is the origin of the coordinates; the positive direction of  $x_E$  points to the east; the positive direction of  $y_E$  points to the south; and the positive direction of  $z_E$  is determined to be perpendicular to the horizontal plane, pointing to the geocenter according to the right-hand rule.

### 2. Body coordinate system $C_B$

The body coordinate system is defined as  $C_B : (O_B, x_B, y_B, z_B)$ , which is connected with the quadcopter.  $O_B$  is the origin of coordinates, which coincides with the center of the quadcopter (all devices are included) and with  $O_E$  at the initial moment. The  $x_B$  axis is in the plane that passes through the centroid and is perpendicular to the connection between motors No. 1 and No. 2, pointing to the

front of the body. The  $y_B$  axis is in the plane that passes through the centroid and is perpendicular to the connection between motors No. 2 and No. 3, pointing to the right of the body. The  $z_B$  axis is in the vertical plane passing through the centroid, pointing to the lower part of the body.

The flight motion of the quadcopter involves linear motion in three directions and angular motion winding around three axes. To be specific, the linear motion should be described in the ground coordinate system, while the angular motion should be described in the body coordinate system. In that case, the ground coordinate system and the body coordinate system should be used simultaneously in the quadcopter system modeling for describing and modeling the flight motion.

The ground coordinate system  $C_E$  and the body coordinate system  $C_B$  used in the quadcopter system modeling are presented in Figure 5. It should be noted that the origins of the two coordinate systems are coincident in the initial state. Moreover, the quadcopter plane simplified structure is presented in Figure 5 to precisely demonstrate the significance of various parameters in the quadcopter model.

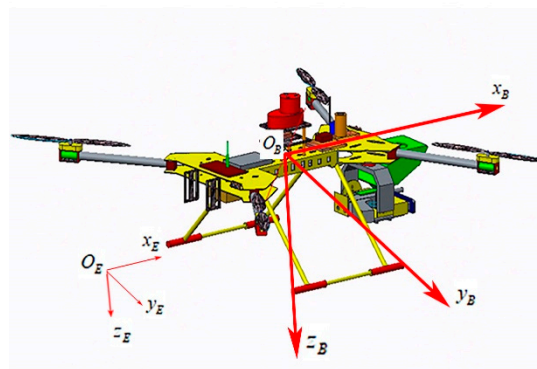


Figure 5. Quadcopter coordinate system.

$l_{txi}$  and  $l_{tyi}$  represent the distance between the motor No. $i$  and the  $x_B$  axis and the  $y_B$  axis, respectively, in the quadcopter plane simplified structure shown in Figure 6. In general, the center of a multirotor quadcopter is assumed to be at the geometric center for modeling the multirotor quadcopter in the majority of the existing literature, obtaining  $l_{tx1} = l_{tx2} = l_{tx3} = l_{tx4}$  and  $l_{ty1} = l_{ty2} = l_{ty3} = l_{ty4}$ . As a matter of fact, the center position deviates from the geometric center, affecting the control effect of the multirotor quadcopter flight control system. Therefore, considering that the center position deviates from the geometric center, the distance between each motor and the  $x_B$  axis and the  $y_B$  axis possesses the following relationship:  $l_{tx1} = l_{tx4}$ ,  $l_{tx2} = l_{tx3}$ ,  $l_{ty1} = l_{ty2}$ , and  $l_{ty3} = l_{ty4}$ .

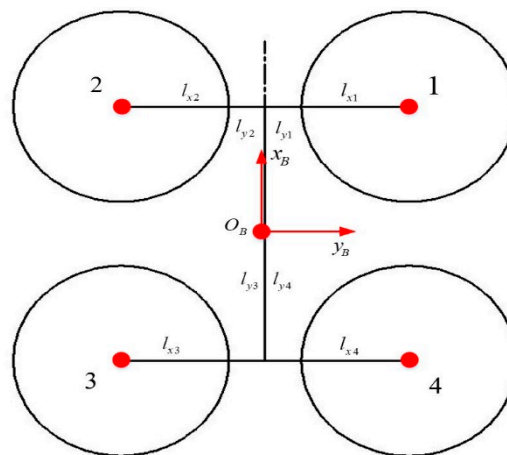


Figure 6. Quadcopter plane: simplified structure.

Based on the quadcopter coordinate system shown in Figure 5, the attitude angle of quadcopter can be defined by the relationship between the body coordinate system  $C_B$  and the ground coordinate system  $C_E$ . The attitude angle of the quadcopter in this paper can be defined as follows:

1. Roll angle  $\phi$ : It is the angle between the body axis  $O_B y_B$  and the horizontal plane, and the right roll is positive.
2. Pitch angle  $\theta$ : It is the angle between the body axis  $O_B x_B$  and the horizontal plane, and the up is positive.
3. Yaw angle  $\psi$ : It is the angle between the projection of the body axis  $O_B x_B$  on the horizontal plane and the ground coordinate system axis  $O_E x_E$ , and the right yaw is positive.

### 2.2.3. Euler Angles

The body coordinate system  $C_B$  can be considered as the coordinate system, being fixed to the quadcopter body in the ground coordinate system  $C_E$ .  $p = [x, y, z]^T$  can be employed to present the position vector of the quadcopter centroid. What is more, the three attitude angles  $a = [\phi, \theta, \psi]^T$  of the quadcopter can represent the position vector of the quadcopter attitude.

The conversion matrix converting from the body coordinate system to the ground coordinate system is:

$$R_{BE} = R_\psi R_\theta R_\phi = \begin{bmatrix} \cos \theta \cos \psi & \sin \phi \sin \theta \cos \psi - \cos \phi \sin \psi & \cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi \\ \cos \theta \sin \psi & \sin \phi \sin \theta \sin \psi + \cos \phi \cos \psi & \cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi \\ -\sin \theta & \sin \phi \cos \theta & \cos \phi \cos \theta \end{bmatrix} \quad (4)$$

### 2.2.4. Kinetic Equations

The multicopter quadcopter flight motion in the air can be decomposed into the quadcopter translation relative to the ground coordinate system and the rotation relative to the body coordinate system.

The translational motion equation of quadcopter is presented as follows:

$$\begin{cases} \dot{p}_t = v_t \\ \dot{v}_t = g \vec{z}_E - \frac{1}{m_t} R_{BE} F \end{cases} \quad (5)$$

where  $p_t = [x_t, y_t, z_t]^T$  and  $v_t = [u_t, v_t, w_t]^T$  represent the position and velocity of the quadcopter in the ground coordinate system, respectively;  $m_t$  is the quadcopter mass;  $g$  is the acceleration of gravity;  $\vec{z}_E = [0, 0, 1]^T$  is the unit vector in the ground coordinate system;  $R_{BE}$  is the conversion matrix from the body coordinate system to the ground coordinate system; and  $F$  is the combined external forces acting on quadcopter, except for gravity.

The attitude motion equation of quadcopter is presented as follows:

$$\begin{cases} \dot{a}_t = \Phi_t(a_t) \omega_t \\ J_t \dot{\omega}_t = -(\omega_t \times J_t \omega_t) + G e_t + M \end{cases} \quad (6)$$

where  $a_t = [\phi_t, \theta_t, \psi_t]^T$  represents the attitude angle of quadcopter in the ground coordinate system;  $\Phi_t(a_t)$  is the conversion matrix between the angular velocity in the body coordinate system and the Euler angular velocity in the ground coordinate system;  $\omega_t = [p_t, q_t, r_t]^T$  is the component of the rotation angular velocity of the body coordinate system on the axis of the body coordinate system relative to the ground coordinate system;  $J_t$  is the quadcopter rotational inertia matrix;  $G e_t$  is the rotor-gyroscopic effect stress torque, as shown in Equation (3); and  $M$  is the combined external torque acting on the quadcopter, except for the gyroscopic effect stress torque.

By combining Equations (5) and (6), the six-degree-of-freedom nonlinear mathematical model of quadcopter can be obtained:

$$\begin{cases} \dot{p}_t = v_t, \dot{v}_t = g\vec{z}_E - \frac{1}{m_t}R_{BE}F \\ \dot{a}_t = \Phi_t(a_t)\omega_t, J_t\dot{\omega}_t = -(\omega_t \times J_t\omega_t) + Ge_t + M \end{cases} \quad (7)$$

### 2.3. PTZ Modeling

#### 2.3.1. Target Expulsion Device

The ejection device of the launch system adopts the idea of a gel ball blaster, and is manufactured by 3D printing, with the features shown in Figures 7 and 8. It is composed of a high-speed motor, a spring, a piston, a cylinder, a gear set, and a power supply. The launcher uses safe and environmentally friendly gel balls to expel targets, as shown in Figure 9.

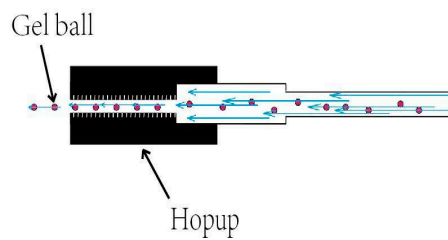


Figure 7. Gel ball blaster.

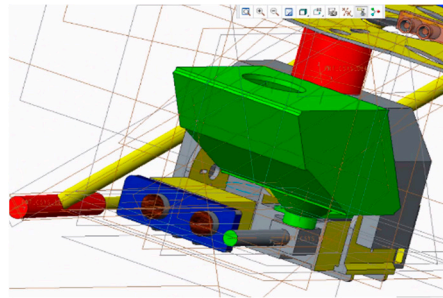


Figure 8. Assembly drawing of continuous gel balls feeding device.



Figure 9. Environmentally friendly gel balls [17].

#### 2.3.2. Design of High-Speed Steering Gear PTZ

Due to the limitations of the load and power consumption of a quadcopter, the requirements for the PTZ are being light in weight, small in volume, and simple in structure. Based on the above principles, a two-degrees-of-freedom PTZ controlled by the steering gear is designed, as shown in Figure 10, including yaw axis steering gear, a yaw axis adapter plate, pitch axis steering gear, a pitch axis adapter plate, a lens mounting base and lens, and a launcher mounting base. The model of



the steering gear is ds8611, while its torque is 7.4 V 18 kg/cm and speed is 0.14 s/60°. The bracket is made using 3D printing. In order to ensure the continuous supply of gel balls for the launching device, a continuous gel ball feeding device is designed, as shown in the Figure 8. The gel balls slide downward under the action of gravity and enter the feeding hole. The designed projectile supply device can store up to 200 gel balls at a time, which can meet the requirements of expelling targets on the flight mission (see Figure 10).



Figure 10. PTZ launcher.

The actuator design is the model obtained by identifying the test results. The operating characteristics of the steering gear include the dead band, linear zone, and nonlinear zone. The steering gear working in the linear zone can be approximately represented as a second-order model as follows:

$$G(s) = \omega_n^2 / (s^2 + 2\xi\omega_n s + \omega_n^2) \quad (8)$$

Since the calculation is discrete, the above equation should be discretized as follows:

$$G(z) = \frac{y(z)}{u(z)} = \frac{b_1 z^{-1} + b_2 z^{-2}}{1 - a_1 z^{-1} - a_2 z^{-2}} \quad (9)$$

where  $u(z)$  is the control signal of the actuator;  $y(z)$  is the feedback signal of the actuator; and  $a_1$ ,  $a_2$ ,  $b_1$ , and  $b_2$  are the parameters obtained through identification. Actuator identification is performed using the adaptive genetic algorithm [18], obtaining  $a_1 = 0.7349$ ,  $a_2 = -0.0076$ ,  $b_1 = -0.0986$ , and  $b_2 = -0.1787$ .

#### 2.4. Ground Station

The ground monitoring computer is the basis of the reliable operation for the identification, tracking, and expelling system. With the help of the ground system, the remote control and monitoring of the equipment are realized. The key functions are as follows: (1) Planning the flight route of the UAV and (2) monitoring the status of the UAV in real time, while also sending instructions, as shown in Figure 11.



Figure 11. Ground station.

### 3. Perception and Control

#### 3.1. Design of Backstepping Quadcopter Controller Based on Interference Compensation

The attitude controller is the most important part of multi-rotor UAV flight control system, because the performance of the attitude controller directly affects the performance of trajectory tracking. Therefore, an attitude controller with superior performance is a prerequisite for stable flight of a multi-rotor UAV flight controller. At the same time, due to the complex structure of multi-rotor UAV, it is difficult to build a precise six-freedom nonlinear mathematical model, and the flight performance of multi-rotor UAV would be affected by the uncertainties of kinetic model, external interference and other common factors. Control methods that are frequently applied nowadays include PID control [19], sliding model control [20], backstepping control [21], feedback linearization control [22], and robust control [23]. The backstepping method is especially suitable for some under-actuated systems. It converts problems related to the design of complicated and high-order controlled system into problems of low-order system decomposed from it. The method designs a virtual and intermediate control variable for each decomposed subsystem to stabilize them, and to thereby obtain the actual control input of the system through iteration. The backstepping method combines the selection of the Lyapunov function well with the solving of the control law, which proves Lyapunov liability. When designing the quadcopter controller, this thesis had fully considered the uncertainties of the kinetic model as well as the influence of external interference, and used radial basis function neural network (RBFNN) to approximate the uncertain items of the model. As for the inference brought by changes of the mass of gel balls after launching from the quadcopter as well as changes of external air current, we had used nonlinear observer to make compensation, and designed the backstepping controller based on interference compensation. Stability of the system was proved with the analysis method in Lyapunov stability theory. At the end, verification was conducted on the simulation platform, and flight test was conducted.

##### 3.1.1. Quadcopter Modeling and Processing

The nonlinear model of 6-DOF quadcopters is expressed by Equation (10):

$$\begin{cases} \dot{p}_t = v_t \\ \dot{a}_t = \Phi(t)\omega_t \\ \dot{v}_t = g \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} - \frac{1}{m_t} R_{BE} \begin{bmatrix} 0 \\ 0 \\ u_{t1} \end{bmatrix} - \frac{k_{tr}}{m_t} \begin{bmatrix} u_t^2 \\ v_t^2 \\ w_t^2 \end{bmatrix} \\ J_t \dot{\omega}_t = -(\omega_t \times J_t \omega_t) + G e_t + M_t \end{cases} \quad (10)$$

in which  $p_t = [x_t, y_t, z_t]^T$  is the position state vector of quadcopter in the ground coordinate system;  $v_t = [u_t, v_t, w_t]^T$  is the speed state vector of the quadcopter in the ground coordinate system;  $a_t = [\phi_t, \theta_t, \psi_t]^T$  is the attitude state vector of the quadcopter in the ground coordinate system;  $\omega_t = [p_t, q_t, r_t]^T$  is the state vector of the attitude angular velocity vector of the quadcopter in the body coordinate system;  $\Phi_t(t)$  is the transform matrix between attitude angular velocity and Eulerian angular velocity, as shown by Equation (11);  $g$  is the local gravitational acceleration,  $g = 9.81 \text{ m/s}^2$ ;  $m_t$  is the mass of the quadcopter,  $m_t = 1.94 \text{ kg}$ ;  $R_{BE}$  is the transform matrix between the body coordinate system and the ground coordinate system, as shown by Equation (12);  $k_{tr}$  is the air drag coefficient of quadcopter, which is given by Equation (13);  $J_t$  is the moment of inertia matrix of quadcopter,  $J_t = \text{diag}[2.37 \times 10^{-3} \quad 3.51 \times 10^{-3} \quad 5.31 \times 10^{-3}] \text{ kg}\cdot\text{m}^2$ ;  $G e_t$  is the gyroscopic moment, as expressed by Equation (14);  $J_{tr}$  is the rotational inertia of motor and rotary wing around the motor shaft,  $J_{tr} = 4.92 \times 10^{-6} \text{ kg}\cdot\text{m}^2$ ;  $M_t = [u_{t1}, u_{t2}, u_{t3}, u_{t4}]^T$  is the resultant moment excluding the gyroscopic

moment, with the control variables  $u_{t1}, u_{t2}, u_{t3}, u_{t4}$  being defined by Equation (15); and  $l_t$  is the length of the quadcopter's frame arm,  $l_t = 0.265$  m. As shown in Table 2.

$$\Phi_t(t) = \begin{bmatrix} 1 & \sin \phi_t \tan \theta_t & \cos \phi_t \tan \theta_t \\ 0 & \cos \phi_t & -\sin \phi_t \\ 0 & \sin \phi_t \sec \theta_t & \cos \phi_t \sec \theta_t \end{bmatrix} \quad (11)$$

$$R_{BE} = \begin{bmatrix} \cos \theta_t \cos \psi_t & \sin \phi_t \sin \theta_t \cos \psi_t - \cos \phi_t \sin \psi_t & \cos \phi_t \sin \theta_t \cos \psi_t + \sin \phi_t \sin \psi_t \\ \cos \theta_t \sin \psi_t & \sin \phi_t \sin \theta_t \sin \psi_t + \cos \phi_t \cos \psi_t & \cos \phi_t \sin \theta_t \sin \psi_t - \sin \phi_t \cos \psi_t \\ -\sin \theta_t & \sin \phi_t \cos \theta_t & \cos \phi_t \cos \theta_t \end{bmatrix} \quad (12)$$

$$\begin{cases} k_{trx} = -9.56 \times 10^{-6} \times \phi_t^2 + 1.26 \times 10^{-3} \times \phi_t + 2 \times 10^{-3} \\ k_{try} = -1.16 \times 10^{-5} \times \theta_t^2 + 1.37 \times 10^{-3} \times \theta_t + 1 \times 10^{-4} \\ k_{trz} = 0.0596 \end{cases} \quad (13)$$

$$Ge_t = \begin{bmatrix} J_{tr} q_t (\omega_{t1} + \omega_{t3} - \omega_{t2} - \omega_{t4}) \\ -J_{tr} p_t (\omega_{t1} + \omega_{t3} - \omega_{t2} - \omega_{t4}) \\ 0 \end{bmatrix} \quad (14)$$

$$\begin{cases} u_{t1} = k_{tl} (\omega_{t1}^2 + \omega_{t2}^2 + \omega_{t3}^2 + \omega_{t4}^2) \\ u_{t2} = k_{tl} l_t (\omega_{t2}^2 + \omega_{t3}^2 - \omega_{t1}^2 - \omega_{t4}^2) \\ u_{t3} = k_{tl} l_t (\omega_{t1}^2 + \omega_{t2}^2 - \omega_{t3}^2 - \omega_{t4}^2) \\ u_{t4} = k_{tl} (\omega_{t1}^2 + \omega_{t3}^2 - \omega_{t2}^2 - \omega_{t4}^2) \end{cases} \quad (15)$$

$$\begin{cases} \dot{x}_t = u_t, \dot{y}_t = v_t, \dot{z}_t = w_t \\ \dot{\phi}_t = p_t + q_t \sin \phi_t \tan \theta_t + r_t \cos \phi_t \tan \theta_t \\ \dot{\theta}_t = q_t \cos \phi_t - r_t \sin \phi_t \\ \dot{\psi}_t = q_t \sin \phi_t \sec \theta_t + r_t \cos \phi_t \sec \theta_t \\ \dot{u}_t = -\frac{(\cos \phi_t \sin \theta_t \cos \psi_t + \sin \phi_t \sin \psi_t) u_{t1} + k_{trx} u_t^2}{m_t} + d_{tu} \\ \dot{v}_t = -\frac{(\cos \phi_t \sin \theta_t \sin \psi_t - \sin \phi_t \cos \psi_t) u_{t1} + k_{try} v_t^2}{m_t} + d_{tv} \\ \dot{w}_t = g - \frac{\cos \phi_t \cos \theta_t u_{t1} + k_{trz} w_t^2}{m_t} + d_{tw} \\ \dot{p}_t = \frac{q_t r_t (J_y - J_z) + u_{t2} + J_{tr} q_t (\omega_{t1} + \omega_{t3} - \omega_{t2} - \omega_{t4})}{J_x} + d_{tp} \\ \dot{q}_t = \frac{p_t r_t (J_z - J_x) + u_{t3} - J_{tr} p_t (\omega_{t1} + \omega_{t3} - \omega_{t2} - \omega_{t4})}{J_y} + d_{tq} \\ \dot{r}_t = \frac{p_t q_t (J_x - J_y) + u_{t4}}{J_z} + d_{tr} \end{cases} \quad (16)$$

Table 2. Quadcopter parameters.

Parameters	Values	Units
$m$	1.94	kg
$g$	9.81	m/s <sup>2</sup>
$l_t$	265	mm
$J_{tr}$	$4.92 \times 10^{-6}$	kg·m <sup>2</sup>
$J_{xx}$	$2.37 \times 10^{-3}$	kg·m <sup>2</sup>
$J_{yy}$	$3.51 \times 10^{-3}$	kg·m <sup>2</sup>
$J_{zz}$	$5.31 \times 10^{-3}$	kg·m <sup>2</sup>

Substituting the above equations into Equation (10) and introducing the external disturbance term  $d_{tu}, d_{tv}, \dots, d_{tr}$ , we can obtain the nonlinear model of 6-DoF quadcopters, as shown in Equation (16).

We selected the three virtual control variables as follows:

$$\begin{cases} u_{tx} = \frac{(\cos \phi_t \sin \theta_t \cos \psi_t + \sin \phi_t \sin \psi_t) u_{t1}}{m_t} \\ u_{ty} = \frac{(\cos \phi_t \sin \theta_t \sin \psi_t - \sin \phi_t \cos \psi_t) u_{t1}}{m_t} \\ u_{tz} = \frac{\cos \phi_t \cos \theta_t u_{t1}}{m_t} - g \end{cases} \quad (17)$$

Now, its translation dynamical equation can be rewritten as:

$$\begin{cases} \dot{u}_t = -u_{tx} - \frac{k_{trx} u_t^2}{m_t} + d_{tu} \\ \dot{v}_t = -u_{ty} - \frac{k_{try} v_t^2}{m_t} + d_{tv} \\ \dot{w}_t = -u_{tz} - \frac{k_{trz} w_t^2}{m_t} + d_{tw} \end{cases} \quad (18)$$

where the three virtual control variables,  $u_{tx}$ ,  $u_{ty}$ , and  $u_{tz}$ , and the desired yaw angle  $\psi_{td}$  can be used to solve the desired control variable  $u_{t1}$ , desired roll angle  $\phi_{td}$ , and desired pitch angle  $\theta_{td}$ .

$$\begin{cases} u_{t1} = m_t \sqrt{u_{tx}^2 + u_{ty}^2 + (u_{tz} + g)^2} \\ \phi_{td} = \arcsin\left(\frac{m_t (u_{tx} \sin \psi_{td} - u_{ty} \cos \psi_{td})}{u_{t1}}\right) \\ \theta_{td} = \arctan\left(\frac{u_{tx} \cos \psi_{td} + u_{ty} \sin \psi_{td}}{u_{tz} + g}\right) \end{cases} \quad (19)$$

As can be seen from the nonlinear mathematical model of 6-DoF quadcopters expressed by Equation (16), the resultant external moment acting on a quadcopter leads to changes in its attitude angular velocity and thus in the attitude angle, which ultimately causes changes in the spatial position of the quadcopter. Therefore, similar to the case with the quadcopter, the trajectory tracking controller is also designed with a double closed-loop control structure comprised of an attitude controller (inner loop) and a trajectory tracking controller (outer loop). In the design of the inner-loop controller, a RBFNN is adopted to approximate uncertainties of the model, and a nonlinear observer is adopted to compensate for the influence caused by external disturbances.

### 3.1.2. Back-Stepping Attitude Controller Design Based on Disturbance Compensation

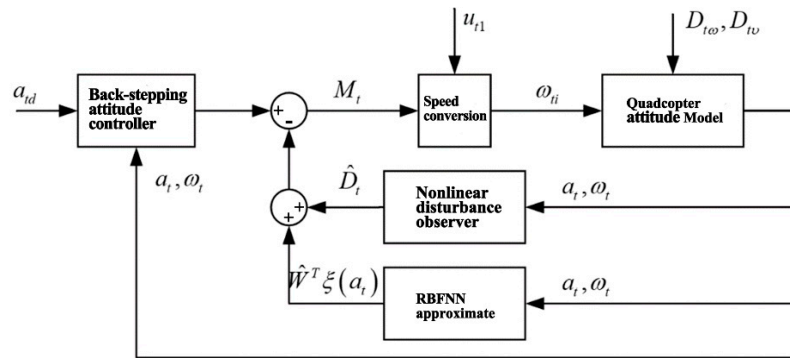
The design and stability analysis of RBFNN and nonlinear disturbance observer in quadcopter trajectory tracking controller, as well as the demonstration of their stability, are omitted. The estimations of external disturbance terms  $D_{tv}$  Equation (20) and  $D_{t\omega}$  Equation (21) in the quadcopter model can be directly given by equations. The detailed process is omitted here.

$$\begin{cases} \hat{D}_{tv}(t) = L_{tv}(v_t - z_{tv}) \\ \dot{z}_{tv} = -u_t - \frac{k_{pr} v_t^2}{m_t} + \hat{D}_{tv}(t) \end{cases} \quad (20)$$

$$\begin{cases} \hat{D}_{t\omega}(t) = L_{t\omega}(\omega_t - z_{t\omega}) \\ \dot{z}_{t\omega} = -(\omega_t \times J_t \omega_t) + G_e t + M_t + \hat{D}_{t\omega}(t) \end{cases} \quad (21)$$

where  $u_t = [u_{tx}, u_{ty}, u_{tz}]^T$ ,  $L_{tv} > 0$  and  $L_{t\omega} > 0$  are all parameters to be designed for the observer.

Similarly, the quadcopter attitude controller is also designed with a double closed-loop structure that is comprised of an attitude angular velocity controller (inner loop) and an attitude angle controller (outer loop). The structural diagram of the attitude controller is shown in Figure 12.



**Figure 12.** Structural diagram of quadcopter attitude controller.

Next, the process by which the attitude angle controller is designed using the back-stepping method is described.

The error between the attitude angle of the quadcopter  $Y_t$  and the desired attitude angle  $Y_d$  is defined as:

$$e_Y = Y_t - Y_d \quad (22)$$

where  $Y_t = [\phi_t, \theta_t, \psi_t]^T$  is the attitude angle of the quadcopter and  $Y_d = [\phi_d, \theta_d, \psi_d]^T$  is the desired attitude angle.

Select a Lyapunov function for the quadcopter attitude angle system, as follows:

$$V_Y = \frac{1}{2} e_Y^T e_Y \quad (23)$$

Taking the derivative of the function, and substituting into Equation (22), we obtain:

$$\dot{V}_Y = e_Y^T \dot{e}_Y = e_Y^T (\dot{Y}_t - \dot{Y}_d) = e_Y^T (\Phi_t(t) X_t - \dot{Y}_d) \quad (24)$$

Let the command for attitude angular velocity be:

$$X_c = -\Phi_t^{-1}(t) (\Lambda_Y e_Y - \dot{Y}_d) \quad (25)$$

where  $\Lambda_Y = \Lambda_Y^T > 0$  is the matrix to be designed.

Substituting Equation (25) into Equation (24), we obtain:

$$\dot{V}_Y = -e_Y^T \Lambda_Y e_Y \leq 0 \quad (26)$$

That is, when adopting the command for attitude angular velocity designated by Equation (25), the attitude angle system of the quadcopter is stabilized.

The error between the attitude angular velocity of the quadcopter  $X_t$  and the desired attitude angular velocity  $X_d$  is defined as:

$$e_X = X_t - X_d \quad (27)$$

where  $X_t = [p_t, q_t, r_t]^T$  is the attitude angular velocity of the quadcopter and  $X_d = [p_d, q_d, r_d]^T$  is the desired attitude angular velocity.

Taking the derivative of the above function, we can substitute it into the quadcopter attitude model as follows:

$$\begin{cases} \dot{Y}_t = \Phi_t(t) X_t \\ \dot{X}_t = F(Y_t) + G(Y_t) M_t(t) + L^{-1} W^* \xi(Y_t) + D_t(t) \end{cases} \quad (28)$$



in which,  $L = L^T > 0$  is the parameter of the auxiliary system to be designed;  $\hat{D}_t(t)$  represents the estimation of external unknown interference  $D$ ;  $\hat{W}^T \xi(Y_t)$  represents the RBFNN estimation of the function. Among which,  $Y_t = [\phi_t, \theta_t, \psi_t]^T$ ,  $X_t = [p_t, q_t, r_t]^T$ ,  $\Phi_t(t)$  is as shown in Equation (1);  $d_a(t) = [d_{pp}, d_{pq}, d_{pr}]^T$  represents unknown external interference.

We obtain:

$$\dot{e}_X = \dot{X}_t - \dot{X}_d = F(Y_t) + G(Y_t)M_t(t) + L^{-1}W^{*T}\xi(Y_t) + D_t(t) - \dot{X}_d \quad (29)$$

Letting

$$M_t(t) = J_t(\dot{X}_d - F(Y_t) - L^{-1}\hat{W}^T\xi(Y_t) - \hat{D}_t(t) - \Lambda_X e_X) \quad (30)$$

we have:

$$\begin{aligned} e_X^T \dot{e}_X &= -e_X^T \Lambda_X e_X + e_X^T L^{-1} \tilde{W}^T \xi(Y_t) + e_X^T \tilde{D}_t(t) \\ &\leq -e_X^T (\Lambda_X - 0.5I_{3 \times 3}) e_X + e_X^T L^{-1} \tilde{W}^T \xi(Y_t) + 0.5 \|\tilde{D}_t(t)\|^2 \end{aligned} \quad (31)$$

Select a Lyapunov function for the quadcopter attitude angular velocity system, as follows:

$$V_X = \frac{1}{2} e_X^T e_X + \frac{1}{2} e_x^T e_x + \frac{1}{2} \text{tr}(\tilde{W}^T \Lambda_W \tilde{W}) + \frac{1}{2} \tilde{D}_t^T(t) \tilde{D}_t(t) \quad (32)$$

Taking the derivative of the above function, and substituting it into Equation (31), we obtain:

$$\begin{aligned} \dot{V}_X &= e_X^T \dot{e}_X + e_x^T \dot{e}_x + \text{tr}(\tilde{W}^T \Lambda_W \dot{\tilde{W}}) + \tilde{D}_t^T(t) \dot{\tilde{D}}_t(t) \\ &\leq -e_X^T (\Lambda_X - 0.5I_{3 \times 3}) e_X + e_X^T L^{-1} \tilde{W}^T \xi(Y_t) + \|\tilde{D}_t(t)\|^2 \\ &\quad - e_x^T (\Gamma - 0.5I_{3 \times 3}) e_x + e_x^T L^{-1} \tilde{W}^T \xi(Y_t) + 0.5\rho^2 + \frac{1}{2\mu} \|\tilde{W}\|^2 \\ &\quad - \tilde{D}_t^T(t) (L - (0.5 + 0.5\mu\tau^2)I_{3 \times 3}) \tilde{D}_t(t) + \text{tr}(\tilde{W}^T \Lambda_W \dot{\tilde{W}}) \end{aligned} \quad (33)$$

Observe and analyze the above equation, and select the adaptive rate of  $\hat{W}$ , as shown below:

$$\dot{\hat{W}} = \Lambda_W^{-1} (\xi(Y_t)(e_x + e_X)^T L^{-1} - \mu_W \hat{W}) \quad (34)$$

where  $\Lambda_W = \Lambda_W^{-1} > 0$  and  $\mu_W > 0$  are both parameters to be designed.

Substituting Equation (34) into Equation (33), we obtain:

$$\begin{aligned} \dot{V}_X &\leq -e_X^T (\Lambda_X - 0.5I_{3 \times 3}) e_X - \tilde{D}_t^T(t) (L - (1.5 + 0.5\mu\tau^2)I_{3 \times 3}) \tilde{D}_t(t) \\ &\quad - e_x^T (\Gamma - 0.5I_{3 \times 3}) e_x + 0.5\rho^2 + \frac{1}{2\mu} \|\tilde{W}\|^2 - \mu_W \text{tr}(\tilde{W}^T \hat{W}) \end{aligned} \quad (35)$$

In the meantime, the following transformation should also be taken into account:

$$2\text{tr}(\tilde{W}^T \hat{W}) = \|\tilde{W}\|^2 + \|\hat{W}\|^2 - \|W\|^2 \geq \|\tilde{W}\|^2 - \|W\|^2 \quad (36)$$

Substituting Equation (36) into Equation (35), we obtain:

$$\begin{aligned} \dot{V}_X &\leq -e_X^T (\Lambda_X - 0.5I_{3 \times 3}) e_X - \tilde{D}_t^T(t) (L - (0.5 + 1.5\mu\tau^2)I_{3 \times 3}) \tilde{D}_t(t) \\ &\quad - e_x^T (\Gamma - 0.5I_{3 \times 3}) e_x + 0.5\rho^2 - \left(\frac{\mu_W}{2} - \frac{1}{2\mu}\right) \|\tilde{W}\|^2 + \frac{\mu_W}{2} \|W\|^2 \\ &\leq -KV_X + C \end{aligned} \quad (37)$$

where

$$K = \min \left\{ \lambda_{\min}(\Lambda_X - 0.5I_{3 \times 3}), \lambda_{\min}(\Gamma - 0.5I_{3 \times 3}), \lambda_{\min} \left( \left( \frac{\mu_W}{2} - \frac{1}{2\mu} \right) / \lambda_{\max}(\Lambda) \right), \lambda_{\min} \left( L - (0.5 + 1.5\mu\tau^2)I_{3 \times 3} \right) \right\} \quad (38)$$

$$C = 0.5\rho^2 + \frac{\mu_W}{2}\|W\|^2 \quad (39)$$

To ensure the stability of the attitude angular velocity system of the quadcopter, the parameters of the controller and relevant auxiliary systems should at least satisfy the following conditions:

$$\Lambda_X - 0.5I_{3 \times 3} > 0, \Gamma - 0.5I_{3 \times 3} > 0, \quad (40)$$

$$\frac{\mu_W}{2} - \frac{1}{2\mu} > 0, L - (0.5 + 1.5\mu\tau^2)I_{3 \times 3} > 0.$$

Meanwhile, as known from literature [24,25],

$$0 \leq V_X \leq \frac{C}{K} + \left(V_X(0) - \frac{C}{K}\right)e^{-Kt} \quad (41)$$

where  $V_X$  is convergent and  $\lim_{t \rightarrow \infty} V_X = \frac{C}{K}$ .

Thus, under the effect of the control moment for attitude angle, as shown by Equation (30), the attitude angular velocity system of the quadcopter is stable.

### 3.1.3. Trajectory Tracking Control Based on the Back-Stepping Method

As can be known from the above, the model of the quadcopter position system is as follows:

$$\begin{cases} \dot{x}_t = u_t \\ \dot{y}_t = v_t \\ \dot{z}_t = w_t \\ \dot{u}_t = -\frac{(\cos \phi_t \sin \theta_t \cos \psi_t + \sin \phi_t \sin \psi_t)u_{t1} + k_{trx}u_t^2}{m_t} + d_{tu} \\ \dot{v}_t = -\frac{(\cos \phi_t \sin \theta_t \sin \psi_t - \sin \phi_t \cos \psi_t)u_{t1} + k_{try}v_t^2}{m_t} + d_{tv} \\ \dot{w}_t = g - \frac{\cos \phi_t \cos \theta_t u_{t1} + k_{trz}w_t^2}{m_t} + d_{tw} \end{cases} \quad (42)$$

The quadcopter position system shown by Equation (42) is divided into three subsystems along the longitudinal and horizontal movements, which are described in the state space:

$$\begin{cases} \dot{y}_1 = \dot{x}_t = u_t = x_1 \\ \dot{x}_1 = \dot{u}_t = -\frac{(\cos \phi_t \sin \theta_t \cos \psi_t + \sin \phi_t \sin \psi_t)u_{t1} + k_{trx}u_t^2}{m_t} + d_{tu} \\ \dot{y}_2 = \dot{y}_t = v_t = x_2 \\ \dot{x}_2 = \dot{v}_t = -\frac{(\cos \phi_t \sin \theta_t \sin \psi_t - \sin \phi_t \cos \psi_t)u_{t1} + k_{try}v_t^2}{m_t} + d_{tv} \\ \dot{y}_3 = \dot{z}_t = w_t = x_3 \\ \dot{x}_3 = \dot{w}_t = g - \frac{\cos \phi_t \cos \theta_t u_{t1} + k_{trz}w_t^2}{m_t} + d_{tw} \end{cases} \quad (43)$$

The following part takes the height controller as an example, where the back-stepping method is adopted in design.

The height error of the quadcopter is defined as follows:

$$e_{y3} = y_{3d} - y_3 \quad (44)$$

where  $y_{3d}$  is the desired height of the quadcopter and  $y_3$  is its actual height.

Select a Lyapunov function for the quadcopter height subsystem, as follows:

$$V_{y3} = \frac{1}{2}e_{y3}^2 \quad (45)$$

Taking the derivative of the above Lyapunov function, we obtain:

$$\dot{V}_{y3} = e_{y3}\dot{e}_{y3} = e_{y3}(\dot{y}_{3d} - \dot{x}_3) = e_{y3}(\dot{y}_{3d} - w_p) \quad (46)$$

Design the following virtual control variable according to Equation (46):

$$w_{pd} = \dot{z}_{pd} + \Lambda_{y3}e_{y3} \quad (47)$$

where  $\Lambda_{y3} > 0$  is the controller parameter to be designed.

Substituting Equation (47) into Equation (46), we obtain:

$$\dot{V}_{y3} = e_{y3}(\dot{y}_{3d} - \dot{y}_{3d} - \Lambda_{y3}e_{y3}) = -\Lambda_{y3}e_{y3}^2 \leq 0 \quad (48)$$

that is, the virtual control variable designed according to Equation (47) enables the error of quadcopter height to converge.

The longitudinal speed error of the quadcopter is defined as follows:

$$e_{x3} = x_3 - x_{3d} = x_3 - \dot{z}_{pd} - \Lambda_{y3}e_{y3} \quad (49)$$

where  $x_{3d}$  is the desired longitudinal speed of the quadcopter and  $x_3$  is its actual longitudinal speed.

Taking the derivative of the longitudinal speed of the quadcopter, we obtain:

$$\dot{e}_{x3} = \dot{x}_3 - \ddot{z}_{td} - \Lambda_{y3}\dot{e}_{y3} = -u_{tz} - \frac{k_{trz}w_t^2}{m_t} + d_{tw} - \ddot{z}_{td} - \Lambda_{y3}\dot{e}_{y3} \quad (50)$$

Furthermore, we have:

$$\dot{e}_{y3} = \dot{y}_{3d} - \dot{y}_3 = w_{td} - w_t - \Lambda_{y3}e_{y3} = -e_{x3} - \Lambda_{y3}e_{y3} \quad (51)$$

Substituting Equation (51) into Equation (50), we obtain:

$$\dot{e}_{x3} = \dot{x}_3 - \ddot{z}_{td} - \Lambda_{y3}\dot{e}_{y3} = -u_z - \frac{k_{prz}w_p^2}{m_p} + d_{pw} - \ddot{z}_{pd} + \Lambda_{y3}(e_{x3} + \Lambda_{y3}e_{y3}) \quad (52)$$

where  $\Lambda_{x3} > 0$  is the controller parameter to be designed.

Substituting Equation (52) into Equation (51), we obtain:

$$\dot{e}_{x3} = -\Lambda_{x3}e_{x3} + e_{y3} + d_{tw} - \hat{d}_{tw} \quad (53)$$

Select a Lyapunov function for the quadcopter longitudinal speed subsystem, as follows:

$$V_{x3} = \frac{1}{2}e_{y3}^2 + \frac{1}{2}e_{x3}^2 + \frac{1}{2}\tilde{d}_{tw}^2 \quad (54)$$

where  $\tilde{d}_{tw} = d_{tw} - \hat{d}_{tw}$  is the estimated error of the nonlinear observer.

Taking the derivative of the Lyapunov function, as shown by Equation (54), we obtain:

$$\begin{aligned} \dot{V}_{x3} &= e_{y3}\dot{e}_{y3} + e_{x3}\dot{e}_{x3} + \tilde{d}_{tw}\dot{\tilde{d}}_{tw} \\ &= e_{y3}(-e_{x3} - \Lambda_{y3}e_{y3}) + e_{x3}(\tilde{d}_{tw} - \Lambda_{x3}e_{x3} + e_{y3}) + \tilde{d}_{tw}\dot{\tilde{d}}_{tw} \\ &= -\Lambda_{y3}e_{y3}^2 - \Lambda_{x3}e_{x3}^2 + e_{x3}\tilde{d}_{tw} + \tilde{d}_{tw}\dot{\tilde{d}}_{tw} \\ &\leq -\Lambda_{y3}e_{y3}^2 - \Lambda_{x3}e_{x3}^2 + 0.5e_{x3}^2 + 0.5\tilde{d}_{tw}^2 + \tilde{d}_{tw}\dot{\tilde{d}}_{tw} \end{aligned} \quad (55)$$

In the meantime, we use the nonlinear observer to estimate the external longitudinal disturbance term  $d_{tw}$ . The specific definition is shown below. The demonstration process of its stability is omitted here.

$$\begin{cases} \hat{d}_{tw}(t) = L_w(w_t - z_w) \\ \dot{z}_w = -u_z - \frac{k_{trz}w_t^2}{m_t} + \hat{d}_{tw}(t) \end{cases} \quad (56)$$

According to Equation (56), we can obtain:

$$\begin{aligned} \dot{\hat{d}}_{tw}(t) &= L_w(\dot{w}_t - \dot{z}_w) = L_w\left(\dot{w}_t + u_z + \frac{k_{trz}w_t^2}{m_t} - \hat{d}_{tw}(t)\right) \\ &= L_w(d_{tw}(t) - \hat{d}_{tw}(t)) = L_w\tilde{d}_{tw}(t) \end{aligned} \quad (57)$$

Substituting Equation (57) into Equation (56), we obtain:

$$\begin{aligned} \dot{V}_{x3} &\leq -\Lambda_{y3}e_{y3}^2 - \Lambda_{x3}e_{x3}^2 + 0.5e_{x3}^2 + 0.5\tilde{d}_{tw}^2 + \tilde{d}_{tw}\dot{\tilde{d}}_{tw} \\ &\leq -\Lambda_{y3}e_{y3}^2 - \Lambda_{x3}e_{x3}^2 + 0.5e_{x3}^2 + 0.5\tilde{d}_{tw}^2 + \tilde{d}_{tw}(\dot{d}_{tw} - \dot{\hat{d}}_{tw}) \\ &\leq -\Lambda_{y3}e_{y3}^2 - \Lambda_{x3}e_{x3}^2 + 0.5e_{x3}^2 + 0.5\tilde{d}_{tw}^2 + \tilde{d}_{tw}(\dot{d}_{tw} - L_w\tilde{d}_{tw}) \\ &\leq -\Lambda_{y3}e_{y3}^2 - \Lambda_{x3}e_{x3}^2 + 0.5e_{x3}^2 + 0.5\tilde{d}_{tw}^2 + \tilde{d}_{tw}\dot{d}_{tw} - L_w\tilde{d}_{tw}^2 \\ &\leq -\Lambda_{y3}e_{y3}^2 - \Lambda_{x3}e_{x3}^2 + 0.5e_{x3}^2 + 0.5\tilde{d}_{tw}^2 + 0.5\dot{\tilde{d}}_{tw}^2 - L_w\tilde{d}_{tw}^2 \\ &\leq -\Lambda_{y3}e_{y3}^2 - (\Lambda_{x3} - 0.5)e_{x3}^2 - (L_w - 0.5)\tilde{d}_{tw}^2 + 0.5\mu_{dw}^2 \end{aligned} \quad (58)$$

To ensure the convergence of the quadcopter's longitudinal speed error, the following condition should be satisfied:

$$\Lambda_{y3} > 0, \Lambda_{x3} - 0.5 > 0, L_w - 0.5 > 0 \quad (59)$$

By the same token, we can derive the virtual control variables for the two subsystems along horizontal movements, as shown below:

$$\begin{aligned} u_{td} &= \dot{x}_{td} + \Lambda_{y1}e_{y1} \\ u_x &= \ddot{x}_{td} + e_{y1} - \Lambda_{y1}(e_{x1} + \Lambda_{y1}e_{y1}) - \Lambda_{x1}e_{x1} + \frac{k_{trx}u_t^2}{m_t} - \hat{d}_{tu} \\ v_{td} &= \dot{y}_{td} + \Lambda_{y2}e_{y2} \\ u_y &= \ddot{y}_{td} + e_{y2} - \Lambda_{y2}(e_{x2} + \Lambda_{y2}e_{y2}) - \Lambda_{x2}e_{x2} + \frac{k_{try}v_t^2}{m_t} - \hat{d}_{tv} \end{aligned} \quad (60)$$

where  $e_{y1} = y_{1d} - y_1$  and  $e_{y2} = y_{2d} - y_2$  are horizontal position errors;  $e_{x1} = x_1 - x_{1d}$  and  $e_{x2} = x_2 - x_{2d}$  are horizontal speed errors; and  $\Lambda_{x2} - 0.5 > 0$ ,  $\Lambda_{x1} - 0.5 > 0$ ,  $\Lambda_{y2} > 0$  and  $\Lambda_{y1} > 0$  are controller parameters to be designed.

### 3.2. Real-Time Image Recognition, Location, and Tracking Based on the YOLO and KCF Algorithms

#### 3.2.1. Target Recognition, Location

The deep learning model has become a hot research topic in computer vision due to its strong presentation ability, data accumulation, and computing power progress. It is mainly divided into three levels: classification, detection, and segmentation.

The classification task focuses on the whole, giving the content description of the whole picture, while the detection focuses on the specific object target, requiring the simultaneous acquisition of the class information and location information of the target. Compared with classification, detection is the understanding of the foreground and background of a picture. We need to isolate an interesting target from the background and determine the target description (class and location). Therefore, the output of a detection model is a list, and each item in the list uses a dataset, giving the class and position of the detected target (often expressed by coordinates of a rectangular detection box).

With the development of Deep Neural Networks (DNN), the Convolutional Neural Network (CNN) has been widely used in image recognition. When a video collected by the camera is subject to frame extracting to obtain the image, the first task is to detect all kinds of aircrafts in the image, that is, target detection of the corresponding image processing task. However, CNN can only judge if the target object appears in the image, but cannot locate the position of the target object in the image. Ross et al. [26], from Berkeley University, proposed a new network structure named Regions with Convolutional Neural Network Features (R-CNN) in 2014, which realized the functions of image recognition and item location. Girshick proposed a new Faster R-CNN [27] in 2016. In structure, Faster R-CNN has already integrated feature extraction, proposal, bounding box regression, and classification in one network, greatly improving the overall performance, especially in detection speed. The improved versions of Fast R-CNN [28] and Faster R-CNN realized higher recognition accuracy, but the application of classifiers in the R-CNN series network caused the processing speed of Fast R-CNN to be 0.5 frames per second (FPS), while that of Faster R-CNN reached 7 FPS. This is a great improvement compared to R-CNN, but still fails to meet the real-time requirement.

In 2016, Redmon et al. [29–31] from the University of Washington proposed a real-time object detection network YOLO (You Only Look Once). The YOLO algorithm is also a classic algorithm in the field of target detection. The core idea is to use the entire picture as an input to the network, directly returning the location of the bounding box and its class in the output layer. The object detection problem is transformed into a regression problem for processing and a single neural network can be used to obtain the position coordinates and relative size of the object from one image. On a computer with a GPU, YOLO in its standard version can process an image on the real-time basis at a rate of 45 FPS, but the YOLO fast version can reach a speed of 155 FPS, doubling the average accuracy of any other real-time object detection method.

We compared the similarities and differences of Faster R-CNN and YOLOv3 in target detection. With regard to the traditional multiclass detection task, the target detection task in high-speed field is relatively simple (for aircraft detection only). In terms of accuracy, YOLOv3 has better performance; in terms of processing speed, Faster R-CNN algorithm runs more slowly: the time spent on one image is 5–6 s, which does not meet the speed requirement in actual testing tasks. However, the YOLOv3 algorithm takes only 0.1–0.2 s to process one image and satisfies the requirement better in actual conditions. To improve the accuracy of lane detection in complex scenarios, an adaptive lane feature learning algorithm that can automatically learn the features of a lane in various scenarios is proposed [32]. As a result, the YOLOv3 algorithm is adopted for target detection.

This paper aims to propose a real-time image recognition, location, and tracking system on the basis of a YOLO network. YOLO divides the whole input image into a  $S \times S$  grid that can predict the normalized relative coordinates  $(x, y)$ , normalized relative length and width  $(w, h)$ , and confidence level *conf* of the central position of  $R$  bounding boxes, as well as  $C$  conditional probabilities, that is, the probability that this object belongs to one class when this grid contains the target object. The YOLO network structure is shown in Figure 13, including 24 convolutional layers and two fully connected layers. Convolutional layers are used to extract the features in the image, while the fully connected layers are used to build the relationship between the features and the probability of image position and target. The output of the YOLO network is a  $S \times S \times (5 \times R + C)$  tensor, where each  $1 \times 1 \times (5 \times R + C)$  tensor corresponds to one  $S \times S$  grid from the image, including conditional probability, bounding box size, and coordinate information.





layer is added to the YOLO network to build the relationship between the coordinate and size of the bounding box and the actual location and distance of the aircraft.

In network training, 2000 images are collected as the specimens, and the locations of the targets in sample images are marked. Some specimens are shown as Figures 14–17. Figure 18 provides the recognition of targets by YOLO after training.

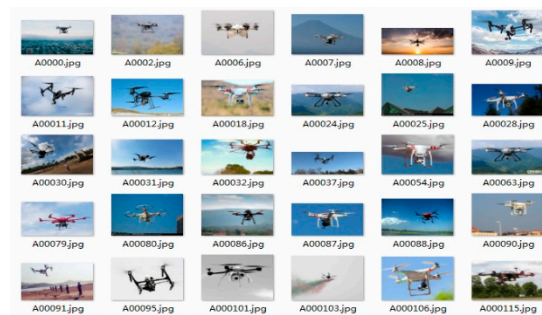


Figure 14. Rotorcraft positive specimen training set.



Figure 15. Balloon positive specimen training set.



Figure 16. Fixed-wing aircraft positive specimen training set.

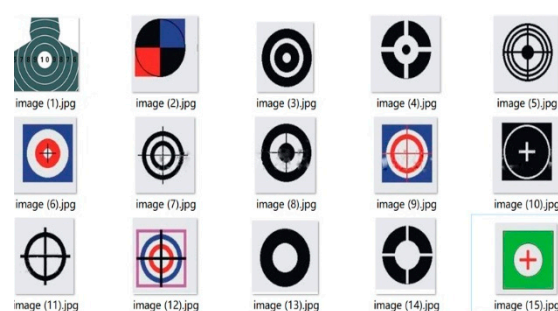
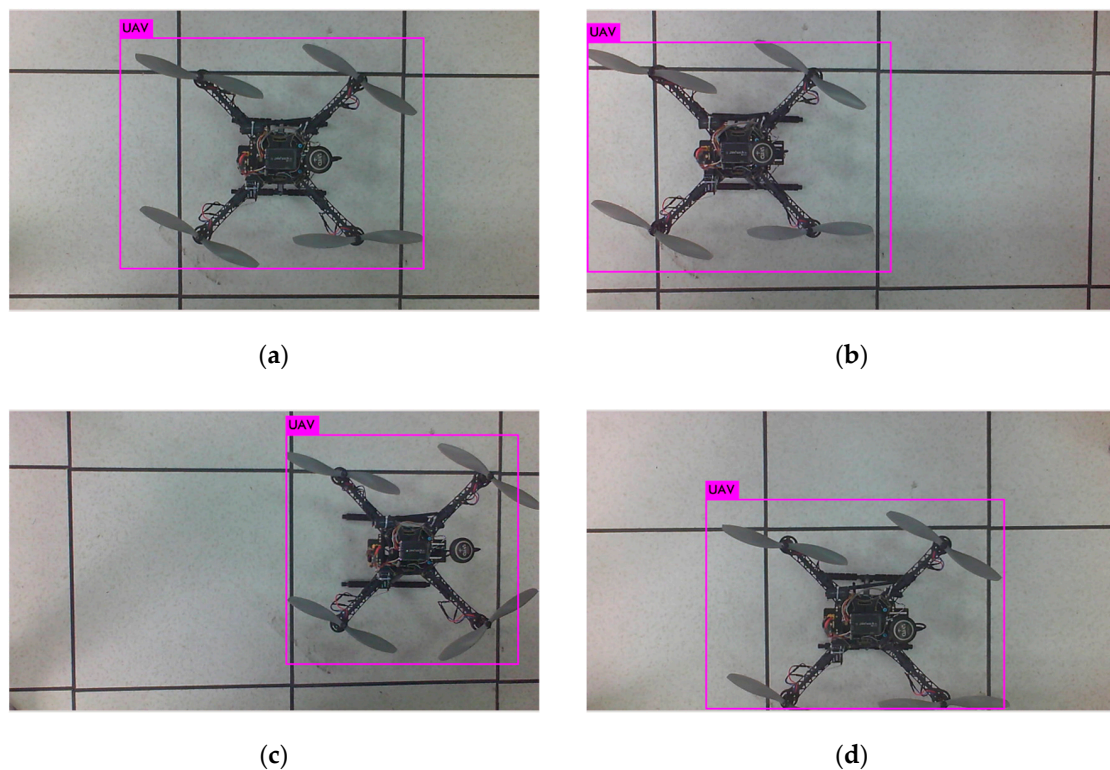


Figure 17. Target positive specimen training set.



**Figure 18.** YOLO real-time recognition location UAV. (a–d) Four sets of localization experiments.

As seen in Table 3, when the UAV is away from the image boundary, it can be located accurately, but when the UAV approaches the image boundary or goes beyond the image range, the accuracy of the location may decrease.

**Table 3.** You Only Look Once (YOLO) real-time recognition location results.

Parameters	a	b	c	d
Bounding box $x$ coordinate	0.5028	0.2920	0.7434	0.5055
Bounding box $y$ coordinate	0.4783	0.4867	0.4675	0.6494
Bounding box length	0.5752	0.5758	0.4379	0.5634
Bounding box height	0.7651	0.7626	0.7602	0.6964
Actual $x$ coordinate (cm)	0	−30	30	0
Actual $y$ coordinate (cm)	0	0	0	10
Actual altitude (cm)	100	100	100	100
Recognition $x$ coordinate (cm)	0.0354	−26.30	31.21	0.0707
Recognition $y$ coordinate (cm)	−0.1591	−0.2299	−0.2387	10.96
Recognition altitude (cm)	98.10	98.16	133.2	102.8

### 3.2.2. Target Tracking

In the early stage, image tracking algorithms such as Camshift, light stream, and background subtraction were very popular, and were applied successfully in static background conditions. After 2008, such methods were gradually abandoned, with the research focus shifting to the study of image tracking with a dynamic or complex background.

Currently, OpenCV provides eight algorithms in target tracking: Boosting, Channel and Spatial Reliability Tracking (CSRT), GOTURN, KCF, Median Flow, Multiple Instance Learning (MIL), Minimum Output Sum of Squared Error (MOSSE), and Tracking Learning Detection (TLD), including the classic algorithm and the current advanced tracking algorithm.

The main contributions of the KCF algorithm are as follows: (1) Positive and negative samples are collected by using the cyclic matrix of the surrounding area of the target, and the target detector is

trained by using ridge regression. The operation of a matrix is transformed into a Hadamard product of vector by the diagonalization property of a cyclic matrix in Fourier space, i.e., the dot product of an element, greatly reducing the amount of computation, improving the speed of operation, and making the algorithm meet the real-time need. (2) The ridge regression of linear space is mapped to the nonlinear space by a kernel function. In the nonlinear space, by solving a dual problem and some common constraints, the calculation can also be simplified by diagonalizing the cyclic matrix Fourier space. (3) A way to integrate multichannel data into the algorithm is presented. The histogram of oriented gradient (HOG) feature is used when the features of a targeted area are extracted. The HOG feature divides the image into smaller parts called cells. Gradient information is extracted from the cells, and a gradient orientation histogram is drawn to reduce the influence of light. By gathering the orientation histograms of several cells for block normalization, all orientation histograms of cells are connected in a series to get the features of the image.

The accelerating methods used in KCF are as follows: (1) Detection: the cyclic matrix + Fourier transform calculation response diagram are used; the original  $O(N^3)$  algorithm needs only  $O(n^* \log(n))$ . (2) Training: The cyclic matrix property is used for training in the frequency domain. (3) Kernel regression acceleration: For a kernel function, it can also be converted to the frequency domain for training and detection, greatly improving the speed. (4) Special kernel functions are further accelerated: for a Gaussian kernel, a polynomial kernel can further use a cyclic matrix to calculate the cyclic matrix of kernel functions.

We have built the OpenCV running environment on the Jetson nano and deployed the YOLO algorithm and KCF tracker [33]. Then we prepared for the next test.

### 3.3. Automatic Tracking Strategy and PTZ Control

#### 3.3.1. Principle of Visual Feedback Servo Tracking

Through image acquisition and comparison of successive frames, computer vision feedback can be realized. The difference between the target position extracted according to the next image information and the position information extracted from the previous image is used as the input signal of PTZ position control; the space movement of the moving target is converted into frame image plane coordinates, and the angle of rotation required for PTZ aiming is calculated to form the visual feedback.

As shown in Figure 19, the optical center point of the camera mounted on the PTZ is used as the reference point  $O_e$ , and a space reference coordinate system  $O_e x_e y_e z_e$  is built. By means of the space reference coordinate system and image plane coordinate transformation, the coordinate of the target in the camera imaging plane, the image plane coordinate, is determined. Assuming that at moment  $t$ , the coordinate of the target object in the reference coordinate system  $O_e x_e y_e z_e$  is  $A(x_t, y_t, z_t)$ , after image plane coordinate transformation, its location coordinate  $A(x_{t1}, y_{t1})$  in the image plane can be determined. After a very short time  $\Delta t$ , PTZ does not act; the space reference coordinate system remains unchanged and the coordinate of the target object in the reference coordinate system is  $A(x_{t+\Delta t}, y_{t+\Delta t}, z_{t+\Delta t})$ . After image plane coordinate transformation, its location coordinate  $A(x_{t1+\Delta t}, y_{t1+\Delta t})$  in the image plane can be determined. After calculation, within the time interval  $\Delta t$ , the coordinate of the target object turns  $\alpha$  and  $\beta$  relative to the  $x_e$  axis and  $y_e$  axis of the space reference coordinate system.

Assuming that the execution time of the PTZ action is 0, after the PTZ location adjustment, the space datum coordinate system becomes  $O'_{e+\Delta t} x'_{e+\Delta t} y'_{e+\Delta t} z'_{e+\Delta t}$ . The coordinate of the target in relative to the new space coordinate system is  $S'_{t+\Delta t}(x'_{t+\Delta t}, y'_{t+\Delta t}, z'_{t+\Delta t})$ , and the image plane coordinate is  $A'_{t+\Delta t}(a'_{x,t+\Delta t}, a'_{y,t+\Delta t})$ . Assuming that before every rotation in this process, the camera coordinate system is the space coordinate system, its mathematical model can be described as shown in Equation (66):

$$O'_{e+\Delta t} = R_{e+\Delta t} * O_e \quad (66)$$

where  $O_e$  is the set space coordinate at moment  $t$ , i.e., the standard coordinate system;  $O'_{e+\Delta t}$  is the camera coordinate after rotation at moment  $\Delta t$ , equivalent to the transition coordinate system introduced for calculating the rotation. In  $R_{e+\Delta t} = R_\alpha \times R_\beta$ ,  $R_{e+\Delta t}$  is the rotation matrix. Its coordinate transformation, i.e., the process of movement, is as shown in Figure 19. The target object moves from Point A to Point B, and yaw angle  $\alpha$  and pitch angle  $\beta$  are adjusted through PTZ to ensure that the visual axis is aligned with the target object and achieve target tracking.

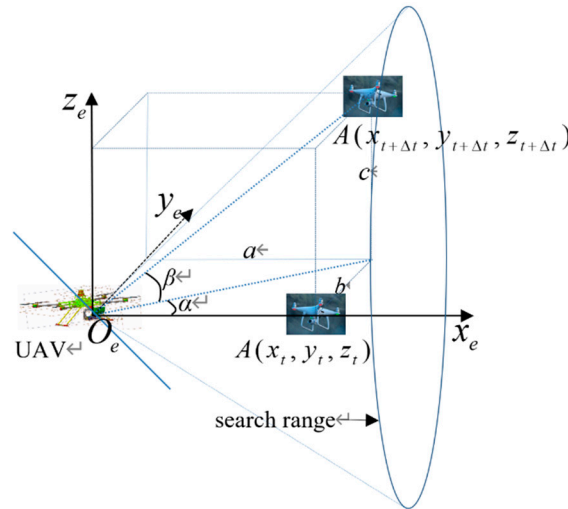


Figure 19. 2-DOF PTZ tracking coordinate diagram.

The 2-DOF PTZ angle is adjusted to ensure the coincidence of optical center and rotation center; the tracking motion of the visual axis can be broken into the rotation motion around  $x$  axis and  $y$  axis in the camera coordinate system. The matrix is shown in Equation (67):

$$R_{e+\Delta t} = \begin{bmatrix} \cos \Delta\beta & 0 & -\sin \Delta\beta \\ -\sin \Delta\beta \sin \Delta\alpha & \cos \Delta\alpha & -\sin \Delta\alpha \cos \Delta\beta \\ -\sin \Delta\beta \cos \Delta\alpha & \sin \Delta\alpha & \cos \Delta\alpha \cos \Delta\beta \end{bmatrix} \quad (67)$$

The workflow is as follows: in coordinate system  $o$ , the starting position of the UAV is the origin of coordinates, the starting position of target is  $A(x_t, y_t, z_t)$  and the detection angle of airborne camera is  $120^\circ$ . After a short time  $\Delta t$ , the target position is  $A(x_{t+\Delta t}, y_{t+\Delta t}, z_{t+\Delta t})$ . (1) The Jetson Nano collects a video stream through the UAV front-end camera. (2) When the target is found, it marks the target and sends the target frame coordinates and color images to the target tracking algorithm for initialization. (3) The color image is sent to the target tracking algorithm for iterative updating, and the next color image is re-executed in the third step. (4) According to the color image and the corresponding target frame coordinate information, the UAV flight height, speed, and attitude angle are controlled in order to enable the UAV to approach the target. The Jetson Nano detects distance via the front-end ultrasonic ranging module to keep a safe distance from the target. (5) After entering the range, the Arduino control board executes the fire control program, drives the two-degrees-of-freedom steering gear pan tilt, quickly corrects angles  $\alpha$  and  $\beta$  between the collimation and the target, automatically executes the shooting command after aiming, drives the gel ball blaster unit motor to launch through the relay, and completes its task of hitting the target.

In ideal conditions, the new coordinate should be the same as the image plane coordinate at moment  $t$ . In the actual movement process, due to the extraction error of image plane coordinates in the target recognition process and the tracking error of the control system, an because of the continuous movement of the target during the tracking process, the visual feedback system suffers an upper



limit of tracking speed. In order to further improve the speed and accuracy of target tracking, higher requirements are put forward for the design of a PTZ controller.

### 3.3.2. Anti-Windup PID Control Algorithm

In the design process of the pan tilt control system, the structural strength and response speed of the steering gear are limited; in general, this is referred to as plant input limitation. In addition, the PTZ module requires frequent switches to different modes, such as from follow mode to target attack mode, which is known as plant input substitution. Due to the existence of input limitation and displacement, the input and output of the control system are sometimes unequal, which leads to further variation in the closed-loop response of the control system, resulting in the windup phenomenon. The PID controller is widely used across various aspects of control system design. In order to eliminate static error, the windup phenomenon is inevitable in the integral part of the controller. The fast-tracking task results in higher requirements in the design of the PTZ control system. Typically, the control system takes a small signal as input in the process of debugging and operation. When the PTZ of the steering gear quickly follows and adjusts the firing angle, the control signal input is given a large range of sudden change, which is prone to large overshoots and vibrations, affecting the stability of the entire flight control system.

In view of the windup phenomenon in the PTZ control system, we have established the PTZ model of the steering gear, analyzed the influence of the structure, speed and force of the steering gear on tracking and proposed an Anti-Windup PID controller to reduce the influence of actuator saturation and improve the dynamic response performance of the steering gear PTZ. Firstly, we ignore the nonlinear effect of saturation caused by the actuator limitation of the steering gear and take the deviation between the expected position of the aim point and the actual position as the input value, integrate the saturation error, and weaken the saturation effect by adjusting the adaptive coefficient. When the pan tilt of the steering gear is adjusted slightly, the compensator does not work. When the PTZ of the steering gear is adjusted rapidly and at a large angle, the path and time information and large signal are taken as input values. The PID controller with anti-integral saturation compensation will play a role in ensuring the control performance as the system is saturated. The input video stream resolution is  $1080 \times 720$  pixels. The center abscissa (540,360) of the frame image pixel of the video stream is taken as the given value, and the center abscissa of the target frame is taken as the output value and negative feedback, all of which form a closed-loop control loop. After the difference between the given value and the feedback value is passed through the Anti-Windup PID controller, it is sent to the Arduino control board through the serial port. From this, the angular speed of rotation is calculated in order to control the rotation of the pan tilt of the steering gear so that the target is in the center of the image.

In this simulation experiment, the Anti-Windup PID controller adopts back-calculation, and the structure is shown in Figure 20.

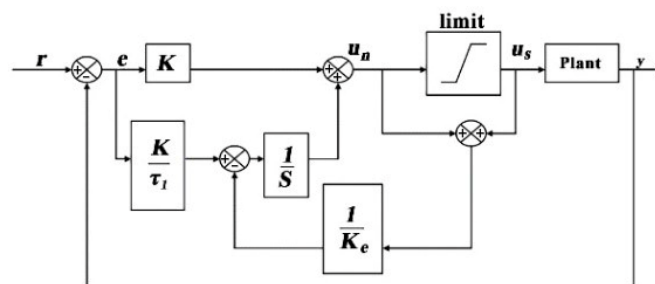


Figure 20. Back-calculation anti-windup Proportional Integral Derivative (PID) structure.

Anti-Windup PID controller output expression:

$$u_n = \frac{KK_e(\tau_1 s + 1)}{\tau_1(K_e s + 1)} e + \frac{1}{K_e s + 1} u_s \quad (68)$$

In the determination of distance from the target, we adopted the ultrasonic ranging scheme. The threshold value is set to 10 m. When the ultrasonic device detects that it is 10 m from the target, the UAV stops moving toward it. The 10 m distance can effectively ensure the safety of the UAV, and at the same time, remain within effective expelling range. In this paper, the binocular camera is not used for video streaming and target depth information collection, because the data volume of the binocular camera is too large, and the maximum processing speed of the airborne processor Jetson Nano can only reach three frames per second. The real-time performance of the program is poor.

#### 4. Experimental Results

The experiment is divided into five stages:

In the first stage, in Sections 2.2 and 3.1, the back-stepping method was applied to design the attitude controller and trajectory tracking controller of quadcopters, and the Lyapunov stability of the said controllers was demonstrated. An experimental simulation based on the quadcopter model was conducted to validate the correctness of the designed controllers. To demonstrate the advanced nature of the controllers, the traditional PID controller was used as a comparable counterpart. The diagram of the trajectory tracking controller designed for the quadcopter is shown in Figure 21.

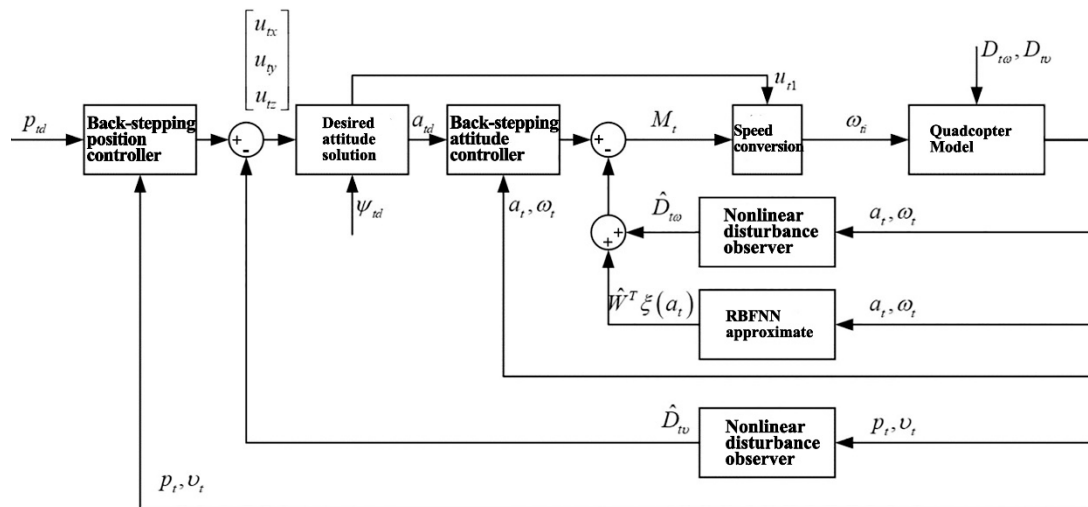


Figure 21. Structural diagram of quadcopter trajectory tracking controller.

Under the Matlab2018/Simulink environment, the trajectory tracking of the designed controller was simulated, where a fixed step size of 0.001 s was adopted. The specified trajectory was given by Equation (69), and the initial values for the attitude angle, its angular velocity, position, and speed

were all zero in the initial state. The following parameters for the controller were selected, as shown in Table 4.

$$\begin{cases} x_d = 0, y_d = 0, z_d = -0.2t; t \in (0, 5] \\ x_d = 0.5 \times (t - 5), y_d = 0, z_d = -1; t \in (5, 10] \\ x_d = 2.5, y_d = 0.3 \times (t - 10), z_d = -1; t \in (10, 15] \\ x_d = 2.5 - 0.5 \times (t - 15), y_d = 1.5, z_d = -1; t \in (15, 20] \\ x_d = 0, y_d = 1.5 - 0.3 \times (t - 20), z_d = -1; t \in (20, 25] \\ x_d = 0, y_d = 0, z_d = -1 + 0.2 \times (t - 25); t \in (25, 30] \\ \psi_d = 0 \text{ rad}, t \in (0, 5] \\ \psi_d = 0.25 \text{ rad}, t \in (5, 30] \end{cases} \quad (69)$$

Table 4. Simulation parameters.

Back-Stepping Controller	Nonlinear Disturbance Observer	PID Controller
$\Lambda_{ta} = \text{diag}\{5, 5, 5\}$	$L_{tv} = \text{diag}\{40, 40, 40\}$	$K_{xp} = 5, K_{xi} = 5, K_{xd} = 5$
$\Lambda_{tw} = \text{diag}\{10, 10, 10\}$	$L_{tw} = \text{diag}\{50, 50, 50\}$	$K_{yp} = 5, K_{yi} = 5, K_{yd} = 5$
$\Lambda_{tp} = \text{diag}\{1, 1, 0.5\}$		$K_{zp} = 10, K_{zi} = 3, K_{zd} = 10$
$\Lambda_{tv} = \text{diag}\{5, 5, 10\}$		$K_{\phi p} = 5, K_{\phi i} = 3, K_{\phi d} = 4$
		$K_{\theta p} = 5, K_{\theta i} = 3, K_{\theta d} = 4$
		$K_{\psi p} = 5, K_{\psi i} = 2, K_{\psi d} = 1.5$

The external disturbance terms  $D_{tv}(t)$  and  $D_{tw}(t)$  are shown by Equation (70) and Equation (71), respectively.

$$d_a(t) = \begin{bmatrix} -0.6785 \cos(1.2t + 30^\circ) \\ -0.8422 \sin(1.5t + 50^\circ) \\ -0.5114 \cos(1.3t + 20^\circ) \end{bmatrix} \quad (70)$$

$$\begin{bmatrix} d_{tu}(t) \\ d_{tv}(t) \\ d_{tw}(t) \end{bmatrix} = \begin{bmatrix} -1.3187 \cos(0.7t + 40^\circ) \\ -1.3562 \sin(0.5t + 20^\circ) \\ -2.5716 \cos(0.8t + 50^\circ) \end{bmatrix} \quad (71)$$

The simulation results are shown in Figure 22.

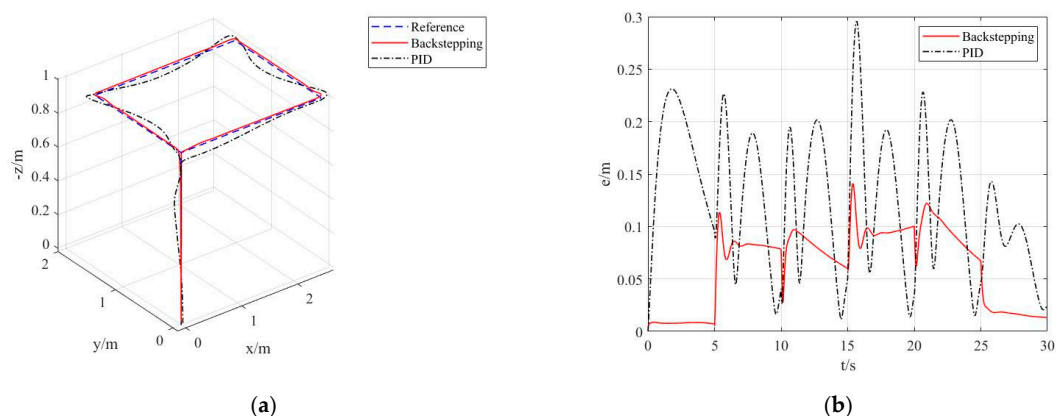
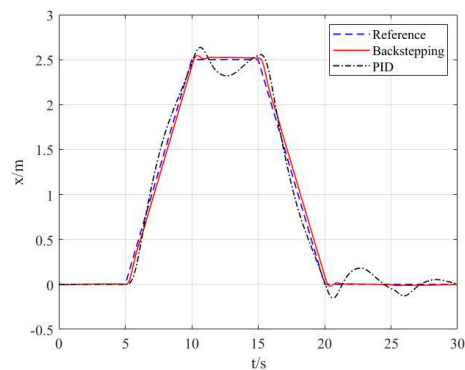
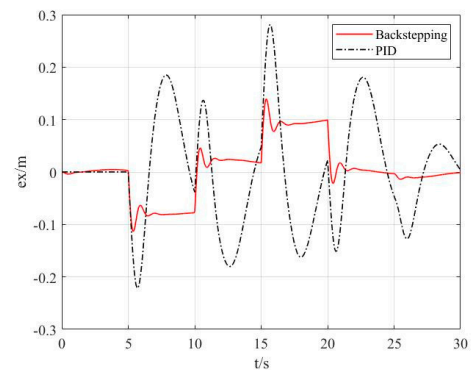


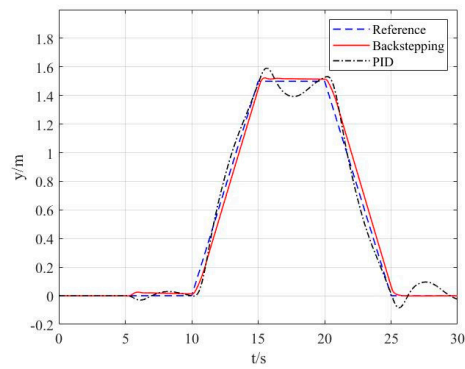
Figure 22. Cont.



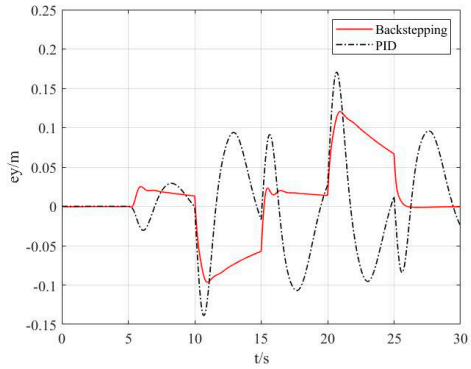
(c)



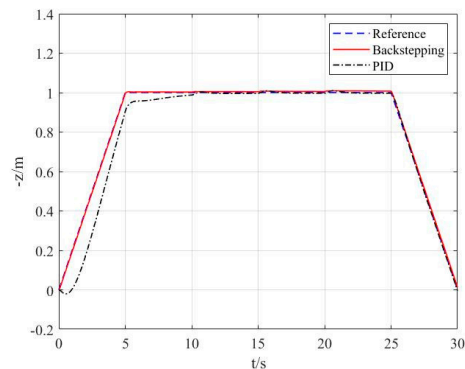
(d)



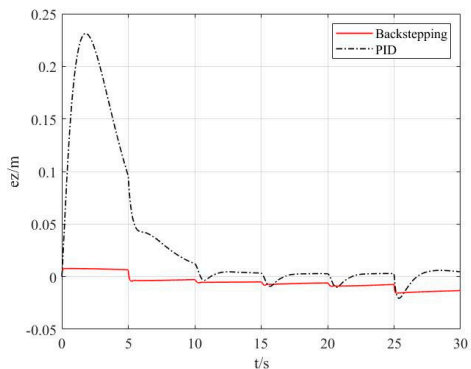
(e)



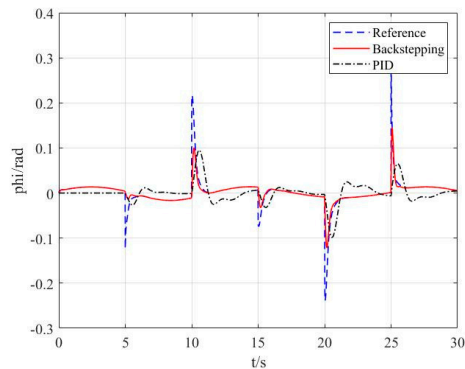
(f)



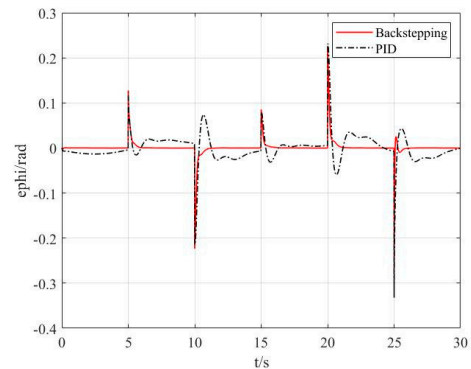
(g)



(h)

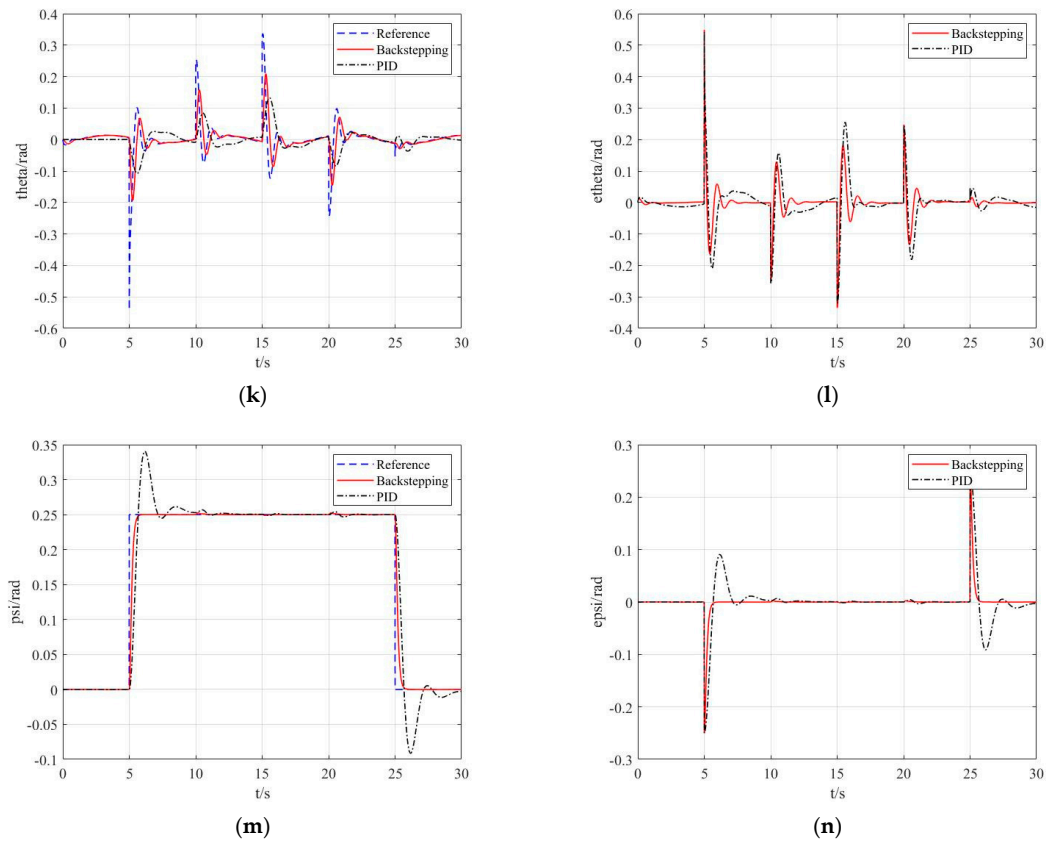


(i)



(j)

Figure 22. Cont.



**Figure 22.** Curve of quadcopter trajectory tracking responses. (a) 3D curve of trajectory tracking. (b) Curve of trajectory tracking errors. (c) Curve of trajectory tracking along the  $x$ -axis. (d) Curve of trajectory tracking errors along the  $x$ -axis. (e) Curve of trajectory tracking along the  $y$ -axis. (f) Curve of trajectory tracking errors along the  $y$ -axis. (g) Curve of trajectory tracking along the  $z$ -axis. (h) Curve of trajectory tracking errors along the  $z$ -axis. (i) Curve of roll angle responses. (j) Curve of roll angle tracking errors. (k) Curve of pitch angle responses. (l) Curve of pitch angle tracking errors. (m) Curve of yaw angle responses. (n) Curve of yaw angle tracking errors.

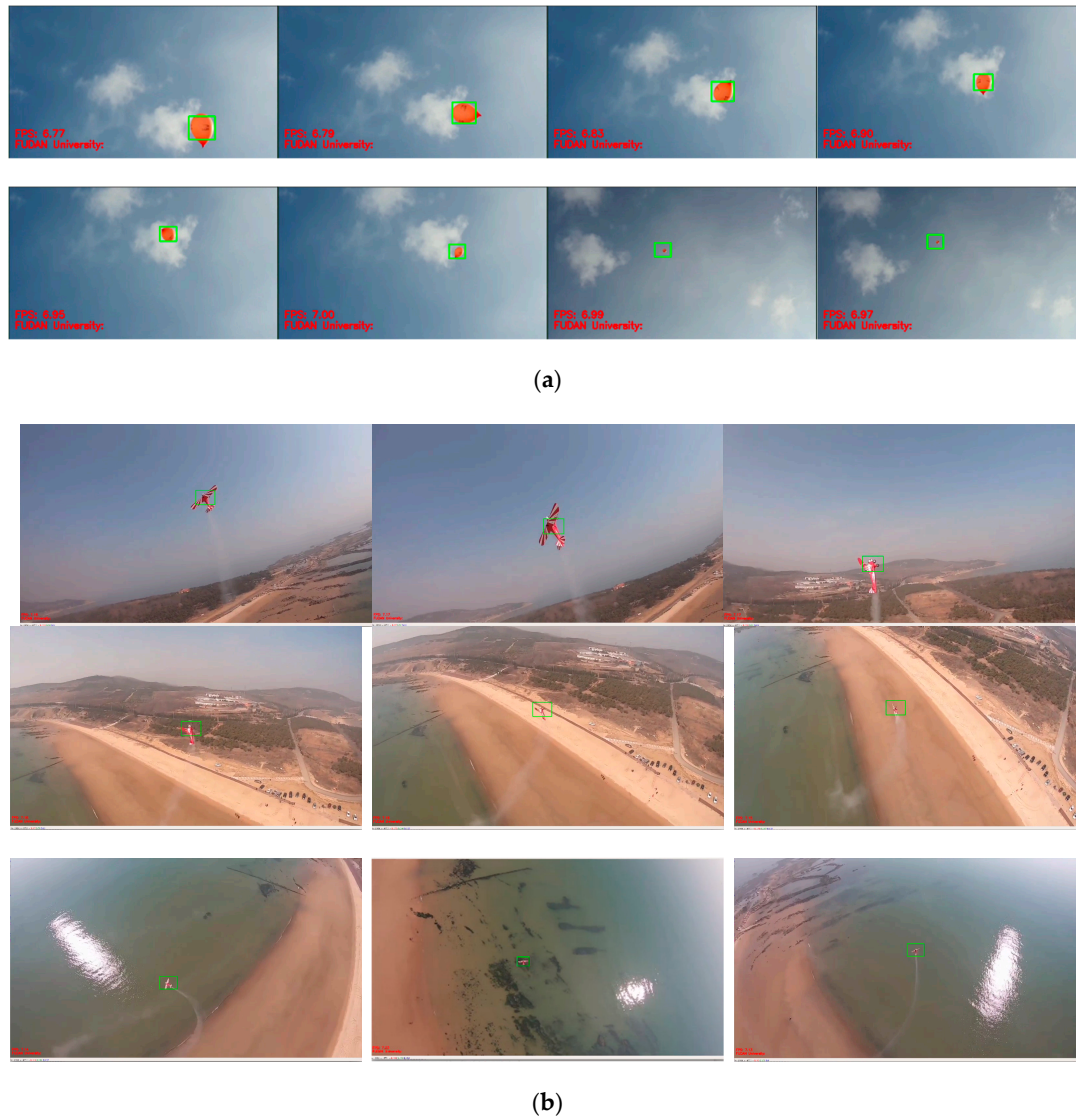
As can be seen from the simulation results in Figure 22, the quadcopter has fairly good performance in terms of tracking the desired trajectories under the designed trajectory tracking controller. As can be seen from Figure 22b, when tracking 3D trajectories, the maximum error is kept within 0.15 m; comparatively, the maximum error under the PID controller reaches 0.3 m. Furthermore, the average error of the designed controller is also lower than that of the PID controller. As seen in Figure 22c–f, the maximum tracking errors along the  $x$ - and  $y$ -axis are around 0.1 m, and both the maximum and average errors are lower than those of the PID controller. As seen in Figure 22g–h, the tracking error remains within 0.02 m along the  $z$ -axis—an evidently better performance than the PID controller.

As seen in Figure 22i,n, both the designed attitude controller and the PID attitude controller can effectively maintain the quadcopter's attitude. However, the attitude controller designed in this paper can enable the three attitude angles of the quadcopter to rapidly converge towards the desired angles.

In the meantime, as seen in Equation (69), the desired trajectories of the quadcopter when  $t \in (0, 5]s$  and  $t \in (25, 30]s$  are straight upwards and straight downwards. Under the effect of external disturbances, the tracking errors along the  $x$ - and  $y$ -axis remain within 0.01 m, and around 0.02 m along the  $z$ -axis. These lay the foundation for quadrotor flight stability. During outdoor testing with disturbance factors like varied wind force, wind direction, and mass, the controller also delivered fairly good stability and reliability.

In the second stage, we ran a test for target tracking algorithm. The test was conducted with the selected balloon in the air and the fixed-wing model aircraft in flight. The result showed that fast and

accurate tracking can be realized in airborne image processing with a velocity of around 7 FPS when the size, speed, posture, and background of the target change. As shown in Figure 23. Table 5 listed the tracking rate of different types of targets.



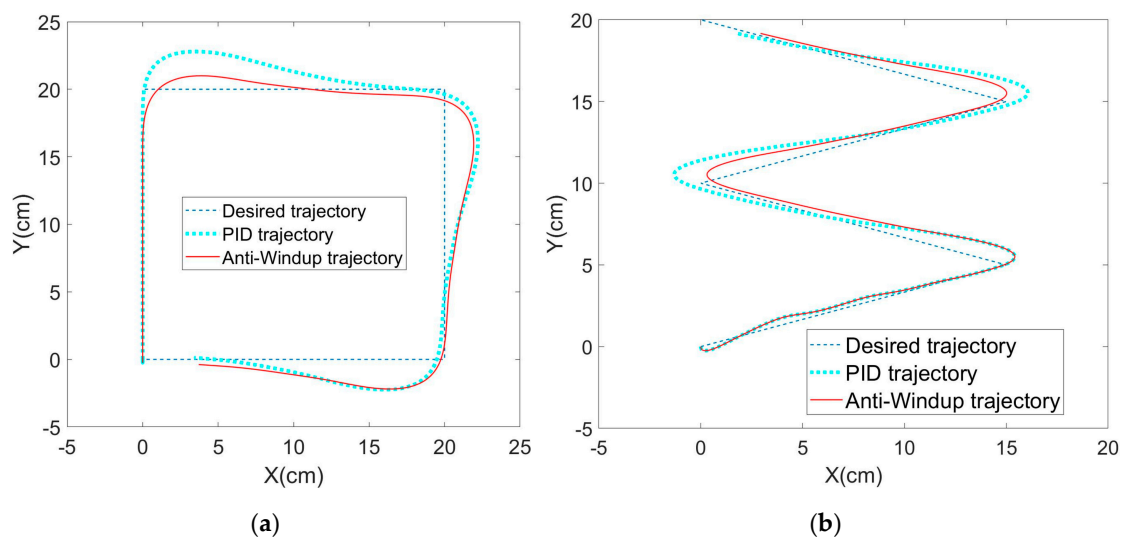
**Figure 23.** Target tracking test: (a) balloon in the air; (b) fixed-wing model aircraft in flight.

**Table 5.** Tracking rates of different types of targets.

Target Type	Balloon	Fixed-Wing Model	DJI Phantom4	DJI Mavic Pro	DJI Mavic Air
Frame rate (Fps)	6.83	7.17	6.52	6.14	6.10

In the third stage, we ran a trajectory tracking MATLAB simulation experiment of the steering gear PTZ. Expected trajectories in shapes of square and Z were set up to test the performance of Anti-Windup PID controller. Compared with a traditional PID controller, the results showed that, under the large signal input with rapid changes, the overshoot of the Anti-Windup PID controller is smaller with faster shrink, and its PTZ controlling performance is better than that of the traditional PID controller, as shown in Figure 24.





**Figure 24.** Simulation test of PTZ controller trajectory tracking: (a) shape of square; (b) shape of Z.

In the fourth stage, we ran an indoor static test. The UAV was placed on a box with a height of 0.5 m. The target was set 1.5 m high and 8 m away from the UAV. When the target was calibrated, Jetson nano, Arduino, and the steering gear PTZ located the target within 2 s, corrected the shooting trajectory, and triggered the shooting procedure. The gel ball trajectory was scattered within the range of 10 cm. High-speed cameras had been used to shoot the target, and then the number of impact points in each target area was counted through slow-motion playback to calculate the accuracy of PTZ launch. Hit rate of indoor static test (deviation <20 cm) was 94.44%. An expected result was achieved in the experiment. Indoor accuracy test, as shown in Table 6 and Figure 25.

**Table 6.** Indoor accuracy test.

Distance from Target Center (cm)	0–5	6–10	11–15	16–20	20+
Number of impact points	32	28	17	8	5



**Figure 25.** Indoor static test.

In the fifth stage, we launched an outdoor dynamic expelling test. As shown in Table 7 and Figure 26. The target was fixed on the DJI (DJI-Innovations, Shenzhen, China) UAV, which could move randomly in the air. The UAV we designed could follow the flight with a response speed of only 0.5 s. Meanwhile, it processed the target tracking program at high speed (the rate in testing environment was 7.2 Fps), and executed the trajectory correcting program (10 times/s) and the launching program (15 times/s). The gel ball dispersion was within 20 cm in field tests. Figure 27 is a close-up shot of the target, and Figure 28 is the gray-scale image of the target, which shows the distribution of impact



points visually. The test method is the same as above. Hit rate of outdoor dynamic test (deviation  $<20$  cm) was 82.98%. In the outdoor dynamic shooting test, fast and effective expulsion of the target UAV was achieved.

**Table 7.** Outdoor accuracy test.

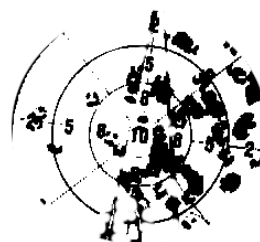
Distance from Target Center (cm)	0–5	6–10	11–15	16–20	20+
Number of impact points	12	19	26	21	16



**Figure 26.** Outdoor dynamic expelling test.



**Figure 27.** Target image.



**Figure 28.** Target in grayscale.

## 5. Conclusions and Future Work

We have designed an integrated identification, tracking, and expelling quadcopter UAV system based on computer vision, proposed a real-time image recognition, location, and tracking plan based on the YOLO and KCF algorithms, and designed corresponding target tracking and expelling strategies. We have made a prototype UAV, conducted target recognition and tracking tests, a PTZ control algorithm test, an indoor static test, and outdoor dynamic expelling testing, and assessed the design plan in different scenarios. According to the experiment, an automatic target identification and tracking system is designed based on the YOLO and KCF algorithms, and the ability to identify and track the target with high speed in a complex environment is realized. Additionally, the high-speed steering gear PTZ that is designed by the Anti-Windup algorithm is adopted to effectively ensure the ability of operating the aiming and expelling device with high speed and precision. The UAV system based on

computer vision introduced in this paper can identify, track, and expel small targets and changeable targets quickly in complex environments, thus realizing the air target control with high efficiency and low cost.

We have demonstrated that the computer vision technology has great feasibility and significance in the design of an integrated identification, tracking and expelling UAV, with broad application prospects in civil fields. However, some problems still deserve further research:

1. Meteorological conditions may have a great influence on target recognition, especially the light intensity, dust, smoke, and other interference factors.
2. Based on the multiple-target and tracking studies of the YOLO and KCF algorithms, an estimation of the moving speed and trajectory of multiple targets, and the prediction of the hazard level.
3. The impact of vibration on PTZ and attack precision of targets in the operating process of the trigger mechanism. The electrically controlled simulated launch device used in this experiment has a small recoil force, so the impact of vibration is not considered for the time being. In future practical applications, the vibrations created by the target expulsion device cannot be ignored, and a damping device should be designed.
4. Unlike high-cost schemes, this paper focuses on a low-cost target-expelling scheme, using gel balls as combat ammunition. In a real flight environment, wind power, wind direction, and ammunition weight are the factors that must be considered, and an analysis of hitting points spread across different ranges is needed.

The design of the obstacle avoidance system based on multisensor information fusion has been completed and successfully tested. It will be introduced in detail in future articles. A video (Supplementary Materials) link to the whole experiment has been included at the end of this article and gives part of the obstacle avoidance test. You are welcome to click on it and check our results.

**Supplementary Materials:** Video of the whole experiment: <https://youtu.be/IPUHHmi-Gto>.

**Author Contributions:** L.W. and J.A. conceived the methodology; L.W. designed the experiments; L.W., L.Z. and Z.X. performed the experiments and analyzed the data; J.A. provided valuable advice and guidance; L.W. wrote the paper. All authors have read and agreed to the published version of the manuscript

**Funding:** This research received no external funding.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Abbreviation

UAV	Unmanned Aerial Vehicle
DOF	Degrees of Freedom
PTZ	Pan/Tilt/Zoom
YOLO	You Only Look Once
KCF	Kernel Correlation Filter
PID	Proportional Integral Derivative
RF	Radio Frequency
FMCW	Frequency Modulated Continuous Wave
MIMO	Multiple Input Multiple Output
DOA	Direction of Arrival
RSS	Received-Signal-Strength
HOG	Histogram of Oriented Gradient
3D	Three-Dimensional
EMI	Electromagnetic Interference
RBFNN	Radial Basis Function Neural Network
DNN	Deep Neural Network
CNN	Convolutional Neural Network
R-CNN	Regions with Convolutional Neural Network Features

FPS	Frames Per Second
CSRT	Channel and Spatial Reliability Tracking
MIL	Multiple Instance Learning
MOSSE	Minimum Output Sum of Squared Error
TLD	Tracking Learning Detection
HOG	Histogram of Oriented Gradient

## References

1. Gatwick Airport: Drones Ground Flights. Available online: <https://www.bbc.com/news/uk-england-sussex-46623754> (accessed on 6 April 2020).
2. Nowak, A.; Naus, K.; Maksimiuk, D. A method of fast and simultaneous calibration of many mobile FMCW radars operating in a network anti-drone system. *Remote Sens.* **2019**, *11*, 2617. [\[CrossRef\]](#)
3. Multerer, T.; Ganis, A.; Prechtel, U.; Miralles, E.; Meusling, A.; Mietzner, J.; Vossiek, M.; Loghi, M.; Ziegler, V. Low-cost jamming system against small drones using a 3D MIMO radar based tracking. In Proceedings of the 2017 European Radar Conference (EURAD), Nuremberg, Germany, 11–13 October 2017; pp. 299–302.
4. Lee, J.; Go, Y.-J.; Kim, S.; Choi, J.-S. Flight Path Measurement of Drones Using Microphone Array and Performance Improvement Method Using Unscented Kalman Filter. *J. Korean Soc. Aeronaut. Space Sci.* **2018**, *46*, 975–985.
5. Zhang, Z.; Cao, Y.; Ding, M.; Zhuang, L.; Yao, W. An intruder detection algorithm for vision based sense and avoid system. In Proceedings of the 2016 International Conference on Unmanned Aircraft Systems (ICUAS), Arlington, VA, USA, 7–10 June 2016; pp. 550–556.
6. Ganti, S.R.; Kim, Y. Implementation of detection and tracking mechanism for small UAS. In Proceedings of the 2016 International Conference on Unmanned Aircraft Systems (ICUAS), Arlington, VA, USA, 7–10 June 2016; pp. 1254–1260.
7. Ju, M.; Luo, H.; Wang, Z.; Hui, B.; Chang, Z. The application of improved YOLO V3 in multi-scale target detection. *Appl. Sci.* **2019**, *9*, 3775. [\[CrossRef\]](#)
8. Lan, W.; Dang, J.; Wang, Y.; Wang, S. Pedestrian detection based on yolo network model. In Proceedings of the 2018 IEEE International Conference on Mechatronics and Automation (ICMA), Changchun, China, 5–8 August 2018; pp. 1547–1551.
9. Oltean, G.; Florea, C.; Orghidan, R.; Oltean, V. Towards Real Time Vehicle Counting using YOLO-Tiny and Fast Motion Estimation. In Proceedings of the 2019 IEEE 25th International Symposium for Design and Technology in Electronic Packaging (SIITME), Cluj-Napoca, Romania, 23–26 October 2019; pp. 240–243.
10. Lestari, D.P.; Kosasih, R.; Handhika, T.; Sari, I.; Fahrurrozi, A. Fire Hotspots Detection System on CCTV Videos Using You Only Look Once (YOLO) Method and Tiny YOLO Model for High Buildings Evacuation. In Proceedings of the 2019 2nd International Conference of Computer and Informatics Engineering (IC2IE), Banyuwangi, Indonesia, 10–11 September 2019; pp. 87–92.
11. Zhang, R.; Yang, Y.; Wang, W.; Zeng, L.; Chen, J.; McGrath, S. An algorithm for obstacle detection based on YOLO and light filed camera. In Proceedings of the 2018 12th International Conference on Sensing Technology (ICST), Limerick, Ireland, 4–6 December 2018; pp. 223–226.
12. Yoon, Y.; Hwang, H.; Choi, Y.; Joo, M.; Oh, H.; Park, I.; Lee, K.-H.; Hwang, J.-H. Analyzing Basketball Movements and Pass Relationships Using Realtime Object Tracking Techniques Based on Deep Learning. *IEEE Access* **2019**, *7*, 56564–56576. [\[CrossRef\]](#)
13. Awad, S.; Alkashash, A.; Fatima, H.; Eltoum, I. Yolov3 Pap Test: The Use of Publically Available Neural Networks for Screening Pap Test. In *Laboratory Investigation*; Nature Publishing Group: New York, NY, USA, 2020; pp. 327–328.
14. Orsag, M.; Korpela, C.; Oh, P. Modeling and control of MM-UAV: Mobile manipulating unmanned aerial vehicle. *J. Intell. Robot. Syst.* **2013**, *69*, 227–240. [\[CrossRef\]](#)
15. Thomas, J.; Polin, J.; Sreenath, K.; Kumar, V. Avian-inspired grasping for quadrotor micro UAVs. In Proceedings of the ASME 2013 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Portland, OR, USA, 4–7 August 2013.

16. Bellicoso, C.D.; Buonocore, L.R.; Lippiello, V.; Siciliano, B. Design, modeling and control of a 5-DoF light-weight robot arm for aerial manipulation. In Proceedings of the 2015 23rd Mediterranean Conference on Control and Automation (MED), Torremolinos, Spain, 16–19 June 2015; pp. 853–858.
17. Hua, K.C. Water Crystal Paintball Soft Bullet Gun Unboxing & Testing. Available online: <https://www.youtube.com/watch?v=54Zz3q-LouI> (accessed on 3 March 2020).
18. Sangdani, M.; Tavakolpour-Saleh, A.; Lotfavar, A. Genetic algorithm-based optimal computed torque control of a vision-based tracker robot: Simulation and experiment. *Eng. Appl. Artif. Intell.* **2018**, *67*, 24–38. [[CrossRef](#)]
19. Rabah, M.; Rohan, A.; Han, Y.-J.; Kim, S.-H. Design of fuzzy-PID controller for quadcopter trajectory-tracking. *Int. J. Fuzzy Log. Intell. Syst.* **2018**, *18*, 204–213. [[CrossRef](#)]
20. Nguyen, N.P.; Hong, S.K. Fault-tolerant control of quadcopter UAVs using robust adaptive sliding mode approach. *Energies* **2019**, *12*, 95. [[CrossRef](#)]
21. LOUBAR, H.; Boushaki, R.Z.; ARIBI, Y.; Abdellah, K. Altitude Back-stepping Control of Quadcopter. In Proceedings of the 2019 International Conference on Applied Automation and Industrial Diagnostics (ICAAID), Elazig, Turkey, 25–27 September 2019; pp. 1–11.
22. Kuantama, E.; Tarca, I.; Tarca, R. Feedback linearization LQR control for quadcopter position tracking. In Proceedings of the 2018 5th International Conference on Control, Decision and Information Technologies (CoDIT), Thessaloniki, Greece, 10–13 April 2018; pp. 204–209.
23. Talha, M.; Asghar, F.; Rohan, A.; Rabah, M.; Kim, S.H. Fuzzy logic-based robust and autonomous safe landing for UAV quadcopter. *Arab. J. Sci. Eng.* **2019**, *44*, 2627–2639. [[CrossRef](#)]
24. Ren, B.; Ge, S.S.; Tee, K.P.; Lee, T.H. Adaptive neural control for output feedback nonlinear systems using a barrier Lyapunov function. *IEEE Trans. Neural Netw.* **2010**, *21*, 1339–1345. [[PubMed](#)]
25. Ge, S.S.; Wang, C. Adaptive neural control of uncertain MIMO nonlinear systems. *IEEE Trans. Neural Netw.* **2004**, *15*, 674–692. [[CrossRef](#)] [[PubMed](#)]
26. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
27. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2015; pp. 91–99.
28. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 11–18 December 2015; pp. 1440–1448.
29. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
30. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7263–7271.
31. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
32. Zhang, X.; Yang, W.; Tang, X.; Liu, J. A fast learning method for accurate and robust lane detection using two-stage feature extraction with YOLO v3. *Sensors* **2018**, *18*, 4308. [[CrossRef](#)] [[PubMed](#)]
33. Howse, J. *OpenCV Computer Vision with Python*; Packt Publishing Ltd.: Birmingham, UK, 2013.

