2020 IEEE International Conference on Robotics and Automation (ICRA)
31 May - 31 August, 2020. Paris, France

An Autonomous Intercept Drone with Image-based Visual Servo

Kun Yang and Quan Quan

***Abstract*— For most people on the ground, facing an unwanted drone buzzing around overhead, there is not a lot that we can do, especially if it is out of gun (radio wave gun or shotgun) range. A solution to this is to use intercept drones that seek out and bring down other drones. In order to make the interception autonomous, an image-based visual servo algorithm is designed with a forward-looking monocular camera. The control command, namely the angular velocity and thrust, is generated for intercept drones to implement accurate and fast interception. The proposed method is demonstrated in both hardware-in-the-loop simulation and demonstrative flight experiments.**

I. Introduction

Micro-small drones in low-altitude airspace have been widely used in the fields of search-and-rescue work, remote sensing, aerial photography, surveillance, and especially in military applications [1]. However, some prohibited or hostile drones may enter into no-fly zones such as airports, nuclear power plants, and populated areas [2]. What can be done is limited, especially if it is out of an anti-drone device (radio wave gun or shotgun or GPS interference [3], [4] or Laser weapon [5], [6], [7]) range. Even if the intruder drones are nearby, shooting or interference will also cause the secondary hazards as the debris will fall down to hit the people on the ground [8] or disturb other aircrafts, especially those at the airport. A solution to this problem is to use intercept drones that seek out and bring down other drones by the net gun [9]. Such interceptions by multicopters were reported based on the operation by one or two remote pilots on the ground. However, this method is difficult to extend as it will put too much workload on remote pilots. Therefore, an intercept quadcopter should be with the abilities of autonomy and aggressive maneuver so that it can catch intruders fast without too much workload on remote pilots.

This interception problem is a kind of pursuit-evasion problem [10]. Different from traditional "one-sided search" and "adversarial search" methods, our approach aims to perform the interception autonomously once the intruder is detected. Therefore, an Image-Based Visual Servo (IBVS) algorithm is designed with a forward-looking monocular camera on an intercept quadcopter. Visual servo control methods can be divided into two main categories, including IBVS [11] and Position-based Visual Servo (PBVS) [12]. IBVS control consists of a feedback signal that is composed of pure image space information (i.e., the control objective is defined in terms of an image pixel error). IBVS is more likely to keep the intruder in the field of view during the whole interception process than PBVS, because its feedback is directly obtained from the image without transferring the image-space measurement to another space. By PBVS, two cameras on an interceptor are required theoretically. However, in practice, the baseline distance of two cameras is too short to detect the size accurately and robustly. By contrast, the intruders' size is not necessary for IBVS, which makes the detection result accurately and robustly. As far as we know, the applications of IBVS algorithms often focus on fixed/slow target approaching [13], [14], or following [15] with down-looking cameras. The challenges also contributions here are in the following.

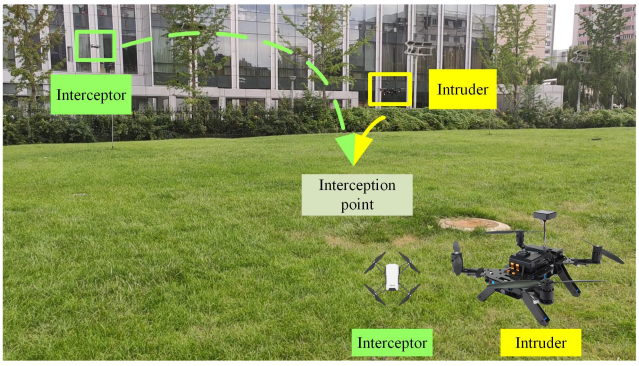

Fig. 1. A demonstrative interception process.

1) An IBVS model with a forward-looking monocular camera mounted on a quadcopter has to be established first, where the underactuated characteristics of motion of the quadcopter are combined with the Jacobian matrix related to the derivative of the image space measurements to the camera's linear and angular velocities.
2) Based on the established model, an attitude controller rather than a velocity controller is designed for fast target approaching. The effectiveness of the proposed controller is analyzed.

Fig. 1 depicts the interception process. Once an intruder enters the field of view of the monocular camera mounted on the interceptor, the intruder will be automatically identified. When the operator confirms the intruder, the interceptor will intercept it immediately. To demonstrate the effectiveness, Hardware-in-the-Loop (HIL) simulation with Pixhawk and AirSim platform [16] and outdoor flight experiments are performed.

The paper is organized as follows: in Sec. II, a brief summary of the assumptions and problems is proposed for leading the following investigation. In Sec. III, the visual

This work was supported by the National Natural Science Foundation of China under Grant 61973015.

K. Yang and Q. Quan are with School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China (e-mail: {yangkun_buaa, qq_buaa}@buaa.edu.cn).

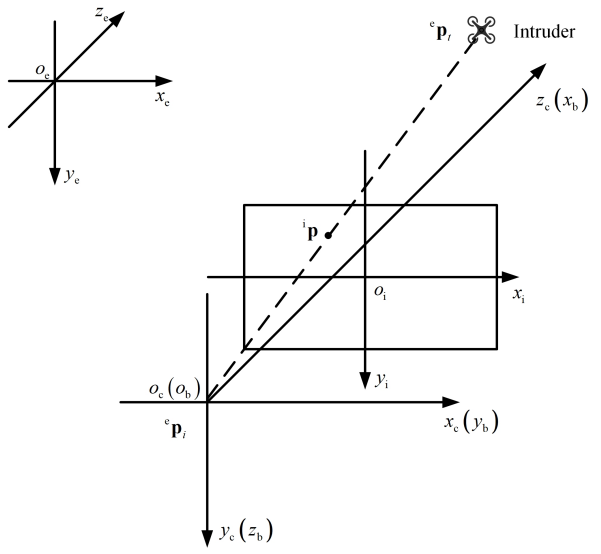

Fig. 2. Relationship among coordinate systems. $o_{\mathrm{e}}-x_{\mathrm{e}}y_{\mathrm{e}}z_{\mathrm{e}}$ is the inertial coordinate system $\{\mathrm{e}\}$, $o_{\mathrm{b}}-x_{\mathrm{b}}y_{\mathrm{b}}z_{\mathrm{b}}$ is the body coordinate system $\{\mathrm{b}\}$, and the conversion from $\{\mathrm{b}\}$ to $\{\mathrm{e}\}$ is $\mathbf{R}_{\mathrm{b}}^{\mathrm{e}},\mathbf{t}_{\mathrm{b}}^{\mathrm{e}}$. $o_{\mathrm{c}}-x_{\mathrm{c}}y_{\mathrm{c}}z_{\mathrm{c}}$ is the camera coordinate system $\{\mathrm{c}\}$, and the conversion from $\{\mathrm{c}\}$ to $\{\mathrm{b}\}$ is $\mathbf{R}_{\mathrm{c}}^{\mathrm{b}},\mathbf{t}_{\mathrm{c}}^{\mathrm{b}}$. $o_{\mathrm{i}}-x_{\mathrm{i}}y_{\mathrm{i}}$ is the image coordinate system $\{\mathrm{i}\}$. The coordinate of the intruder in the coordinate system $\{\mathrm{e}\}$ is ${}^{\mathrm{e}}\mathbf{p}_t$, and the interceptor is ${}^{\mathrm{e}}\mathbf{p}_i$.

servo model and visual servo control are proposed and proved theoretically. Then a HIL simulation and a demonstrative experiment are shown in Sec. IV. This paper draws conclusions in Sec. V.

II. Problem Formulation

In this paper, we will design an IBVS based controller for the interceptor. The interceptor will work in a range that is neither too far nor too close to the intruder so that the mounted net gun can shoot to capture the intruder. To set a shooting distance, we need to use the coverage rate (the default value is 50%) of the intruder in picture. In this paper, we chose a multicopter as the interceptor, which is modeled as a rigid body with forces and torques only from the rotors and gravity. The relationship among the coordinate systems $\{\mathrm{e}\}$, $\{\mathrm{b}\}$, $\{\mathrm{c}\}$, and $\{\mathrm{i}\}$ is shown in Fig. 2.

According to [17], a simplified multicopter dynamics model is used:

- horizontal position channel model

$$ {}^{\mathrm{e}}\ddot{\mathbf{p}}_h = -g\mathbf{A}_\psi\Theta_h \tag{1}$$

- height channel model

$$ {}^{\mathrm{e}}\ddot{p}_z = g - \frac{f}{m}\cos\theta \tag{2}$$

- attitude model

$$ \dot{\Theta} = {}^{\mathrm{b}}\omega \tag{3}$$

where ${}^{\mathrm{e}}\mathbf{p}_h = \begin{bmatrix} {}^{\mathrm{e}}p_x & {}^{\mathrm{e}}p_y \end{bmatrix}^{\mathrm{T}}$ and ${}^{\mathrm{e}}p_z$ are the horizontal and vertical position of the interceptor under the inertial system, respectively; $\Theta = \begin{bmatrix} \phi & \theta & \psi \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^3$ $(\Theta_h = \begin{bmatrix} \phi & \theta \end{bmatrix}^{\mathrm{T}})$ and ${}^{\mathrm{b}}\omega \in \mathbb{R}^3$ are Euler angles and the angular velocity the body coordinate system respectively;

$$ \mathbf{R}_\psi = \begin{bmatrix} \cos\psi & -\sin\psi \\ \sin\psi & \cos\psi \end{bmatrix}, \quad \mathbf{A}_\psi = \mathbf{R}_\psi \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}; $$

$\tau = \begin{bmatrix} \tau_x & \tau_y & \tau_z \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^3$ represents the torque generated by propellers, and f is the total thrust generated by propellers.

To simplify the design, two assumptions are made in the following.

Assumption 1. The origin of the camera coordinate system is aligned with the origin of the body coordinate system. In other words, their displacement $\mathbf{t}_{\mathrm{c}}^{\mathrm{b}} = \mathbf{0}$ and the rotation matrix is

$$ \mathbf{R}_{\mathrm{c}}^{\mathrm{b}} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}. \tag{4}$$

Assumption 2. The visual tracking module of the interceptor is able to accurately capture the intruder's central coordinate $\left({}^{\mathrm{i}}x,{}^{\mathrm{i}}y\right)$ from images in real-time.

Remark 1. The camera we used in this paper is mounted looking forward. So, the optical axis is in the same direction as the multicopter head. As shown in Fig. 2, $\mathbf{R}_{\mathrm{c}}^{\mathrm{b}}$ in the form (4) holds. With it, we have

$$ {}^{\mathrm{b}}\mathbf{v} = \mathbf{R}_{\mathrm{c}}^{\mathrm{b}} \cdot {}^{\mathrm{c}}\mathbf{v} $$
$$ {}^{\mathrm{b}}\omega = \mathbf{R}_{\mathrm{c}}^{\mathrm{b}} \cdot {}^{\mathrm{c}}\omega. $$

We set the camera's optical center as the origin to establish the body coordinate system, so $\mathbf{t}_{\mathrm{c}}^{\mathrm{b}} = \mathbf{0}$. In this case, the convergence center of the feature point is the origin o_{i} of the image coordinate system.

Remark 2. In the experimental section, we will show the network we designed can perform reliable tracking in 20Hz. The image tracking error $\mathbf{e}$ in the image is defined as

$$ \mathbf{e} = \begin{bmatrix} e_x \\ e_y \end{bmatrix} \triangleq \begin{bmatrix} {}^{\mathrm{i}}x - {}^{\mathrm{i}}x_o \\ {}^{\mathrm{i}}y - {}^{\mathrm{i}}y_o \end{bmatrix} \tag{5}$$

where $\left({}^{\mathrm{i}}x_o,{}^{\mathrm{i}}y_o\right)$ is a fixed point in the image called the convergence point of image. Based on **Assumption 1**, $\left({}^{\mathrm{i}}x_o,{}^{\mathrm{i}}y_o\right)$ is aligned with o_{i}.

Based on **Assumptions 1**,**2**, the **Interception Control Problem** is stated in the following.

Interception Control Problem. Under **Assumptions 1**,**2**, we suppose that the state information of the multicopter can be accessible by the Inertial Measurement Unit (IMU), while the positions of intruder and interceptor in the inertial system are unknown. For the interceptor model (1), (2) and (3), design controllers for f, ${}^{\mathrm{b}}\omega$ such that the image tracking error converges to zero, and the interceptor multicopter approaches a given intruder, i.e., $\mathbf{e} \to \mathbf{0}$, $d \to 0$. Here, the distance between the interceptor and the intruder is defined as $d = \|{}^{\mathrm{e}}\mathbf{p}_i - {}^{\mathrm{e}}\mathbf{p}_t\|$.

III. Interception Control of Multicopter with IBVS

No matter which kind of method is used (IBVS, PBVS, or homography-based visual servo control), it is supposed to calculate desired linear velocity and angular velocity (six

control variables) through the motion of image features. However, a multicopter is an underactuated system, which has six degrees of freedom controlled by four control variables, namely three-axis angular velocities ${}^{\mathrm{b}}\boldsymbol{\omega}$ and one thrust f. Therefore, the six degrees of freedom of multicopter is coupled. Here is the obstacle while using IBVS. In this section, an IBVS model with a forward-looking monocular camera mounted on a multicopter is established first with a longitudinal channel model and a lateral channel model derived. Then, based on the two obtained models, the controllers are designed separately.

A. Visual servo model

The relationship between the camera coordinate system and the image coordinate system is described in Fig. 2. Visual servo-based solutions aim to reduce image tracking error. In IBVS, $\mathbf{s} = \begin{bmatrix} {}^{\mathrm{i}}x & {}^{\mathrm{i}}y \end{bmatrix}^{\mathrm{T}}$ is the center of the intruder in the image. In order to design a velocity controller, the relationship between $\mathbf{s}$ and camera velocity ${}^{\mathrm{c}}\mathbf{v}$ needs to be obtained. According to the basic equation of IBVS [18], the relationship between $\dot{\mathbf{s}}$ and ${}^{\mathrm{c}}\mathbf{v}$ is

$$\dot{\mathbf{s}} = \mathbf{L}_{\mathbf{s}}{}^{\mathrm{c}}\tilde{\mathbf{v}} \tag{6}$$

where $\mathbf{L}_s \in \mathbb{R}^{2\times 6}$ is called the Jacobi matrix in the form as

$$\mathbf{L}_s = \begin{bmatrix} -\frac{1}{{}^{\mathrm{c}}z} & 0 & \frac{{}^{\mathrm{i}}x}{{}^{\mathrm{c}}z} & {}^{\mathrm{i}}x{}^{\mathrm{i}}y & -(1+{}^{\mathrm{i}}x^2) & {}^{\mathrm{i}}y \\ 0 & -\frac{1}{{}^{\mathrm{c}}z} & \frac{{}^{\mathrm{i}}y}{{}^{\mathrm{c}}z} & 1+{}^{\mathrm{i}}y^2 & -{}^{\mathrm{i}}x{}^{\mathrm{i}}y & -{}^{\mathrm{i}}x \end{bmatrix} \tag{7}$$

and ${}^{\mathrm{c}}z$ is the distance between the center of the intruder and the optical center along the optical axis; ${}^{\mathrm{c}}\tilde{\mathbf{v}} = \begin{bmatrix} {}^{\mathrm{c}}\mathbf{v}^{\mathrm{T}} & {}^{\mathrm{c}}\boldsymbol{\omega}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$ with ${}^{\mathrm{c}}\mathbf{v} = \begin{bmatrix} {}^{\mathrm{c}}v_x & {}^{\mathrm{c}}v_y & {}^{\mathrm{c}}v_z \end{bmatrix}^{\mathrm{T}}$ and ${}^{\mathrm{c}}\boldsymbol{\omega} = \begin{bmatrix} {}^{\mathrm{c}}\omega_x & {}^{\mathrm{c}}\omega_y & {}^{\mathrm{c}}\omega_z \end{bmatrix}^{\mathrm{T}}$.

Because the multicopter is an underactuated system, the six variables are not independent of each other, and there are only four free variables. In order to simplify the calculation, we eliminate some unimportant variables. Equation (6) is decomposed into the horizontal and vertical directions. According to equation (5), $\mathbf{e} = \begin{bmatrix} {}^{\mathrm{i}}x & {}^{\mathrm{i}}y \end{bmatrix}^{\mathrm{T}}$, and $\dot{\mathbf{e}} = \begin{bmatrix} {}^{\mathrm{i}}\dot{x} & {}^{\mathrm{i}}\dot{y} \end{bmatrix}^{\mathrm{T}}$.

- In the x_c-z_c plane, the three degrees of freedom are ${}^{\mathrm{c}}v_x, {}^{\mathrm{c}}v_z, {}^{\mathrm{c}}\omega_y$ with ${}^{\mathrm{c}}v_y = 0, {}^{\mathrm{c}}\omega_x = 0$. Equation (6) becomes

$$\dot{e}_x = -\frac{{}^{\mathrm{c}}v_x}{{}^{\mathrm{c}}z} + \frac{{}^{\mathrm{i}}x{}^{\mathrm{c}}v_z}{{}^{\mathrm{c}}z} - \left(1+{}^{\mathrm{i}}x^2\right){}^{\mathrm{c}}\omega_y. \tag{8}$$

- In the y_c-z_c plane, the three degrees of freedom are ${}^{\mathrm{c}}v_y, {}^{\mathrm{c}}v_z, {}^{\mathrm{c}}\omega_x$ with ${}^{\mathrm{c}}v_x = 0, {}^{\mathrm{c}}\omega_y = 0$. Equation (6) becomes

$$\dot{e}_y = -\frac{{}^{\mathrm{c}}v_y}{{}^{\mathrm{c}}z} + \frac{{}^{\mathrm{i}}y{}^{\mathrm{c}}v_z}{{}^{\mathrm{c}}z} + \left(1+{}^{\mathrm{i}}y^2\right){}^{\mathrm{c}}\omega_x. \tag{9}$$

From (8) and (9), the decomposed Jacobian equation can be combined as

$$\dot{\mathbf{e}} = \mathbf{L}_e{}^{\mathrm{c}}\mathbf{v}^*,$$
$$\mathbf{L}_e = \begin{bmatrix} -\frac{1}{{}^{\mathrm{c}}z} & 0 & \frac{{}^{\mathrm{i}}x}{{}^{\mathrm{c}}z} & 0 & -(1+{}^{\mathrm{i}}x^2) \\ 0 & -\frac{1}{{}^{\mathrm{c}}z} & \frac{{}^{\mathrm{i}}y}{{}^{\mathrm{c}}z} & 1+{}^{\mathrm{i}}y^2 & 0 \end{bmatrix} \tag{10}$$

where ${}^{\mathrm{c}}\mathbf{v}^* = \begin{bmatrix} {}^{\mathrm{c}}v_x & {}^{\mathrm{c}}v_y & {}^{\mathrm{c}}v_z & {}^{\mathrm{c}}\omega_x & {}^{\mathrm{c}}\omega_y \end{bmatrix}^{\mathrm{T}}$ refers to the relative velocities in the camera coordinate system.

Combining Equation (2), (3) and (9), we get the *longitudinal channel model* as

$$\begin{cases} {}^{\mathrm{e}}\dot{v}_z = g - \frac{f}{m}\cos\theta \\ \dot{\theta} = {}^{\mathrm{c}}\omega_x \\ \dot{e}_y = -\frac{{}^{\mathrm{c}}v_y}{{}^{\mathrm{c}}z} + \frac{{}^{\mathrm{i}}y{}^{\mathrm{c}}v_z}{{}^{\mathrm{c}}z} + \left(1+{}^{\mathrm{i}}y^2\right){}^{\mathrm{c}}\omega_x. \end{cases} \tag{11}$$

Combining Equation (1), (3) and (9), we get the *lateral channel model* as

$$\begin{cases} {}^{\mathrm{e}}\dot{\mathbf{v}}_h = -g\mathbf{A}_\psi\Theta_h \\ \dot{\Theta}_h = \begin{bmatrix} {}^{c}\omega_z \\ {}^{c}\omega_y \end{bmatrix} \\ \dot{e}_x = -\frac{{}^{\mathrm{c}}v_x}{{}^{\mathrm{c}}z} + \frac{{}^{\mathrm{i}}x{}^{\mathrm{c}}v_z}{{}^{\mathrm{c}}z} - \left(1+{}^{\mathrm{i}}x^2\right){}^{\mathrm{c}}\omega_y. \end{cases} \tag{12}$$

Based on the two models obtained, we can restate the **Interception Control Problem**. It can be divided into the following two sub-problems.

Problem 1 (Longitudinal channel controller design). For the longitudinal channel model (11), design a controller for $f, {}^{\mathrm{c}}\omega_x$, such that the vertical image tracking error converges to zero and the multicopter approaches the target, i.e., $e_y \to 0$, $d \to 0$.

Problem 2 (Lateral channel controller design). For the lateral channel model (12), design a lateral channel controller for $\begin{bmatrix} {}^{c}\omega_z & {}^{c}\omega_y \end{bmatrix}^{\mathrm{T}}$, such that the horizontal image tracking error converges to zero and the multicopter approaches the target, i.e., $e_x \to 0$.

B. Visual servo control

1) Longitudinal channel controller design: For **Problem 1**, we design a controller for the longitudinal channel model (11). First, as for (9), taking ${}^{\mathrm{c}}v_y$ and ${}^{\mathrm{c}}\omega_x$ as the control input, the controller is designed as

$${}^{\mathrm{c}}v_y = k_1 e_y, \quad {}^{\mathrm{c}}\omega_x = -k_2 e_y. \tag{13}$$

With them, Equation (9) becomes

$$\dot{e}_y = -\lambda_1 e_y \tag{14}$$

where

$$\lambda_1 = \left(\frac{k_1}{{}^{\mathrm{c}}z} - \frac{{}^{\mathrm{c}}v_z}{{}^{\mathrm{c}}z} + k_2(1+{e_y}^2)\right).$$

In order to make $\lambda_1 > 0$, we have $k_1 > {}^{\mathrm{c}}v_z$. In other words, if k_1 is chosen sufficiently large, namely $k_1 > \max({}^{c}v_z)$, then $\lambda_1 > 0$. The parameter k_2 is another parameter to reduce the image tracking error component e_y by controlling the pitch angle. Then, we can ensure that $e_y \to 0$. In the following, based on (12), taking $k_1 e_y$ as the desired velocity, we design the thrust as

$$f = \frac{m}{\cos\theta}\left(k_4({}^{\mathrm{c}}v_y - k_1 e_y) + g\right) \tag{15}$$

where k_4 is a proportional parameter for controlling ${}^{\mathrm{c}}v_y$ to reach the desired speed.

Furthermore, we hope the interceptor approaching the intruder ($d \to 0$) so the pitch should have a component (head down) to make ${}^{\mathrm{c}}v_z > 0$. Here, we design

$${}^{\mathrm{c}}\omega_x = -k_2 e_y - k_3 e_\theta \tag{16}$$

$$\theta_d = \max\left\{\theta_{th}, \arctan\left(\frac{e_y}{f_{oc}}\right)\right\} \tag{17}$$

where f_{oc} is the focal length; $e_\theta = \theta - \theta_d$ is the pitch angle error; θ_d is designed to make the interceptor point at the intruder; $\theta_{th} < 0$ is the threshold of the pitch angle, ensuring that the interceptor is flying towards the intruder; k_3 is a proportional parameter for pitch angle tracking θ_d. If k_3 is increased, then the interceptor can track the desired pitch angle fast. Combined with the Equations (15) and (16), the longitudinal channel controller is obtained.

2) Lateral channel controller design: Similarly, for **Problem 2**, we design the controller for the lateral channel model (12). Since the change in the roll angle causes a dramatic change in image, it is expected that the roll angle ϕ is zero. According to Equation (12), we can obtain ${}^{\mathrm{c}}\dot{v}_x = g\phi$. When the roll angle ϕ is nearly zero, ${}^{\mathrm{c}}\dot{v}_x$ is near to zero as well. In order to keep the roll angle near to zero, let $\phi_d = 0$, the controller is designed as

$${}^{\mathrm{c}}\omega_z = k_6(\phi - \phi_d) \tag{18}$$

where k_6 is a proportional parameter of roll angle error.

The proportional control of e_x is designed to get ${}^{\mathrm{c}}\omega_y$, which is

$${}^{\mathrm{c}}\omega_y = k_5 e_x. \tag{19}$$

Therefore, Equation (8) is simplified as

$$\dot{e}_x = -\lambda_2 e_x \tag{20}$$

where

$$\lambda_2 = (-\frac{{}^{\mathrm{c}}v_z}{{}^{\mathrm{c}}z} + k_5(1 + {e_x}^2)).$$

In order to make $\lambda_2 > 0$, we have $k_5 > \frac{{}^{\mathrm{c}}v_z}{{}^{\mathrm{c}}z(1+{e_x}^2)}$. That is, if k_5 is chosen sufficiently large and the distance ${}^{\mathrm{c}}z$ is not too small, then $\lambda_2 > 0$. We can ensure that $e_x \to 0$. Combined with the Equations (18) and (19), the lateral channel controller is obtained.

Mathematically, under **Assumptions 1–2**, suppose that the interceptor model is given by (1), (2) and (3), and visual servo model is got by Equation (10). Then, if the final controller is designed as

$$\begin{cases} {}^{\mathrm{b}}\omega = \mathbf{R}_{\mathrm{c}}^{\mathrm{b}} \begin{bmatrix} -k_2 e_y - k_3(\theta - \theta_d) \\ k_5 e_x \\ k_6(\phi - \phi_d) \end{bmatrix} \\ f = \frac{m}{\cos\theta}(k_4({}^{\mathrm{c}}v_y - k_1 e_y) + g) \end{cases} \tag{21}$$

where $k_i > 0, i = 1, 2, \cdots, 6$ are control gains, the image tracking error will converge to zero, and the interceptor will approach the intruder.

The analysis can be made through a Lyapunov function. A Lyapunov candidate is selected as

$$\begin{aligned} L &= \tfrac{1}{2}\left(\mathbf{e}^{\mathrm{T}}\mathbf{e} + e_\theta^2 + d^2\right) \\ &= \tfrac{1}{2}\left(\left(1 + \tfrac{{}^{\mathrm{c}}z^2}{f_{oc}^2}\right)e_x^2 + \left(1 + \tfrac{{}^{\mathrm{c}}z^2}{f_{oc}^2}\right)e_y^2 + e_\theta^2 + {}^{\mathrm{c}}z^2\right) \geq 0. \end{aligned} \tag{22}$$

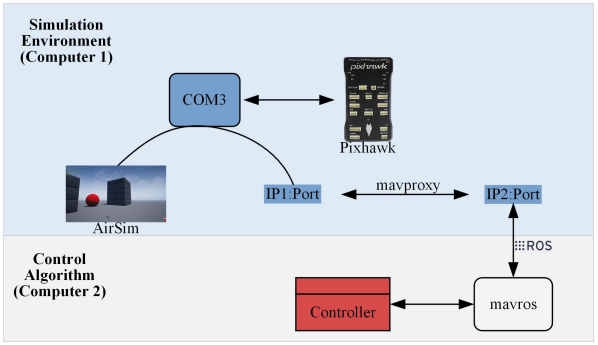

Fig. 3. HIL simulation data flow. The simulation environment runs on a computer (Computer 1), and the IBVS control algorithm runs on another computer (Computer 2), which exchanges data via ROS. The Pixhawk autopilot in the offboard mode will receive the control from Computer 2 and output the PWM control signal for motors. In the simulation environment, AirSim is responsible for receiving various sensor signals from the Pixhawk and forwarding them to IP1:Port. The signal is broadcasted to the port of Computer 2 via "mavproxy" and reaches the IBVS controller in Computer 2 via "mavros". On the other hand, the first-person-view of the drone can be obtained from the simulation environment for the IBVS controller.

Its time derivative can be calculated by

$$\begin{aligned} \dot{L} &= \left(1 + \tfrac{{}^{\mathrm{c}}z^2}{f_{oc}^2}\right)e_x\dot{e}_x + \left(1 + \tfrac{{}^{\mathrm{c}}z^2}{f_{oc}^2}\right)e_y\dot{e}_y + \left(\tfrac{e_x^2+e_y^2}{f_{oc}^2} + 1\right){}^{\mathrm{c}}z\,{}^{\mathrm{c}}\dot{z} \\ &\quad + e_\theta\dot{e}_\theta \\ &= -\lambda_1\left(1 + \tfrac{{}^{\mathrm{c}}z^2}{f_{oc}^2}\right)e_y^2 - \lambda_2\left(1 + \tfrac{{}^{\mathrm{c}}z^2}{f_{oc}^2}\right)e_x^2 - k_3 e_\theta^2 \\ &\quad - \left(\tfrac{e_x^2+e_y^2}{f_{oc}^2} + 1\right){}^{\mathrm{c}}v_z\,{}^{\mathrm{c}}z - \left(k_3\left(1 + e_y^2\right)\left(1 + \tfrac{{}^{\mathrm{c}}z^2}{f_{oc}^2}\right) + k_2\right)e_y e_\theta \\ &\leq 0 \end{aligned} \tag{23}$$

where the interceptor flying to the intruder and keeping the intruder in the field of view ensure that ${}^{\mathrm{c}}v_z > 0$ and ${}^{\mathrm{c}}z > 0$. The design of θ_d according to Equation (17) can guarantee $e_y e_\theta \geq 0$.

Consequently, it can be claimed that each error converges to zero asymptotically. It follows that the interception can be guaranteed.

IV. Simulation and Experiment

A. HIL simulation experiment

In the HIL simulation experiment, the AirSim simulation platform is adopted, and the hardware is a Pixhawk autopilot. The intruder in the simulation scene is set to be a rigid body that can move arbitrarily. We extract the coordinates of the intruder centroid in the image. In HIL mode, the interceptor multicopter in the simulation scene is under the control of the designed visual servo control algorithm. We set up three sets of experiments with intruder speeds at 0, 5m/s, and 10m/s. The parameter setting is shown in Table I, where the attitude angle unit is rad. In the three experiments, the start position of the interceptor and the start position of the intruder does not change. At speeds 5m/s and 10m/s, the two intruders move along vectors $(-2,-1,3)$ and $(1,1,1)$ respectively. The interceptor intercepts at speed up to 15m/s. Three experiments show that the intruders are all intercepted successfully. The stability of the attitude controller with IBVS under high maneuvering is demonstrated.

TABLE I

PARAMETER SETTING OF THE CONTROL ALGORITHM

Parameter	k_1	k_2	k_3	k_4	k_5	k_6	θ_d	ϕ_d
Value	0.02	0.01	3	3	0.01	5	0.3	0

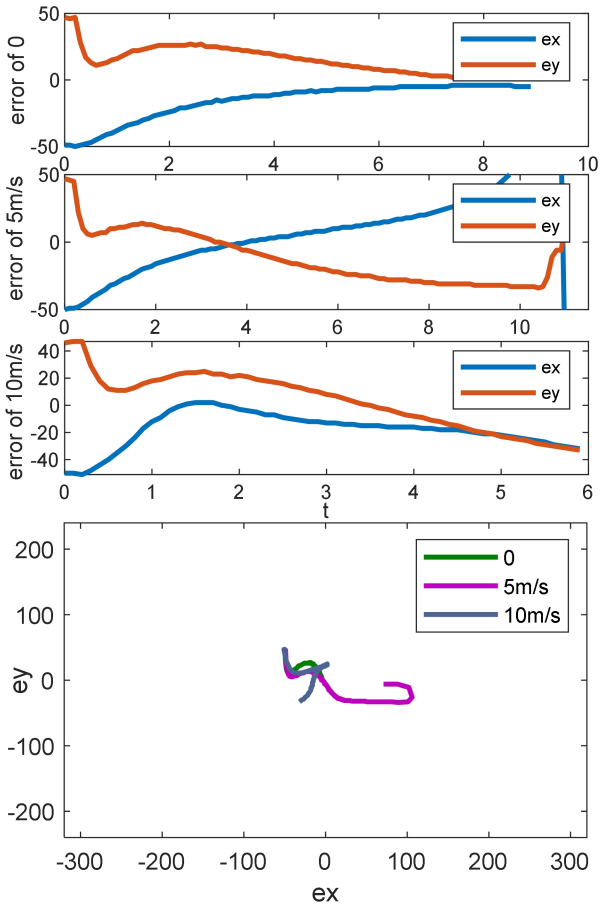

Fig. 4. The image tracking error with time and error walking in image in HIL experiment

That the image tracking error changes with time and that the error walking in the image plane are shown in Fig. 4. The trajectories of the intruder and the interceptor are shown in Fig. 5. It can be observed that the error always tends to converge to zero. The tracking error is confined within the image plane. When the intruder moves with uniform motion in a straight line, the interceptor's trajectory is smooth. As shown in Fig. 6, the simulation experiment is also performed for the multicopter as an intruder.

B. Flight experiment

HIL simulation experiments show that the feasibility of onboard processing. Compared with software-in-the-loop, HIL simulation verifies the correctness of the algorithm on the hardware platform and improves development efficiency. In fact, during the work from the HIL simulation to the real flight test, it only needs to take a few times of debugging work. In the real flight experiment, we used DJI's Tello platform, and the video was sent back to a laptop. The perception and control algorithms run on the laptop with

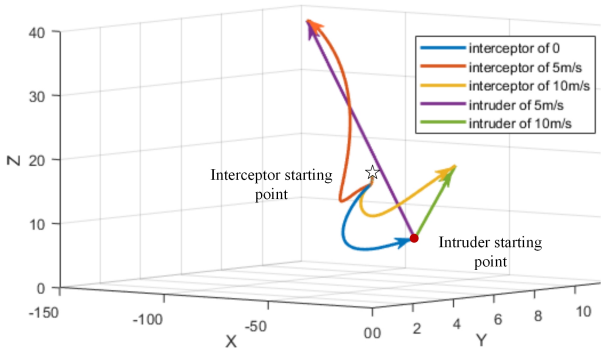

Fig. 5. Trajectories of the intruder and interceptor

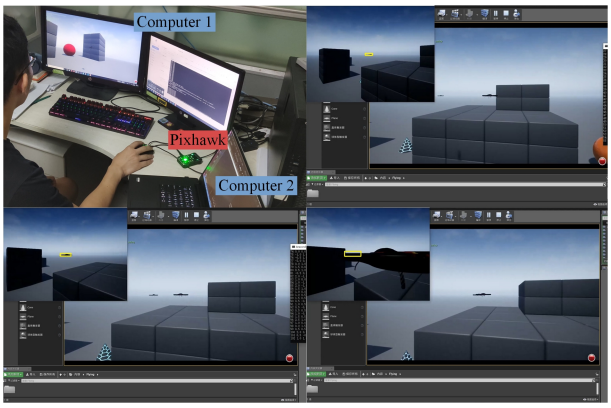

Fig. 6. HIL experiment process. The first picture is the experimental scene of HIL. The second to fourth pictures are the interception process.

an Intel Core i7-7700 (2.80 GHz) processor and a QUAD M1200 (640 CUDA) graphics card. The flow of the system used in the experiment is shown in Fig. 7.

1) Perception network: For fast and accurate detection and visual tracking, we optimized two excellent networks and combined them in tight coupling. We made the detection, Human Machine Interface (HMI), and the tracking part into a pipeline called Drone Perception Network (DroPNet). The structure of DroPNet is shown in Fig. 8. For target detection, we employed the structure of YOLO-v3 [19] as the first part of DroPNet. We built a drone database to retrain DroPNet. The image of the database came from the video stream collected by Tello's forward camera as well as the images from the Internet. The targets in the database were characterized by diverse perspectives, substantial environmental differences, different acquisition equipment, and various drone models.

The second part of DroPNet is the HMI. HMI is mainly used to confirm intercepted objects from candidate targets. This will reduce the error rate and prevents AI from causing uncontrollable damage. In addition to displaying the first-person view of the interceptor in real-time, the HMI also has the function for the operator to select the target and emergency landing interceptor actively.

To track the target in real-time, we introduced DaSiamRPN [20] as a benchmark. DaSiamRPN has shown superior performance in many scenarios. However, redetection after the target being out of the visual field is a difficult

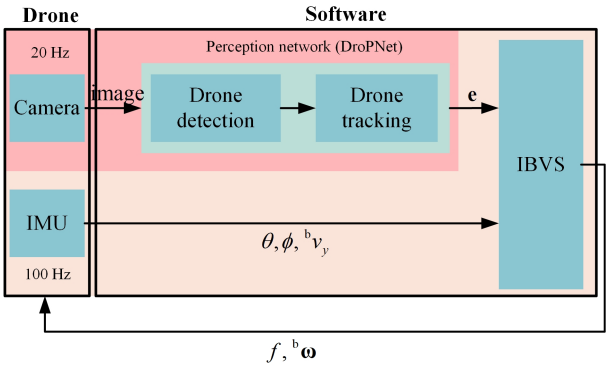

Fig. 7. The relationships among perception network, control algorithm, and hardware platform. The software receives the drone state acquired by IMU and image and gives the command after processing. The perception network works at 20 Hz and the controller works at 100 Hz. Two processes asynchronously update shared memory to implement interprocess communication.

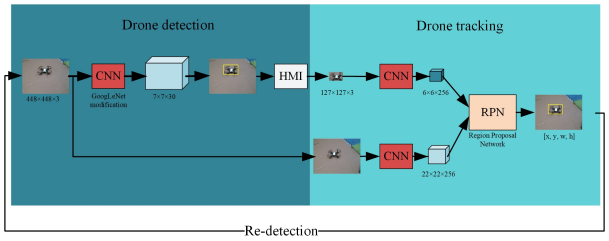

Fig. 8. The structure of DroPNet. The input of DroPNet is a video stream and the output is the bounding box of the target.

problem for target tracking, and DaSiamRPN is no exception. We introduce the forward channel to form a tight coupling with the first two parts of the DroPNet. When the target is lost, expand the search range of the previous step of the tracking process, and use the detection method of the first part of DroPNet to regain the intruder's position.

To train DroPNet, we collected and labeled 5,000 images. The data set was divided into a training set, a valid set, and a test set according to 4000:500:500. After 2000 iterations, the average IOU of the test set was 75.23%. For the 960×720 resolution video stream collected by Tello's forward-looking camera, DroPNet's processing speed is 21.3fps.

2) Interception experiment: We did three sets of experiments with intruder speeds of 0, 1m/s, and 2m/s. The experimental parameters were slightly modified on the basis of Table I. The image tracking error changed with time is shown in Fig. 10. At the beginning of the experiment, the intruder was 20 meters away and the interceptor takes off immediately. After confirming the intruder, the interceptor approached the intruder at 5m/s and successfully intercepted it. It can be seen that when the intruder made an unexcepted action, image tracking error increased. Otherwise, the image tracking error always converges to zero. The image tracking error moves near the center of the image.

V. CONCLUSION

In this paper, an interception algorithm based on IBVS is designed. The system can track an intruder steadily and

Fig. 9. Flight experiment process. The scenes of the three moments of the interception process were captured. The above three pictures are scenes taken from the ground, and the following three pictures are the first-person view of the interceptor with the bounding box given by target tracking.

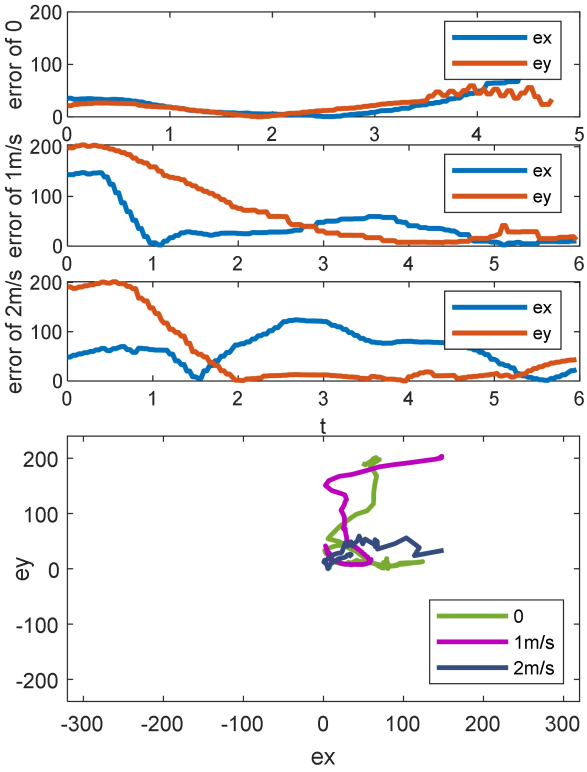

Fig. 10. Image tracking error with respect to time and error walking in the image plane in a flight experiment

implement the interception rapidly. The application of multicopter to intercept intruder provides an airborne solution for anti-drone systems. This method relies on the resolution of the camera and is difficult to detect for objects too far away. In future work, we plan to improve perception capabilities. When the intruder enters the visual sensor network, the surveillance system can obtain its approximate region. In response to the situation of multiple intruders at the same time, a distributed interception strategy will also be taken into consideration.

REFERENCES

[1] V. Roberge, M. Tarbouchi, and G. Labonté, "Fast genetic algorithm path planner for fixed-wing military uav using gpu," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 54, no. 5, pp. 2105–2117, 2018.

[2] S. J. Fox, "Policing: Monitoring, investigating and prosecuting 'drones'," *European Journal of Comparative Law and Governance*, vol. 6, no. 1, pp. 78–126, 2019.

[3] L. Bond, "Overview of gps interference issues," in *GPS Interference Symposium-Volpe National Transportation Systems Center, Boston*, 1998.
[4] A. J. Barry and R. Tedrake, "Pushbroom stereo for high-speed navigation in cluttered environments," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 3046–3052.
[5] G. D. Cubber, "Explosive drones: How to deal with this new threat?" Apr. 2019. [Online]. Available: https://doi.org/10.5281/zenodo.2628752
[6] H. Duan, A. Yang, J. Qu, and J. Liu, "Application of laser suppression weapon and opto-electronic tracking integrative technique in air defense," *Journal of Applied Optics*, vol. 1, 2010.
[7] V. Coffey, "High-energy lasers: new advances in defense applications," *Optics and Photonics News*, vol. 25, no. 10, pp. 28–35, 2014.
[8] P. Basset, A. Tremolet, F. Cuzieux, G. Reboul, M. Costes, D. Tristrant, and D. Petot, "Creation the onera multi-level rotorcraft concepts evaluation tool: the foundations," in *Future Vertical Lift Aircraft Design Conference*, vol. 31, 2012, p. 32.
[9] M. R. Aagaah, E. M. Ficanha, and N. Mahmoudian, "Drone having drone-catching feature," May 25 2017, uS Patent App. 15/332,170.
[10] T. H. Chung, G. A. Hollinger, and V. Isler, "Search and pursuit-evasion in mobile robotics," *Autonomous robots*, vol. 31, no. 4, p. 299, 2011.
[11] K. Hashimoto, T. Kimoto, T. Ebine, and H. Kimura, "Manipulator control with image-based visual servo," in *Proceedings. 1991 IEEE International Conference on Robotics and Automation*. IEEE, 1991, pp. 2267–2271.
[12] P. Martinet, J. Gallice, and D. Khadraoui, "Vision based control law using 3d visual features," in *Proceedings World Automation Congress*, vol. 96. Citeseer, 1996, pp. 497–502.
[13] L. Mejias, S. Saripalli, P. Campoy, and G. S. Sukhatme, "Visual servoing of an autonomous helicopter in urban areas using feature tracking," *Journal of Field Robotics*, vol. 23, no. 3-4, pp. 185–199, 2006.
[14] T. Hamel and R. Mahony, "Visual servoing of an under-actuated dynamic rigid-body system: an image-based approach," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 2, pp. 187–198, 2002.
[15] E. Frew, T. McGee, Z. Kim, X. Xiao, S. Jackson, M. Morimoto, S. Rathinam, J. Padial, and R. Sengupta, "Vision-based road-following using a small autonomous aircraft," in *2004 IEEE Aerospace Conference Proceedings (IEEE Cat. No. 04TH8720)*, vol. 5. IEEE, 2004, pp. 3006–3015.
[16] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "Airsim: High-fidelity visual and physical simulation for autonomous vehicles," in *Field and Service Robotics*, 2017. [Online]. Available: https://arxiv.org/abs/1705.05065
[17] Q. Quan, *Introduction to multicopter design and control*. Springer, 2017.
[18] G. Hu, N. Gans, and W. E. Dixon, "Adaptive visual servo control," *Encyclopedia of Complexity and Systems Science*, pp. 42–63, 2009.
[19] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.
[20] Z. Zhu, Q. Wang, B. Li, W. Wu, J. Yan, and W. Hu, "Distractor-aware siamese networks for visual object tracking," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 101–117.