

Regular Article

Autonomous Navigation and Control System for Capturing A Moving Drone

Hong Tao[✉], Tao Song[✉], Defu Lin[✉], Ren Jin[✉] and Bin Li

Beijing Key Laboratory of UAV Autonomous Control, Beijing Institute of Technology, Beijing, China

Abstract: This paper describes an autonomous navigation and control system for capturing the maneuvering drones. A vision-based navigation method seeks and detects the intruding drone, then, the target trajectory is predicted by fusing onboard vision and inertial-measurement resources. The target's relative position, velocity and acceleration are also obtained at the same time. Then, we present a modified proportional-derivative (PD) algorithm based on the estimated target states. In addition, the boundary constraints of the protected area are considered to avoid a collision. The proposed capture navigation and control system has demonstrated its efficiency both in simulation, flight experiments, and MBZIRC 2020, where our team won the Challenge I competition.

Keywords: Drone capture, trajectory prediction, extended Kalman filter, PD controller, visual detection.

1. Introduction

In recent years, small drones have been widely used due to their convenience and cost-effectiveness. Some have been used for illegal surveillance and malicious actions, which threaten the economic development and public safety (Rogers, 2019). Thus, many researchers are looking for an efficient way to counter invading drones and ensure the safety in urban environment. The method of shooting down the target one by one with a long spear has low accuracy. Attacking target with a missile or laser weapons inevitably produces aerial falling objects and collateral damage to the ground environment. Besides, the weapons are more expensive than the small drones (Brust et al., 2017). Thus, this paper focuses on utilizing the multicopter to accurately capture and carry the target back home. This autonomous system is cost-effective due to its recyclability, and it also can avoid undesirable side-effects.

Capturing small drones requires highly-accurate guidance and control systems. Target-interception and guidance algorithms have been extensively studied and can be applied to the control phase of drone capture. Among them, proportional navigation (PN) algorithm has been widely used for interception of low-speed targets due to its optimality (Zarchan, 2012; Bing et al., 2016), but the PN algorithm cannot handle situations where the target drone is faster than the pursuer (Becker, 1990). Yoon et.al. (Yoon et al., 2008) proposed a two-point, pursuit-guidance

Received: 30 September 2020; revised: 07 May 2021; accepted: 26 April 2021; published: 08 March 2022.

Correspondence: Tao Song, Beijing Key Laboratory of UAV Autonomous Control, Beijing Institute of Technology, Beijing, China, Email: songtao_BIT@126.com

This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Copyright © 2022 Tao, Song, Lin, Jin and Li

method and an adaptive, backstepping controller for the vision-based net-recovery, which approach significantly complicates the control system. However, a trajectory-tracking method can be used to capture the moving drone if the target trajectory is obtained (Yamasaki et al., 2007). For this reason, we have focused on developing an accurate, real-time, trajectory-predicting method. With this design, we can then augment simple PD tracking algorithm with estimated target states to achieve successful interception.

Trajectory-prediction algorithms mainly consist of modeling and data-based methods. The first method must build a target-motion or relative-motion model. Gambs (Gambs et al., 2012) developed the Mobility Markov Chain (MMC) algorithm for the next location prediction, and the accuracy of the method is in the range of 70 to 95 percent. However, the MMC has a high complexity and complicated calculation process. The data-based method has developed rapidly using the Machine Learning approach in these years. The authors of (Yang et al., 2017) proposed a trajectory-prediction model based on bidirectional, long, short-term memory (B-i LSTM) neural network. This approach requires the target pose and relative position as the prediction input, but the pose information for uncooperative target is hard to obtain. In Xu Ximeng's work (Ximeng et al., 2019), the parameters of the output-input feedback Elman network are optimized by the genetic algorithm and then the GA-OIF-Elman neural network is established to predict the target trajectory. Although this method does not depend on the target motion model, it requires a high computing power of the airborne platform.

In this paper, we present a multi-source prediction method that utilizes the information of onboard machine vision and IMU. In order to obtain the relative motion information of the target and the pursuer, a two-axis electro-optical pod (EOP) utilized and its output is fused into the prediction system via an Extended Kalman Filter (EKF) (Zarchan and Musoff, 2005). The EKF process can provide the velocity and acceleration of the target, which are then compensated into the PD algorithm. Our modified PD approach relaxes the maneuver requirements on the pursuer drone, thus enabling successful intercepts even when the intruder performs irregular maneuver patterns. Finally, since we have prior details of the competition arena, we add appropriate boundary constraints to our control rules to avoid hitting walls or other obstacles.

The contribution of this paper was applied and verified in the competition of the Mohamed Bin Zayed International Robotics Challenge (MBZIRC) 2020. In Challenge I, a small, detachable ball is suspended under a drone, which flies around the competition arena in a figure-eight pattern of randomly varying altitude. The Challenge I is to capture the target ball and carrying it back to the landing point. Contact and collision with the target drone are not allowed during the capturing process. This mission is obviously more difficult than intercepting a drone because the ball is only 13cm in diameter and swings in a pendulum fashion beneath the target drone. Despite numerous difficulties, we successfully completed the ball-capture task with full marks and won the championship of the challenge.

The rest of this paper is arranged as follows. The first section introduces the problem formulation and the hardware components of our system. Software framework and state machine are also presented. The target search and detection algorithm are discussed in Section 3, which is followed by the trajectory-prediction method. Section 5 explains the control strategy for autonomously capturing the target ball. The boundary constraint is also considered in this section. Section 6 provides the results of the conducted simulations and flight experiments and is followed by conclusions about our design, methods, and system performance. Lessons learned and future research are presented in the last section.

2. Preliminaries

In the scenario, a target drone is assumed to invade the arena whose size is approximately $100 \times 40m$. The target drone hangs a 13cm ball and moves along a random 8-shape trajectory. The schematic diagram of the arena and flight trajectory of the target drone is presented in Figure 1a. Figure 1b shows the target drone and the hanging ball.

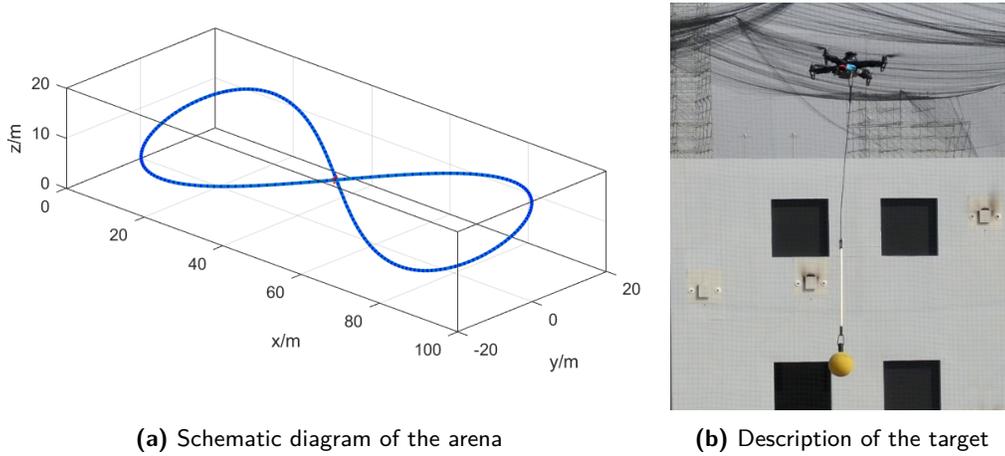


Figure 1. Description of the competition in MBZIRC Challenge 1.

Before studying control algorithms, the relative motion analysis between the target and pursuer is necessary. The hardware components of our pursuer multicopter are introduced in detail. The overview of the proposed framework is finally offered.

2.1. Motion geometry description

In order to establish the three-dimensional relative motion of the target and the pursuer, the relevant coordinate systems are introduced as follows.

The earth-fixed coordinate system ($O_e X_e Y_e Z_e$). The coordinate origin O_e of the $O_e X_e Y_e Z_e$ system is chosen as the initial position of the multicopter, which is usually determined at power-up. The $O_e X_e$ axis points to the direction of geographic east, and the $O_e Y_e$ points to the direction of north. The $O_e Z_e$ is determined according to the right-hand rule.

The multicopter-body coordinate system ($O_b X_b Y_b Z_b$). The $O_b X_b Y_b Z_b$ system is fixed to the multicopter and takes the pursuer's center of gravity (CoG) as the origin. For the X-configuration multicopter, the $O_b X_b$ axis points to the nose direction in the symmetric plane of the multicopter. The $O_b Z_b$ axis is in the symmetric plane of the multicopter, perpendicular to the $O_b X_b$ and pointing upward. The $O_b Y_b$ axis is determined according to the right-hand rule. The rotation relationship between the $O_e X_e Y_e Z_e$ and the $O_b X_b Y_b Z_b$ is expressed by Euler angles (γ, θ, ψ) .

The light-of-sight (LOS) coordinate system ($O_L X_L Y_L Z_L$). The LOS coordinate ($O_L X_L Y_L Z_L$) is introduced to describe the relative motion between two vehicles, whose origin coincides with the pursuer's center of gravity. The $O_L X_L$ axis points to the target and is consistent with the line of sight, $O_L Z_L$ lies in the vertical plane which contains the $O_L X_L$ and is perpendicular to $O_L X_L$, pointing upward. The Y_L axis is determined according to the right-hand rule. The pitch and yaw LOS angle q_y and q_z denote the rotation transformation from $O_e X_e Y_e Z_e$ to $O_L X_L Y_L Z_L$.

The three-dimensional relative motion model between target and pursuer is illustrated in Figure 2. The position vector of target and multicopter referred to $O_e X_e Y_e Z_e$ frame are defined as P_t and P_m , respectively; V_t and a_t are the velocity vector and acceleration vector of target, while V_m and a_m represent multicopter's velocity and acceleration. The relative motion vectors of the target and the

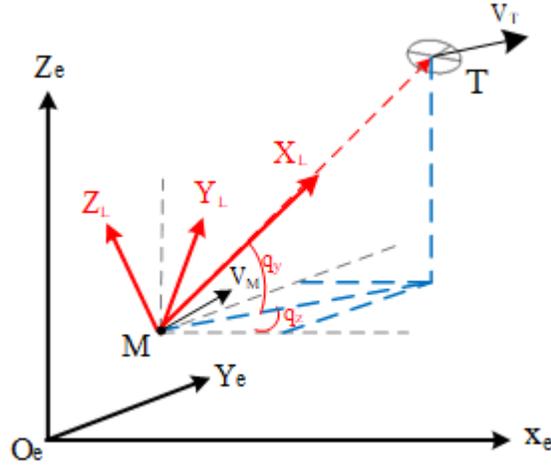


Figure 2. The engagement geometry between the target and the pursuer multicopter.

pursuer in $OX_eY_eZ_e$ system is given

$$\begin{aligned}\Delta P &= P_t - P_m \\ \Delta V &= V_t - V_m \\ \Delta a &= a_t - a_m\end{aligned}\quad (1)$$

Subsequently, the LOS angles can be formulated using the relative position of the target and the pursuer multicopter.

$$\begin{aligned}q_y &= \arctan\left(\frac{-\Delta P_z}{\sqrt{\Delta P_x^2 + \Delta P_y^2}}\right); \quad q_y \in \left(-\frac{\pi}{2}, \frac{\pi}{2}\right) \\ q_z &= \arctan\left(\frac{\Delta P_y}{\Delta P_x}\right); \quad q_z \in (-\pi, \pi)\end{aligned}\quad (2)$$

2.2. Hardware components

The hardware system was designed according to the mission, the limitation for mission time and the multicopter's size in the challenge. The main hardware components and their relationship were shown in Figure 3. In order to achieve the searching task, we chose a two-axis electro-optical pod (EOP) to detect and track the target drone. A 1080p Sony camera is mounted on the gimbal and then integrated into the EOP. The gimbal can rotate 360 degrees, which can help to search targets omnidirectionally. In addition, the gimbal is able to stabilize the camera's image collection for the visual detection. The light-of-sight motion information between the target and the pursuer is also provided by the EOP, and this is helpful for the target trajectory prediction.

An onboard computer, NVIDIA AGX Xavier, was used for image processing and task management. This small module can provide 32 Tops of computing power with a power as low as 10 watts, which is suitable for training and deployment of neural network models. The control nodes for EOP and Flight Control Units (FCU) of the multicopter are also computed in NVIDIA module. We chose the DJI Matrice 210 (M-210) as our flight platform considering the load requirement, in which FCU and Inertial Measurement Units (IMU) are integrated. Mention that our software control algorithms are universally used for other FCU, such as Pixhawk.

The capture mechanism was designed and manufactured by our team, which is made by fishing nets and carbon fiber tubes. The horizontal capture range of the capture mechanism is 0.8 meters.

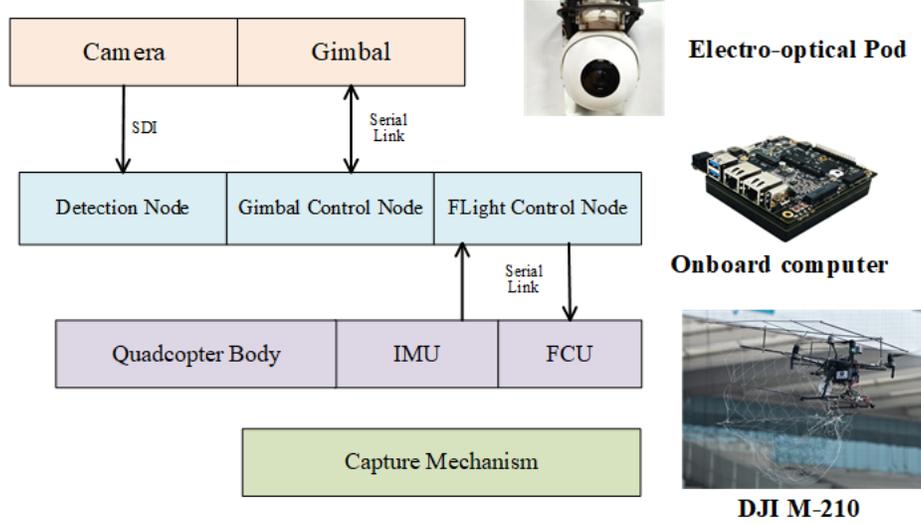


Figure 3. Components description of our hardware platform.

It can not only release the control accuracy of the control algorithm, but also carry the target back to the required drop zone.

2.3. Software framework

This paper aims at introducing an autonomous navigation and control system for capturing the maneuvering target. The capture state machine (Figure 4) starts form autonomously takeoff and searching for the target. After the target is stably detected, the pursuer multicopter will implement capture control and intercept the target. The related core software algorithms are target search and detection, target trajectory prediction and capture control, which will be described in the following sections.

3. Target Search and detection

3.1. Searching algorithm

The search algorithm is necessary because the camera has a limited field of view (FOV) and the target drone moves randomly in the $100 \times 40m$ arena. In this case, we designed a search method for the EOP, which can expand the detection field of view by rotating the camera. For the square field of view, the search area of the square trajectory is larger than that of the circular trajectory. Thus, we chose the square trajectory for the optical axis. Figure 5a presents area scanned by the camera's FOV following the movement of the optical axis. L, W indicate the length and width of the square while the subscript t, f, s denote the square of the search trajectory, camera's FOV and scanned area, respectively. The movement trajectory is designed according to the range of target movement in vertical and horizontal direction. Given the estimated maximum movement range δ_{tx} of the target and the minimum distance D_{min} , the relationship between δ_{tx} and scanned length L_s is illustrated in Figure 5b. We can get

$$L_s = \frac{\delta_{tx}f}{D_{min}} = L_t + L_f \quad (3)$$

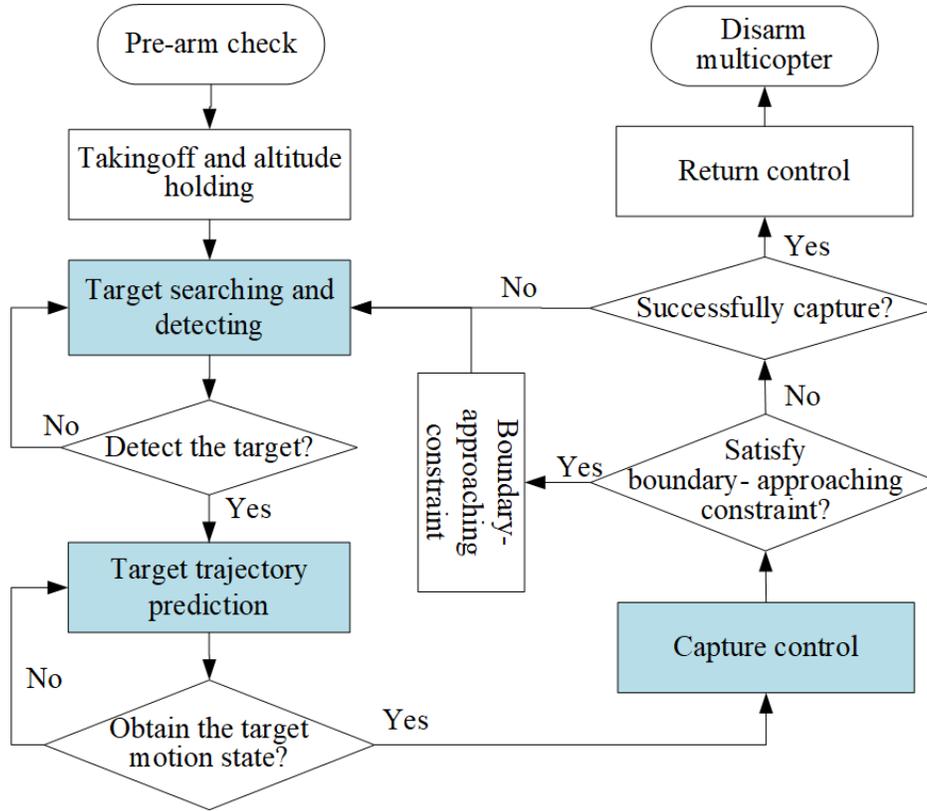


Figure 4. The flow diagram of the capture state machine.

Where f denotes the focal length. The FOV of the utilized camera is $67 \text{ deg} * 37.6 \text{ deg}$. Subsequently, we can derive the desired horizontal movement angle of optical axis

$$\alpha_c = \tan \frac{L_t}{f} = \tan \frac{\delta_{tx} f - L_f D_{min}}{f D_{min}} \quad (4)$$

In the same way, we can get the desired vertical movement angle β_c . The angle commands (α_c, β_c) are implemented in the EOP gimbal. To ensure that the target can appear in the FOV no matter which direction the target is flying from, the search period of the EOP can be designed to be one-half of the target motion period.

3.2. Object detection

Since the intruding drone was uncooperative and moving quickly, we used the visual perception algorithm to detect the target and provide the navigation information. The target drone and the small ball hanging below the drone can form an associated system. Aiming at this highly dynamic, composite target, our team proposed a single-stage rapid collaborative target detection algorithm between the parent vehicle and the depending object.

Target annotation can increase the learning of the detection network's overall representation ability of the target, thereby improving the final detection accuracy. Thus, we collected and labeled thousands of target images to augment the dataset. We made an equal scale model according to the rules, and then collected the data around the suburban flight test site and labeled the collected images manually, including the ball, the drone and the entire target. The annotations of the image is shown in Figure 6. Detecting the entire target at the long distance can greatly increase the detection

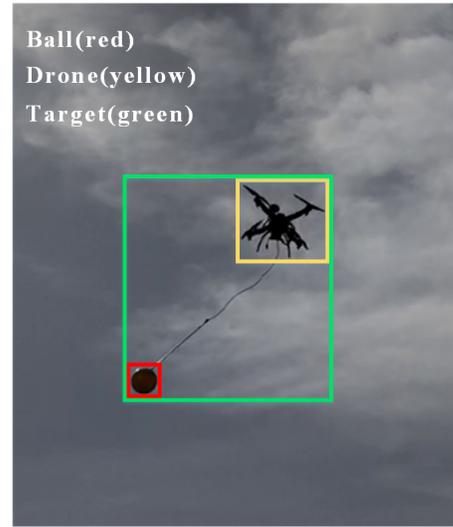
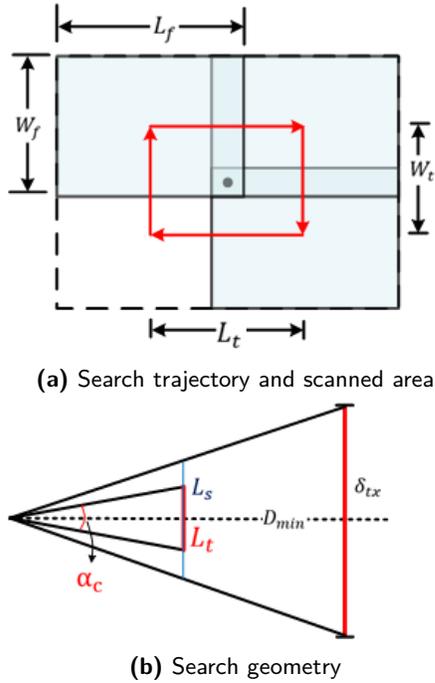


Figure 6. Annotations of the target image.

Figure 5. Working sketch of EOP search field.

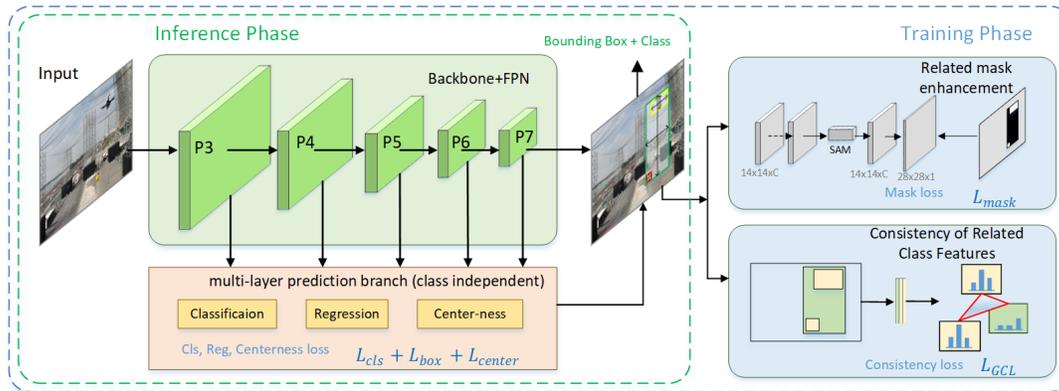


Figure 7. Illustration of the network architecture for target detection.

distance, while only detecting the hanging ball will improve the detection accuracy when the target is close to the pursuer.

As for the network model, our team made three improvement for the related category detection based on the Fully Convolutional One-Stage Object Detection (FCOS) algorithm(Law and Deng, 2018). Firstly, the parallel independent prediction branch is introduced to solve the conflict between the target classification and the bounding box regression. Then, we utilized the correlation class mask enhancement(He et al., 2017; Fu et al., 2019) to help the detector learn the target feature better and reduce the interference of background features in the rectangular box of the target. Finally, the correlation class feature relation constraint, which use the correlation of different class features, is applied to improve detection accuracy. The illustration of the network architecture used in this paper is shown in Figure 7.

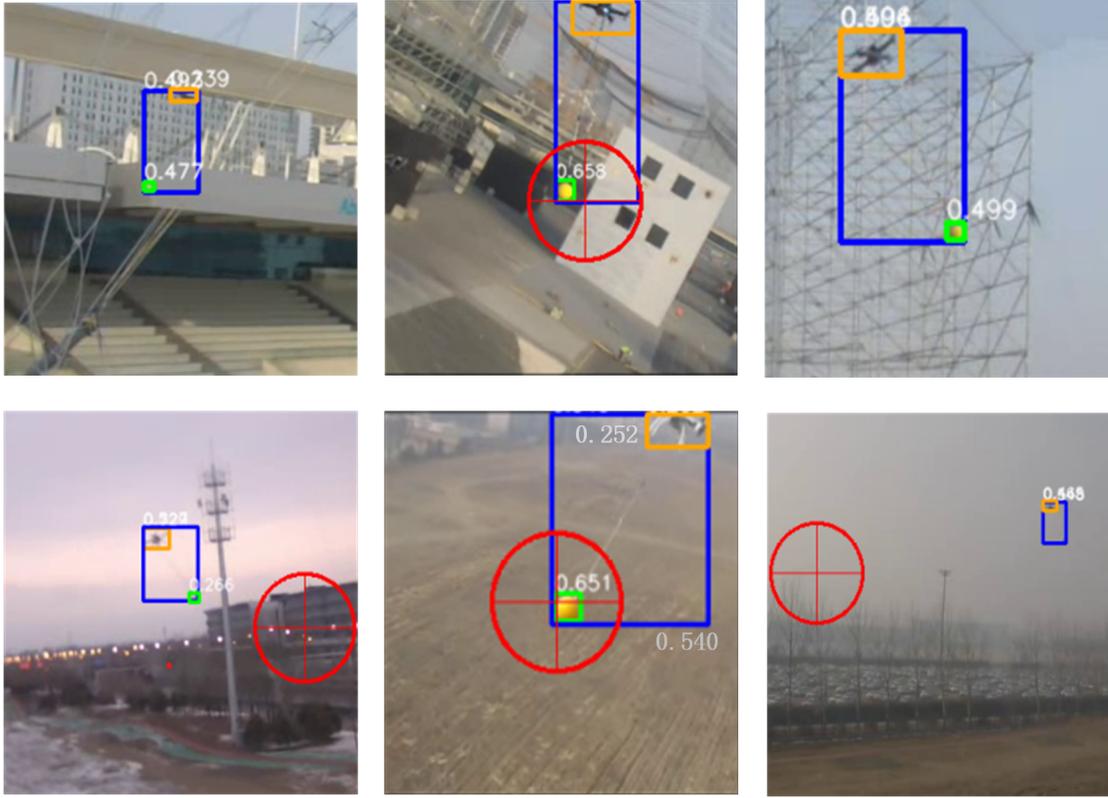


Figure 8. The target detection results in different scenes.

This object detection algorithm can achieve real-time processing (25FPS) on the onboard computer (NVIDIA AGX Xavier), which is helpful for tracking the maneuvering target. Some representative detection results are shown in Figure 8

The pixel deviations (u, v) in the pixel-scale coordinates are obtained after the target is detected. Subsequently, the angle deviations between the target and the optical axis can be derived using the pixel errors and FOV of the camera. Inputting the angle deviations to the control algorithm of the EOP gimbal, the camera in EOP can lock the target in FOV quickly and stably. Given that the size of both the target drone and the ball are known, we can also derive the distance between the target and the pursuer multicopter using the camera focal length f and the target size in the image plane.

$$D = \frac{S_x^w f}{S_x^{im}} = \frac{S_y^w f}{S_y^{im}} \quad (5)$$

Where the subscript x, y denote the length and height of the target. The superscript w and im indicate the world coordinates and the image plane coordinates, respectively.

4. Target trajectory prediction

Kalman filter approach can optimally estimate the system state through state prediction and update process, thus it is especially suitable for real-time state prediction of high-dynamic moving targets. Considering the non-linearity of the prediction system, we adopt the extended Kalman filter to estimate the target motion state (Daum, 2005).

4.1. Prediction principle of EKF

The extended Kalman filter is developed basing on the Kalman filter (Park and Lee, 2001). The first step is to linearize nonlinear systems, including the state function $f(\bullet)$ and measurement function $h(\bullet)$ in Equation 6. Taylor expansion and small item hypothesis method are usually adopted for linearization. Then we utilize the basic process of Kalman filter to perform recursive calculations and prediction.

$$\begin{aligned} x_{k+1} &= f(x_k, u_k, w_k) \\ z_k &= h(x_k, v_k) \end{aligned} \quad (6)$$

where k is the discrete time. x_k , u_k and z_k are the state vector, control vector and measurement vector of the dynamic system, respectively. Both $f[\bullet]$ and $h[\bullet]$ are the differentiable vector function. w_k is the process noise with zero mean, namely $E[w_k] = 0$. The variance matrix of w_k is $Q = E[w_k w_k^T]$. v_k is the measurement noise with $E[v_k] = 0$, and its variance matrix is $R = E[v_k v_k^T]$. It is assumed that w_k and v_k are uncorrelated white noise, thus we can also get $E[w_j v_k^T] = 0$. The initial state $x(0)$ is a random variable and uncorrelated to w_k and v_k ,

$$\begin{aligned} E[x_0] &= \hat{x}_0 \\ E[(x_0 - \hat{x}_0)(x_0 - \hat{x}_0)^T] &= P_0 \end{aligned} \quad (7)$$

The linearization approach is shown in Equation 8.

$$\begin{aligned} F(k-1) &= \left. \frac{\partial f}{\partial x^T} \right|_{x=\hat{x}(k-1)} \\ H(k) &= \left. \frac{\partial h}{\partial x^T} \right|_{x=\hat{x}(k)} \end{aligned} \quad (8)$$

Then the discretized state transition matrix can be approximated as

$$\Phi_k(\hat{x}) \approx \mathbf{I} + F(\hat{x})T_s \quad (9)$$

The discrete process noise variance matrix Q_k and measurement noise variance matrix R_k can be derived by

$$\begin{aligned} Q_k &= \int_0^{T_s} \Phi(\tau) Q \Phi^T(\tau) dt \\ R_k &= R \end{aligned} \quad (10)$$

After the linearized system is obtained, the filtering steps of the standard discrete Kalman filter can be implemented with the following equations:

State estimate process:

$$\begin{aligned} \hat{x}_{k+1}^- &= f(\hat{x}_k, u_k, 0) \\ P_k^- &= \Phi_k P_{k-1} \Phi_k^T + Q_{k-1} \end{aligned} \quad (11)$$

The correction of estimate process:

$$\begin{aligned} K_k &= P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} \\ \hat{x}_k &= \hat{x}_k^- + K_k (z_k - h(\hat{x}_k^-, 0)) \\ P_k &= (I - K_k H_k) P_k^- \end{aligned} \quad (12)$$

4.2. Estimation model

To estimate the motion state of the target, the state vector for EKF is selected as

$$X = [\Delta P_x, \Delta P_y, \Delta P_z, V_{tx}, V_{ty}, V_{tz}, a_{tx}, a_{ty}, a_{tz}, V_{mx}, V_{my}, V_{mz}] \quad (13)$$

Where, the relative position ΔP_t , the target's velocity V_t , the target's acceleration a_t and the pursuer's velocity V_m are all defined in the earth-fixed coordinate system. Here we introduce the

three-dimensional the pursuer's velocity instead of the acceleration to assist in estimating the target state, which is because the measurement accuracy of the multicopter is higher while the signal-noise ratio (SNR) of the measured acceleration is low.

In this competition scenario, we know that the target moves following the 8-shape trajectory. Thus, the target motion model can be predicted as sinusoidal maneuver. The target motion model is shown in the following equation

$$\begin{cases} P_t = P_{t0} + A \sin(\omega t) \\ \dot{P}_t = A\omega \cos(\omega t) \\ \ddot{P}_t = -A\omega^2 \sin(\omega t) \\ \dddot{P}_t = -A\omega^3 \cos(\omega t) = -\omega^2 \dot{X}_t \end{cases} \quad (14)$$

where $P_t = (P_{tx}, P_{ty}, P_{tz})$, which means the target moves sinusoidally in three-dimensional space. ω denotes the target maneuver frequency, which can be roughly estimated from the given target's velocity and the size of the flight arena.

Thus, we can get the discrete state equation

$$\dot{X} = \begin{bmatrix} 0_{3 \times 3} & I_{3 \times 3} & 0_{3 \times 3} & -I_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & I_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & -\omega_{t3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \end{bmatrix} X + \begin{bmatrix} 0_{3 \times 1} \\ 0_{3 \times 1} \\ w_{at}(t) \\ w_{vm}(t) \end{bmatrix} \quad (15)$$

where, $w_{at}(t)$ and $w_{vm}(t)$ are the process noises. $\omega_{t3 \times 3}$ can be decomposed into

$$\omega_{t3 \times 3} = \begin{bmatrix} \omega_{tx} & 0 & 0 \\ 0 & \omega_{ty} & 0 \\ 0 & 0 & \omega_{tz} \end{bmatrix} \quad (16)$$

4.3. Measurement model

With the hardware sensors mentioned in subsection 2.2, we can extract several information to characterizes the relative motion. In this paper, we make good use of the LOS information provided by the EOP and the pursuer's velocity measured by the onboard IMU. The measurement states for this prediction method are chosen as

$$Z = [D, q_y, q_z, \dot{q}_y, \dot{q}_z, V_{mx}, V_{my}, V_{mz}] \quad (17)$$

Where, q_y, q_z represent the rotation angles in the pitch and yaw direction between inertial and LOS coordinates, respectively. \dot{q}_y, \dot{q}_z are the corresponding LOS angular rates in LOS coordinates. (V_{mx}, V_{my}, V_{mz}) is the inertial velocity vector of the pursuer. The measurement accuracy of these states will be described below.

Distance measurement. The distance D from the target to the pursuer can be obtained by using the onboard visual detection algorithm, Equation 5. In this ball capture task, we chose to calculate the distance from the pursuer to the target ball instead of the target drone. Using the object detection method proposed in this paper, the camera can detect the targets within 50 meters. The update frequency of the distance measurement is 25Hz. If the target is lost during the tracking mode, the distance value will keep the value at the last moment. Taking the RTK information as the true value, the relative error of the measured distance is 10% and the measured accuracy of D was 1 meter.

LOS motion measurement. The EOP can track the target in real time and measure the angle and angular rate between the target and the camera optical axis. Thus, the raw measurements provided by EOP are frame angles and angular rates in EOP's internal frame of reference. Coordinate

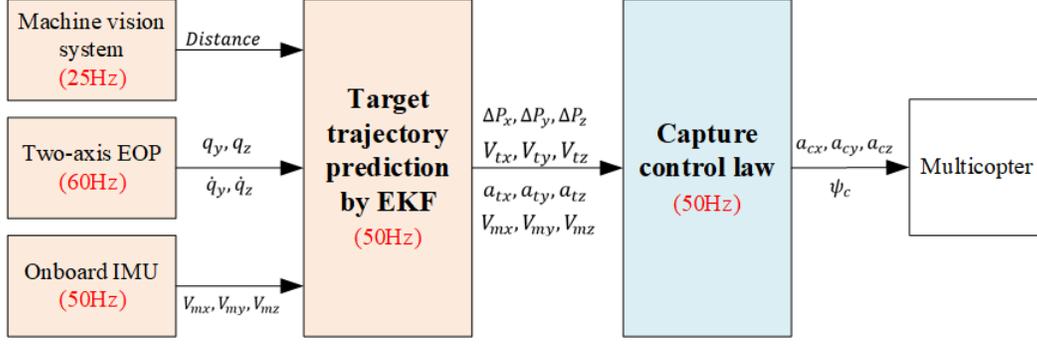


Figure 9. The architecture of the target prediction method.

system transformation is necessary since we need the LOS angle q_y, q_z and LOS angular rate \dot{q}_y, \dot{q}_z . The conversion for EOP information is very basic, so we won't go into details here. A three-axis turntable (Huckridge et al., 2016) was utilized to calibrate the measured accuracy of the LOS motion information, which are 5 deg and $0.6rad/s$ respectively for LOS angle and angular rate. The update frequency of the EOP output is $60Hz$.

Velocity measurement. The onboard Inertial measurement unit (IMU) integrate angular rate gyroscope, accelerator and barometer. By fusing the measured value of these sensors and the longitude and latitude information of GPS with EKF method in the FCU module, the velocity of the multicopter can be obtained. The onboard inertial measurement units are able to provide the velocity information of the pursuer. The accuracy is $0.5m/s$ and the update frequency is $50Hz$. The measurements noises are considered as Gaussian white noises after statistical analysis.

Combining the above measurement equations, we can get the measurement model for the EKF process.

$$h(x) = \begin{bmatrix} D \\ q_y \\ q_z \\ \dot{q}_y \\ \dot{q}_z \\ V_{mx} \\ V_{my} \\ V_{mz} \end{bmatrix} = \begin{bmatrix} \sqrt{\Delta P_x^2 + \Delta P_y^2 + \Delta P_z^2} \\ \arctan\left(\frac{-\Delta P_z}{\sqrt{\Delta P_x^2 + \Delta P_y^2}}\right) \\ \arctan\left(\frac{\Delta P_y}{\Delta P_x}\right) \\ -\frac{\Delta v_z(\Delta P_x^2 + \Delta P_y^2) - \Delta P_z(\Delta P_x \Delta v_x + \Delta P_y \Delta v_y)}{(\Delta P_x^2 + \Delta P_y^2 + \Delta P_z^2)\sqrt{\Delta P_x^2 + \Delta P_y^2}} \\ \frac{\Delta P_x \Delta v_y - \Delta P_y \Delta v_x}{\sqrt{\Delta P_x^2 + \Delta P_y^2}\sqrt{\Delta P_x^2 + \Delta P_y^2 + \Delta P_z^2}} \\ V_{mx} \\ V_{my} \\ V_{mz} \end{bmatrix} \quad (18)$$

Utilizing Equation 8, 11 and 12, the relative position, target velocity and acceleration are estimated. Moreover, we can also obtain the target position in $O_e X_e Y_e Z_e$ coordinate since the pursuer's position is provided by the onboard IMU. As a result, the flight trajectory is predicted. The information flow chart for the target trajectory prediction method is shown in Figure 9.

5. Control strategy

The take-off and return control can easily use a simple position control, whose application is very mature. Thus, we focus on describing the control algorithm for the drone capture using the estimated information in this section.

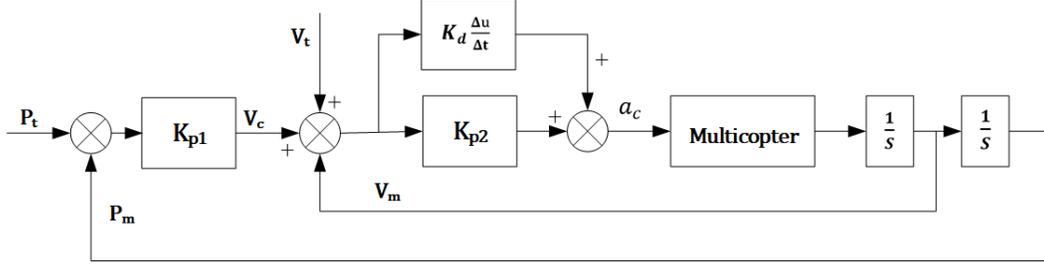


Figure 10. The architecture of the capture control model.

5.1. Capture control algorithm

The proposed trajectory prediction method in Section 4 has provided extremely beneficial information for the control system, including the relative position ΔP , velocity V_t , acceleration a_t of the target. We adopted a simple capture control algorithm basing on the traditional Proportional-Derivative (PD) method of relative velocity (Figure 10). The PD control formula can be expressed as

$$a_c = \left(k_{p2} + k_d \frac{\Delta}{\Delta t} \right) (V_d + V_c - V_m) \quad (19)$$

Where $V_c = k_{p1}(P_t - P_m)$. Substituting V_c into Equation 19, we can get

$$\begin{aligned} a_c &= k_{p1}k_{p2}(P_t - P_m) + (k_{p2} + k_{p1}k_d)(V_t - V_m) + k_d(a_t - a_m) \\ &= k_p(P_t - P_m) + k_v(V_t - V_m) + k_d(a_t - a_m) \end{aligned} \quad (20)$$

Mention that most of the onboard accelerator of the multicopter are noisy and inaccurate, so we only compensate the estimated a_t instead of $a_t - a_m$. As a result, the capture control algorithm yields

$$a_c = k_p(P_t - P_m) + k_v(V_t - V_m) + k_a a_t \quad (21)$$

Where k_p, k_v, k_a are the control gains for position, velocity and acceleration item, respectively. Many methods have been proposed to tune the gains of PID controllers, such as frequency loop-shaping method (Grassi and Tsakalis, 2015), adaptive-tuning method (Achour et al., 2015) and optimal fuzzy method with stochastic algorithms (Pan et al., 2010). However, most of the tuning methods require the accurate model of the object and do not have simplicity for practical applications. In this paper, we determined the control gains both in simulation and flight experiment based on the attenuation curve method (Qingyan et al., 2018), which is based on the test data that the attenuation ratio of the response curve of the control system transition process is 4:1. Since the pursuer multicopter is a symmetrical model, and its forward and lateral control models are basically the same.

In the object capture mission, a grabbing mechanism is usually utilized to capture the moving target. The installation position of the capture mechanism cannot coincide with the position of the optical axis of the camera, so as not to avoid obstructing the field of view. Thus, we need to add a bias term to the capture control algorithm. This bias term ξ^b represents the three-dimensional position of the grabbing device relative to the camera center in the pursuer's body coordinate. Since the control command is described in the earth-fixed frame, ξ^b needs to make the coordinate transformation using Equation 22

$$\xi^i = C_b^i \xi^b \quad (22)$$

Where C_b^i indicates the transformation matrix from the earth-fixed coordinate to the pursuer's body coordinate, which is derived using the Euler angles (ψ, θ, ϕ) .

$$C_b^i = \begin{bmatrix} \cos \theta \cos \psi & \sin \phi \sin \theta \cos \psi - \cos \phi \sin \psi & \cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi \\ \cos \theta \sin \psi & \sin \phi \sin \theta \sin \psi + \cos \phi \cos \psi & \cos \phi \sin \theta \sin \psi \sin \phi \cos \psi \\ -\sin \theta & \sin \phi \cos \theta & \cos \phi \cos \theta \end{bmatrix} \quad (23)$$

As a result, we get the bias modified PD control algorithm (Equation 24) to intercept the target. It is simple but can greatly improve the capture performance for the maneuvering target due to the compensation of estimated a_t .

$$a_c = k_p(P_t - P_m - \xi^i) + k_v(V_t - V_m) + k_a a_t \quad (24)$$

Considering that the vertical mobility of the target drone is weak, we only add the acceleration compensation in horizontal plane. In order to ensure the safety of the pursuer without colliding walls or other obstacles, a boundary-approaching constraint is considered under the condition that the position information of the obstacle walls is known. The obstacle information can be obtained from the visual obstacle avoidance algorithm or geofence. Setting the distance between the obstacle and the pursuer is R_o , then the safety distance R_s corresponding to the multicopter velocity and maximum acceleration a_{max} can be derived by $R_s = \frac{V_m^2}{2a_{max}}$. As a result, the constraint control yields

$$a_{cb} = -\frac{V_m^2}{2R_o}, R_o \leq R_s \quad (25)$$

5.2. Yaw control

To ensure that the nose of the multicopter points to the target, we keep the yaw angle of the pursuer tracking the LOS yaw angle. Thus, the yaw control for the pursuer multicopter is given

$$\psi_c = q_z \quad (26)$$

Combining Equation 24 and 26, we can derive the commands for pitch, roll and thrust channel (Quan, 2017). The dynamic model of the multicopter and the transformation can refer to (McNamee and Barrett-Gonzalez, 2020). Since the transformation from the acceleration commands to the attitude commands are programmed in most of the flight control unit (FCU), the control node in our system only needs to output a_c and ψ_c to the FCU. Then, the multicopter will be controlled to intercept the target.

6. Experimental verification

This section presents the simulation and flight experiment results of the proposed control system, from which the capture performance is analyzed. We chose a self-developed multicopter based on the Pixhawk firmware at the beginning of the preparation but replaced it with DJI M-210 platform after considering the load problem of the capture mechanism. Thus, the proposed control methods are applicable both for DJI FCU and Pixhawk FCU.

6.1. Simulation work

The simulation work was conducted using the multi-vehicles simulation environment in the ROS/Gazebo robotic simulator (Figure 11), which supports the hardware-in-the-loop simulation with Pixhawk firmware.

To observe the accuracy of the predicted target trajectory compared to the true value rather than the measured trajectory, we designed an estimating process in Gazebo simulator, in which the

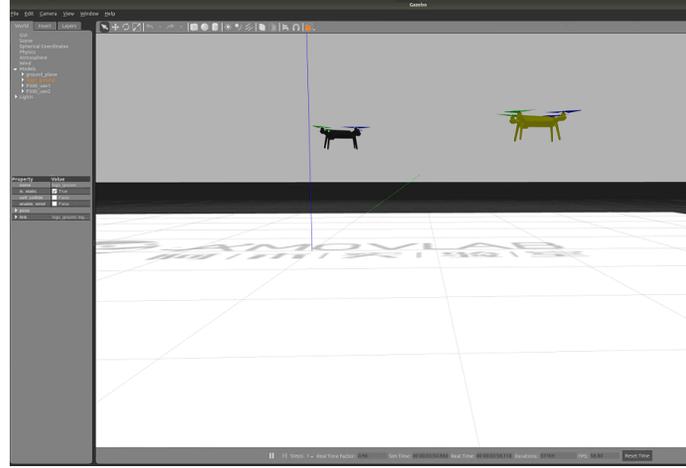


Figure 11. Simulation environment in the ROS/Gazebo robotic simulator.

Table 1. Measurements noise variance for EKF.

| Measurements | $D(m)$ | $q_y(deg)$ | $q_z(deg)$ | $\dot{q}_y(deg)$ | $\dot{q}_z(deg)$ | $V_{mx}(m/s)$ | $V_{my}(m/s)$ | $V_{mz}(m/s)$ |
|----------------|--------|------------|------------|------------------|------------------|---------------|---------------|---------------|
| Noise Variance | 1 | 25 | 25 | 0.36 | 0.36 | 0.25 | 0.25 | 0.25 |

measurement state mentioned in Equation 17 were calculated numerically and then added the noises measured in the flight experiment (refer to Table 1).

A three-dimensional 8-shape trajectory for the target was designed in this scenario. The period of a trajectory was set as $T_t = 24s$. The center of the trajectory was $(x_{t0}, y_{t0}, z_{t0}) = (0m, 0m, 15m)$, which was also the initial location of the target. The target motion model was determined by Equation 27. The pursuer multicopter was hovering at $(0m, 0m, 5m)$ and observing the target.

$$\begin{cases} P_{xt} = 40 \sin(\frac{2\pi}{24}t) + P_{xt0} \\ P_{yt} = 15 \sin(\frac{2\pi}{12}t) + P_{yt0} \\ P_{zt} = 4 \sin(\frac{2\pi}{24}t) + P_{zt0} \end{cases} \quad (27)$$

6.1.1. Simulation for target trajectory prediction

Referring to the EKF method described in Section 4, we derived the initial relative position deviation $(\Delta P_{x0}, \Delta P_{y0}, \Delta P_{z0})$ from Equation 18. Thus, the initial estimating state was set as $X_0 = [\Delta P_{x0}, \Delta P_{y0}, \Delta P_{z0}, 0, 0, 0, 0, 0, V_{mx0}, V_{my0}, V_{mz0}]$. Then, we selected the initial covariance matrix as $P_0 = \text{diag}([0, 0, 0, 25, 25, 25, 9, 9, 9, 0, 0, 0])$. The maneuver frequency of the target was roughly set as $\omega_t = (\frac{2\pi}{30}, \frac{2\pi}{15}, \frac{2\pi}{30})$. The noises variance of the measurements are presented in Table 1. By analyzing the system process noise, Q is adjusted and finally determined.

As the Figure 12 shown, the predicted target trajectory fitted well with the true path. The RMS errors of the estimated target position in Figure 13d were converged below $0.5m$ in $2.5s$. The proposed EKF method also provided a satisfactory target velocity with an accuracy of $0.4m/s$. In Figure 13c, the curve of the estimated target accelerations tracked the measured a_t well but the RMS errors of a_t were slightly noisy. However, the results were reliable for the control algorithm.

6.1.2. Simulation for capture control

In order to verify the random capture performance of the proposed modified PD algorithm and find the optimal intercept trajectory, we selected four different initial locations for the pursuer. Monte Carlo technique (AIAA, 1990; Yan and Shu, 2005) was introduced to statistically examine the control accuracy of the proposed method. Each case has performed 50 simulation runs in the

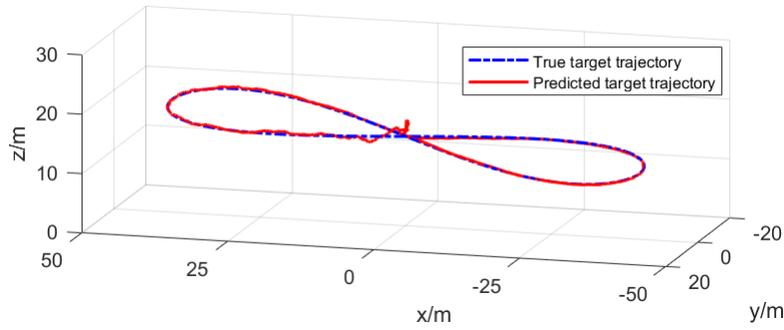


Figure 12. Predicted target trajectory of simulation.

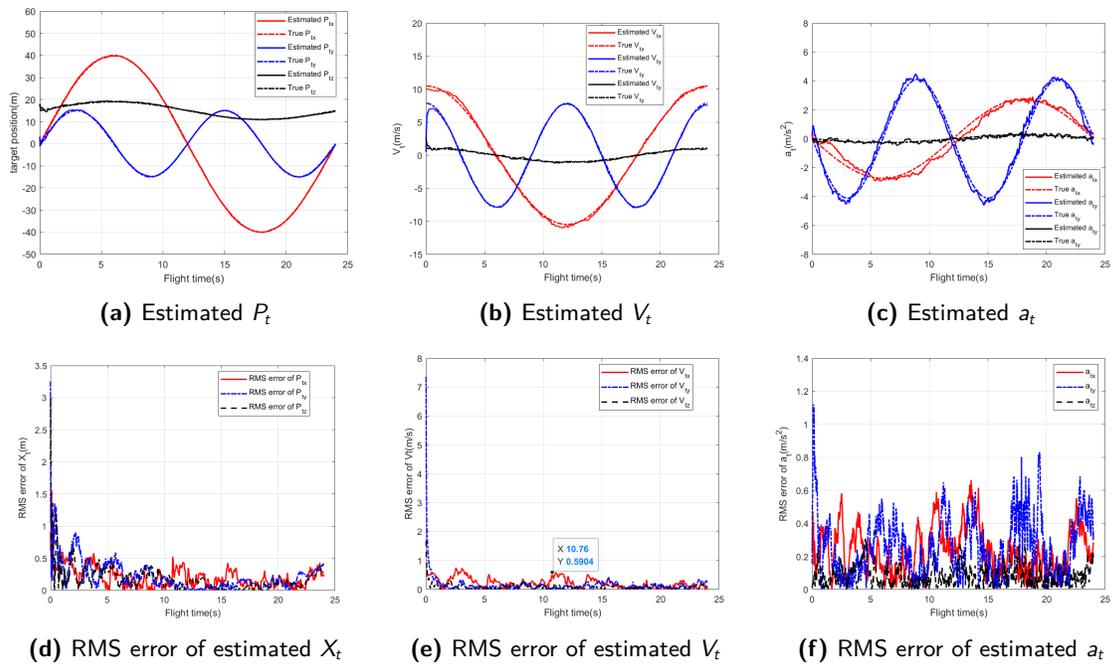


Figure 13. The estimation results of target state.

simulation environment. Spherical Error Probability (SEP) is introduced to evaluate the accuracy of the three-dimensional impact points (Tian-li and Wei-lian, 2006). The mean energy consumption $J = \frac{1}{2} \int_t^{t_f} a_c^2(\tau) d\tau$ and capture time consuming T_g are also utilized to find the optimal control trajectory.

The capture trajectories of four cases in horizontal plane were depicted in Figure 14. All the four locations realized interception with the maneuvering target, and the three-dimensional control accuracies were all below $0.3m$. The control performance of each initial location in Table 2 also suggested that location B was the optimal starting position for capture phase due to the least energy consumption and shortest interception time. Thus, we chose location B as the hovering position to search for the target. With the help of the EOP search method, the camera can detect the target no matter which direction the target is flying from.

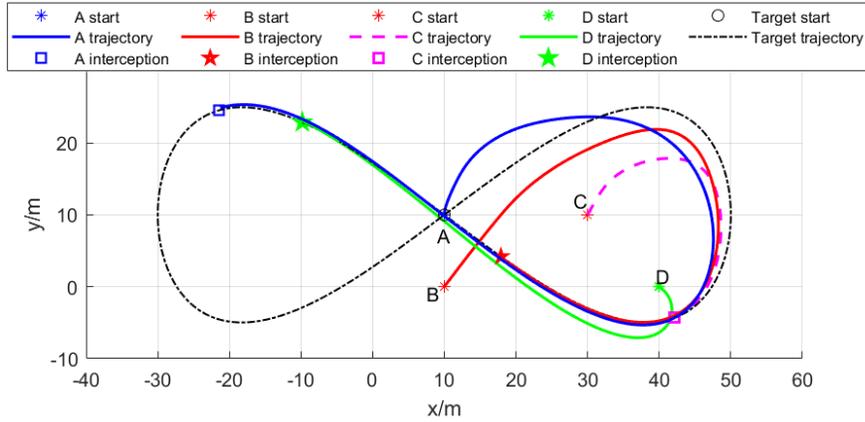


Figure 14. The capture trajectories of different pursuer initial locations.

Table 2. Control performance of different pursuer initial locations.

| Case | initial location(m) | SEP | J_{mean} | T_g |
|------|---------------------|--------|------------|-------|
| A | (10,10,5) | 0.1345 | 851.80 | 19.4 |
| B | (10,0,5) | 0.1276 | 656.03 | 12.06 |
| C | (30,10,5) | 0.2839 | 603.45 | 9.58 |
| D | (40,0,5) | 0.2038 | 878.36 | 18.42 |

6.2. Flight experiment

The scenario in this section was designed based on the challenge I of MBZIRC 2020. Thus, the mission was capturing the 13cm ball hung below the target drone. Referring to the maximum SEP results of 0.3m in Table 2, we designed a grabbing mechanism (Figure 3) which can release the control accuracy requirement to 0.8m. The DJI F-450 multicopter acted as the target and moved following the trajectory in Equation 27. DJI M-210 multicopter was utilized to control the flight of the F450 drone. The target position, velocity and acceleration can be measured by the onboard IMU of DJI F-450. Both the measured accuracy of the target position and velocity were below 0.5m. The target acceleration was measured with noise of $1m/s^2$, so we conducted a low-pass filter to the measured a_t .

According to the analysis of Table 2 in simulation part, the pursuer was hovering at location B and rotating the EOP to search target. When the pursuer detected the target and state estimation converged, the multicopter would start to track the target. For our flight platform, the control gains of velocity term were selected as 3.6, which matched the values in the simulation case. Considering the wind disturbance, the position gains for flight experiment could be slightly bigger than 1.6 of the simulation. The acceleration gains were tuned to 0.6.

The measurements during the capture phase were presented in Figure 15. The relative distance (Figure 15a) was stably obtained from the onboard visual algorithm. With the help of the grabbing device, the pursuer caught the target ball when the depth reached 0.384m. The measured yaw LOS angle (Figure 15b) and angular rate (Figure 15c) oscillated after 9 seconds, which is because the slight movement of the suspended target ball will cause a large change in the LOS information when approaching the target. The pitch LOS angle line converged to zero at $t = 4s$ indicates that the capture mechanism installed on the pursuer pointed towards the target ball.

The real-time estimated target motion states during the capture phase were presented in Figure 16. The estimated relative position in (Figure 16a) showed a good accuracy, which was consistent with the simulation results. The deviation of all the results before 2 seconds were slightly

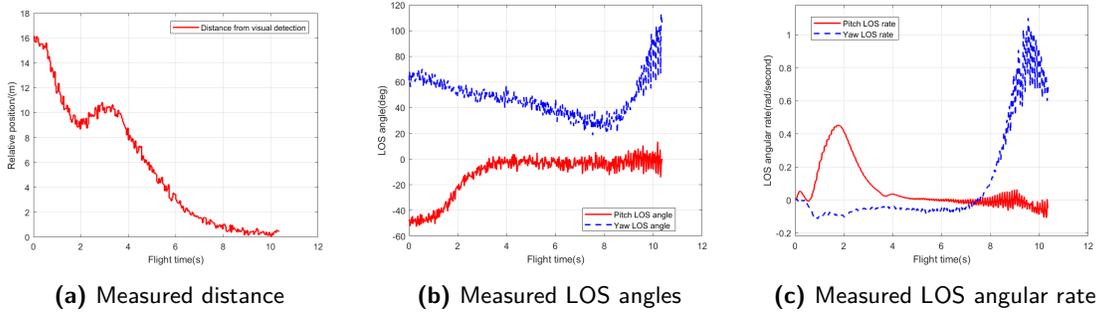


Figure 15. Measurements for EKF.

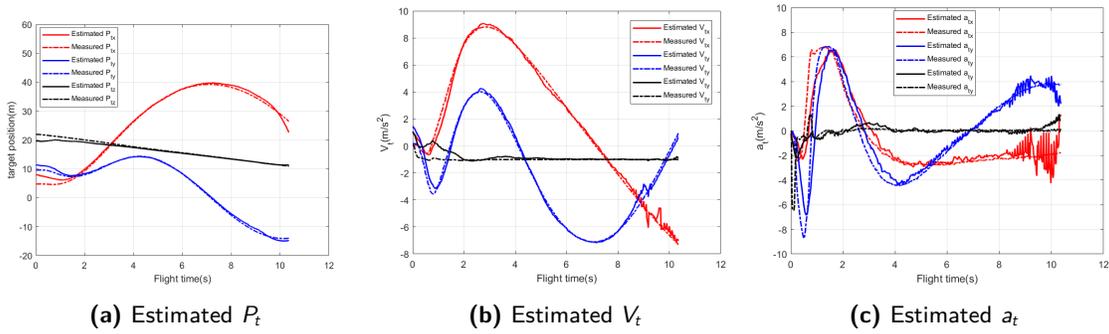


Figure 16. The estimation results of target state.

larger, because the distance detection accuracy was relatively large when the target is far away. The comparison of the predicted (black dotted line) and measured target trajectory (blue dotted line) in Figure 17b also illustrated the phenomenon. The acceleration curves (Figure 16c) appeared obvious oscillations in the end due to the divergence of LOS angular rate. To avoid the control commands for the pursuer oscillating due to the estimated target acceleration, we performed a low-pass filter on the acceleration commands.

Figure 17a showed a capture picture of this flight experiment. As shown in the capture trajectory in Figure 17b, the pursuer realized the interception in the middle straight-line segment, which corresponds to the small acceleration stage in Figure 16c. Thus, this stage is extremely good for grabbing the target ball. In Figure 18a and 18b, we can find that the relative position and velocity error finally converged to zero. Due to the compensation of the estimated a_t , the proposed control algorithm made full use of the pursuer's overload performance in the initial segment, thus it can finally chase the target in a straight trajectory. The phenomenon was also verified in Figure 18c.

The lateral acceleration command a_{cy} reached the maximum limit value of $7m/s^2$ before 5 seconds, and then gradually converged. In addition, the boundary constraint was not met due to the good tracking performance of the proposed control algorithm. We have conducted numerous flight experiments at different times and scenes, the capture performance for each time was good. Flight videos are available at <https://b23.tv/Y0qFN1>.

6.3. Control performance in MBZIRC2020

Basing on the DJI M210 platform, our control system was applied to challenge I of the final MBZIRC 2020 competition. We successfully finished the whole mission within two minutes and 40 seconds. Figure 19 shows the photos of our performance during the competition. Related video of the capture

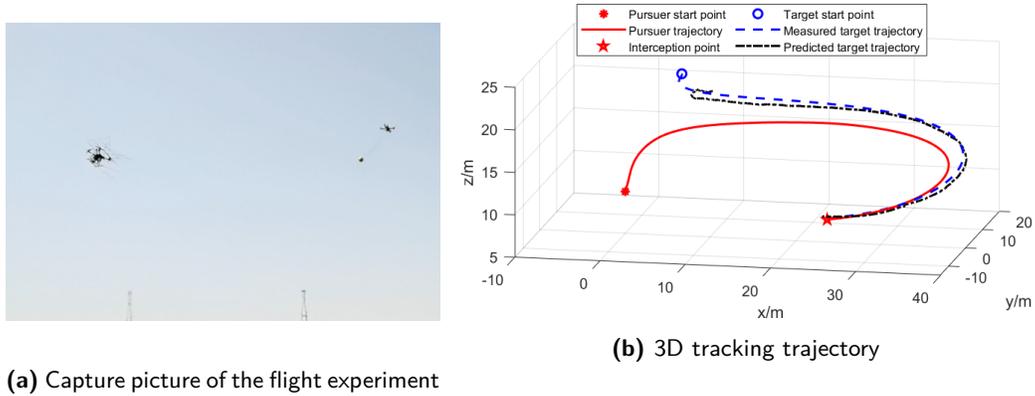


Figure 17. Outdoor capture experiment.

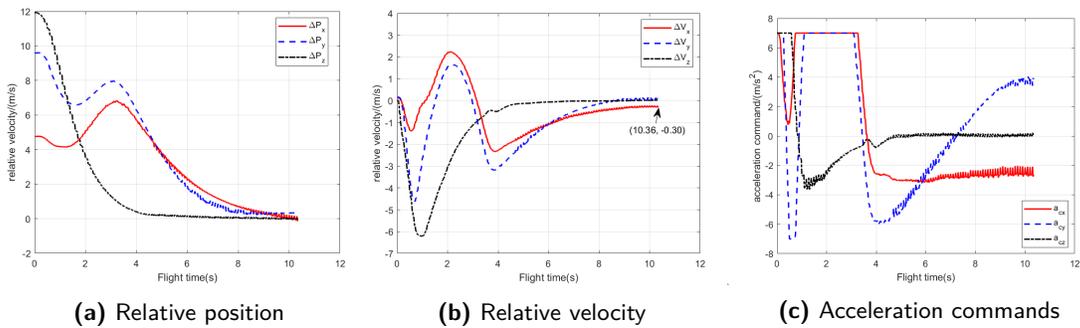


Figure 18. Motion states of the capture phase.

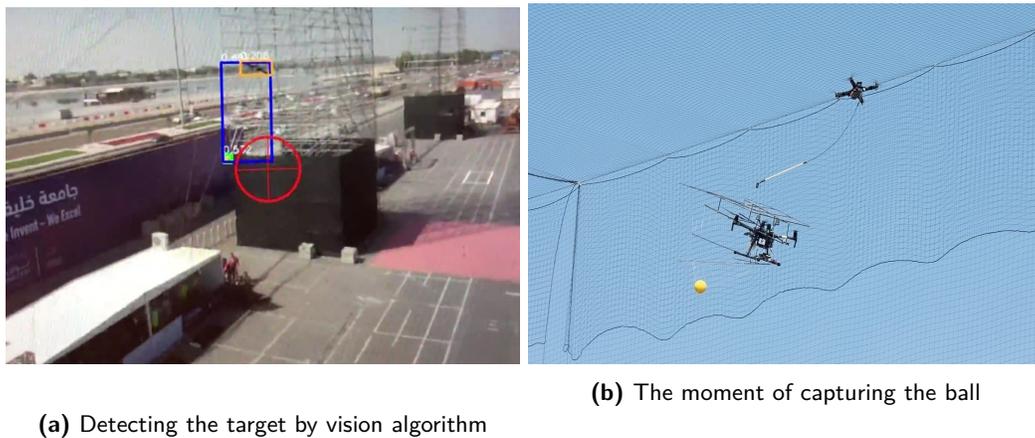


Figure 19. The capture moment in MBZIRC competition.

process is available at <https://b23.tv/TQGWUZ>. We were the only team which completed all tasks autonomously in the shortest time. As a result, we won the championship of challenge I in MBZIRC 2020(Figure 20). This success of our team has also developed a significant approach for the aerial safety protection in urban environments.



Figure 20. Award photo of our team.

7. Conclusion

In this paper, we present an autonomous navigation and control system for capturing a moving drone target. The vision-based navigation component seeks and detects the intruding drone, then three information sources, namely including the onboard vision algorithm, electro-optical pod, and inertial measurement units are fused to predict the target trajectory in real time. Both the simulation and the outdoor flight experiments have verified that our prediction method can estimate the target's trajectory with the accuracy of $0.5m$, and it can also reliably estimate the target's velocity and acceleration. Thus, the precise relative motion information is obtained and fed to the control algorithm.

The capture performance of the proposed modified PD algorithm has been studied both in hardware-in-the-loop simulation and flight experiments. The results we obtained from the Monte Carlo simulation show a three-dimensional capture accuracy below $0.3m$ for varied starting locations. The simulation gives a good suggestion for the optimal capture trajectory. Then the flight experiment verified that the proposed modified PD algorithm can accurately intercept a moving target by compensating for the estimated target acceleration. It has also been demonstrated the considerable efficiency and robustness in the ball-capture mission of MBZIRC Challenge, in which the task is more difficult than capturing the drone itself.

8. Lessons learned and future research

The proposed navigation and control system autonomously and successfully finished the mission in Challenge I of MBZIRC 2020, whose performance has demonstrated our design's value. However, considering the motivation of Challenge I in MBZIRC 2020, we still need to improve our key performance criteria due to the following aspects. The proposed target prediction method relies on the target size, which is known in the competition. However, we can hardly know the type and size of the invading target. Thus, the versatility of the vision perception and estimation method needs to be improved. Additionally, optimal control algorithms for intercepting the target with random maneuver modes should be studied. The control energy and time are key performance due to the limited endurance capability of drones.

We have learned numerous lessons from MBZIRC 2020 and identified several techniques to improve. In the future research, we will continue to study a cost-effective navigation and control system against invading drones and aerial security maintenance.

Acknowledgments

The authors acknowledge the organizing committee of MBZIRC 2020, which exhibits this competition and develops the research of robotics.

ORCID

Hong Tao  <https://orcid.org/0000-0003-4403-116X>
 Tao Song  <https://orcid.org/0000-0001-7412-7336>
 Defu Lin  <https://orcid.org/0000-0003-2736-5096>
 Ren Jin  <https://orcid.org/0000-0002-4432-3181>

References

- Achour, Z., Rezoug, A., and Hamerlain, M. (2015). Adaptive pid + fuzzy pid controller for birotor uav system. In *International Electrical and Computer Engineering Conference (IECEC 2015)*.
- AIAA (1990). Monte carlo optimization strategies for air-traffic control - aiaa guidance, navigation, and control conference and exhibit (aiaa). *Electronics & Communication Engineering Journal*, 2(5):202 – 208.
- Becker, K. (1990). Closed-form solution of pure proportional navigation. *Aerospace & Electronic Systems IEEE Transactions on*, 26(3):526–533.
- Bing, Z., Xu, J., Zaini, A. H. B., and Xie, L. (2016). A three-dimensional integrated guidance law for rotary uav interception. In *IEEE International Conference on Control & Automation*.
- Brust, M. R., Danoy, G., Bouvry, P., Gashi, D., Pathak, H., and Goncalves, M. P. (2017). Defending against intrusion of malicious uavs with networked uav defense swarms. In *2017 IEEE 42nd Conference on Local Computer Networks Workshops (LCN Workshops)*.
- Daum, F. (2005). Nonlinear filters: beyond the kalman filter. *IEEE Aerospace and Electronic Systems Magazine*, 20(8):57–69.
- Fu, C. Y., Shvets, M., and Berg, A. C. (2019). Retinamask: Learning to predict masks improves state-of-the-art single-shot detection for free.
- Gambs, S., Killijian, M.-O., and del Prado Cortez, M. N. (2012). Next place prediction using mobility markov chains. In *Proceedings of the First Workshop on Measurement, Privacy, and Mobility*, page 3.
- Grassi, E. and Tsakalis, K. (2015). Pid controller tuning by frequency loop-shaping: Application to diffusion furnace temperature control. *Control Systems Technology IEEE Transactions on*, 8(5):842–847.
- He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017). Mask r-cnn. *IEEE Transactions on Pattern Analysis & Machine Intelligence*.
- Huckridge, D. A., Ebert, R., Lee, S. T., Gao, C., Wei, G., Wang, Q., Xiong, Z., Wang, Q., and Long, X. (2016). A three axis turntable’s online initial state measurement method based on the high-accuracy laser gyro sins. 9987:998714.
- Law, H. and Deng, J. (2018). Cornernet: Detecting objects as paired keypoints. *International Journal of Computer Vision*.
- Mcnamee, P. and Barrett-Gonzalez, R. M. (2020). Modeling and simulation of quadcopter dynamics in steady maneuvers. In *AIAA Scitech 2020 Forum*.
- Pan, I., Das, S., and Gupta, A. (2010). Tuning of an optimal fuzzy pid controller with stochastic algorithms for networked control systems with random time delay. *Isa Transactions*, 50(1):28–36.
- Park, S. T. and Lee, J. G. (2001). Improved kalman filter design for three-dimensional radar tracking. *IEEE Transactions on Aerospace & Electronic Systems*, 37(2):727–739.
- Qingyan, W., Zhiyun, Z., Zhong, Y., Jianlong, B., Wei, Z., and Yongping, Y. (2018). Design method of linear motor pid controller based on attenuation curve method.
- Quan, Q. (2017). *Introduction to Multicopter Design and Control*. Springer Singapore.
- Rogers, J. (2019). The dark side of our drone future.
- Tian-li, S. and Wei-lian, A. (2006). Three-dimension accuracy evaluation method of missile. *Journal of Astronautics*.
- Ximeng, X., Rennong, Y., Tao, Z., and Bin, Y. (2019). Trajectory prediction of target aircraft in air combat based on ga-oif-elman neural network. In *2019 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)*.

- Yamasaki, T., Sakaida, H., Enomoto, K., Takano, H., and Baba, Y. (2007). Robust trajectory-tracking method for uav guidance using proportional navigation. In *International Conference on Control*.
- Yan, L. and Shu, C. (2005). The application of monte carlo method to estimation of position acquisition probability of terminal guidance radar. *Tactical Missile Technology*.
- Yang, R., Zhenxing, Z., Ying, Z., Huan, Z., Ming, Y. U., and Guohong, X. (2017). Prediction of aircraft flight performance model based on narx neural network. *Journal of Northwest University(Natural Science Edition)*.
- Yoon, S., Kim, Y., and Kim, S. (2008). Pursuit guidance law and adaptive backstepping controller design for vision-based net-recovery uav.
- Zarchan, P. (2012). *Tactical and strategic missile guidance*. American Institute of Aeronautics and Astronautics, Inc.
- Zarchan, P. and Musoff, H. (2005). *Fundamentals of Kalman Filtering: A Practical Approach*, volume 208. Aiaa.

How to cite this article: Tao, H., Song, T., Lin, D., Jin, R., & Li, B. (2022). Autonomous navigation and control system for capturing a moving drone. *Field Robotics*, 2, 34–54.

Publisher's Note: Field Robotics does not accept any legal responsibility for errors, omissions or claims and does not provide any warranty, express or implied, with respect to information published in this article.