DOI 10.24412/1829-0450-fm-2025-2-17-25
УДК
Поступила: 13.10.2025г.
Сдана на рецензию: 13.10.2025г.
Подписана к печати: 21.10.2025г.

AUTONOMOUS UAV CONTROL BASED ON CAMERA VIDEO

V. Melkonyan

Russian-Armenian (Slavonic) University
vahagn.melkonyan@student.rau.am

ABSTRACT

This paper presents a vision-based autonomous control system for unmanned aerial vehicles (UAVs) designed to intercept and track moving targets using onboard camera data. The proposed approach integrates real-time visual feedback with a dynamic control mechanism that continuously adjusts the UAV's trajectory based on the estimated target position.

Keywords: autonomous control, UAV, vision-based navigation, visual servoing, real-time control.

Introduction

Interception is a challenging control problem for unmanned aerial vehicles (UAVs), where the goal is to engage and hit a moving target accurately. This task requires fast, accurate, and dynamically stable control rather than simple visual tracking. It is relevant to applications such as aerial capture, defense, and autonomous pursuit of moving objects. Vision-based interception enables fully autonomous operation without relying on GPS or external localization systems. The UAV must estimate the target's position directly from onboard camera data and continuously adapt its motion in real time. However, vision introduces inherent uncertainties, including image delay, noise, and limited depth perception, which complicate control system design. In this work, the UAV uses a monocular camera, which simplifies the hardware but demands a robust and responsive control mechanism capable of compensating for limited spatial information. This paper focuses on the control aspect of the interception problem. This work proposes a vision-based control system that combines proportional navigation (PN) with visual servoing principles. The controller dynamically adjusts the UAV's trajectory using the target position estimated from image coordinates, ensuring smooth and accurate terminal interception. The approach is validated both in simulation and real-world flight tests, demonstrating high interception success rates.

Related Work

Recent research has explored various strategies for UAV interception using onboard vision. For instance, the work in [1] applies an image-based visual servo (IBVS) approach for high-speed target pursuit, while [2] utilizes stereo vision and deep learning-based detection for dynamic interception. Other studies, such as [3], integrate proportional navigation with visual feedback to enhance accuracy. Despite these advances, most approaches either rely on complex hardware (e.g., stereo or depth cameras) or struggle with real-time performance and robustness under varying lighting and motion conditions.

The challenge remains to design a lightweight, vision-only control mechanism capable of achieving accurate terminal interception under uncertainty. Our method addresses this problem by unifying PN guidance with visual servo feedback derived from monocular camera data.

Proposed Method

The proposed interception controller uses only the tracked bounding box center from a monocular camera as input and runs both in simulation (PX4 [4] HITL [5] + ROS2 [6] with Gazebo [7]) and on real hardware. For target tracking, we employ a hybrid tracker [8] combining MixFormer [9] and KCF [10], leveraging MixFormer for robust object re-identification and KCF for high-speed local updates. The controller operates in three mutually exclusive cases (close, above, normal) and uses a two-rate control cycle: tick (outer loop) and subtick (inner loop). These cycles compute and apply attitude and thrust commands that lead the UAV to physically contact the target.

A. High-level control flow and cases

Let the tracked bounding box center in image coordinates be (u, v) and the configured hit point in image coordinates as (u_h, v_h), which is fixed before flight but can be changed online via MAVLink [11]. After the operator selects the bounding box (either by manual click or by choosing from pre-detected objects), the controller evaluates the box position and selects one of three modes.

Case 1: Object too close (box center low in image)

If v is below a configurable lower threshold, the interceptor reduces altitude and actively centers the box along the image X-axis. This prevents excessive downward speed that would make attitude control unstable. When a critical height is reached, the controller switches to Case 3.

Case 2: Object above the interceptor (box center high in the image)

If v is above a configurable upper threshold, the UAV climbs to reduce the pitch command that would otherwise cause backward motion. The climb continues until the commanded pitch reaches a predefined critical angle. After reaching this angle, the controller switches to Case 3.

Case 3: Normal interception

When neither Case 1 nor Case 2 holds, the interception process runs in ticks and subticks. Each tick computes yaw and pitch commands to move the bounding box center toward the hit point $(u_h,\ v_h)$. During subticks, PID [12] (Proportional–Integral–Derivative) controllers regulate roll and thrust to minimize pixel-wise errors between the box center and the hit point. Those controllers also have an anti-windup mechanism [13] to prevent the integral term from growing excessively large.

B. Tick and Subtick roles (outer and inner loop)

In each tick of the outer loop, which runs every T frames, the system calculates the desired yaw and pitch angles needed to move the bounding box center toward the hit point (u_h, v_h). It then sets the target attitude setpoints for the inner loop and resets the PID terms to maintain consistent behavior in that loop.

Within each subtick of the inner loop, which runs every t_s frames, the system runs the PID_{Roll} and PID_{Thrust} controllers using the pixel-space errors. These controllers generate a roll command, limited by predefined upper and lower bounds, and a thrust value that stays within the range $[0, 1]$. The resulting roll and thrust commands are then sent to PX4 as attitude and thrust setpoints.

This two-rate control approach allows the outer loop to handle coarse direction planning, while the faster inner loop performs fine, real-time corrections to keep the system stable and responsive.

C. PID controllers and mapping

The **Roll PID controller** reduces the rotational alignment error, which represents the angular difference between the object's center and the drone's dynamically adjusted horizon line. In other words, it keeps the target aligned with the horizon as the drone rolls.

The **Thrust PID controller** minimizes the vertical deviation between the desired interception pitch line and the target's projected position. This allows the drone to maintain a steady pitch trajectory toward the target during engagement, ensuring stable and precise motion.

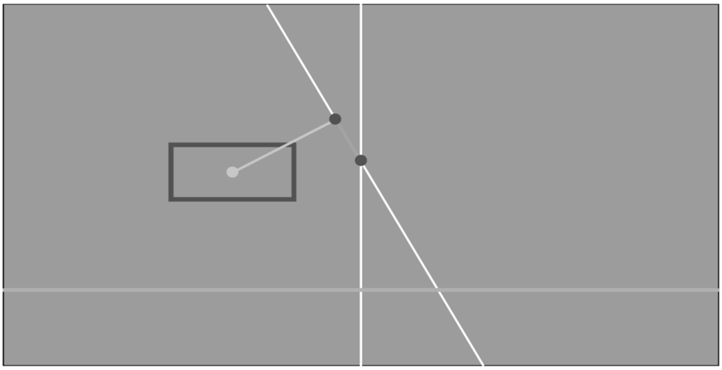

Figure 1. *The visual servoing control loop. The white lines represent the camera's vertical axis and its dynamically rotated vertical line. The green box marks the tracked object, with the yellow dot showing its center. The green dot indicates the projection of the object's center onto the rolled vertical line, while the red dot depicts the optical interception direction. The purple line represents the drone's critical pitch angle threshold.*

Figure 1 shows the visual servoing geometry used by the controller:

- ✓ White lines represent the camera's vertical axis and the dynamically rolled vertical line.
- ✓ The green box and yellow dot mark the tracked object and its center.
- ✓ The green dot is the projection of the object center onto the rolled vertical line.
- ✓ The red dot indicates the direction of optical interception.
- ✓ The yellow line shows the roll error (minimized by PID_{Roll}), the blue line the thrust/pitch error (minimized by PID_{Roll}), and the purple line the critical pitch threshold (for case 1).

Results

This section presents both simulation and real-world results obtained from the proposed autonomous UAV interception system. The evaluation aimed to validate system robustness across varying target speeds, visual conditions, and flight dynamics, using both the PX4-based simulator and multiple physical drone platforms.

A. Simulation

Performance across different target speeds was evaluated using the PX4 HITL (hardware-in-the-loop) Gazebo environment integrated with ROS2 and the proposed visual control system. Targets were simulated as objects moving along predefined trajectories with variable speeds relative to the UAV. Table 1 summarizes interception performance across four target speed categories.

The system demonstrated near-perfect interception performance for stationary and slow targets, maintaining a 100% success rate at 0 m/s and 98% at 5 m/s. A trial was considered successful when the UAV physically made contact with the target. Performance gradually decreased with increasing target speed, primarily due to the higher apparent motion and reduced control time window. At 15 m/s head-on approaches, success dropped to 78%, mainly due to rapid apparent size growth and high optical flow, which occasionally led to late control reactions or ground collisions.

Table 1.

Simulation results: interception success rate by target speed with a fixed 100m distance

Target Speed (m/s)	# Scenarios	Success (%)	Avg. Time (s)	Failures (Ground / Timeout)
0	49	100	8	0 / 0
5	49	98	13	1 / 0
10	49	92	16	3 / 1
15	49	78	21	5 / 6
Total	196	92	15	9 / 7

Overall, the simulation achieved an average success rate of 92% across all tested scenarios. Failures occurred mostly in high-speed head-on cases, where the camera's field of view was quickly saturated and stabilization control reached its limit.

The HITL simulations closely replicated real flight dynamics, confirming that the simulated environment reliably predicted actual flight performance. This setup provided an effective framework for tuning tracker thresholds, control parameters, and mission logic before deploying on real UAVs.

B. Real-world testing

Real-world flight experiments were conducted to validate system performance under practical operational conditions. The primary test platform was the Reptile X500 quadrotor, with additional verification flights on Holybro S500 [14] and Holybro X500 [15] frames to confirm cross-platform compatibility.

Onboard processing was performed using NVIDIA Jetson modules (Nano [16], Xavier NX [17], and Orin Nano [18]), while video input was provided via both USB (Runcam [19]) and CSI (Raspberry Pi [20]) cameras. The camera feed was processed in real time to extract bounding box coordinates and provide feedback to the control loop.

Flights were executed across diverse mission configurations, including:

- ✓ Varying initial yaw, pitch, and roll orientations of the UAV
- ✓ Both stationary (size: $2m \times 2m \times 2m$) and moving targets (size: $1m \times 1m \times 0.5m$)
- ✓ Ground-based and airborne interception scenarios
- ✓ Light to moderate wind, with occasional gusts

In total, approximately 150 real-world flights were conducted. All flights were logged, including MAVLink telemetry, onboard camera video, and control system debug data. These logs were used for post-flight analysis, parameter tuning, and qualitative performance evaluation.

Out of the 150 conducted experiments, 136 flights (90.6%) achieved successful interception, confirming high consistency between simulation and real-world behavior.

Conclusion

This paper presented a vision-based autonomous control system for UAV interception using monocular camera feedback. The proposed method integrates proportional navigation with visual servoing and adaptive PID control to achieve precise target alignment and collision trajectory. Both simulation and real-world experiments demonstrated over 90% interception success, validating the system's effectiveness and robustness. Future work will focus on improving high-speed performance and extending the approach to multi-target and cooperative interception scenarios.

Acknowledgment. *This work was supported by the Science Committee of RA (Research project No 23AA-1B005).*

REFERENCES

1. *Kun Yang, Chenggang Bai, Zhikun She and Quan Quan*, 2025, January. High-speed interception multicopter control by image-based visual servoing. *IEEE Transactions on Control Systems Technology*, 33(1). PP. 119–135. arXiv preprint arXiv:2404.08296v1.
2. *Lyman, T.J., Fields, T.D., and Yakimenko, O.A.,* 2021. Evaluation of Proportional Navigation for Multirotor Pursuit. *Journal of Field Robotics*, 38(6). PP. 1025–1045.
3. *Hailong Yan, Kun Yang, Yixiao Cheng, Zihao Wang and Dawei Li. 2025, April 4.* Precise interception flight targets by image-based visual servoing of multicopter. arXiv preprint arXiv:2409.17497.
4. Open Source Autopilot PX4. [Online]. Available: https://px4.io/
5. Hardware-in-the-loop simulation (HITL). [Online]. Available: https://docs.px4.io/main/en/simulation/hitl.html
6. ROS 2 (Robot Operating System). [Online]. Available: https://docs.ros.org
7. Gazebo Simulator. [Online]. Available: https://gazebosim.org/
8. *Sardaryan A., Sahakyan V., Melkonyan V., Sargsyan S.* 2024. An Accurate Real-Time Object Tracking Method for Resource Constrained Devices. Proceedings of the Institute for System Programming of the RAS, 36(3). PP. 283–294.
9. *Yutao Cui, Cheng Jiang, Limin Wang, Gangshan Wu*, 2022. Mixformer: End-to-end tracking with iterative mixed attention. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. PP. 13608–13618.

10. *João F. Henriques, Caseiro R., Martins P., Batista J.* 2014, April. High-speed tracking with kernelized correlation filters. IEEE Transactions on Pattern Analysis and Machine Intelligence 37(3). PP. 583–596.
11. MAVLink. [Online]. Available: https://mavlink.io/en/
12. *Bennett St.* 1993, December. Development of the PID controller. IEEE Control Systems, 13(6). PP. 58–62.
13. *Lars Rundqwist*, 1990. Anti-reset windup for PID controllers. Control Engineering Practice, 1(1). PP. 37–41.
14. Holybro S500. [Online]. Available: https://holybro.com/products/s500-v2-kit
15. Holybro X500. [Online]. Available: https://holybro.com/products/x500-v2-kits
16. NVIDIA Jetson Nano. [Online]. Available: https://developer.nvidia.com/embedded/jetson-nano
17. NVIDIA Jetson Xavier NX. [Online]. Available: https://developer.nvidia.com/embedded/jetson-xavier-nx
18. NVIDIA Jetson Orin Nano. [Online]. Available: https://developer.nvidia.com/embedded/learn/get-started-jetson-orin-nano-devkit
19. Runcam USB cameras. [Online]. Available: https://www.runcam.com
20. Raspberry Pi CSI camera. [Online]. Available: https://www.raspberrypi.com/products/camera-module-3/

АВТОНОМНОЕ УПРАВЛЕНИЕ БПЛА НА ОСНОВЕ ВИДЕО С КАМЕРЫ

В.Г. Мелконян

Российско-Армянский (Славянский) университет

АННОТАЦИЯ

В данной работе представлена система автономного управления беспилотным летательным аппаратом (БПЛА), основанная на обработке видеоданных с бортовой камеры и предназначенная для перехвата и сопровождения движущихся целей. Предлагаемый подход интегрирует визуальную обратную связь в реальном времени с динамическим механизмом управления, который непрерывно корректирует траекторию полета БПЛА на основе оцененного положения цели.

Ключевые слова: автономное управление, БПЛА, навигация на основе компьютерного зрения, визуальный контроль, управление в реальном времени.