

ՀՀ ԳԱԱ ԻՆՖՈՐՄԱՏԻԿԱՅԻ ԵՎ ԱՎՏՈՄԱՏԱՑՄԱՆ ՊՐՈԲԼԵՄՆԵՐԻ ԻՆՍՏԻՏՈՒՏ

Վահագն Գևորգի Մելքոնյան

Թիրախին հետևող ԱԹՍ ավտոպիլոտի մշակում համակարգչային տեսողության կիրառմամբ

Ե.13.04 - «Հաշվողական մեքենաների, համալիրների, համակարգերի և ցանցերի մաթեմատիկական և ծրագրային ապահովում» մասնագիտությամբ տեխնիկական գիտությունների թեկնածուի գիտական աստիճանի հայցման ատենախոսության

ՍԵՂՄԱԳԻՐ

ԵՐԵՎԱՆ - 2026

INSTITUTE FOR INFORMATICS AND AUTOMATION PROBLEMS OF THE NAS RA

Vahagn Gevorg Melkonyan

Development of an UAV autopilot for tracking targets by using computer vision

SYNOPSIS

of the dissertation for obtaining a Ph.D. degree in Technical Sciences on specialty 05.13.04 "Mathematical and Software Support for Computing Machines, Complexes, Systems, and Networks"

YEREVAN - 2026

Ատենախոսության թեման հաստատվել է Ռուս-Հայկական (Սլավոնական) Համալսարանում

Գիտական ղեկավար՝ Տեխ.գիտ.դոկ. Դ.Գ. Ասատրյան

Պաշտոնական ընդհմախողներ՝ Տեխ.գիտ.դոկ. Հ.Վ. Ասցատրյան

Տեխ.գիտ.թեկ. Ա.Հ. Բաղիյան

Առաջատար կազմակերպություն՝ Երևանի Պետական Համալսարան

Պաշտպանությունը կայանալու է 2026թ. մայիսի 25-ին, ժամը 13:00-ին ՀՀ ԳԱԱ Ինֆորմատիկայի և ավտոմատացման պրոբլեմների ինստիտուտում գրոծող 037 մասնագիտական խորհրդում:

Ատենախոսությանը կարելի է ծանոթանալ ՀՀ ԳԱԱ ԻԱՊԻ գրադարանում:

Սեղմագիրն առաքված է 2026թ. ապրիլի 25-ին:

Մասնագիտական խորհրդի գիտական քարտուղար ֆիզ.մաթ.գիտ.դոկտոր՝



Մ.Ե. Հարությունյան

The topic of the dissertation was approved at the Russian-Armenian (Slavonic) University

Scientific supervisor: Doctor of Tech. sciences D.G. Asatryan

Official opponents: Doctor of Tech. sciences H.V. Astsatryan

Candidate of Tech. sciences A.H. Baghiyan

Leading organization: Yerevan State University

The defense will take place on May 25, 2026, at 13:00, at the Institute for Informatics and Automation Problems of the National Academy of Sciences of the Republic of Armenia, at the acting 037 specialized council.

The dissertation is available in the library of IIAP NAS RA.

The abstract is delivered on April 25, 2026.

Scientific Secretary of the Specialized Council,
D.Ph.M.S.



M.E. Haroutunian

1. Introduction

Over the past two decades, unmanned aerial vehicles (UAVs) have evolved from niche tools in specialized military reconnaissance units into a critical and versatile component of modern technology. Initially, they were primarily tasked with aerial photography, cartography, and intelligence gathering. However, rapid advances in autonomy, computer vision, and onboard processing, combined with the availability of affordable sensors, miniaturized computing platforms, and advanced control algorithms, have enabled UAVs to perform operations without human intervention. As a result, UAVs are now widely deployed across civilian, industrial, and defense applications.

UAV-based interception has proven capable of complementing or even replacing traditional manned or ground-based interception systems. They are capable of detecting, tracking, and physically engaging both airborne and ground-based targets without direct human piloting, by reducing response time and operational risk. This technology can also be applied to a wide range of security and civilian tasks, including border and perimeter protection, such as patrolling and intercepting unauthorized ground or aerial intrusions along national borders.

The objective of this work is to design and develop a lightweight, robust, and efficient system for real-time target tracking, interception, and ground control of UAVs operating in dynamic environments. To achieve this, the **following challenges** must be addressed:

- **Analyze and evaluate existing UAV visual tracking and interception systems** with respect to computational efficiency, robustness, and deployment feasibility on resource-constrained platforms.
- **Design and develop a modular architecture** that integrates perception, tracking, and control into a unified system capable of real-time operation on onboard computing platforms.
- **Design and develop a lightweight and robust visual tracking module** that delivers stable, real-time performance on onboard hardware (with less than 1,280 GFLOPS) under varying environmental conditions and target behaviors, supporting both predefined object classes and class-agnostic tracking to allow dynamic target selection during missions.
- **Design and develop a robust control algorithm for autonomous interception** that accurately guides the UAV toward moving targets while maintaining flight stability, integrating directional estimations, target approach logic, and corrective control adaptable to different target speeds, approach angles, and operational environments.
- **Design and develop a real-time ground control and monitoring system** featuring an extended Ground Control Station (GCS) interface that integrates real-time video streaming, intuitive target selection, in-flight parameter tuning, rapid operator intervention capabilities, and clear visual feedback.
- **Design and implement an automated Hardware-in-the-Loop (HITL) simulation framework** to systematically test and validate the UAV's visual

tracking, guidance, and control algorithms under diverse target behaviors, environmental conditions, and mission scenarios.

By achieving these objectives, the research aims to develop a UAV interception system that balances tracking accuracy, control stability, and user interface intuitiveness, facilitating the practical deployment of such a system in defense, security, and civilian applications.

The following contributions characterize the **scientific novelty of the work**:

- The development of a **UAV interception architecture** that can be deployed on multirotor UAV platforms equipped with flight controllers and onboard computing systems.
- The development of a **real-time hybrid tracking** pipeline designed for embedded UAV hardware and capable of tracking operator-defined arbitrary targets.
- The development of a **PID (Proportional Integral Derivative) control system** with yaw and pitch guidance, independent roll and thrust control, and real-time in-flight parameter tuning via.
- The development of an **evaluation system** combining HITL simulations with diverse target dynamics and real-world field experiments.

From a theoretical standpoint, this research advances the design and implementation principles of autonomous UAV interception systems by proposing a unified architecture for real-time interception tasks.

The system integrates computer vision modules and operator interaction layers into a single distributed ROS 2 framework, ensuring modularity, scalability, and interoperability with existing UAV platforms. The architecture defines clear data flows between perception, decision-making, and actuation layers, enabling other researchers and developers to adapt or extend the system for related UAV autonomy problems.

From a practical perspective, the results of this work provide a ready-to-use technological foundation for deploying autonomous UAV interception capabilities in real operational environments.

The developed system has been validated in both high-fidelity simulation and real-world flight tests, proving its ability to detect, track, and intercept ground and aerial targets under varying weather, lighting, and movement conditions.

Its design emphasizes portability across different UAV airframes, allowing agencies and organizations to integrate the solution into existing fleets without major hardware changes.

The adaptability of the tracking and control modules, combined with the enhanced GCS interface, enables operators to perform interception missions by selecting the target's initial position, ensuring rapid field deployment in defense, security, and public safety operations.

Conference and Competitions

The research and related UAV developments have been presented at various

international and national conferences, workshops, and competitions, earning recognition and awards:

- "Havq" Competition, **Payload Delivery with UAV**, 2nd place, Yerevan, Armenia. 2023
- "Gagarin Science" Conference, "**Optimal Area Coverage Using a Swarm of Drones**", Moscow, Russia. 2024.
- "Gagarin Science" Conference, "**Inertial Navigation for Copters**", Moscow, Russia. 2024.
- 17th International Annual Student Scientific Conference of RAU, "**Optimal Area Coverage Using a Swarm of Drones**", Yerevan, Armenia. 2024.
- 17th International Annual Student Scientific Conference of RAU, "**Inertial Navigation for Copters**", Yerevan, Armenia. 2024.
- United Nations / Philippines Workshop on the Applications of GNSS, "**GPS-based 2D Map Creation using Drone Swarm**", Manila, Philippines. 2024.
- "**Professionals**" Competition, **Unmanned Aerial Vehicle Operations**, 2nd place, Saint Petersburg, Russia. 2024.
- United Nations Workshop on GNSS and Related Space Technologies for Urban Sustainability Challenges, "**Drone-Based GPS Object Localization for Urban and Agricultural Monitoring**", Online. 2024.
- United Nations Workshop on GNSS and Related Space Technologies for Urban Sustainability Challenges, "**GPS-Enhanced Drone Imaging: Stitching and Change Detection Analysis**", Online. 2024.
- 18th International Annual Student Scientific Conference of RAU, "**Autonomous Target Interception with Quadcopters**", Yerevan, Armenia.
- 2025 - Ivannikov Memorial Workshop, "**Real-time Target Localization Using Gimbaled Laser on UAVs**", Irkutsk, Russia. 2024.
- Ivannikov Memorial Workshop, "**Object Re-detection in Aerial Imagery Captured by UAVs**", Irkutsk, Russia. 2025.
- 19th International Annual Student Scientific Conference of RAU, "**Autonomous UAV control based on camera video**", Yerevan, Armenia. 2025.

Grants

The research was supported by several grant-funded projects, which provided both financial and technical resources for the development, testing, and dissemination of the results. These grants include:

- "**ԲԱՋԷ**", Project ՔՏԱՆ-ԴՄ-2021/2-1, funded by the Ministry of High-Tech Industry of the Republic of Armenia. 2021.
- "**Autonomous Aerial Interceptor Drone**", Project 23DP-1B017, funded by the Ministry of Education, Science, Culture, and Sports of the Republic of Armenia. 2022.

- "Development of a UAV Autopilot for Tracking Targets by Using Computer Vision", PhD grant 23AA-1B005, funded by the Ministry of Education, Science, Culture, and Sports of the Republic of Armenia. 2023.
- "Control and Management System for FPV Drone Swarm", Project 25DP-1B005, funded by the Ministry of Education, Science, Culture, and Sports of the Republic of Armenia. 2025.

Publications and Approbation

The results submitted for defense were obtained personally by the applicant. In the published joint works, the formulation and investigation of the problems were carried out through the joint efforts of the co-authors, with the direct involvement of the applicant. In [1], the author evaluated SIFT, SiamTPN, and TransT object tracking algorithms, conducted UAV flight experiments to collect aerial video data, and developed a unified methodology for combining and comparing tracker performance under real-world conditions. In [2], the author designed and implemented a hyperbolic PDE-based convolutional neural network layer, integrated it into existing deep learning frameworks (ResNet), and performed training and evaluation on the CIFAR dataset to assess performance improvements. In [3], the author carried out a comparative analysis of tracking algorithms, implemented benchmarking experiments, and evaluated their performance on limited-resource hardware. In [4], the author implemented the testing pipeline, developed data augmentation and modification procedures, and conducted comparative experiments on change detection methods. In [5], the author built a UAV platform, developed the main system architecture and onboard code, and conducted flight tests for real-time target localization using a gimbaled laser system. In [6], the author is the sole author of the paper and independently developed the autonomous UAV control method based on camera video, implemented the control algorithms, conducted experimental validation, and prepared the manuscript.

Structure and scope

The work is structured to cover both theoretical foundations and practical applications. **Chapter 1** introduces the research problem, objectives, and scientific novelty. **Chapter 2** reviews related works on visual tracking, UAV control, and ground control systems. **Chapter 3** details the overall system architecture, including modular ROS 2-based software, PX4 integration, and hardware requirements. **Chapter 4** focuses on the perception module, covering object detection and the hybrid tracking pipeline combining YOLO and MixFormerV2/KCF. **Chapter 5** describes UAV control and interception mechanisms, including visual servoing and adaptive PID loops. **Chapter 6** presents the experimental setup, HITL simulations, and real-world flight tests, along with results and performance evaluation. **Chapter 7** discusses conclusions, scientific contributions, and future directions.

2. Review of Related Works

An autonomous UAV interception system integrates visual tracking, flight control, and a ground control interface. Because interception technologies are dual-use, no

open-source end-to-end systems are available; therefore, this work reviews relevant subsystems.

Visual tracking is critical for interception. Early correlation-filter methods lacked robustness, while Siamese network trackers improved performance. Transformer-based trackers now achieve state-of-the-art accuracy but are too computationally demanding for lightweight UAVs. To address this, a **hybrid tracker** combining a transformer-based method with a lightweight correlation filter was adopted.

The control system maps visual target information to interception commands. Advanced control strategies were evaluated but found impractical for deployment. A **PID controller** was selected for its simplicity and reliability, with filtering, anti-windup, and axis-specific tuning to ensure stable interception.

QGroundControl was chosen as the Ground Control Station due to its full integration with PX4, compatibility with MAVLink/ROS 2, and extensibility for autonomous interception missions.

3. Overall architecture

The autonomous UAV interception system proposed in this work is built as a tightly integrated hardware-software platform, combining advanced perception modules, robust control logic, and operator supervision tools into a unified operational framework. The architecture follows a layered design, separating perception, decision-making, and actuation components while maintaining low-latency communication between them.

This modular approach enables visual tracking, flight control, and operator interface to be developed, tested, and optimized independently. At the core of the system is a ROS 2-based distributed software architecture running on onboard embedded hardware, which interfaces directly with the PX4 flight stack through uORB in both simulated (HITL) and real-world deployments.

From a hardware perspective, the system leverages a combination of commercial off-the-shelf UAV platforms, high-performance embedded computers, and standard communication interfaces. This ensures that the architecture remains cost-effective, replicable, and adaptable to different UAV configurations without sacrificing real-time performance.

The hardware system is designed with a modular and platform-independent hardware structure, enabling integration with a wide range of multirotor configurations.

The minimum required components are:

- **Pixhawk-series flight controller** (PX4-compatible) for stabilization, navigation, and motor control;
- **NVIDIA Jetson Nano** or higher onboard computer for real-time vision and control processing;
- **USB or CSI camera** for target detection and tracking.

The onboard computer executes detection, tracking, and interception logic, while the Pixhawk handles flight stabilization and low-level control through uORB communication. This modular architecture allows the system to be deployed on any

standard multirotor UAV with minimal hardware adaptation. The onboard and ground system layouts are illustrated in Figures 1 and 2.

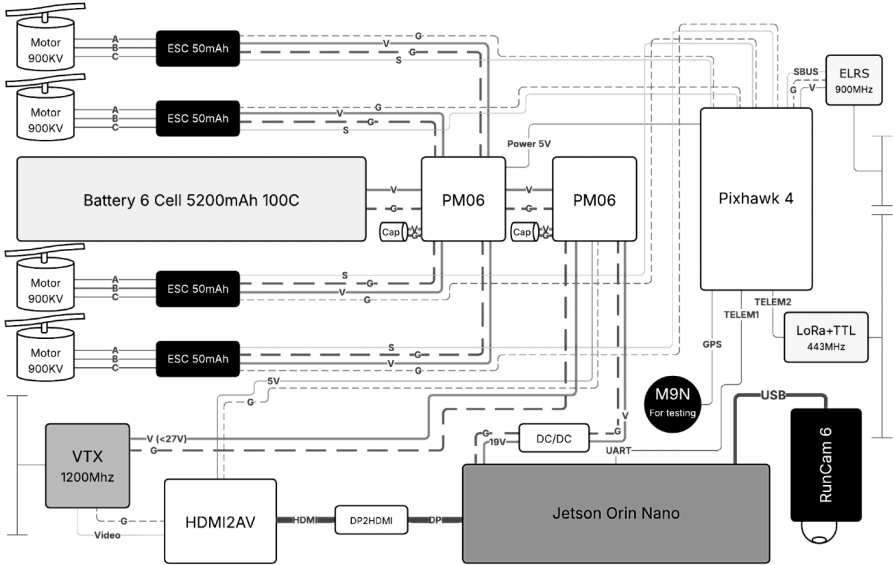


Figure 1: Architecture of the UAV interception platform, including onboard computer (Jetson), flight controller (Pixhawk), camera, power system, and communication interfaces.

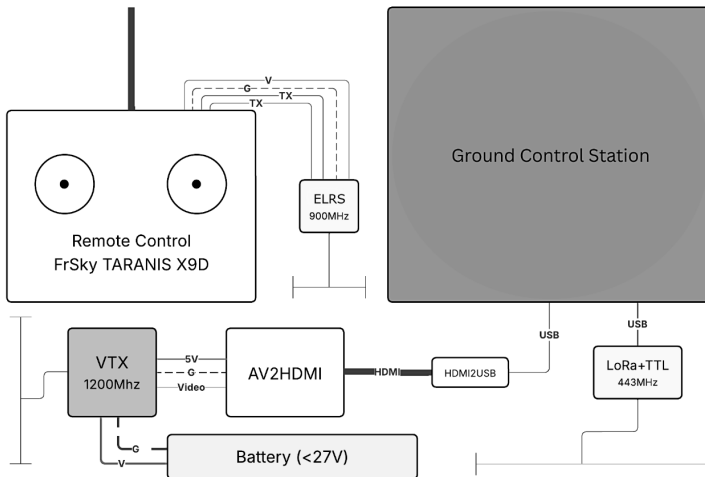


Figure 2: Ground-side system architecture, showing LoRa-based MAVLink telemetry reception, analog video signal conversion chain, and the ground control station.

The software system is built using the ROS 2 Humble middleware. It follows a modular architecture, where each major function (detection, tracking, control) is handled

by an independent node running in its own thread. This design enables real-time parallel execution and flexible system scalability.

A structural overview of the software architecture is shown in Figure 3, where each ROS 2 node is represented by a rectangular block and interconnected via topic-level communication.

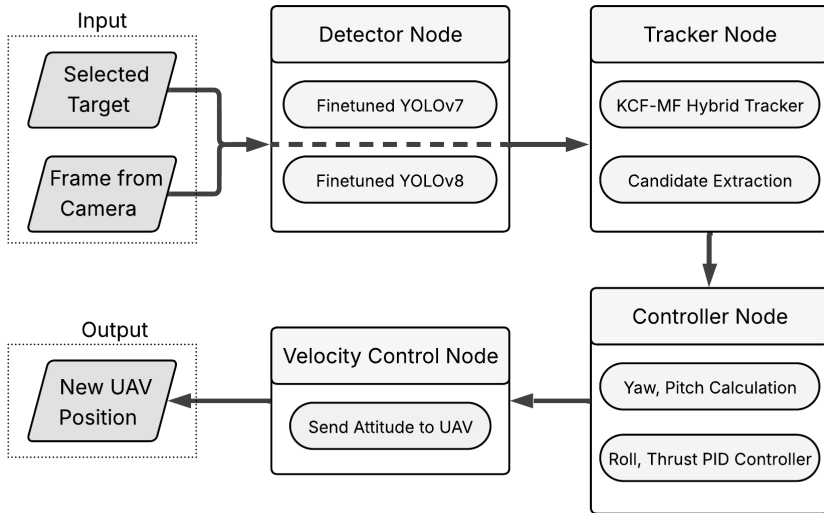


Figure 3: ROS 2 software architecture.

The architecture consists of the following key parts:

Detector Node: Executes a YOLO object detector and publishes bounding boxes with class labels and tolerance metadata to the custom topic.

Tracker Node: Performs continuous target tracking using the selected tracking method and is activated only after target initialization. If the detector is enabled, the target can be initialized by selecting a detected bounding box or by manually defining a custom box. If the detector is disabled, the operator may define a bounding box manually or click on the video frame, in which case a **candidate extraction** module generates candidate regions and selects the one whose center is closest to the click position. The tracker outputs the target's top-left and bottom-right pixel coordinates at each frame.

Candidate Extraction: Generates object candidate bounding boxes around a user-selected image point when no detector output is available. The module crops multiple regions of interest (ROIs) at increasing scales centered on the click position, performs edge-based segmentation using Canny edge detection and morphological filtering, extracts external contours, and converts them into bounding boxes. Each candidate is scored using a weighted combination of shape solidity, edge density (Edge Boxes-inspired score), image sharpness, and distance from the click location. The

highest-scoring candidate across all ROI scales is selected and returned as the initialized target bounding box.

Controller Node: Implements the high-level control logic. It receives tracking data and computes target-relative roll, pitch, yaw, and thrust using the North-East-Down (NED) reference frame and feedback (PID) controllers. These values are passed to the Velocity Control node.

Velocity Control Node: Converts control outputs into PX4 attitude setpoints and publishes them to the uORB topic.

Plotter Node: A debugging and visualization tool that subscribes to controller outputs and system states to produce real-time plots of PID response and tracking performance.

PX4 and uORB Integration

Custom uORB topics were defined for integrating ROS 2 with PX4:

- Used to send target selection coordinates from QGroundControl to the tracker.
- Transmits updated hyperparameters from the GCS to the relevant ROS nodes.

The system uses MicroXRCEAgent to bridge the Jetson's ROS 2 nodes to PX4's uORB messaging system.

4. Visual Perception

The perception module consists of two components: the detector and the tracker. The detector identifies and localizes objects in the video stream, providing initial target information. The tracker maintains the target position during the interception process.

The detection module provides the initial target bounding box for the tracker. At the initial stage of this research, YOLOv7 was selected for its balance between accuracy and real-time performance. After the release of YOLOv8, which represented the state of the art at the time, the module was updated to integrate YOLOv8, benefiting from its improved accuracy and architectural refinements while maintaining high FPS suitable for UAV deployment. Subsequent YOLO release YOLOv9 offered higher accuracy but significantly lower FPS on the embedded hardware¹, making YOLOv8 the preferred choice for maintaining real-time performance.

The detectors were fine-tuned on a combination of Microsoft COCO and VisDrone, selecting only the person and car classes, and a custom AirSim-generated UAV dataset, which was necessary because YOLO's predefined classes do not include UAVs, and tailored data was required for interception scenarios.

The detector is capable of identifying predefined object categories such as vehicles, people, and UAVs. This output not only serves as a direct perception layer for the operator but also initializes the tracker.

The models are converted to TensorRT using FP16 precision to reduce inference latency. Object detection operates asynchronously from the tracking module, publishing bounding boxes to a shared ROS 2 topic. On the GCS screen, the operator can view

¹ Jocher, G., Chaurasia, A., & Qiu, J. (2023). *YOLOv8 vs YOLOv9: A Technical Comparison for Object Detection*. Ultralytics YOLO Docs. Available [Online] <https://docs.ultralytics.com/compare/yolov8-vs-yolov9/>

the detected bounding boxes. The operator can manually select a target by clicking on the desired object on the screen. The algorithm then automatically chooses the detected box whose center is closest to the clicked position, based on Euclidean distance.

The tracking module combines two complementary approaches with different computational characteristics. MixFormerV2-S, a transformer-based tracker, provides high accuracy under occlusion, scale variation, or distractor interference. Its attention mechanism enables context-aware localization but incurs a computational cost of approximately 4.5 GFLOPs per forward pass, which can reduce real-time performance on resource-constrained embedded platforms without hardware-specific optimization. In contrast, the KCF tracker operates at very high frame rates with minimal CPU load and is well-suited for stable-appearance segments of the trajectory.

Adaptive Switching with Perceptual Hashing. To balance accuracy and computational efficiency, an adaptive bidirectional switching mechanism was employed within the frames of this work. At each step:

- ROIs from the current and previous frames are hashed using perceptual hashing (pHash).
- The Hamming distance between hashes is compared to a threshold T .
- If the difference exceeds T , indicating a significant appearance change or potential drift, the system switches to MixFormerV2-S to re-acquire a reliable bounding box.
- If the difference is below T , KCF is used to maintain high-speed tracking.
- When switching from MixFormerV2-S back to KCF, the correlation filter is reinitialized with the most recent high-confidence bounding box.

This ensures that the heavy transformer-based model is activated only when necessary, reducing the average processing time while maintaining high tracking accuracy.

Tracker Evaluation. A two-stage evaluation of tracking algorithms was conducted to determine the most effective approach for real-time UAV interception.

In the first stage, several correlation-filter-based trackers (MOSSE, KCF, CSRT, DSST, ECO) were evaluated on the UAV123 and VisDrone-SOT datasets using a PC equipped with an Intel i7-11th-generation CPU, an NVIDIA RTX 3060 GPU, and 32 GB RAM. This environment enabled rapid prototyping, controlled benchmarking, and extensive parameter tuning, which would have been significantly more difficult and time-consuming if conducted directly on the Jetson platform. Once promising candidates were identified, the same experiments were subsequently replicated on an NVIDIA Jetson Orin Nano to assess performance under embedded hardware constraints.

Implementations were optimized in C++ for maximum performance, with KCF, DSST, and ECO implemented from scratch based on original publications. Evaluation metrics included accuracy, robustness, and frame rate. The results showed that while MOSSE and DSST offered high speed, their accuracy was insufficient for interception tasks. The custom KCF implementation achieved the best balance between accuracy and speed, making it the selected correlation filter for the next stage.

In the second stage, transformer-based trackers were compared, including MixFormerV2, HCAT, and ViT Tracker. Official implementations with CUDA/TensorRT support were used to ensure optimal performance. The evaluation showed that MixFormerV2 provided the best overall accuracy and robustness, although at a high computational cost.

A more detailed analysis of the experimental setup, evaluation methodology, and results is provided in the main text of the thesis.

Table 1. Comparison of the Hybrid tracking method with different thresholds T , with MixFormerV2, KCF, and HCAT on UAV123 and VisDrone-SOT datasets on Jetson Orin Nano.

	UAV123 [CPU/GPU]							VisDrone-SOT [CPU/GPU]						
	% of MF ²	↑A (%)	↑R (%)	↑AUC-A (%)	↑AUC-R (%)	↑DRE (%)	↑FPS	% of MF	↑A (%)	↑R (%)	↑AUC-A (%)	↑AUC-R (%)	↑DRE (%)	↑FPS
KCF	0	53.7	41.8	43.8	34.1	58.1	174	0	60.2	57.5	48.4	46.2	42.5	153
MixFormerV2	100	79.7	80.7	64.1	65	19.3	34	100	68.7	69.6	54.8	55.5	30.4	31
HCAT	0	75.2	76.1	60.7	61.4	23.9	33	0	74.6	74.6	58.3	58.4	25.4	33
KCF-MF (T=2)	91.8	77.8	78.9	63.1	63.6	20.4	46	71.5	67.9	67.4	54.1	54.0	31.2	60
KCF-MF (T=5)	84.1	76.4	77.2	61.3	62	22.8	54	62.2	66.4	66.3	52.3	52.4	33.7	68
KCF-MF (T=10)	72.9	69.5	69.3	55	55	30.6	82	30.4	59.9	58.9	48.1	47.4	41.1	113
KCF-MF (T=15)	28.3	63	62.1	49	48.3	38	110	14.9	54.1	53	44.2	44.1	48	137
KCF-MF (T=20)	16.2	60.8	59.9	46.2	45.5	41.9	128	8.5	52.6	51.8	42.3	42.1	52.7	152

Building on these findings, a hybrid tracking approach was developed that combines the speed of KCF with the robustness of MixFormerV2. Switching between trackers is controlled by a perceptual hashing-based similarity metric, with a tunable threshold determining when MixFormerV2 is used. Threshold values of 5 and 10 were selected for experiments, representing trade-offs between accuracy and speed. Results on PC show that the hybrid method with $T=5$ retains accuracy within 1-2% of MixFormerV2 while increasing FPS by 26-36%.

Experiments on the Jetson Orin Nano platform (Table 1) confirmed the method’s suitability for onboard UAV deployment. While MixFormerV2 operates at ~34 FPS, the hybrid tracker with $T=5$ achieves up to 54-68 FPS with minimal loss in accuracy. Higher thresholds further increase speed, but at the cost of a corresponding drop in accuracy. The hybrid method enables fine-tuning of performance according to mission requirements, making it adaptable for high-speed ground and aerial target interception, where low latency is crucial.

5. Control and Interception Mechanism

The interception behavior of the UAV is governed by a visual servoing architecture that tightly couples object localization in the camera frame with real-time flight control. The control pipeline consists of 3 main stages: **Rotation Vector Calculation**, **Outer Loop** (yaw and pitch calculation), and **Inner Loop** (roll and thrust control).

² % of MF indicates the proportion of frames tracked by MixFormerV2, with the remainder tracked by KCF.

The first stage of the interception mechanism involves transforming image-based target coordinates into 3D angular directions relative to the UAV. This is essential for generating control signals that steer the UAV toward the selected object.

When the tracker identifies a target in the image frame, it provides the center point of the bounding box in pixel coordinates (u, v) . To convert this 2D location into a 3D direction in the UAV's reference frame, a multi-step transformation pipeline is used.

Projection to the Ray in the Camera Frame

Let f_x and f_y be the focal lengths of the camera, with c_x and c_y representing the optical center. The target center pixel (u, v) is projected into a unit direction vector (ray r_c) in the camera's coordinate system using the calibrated pinhole model:

$$r_c = \left[\frac{u-c_x}{f_x} \quad \frac{v-c_y}{f_y} \quad 1 \right]^T \quad (1)$$

Then we can calculate the normalized vector by dividing r_c by its length:

$$\hat{r}_c = \frac{r_c}{\|r_c\|} \quad (2)$$

Transformation to UAV NED Frame

The resulting ray is transformed from the camera frame to the UAV's body-aligned North-East-Down (NED) frame using TF²³ transforms. This transformation accounts for the camera's pose relative to the UAV body, and ensures that all angular calculations are consistent with flight control conventions. We know UAV orientation in space (R_{NED}^{body}) from IMU and camera direction from the UAV's frame (R_{body}^{cam}). So, we can calculate ray coordinates in the UAV's NED system:

$$R_{NED}^{cam} = R_{NED}^{body} \cdot R_{body}^{cam} \in \mathfrak{R}^{3 \times 3} \quad (3)$$

$$\hat{r}_d = R_{NED}^{cam} \cdot \hat{r}_c \in \mathfrak{R}^3 \quad (4)$$

Outer Loop: Yaw and Pitch Calculation

The 3D vector in NED coordinates is normalized (because we only applied rotation on the normalized vector) and used to compute the yaw (ψ) and pitch (θ) angles to the target:

$$\hat{r}_d = [x \ y \ z]^T \quad (5)$$

$$\psi = \text{atan2}(y, x) \quad (6)$$

$$\theta = \text{atan2}(-z, \sqrt{x^2 + y^2}) \quad (7)$$

These angles represent the desired change in UAV orientation required to align with the target direction.

Rotation Process

The resulting yaw (ψ) and pitch (θ) values are fed into a custom rotation

³ <https://docs.ros.org/en/humble/Concepts/Intermediate/About-Tf2.html>

scheduling algorithm proposed in this work, which smoothly adjusts the UAV's orientation over multiple frames. The algorithm operates as follows:

- Limiting the maximum pitch rotation per control cycle to a predefined threshold by decomposing large pitch commands into multiple incremental rotations;
- Deferring yaw adjustment until the previous yaw rotation is completed;
- This estimation loop is executed in real-time and repeated at each control cycle to ensure continuous alignment with the moving target.

These methods are newly developed for this thesis and represent our original contribution to the control of UAV orientation.

While pitch and yaw angles are derived directly from the object's direction vector, roll and thrust commands are generated through feedback-based PID control (inner) loops, which have also been developed as part of this work and represent a new contribution. These loops regulate the UAV's alignment with the target using its position within the camera frame.

Figure 4 illustrates a **discrete PID control loop** that converts pixel error into a corrective control signal (roll or thrust). The proportional path reacts immediately to the current pixel error, while the derivative path uses a smoothed error change to improve stability and reduce noise sensitivity. The integral path accumulates error over time but is conditionally updated to avoid the windup effect when the output is saturated.

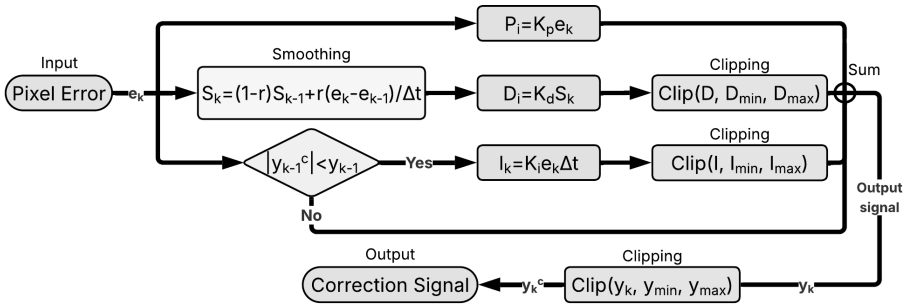


Figure 4. PID control loop

All contributions are clipped to predefined limits and summed to produce a bounded output that drives the system toward minimizing the pixel error. The only distinction between the roll and thrust PID controllers lies in their respective gain coefficients; otherwise, their structure and functionality are identical.

The Roll PID controller minimizes the rotational alignment error, which is the angular deviation between the object's center and the UAV's dynamically rotated horizon line.

The Thrust PID controller minimizes the vertical offset between the expected interception pitch line and the target's projected location.

As shown in Figure 5, the following control relationships are enforced:

- **White lines** represent the camera's vertical axis (line 1) and the dynamically rolled vertical line (line 2).
- **The rectangle and the star mark** the tracked object and its center.
- **The cross** is the projection of the object center onto the rolled vertical line.
- **The central circle** indicates the direction of optical interception.
- **The gray dashed line** (line 3) shows the roll error (minimized by roll PID)
- **The white dot-dashed line** (line 4) shows the thrust error (minimized by thrust PID)
- **The black dotted line** (line 5) is the critical pitch threshold line.

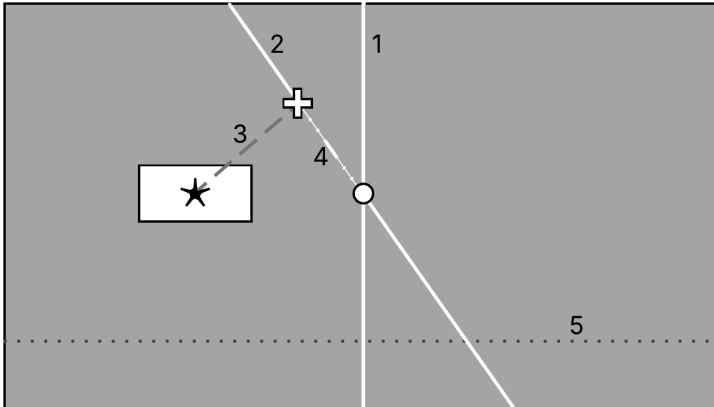


Figure 5. Visual representation of the visual servoing control loop.

In optimal alignment, the star moves toward the cross mark (i.e., the object center aligns with the rolled vertical line), and the cross mark moves toward the central circle (i.e., aligned intercept trajectory). The control system ensures that all three converge, forming a visually stable and geometrically consistent intercept maneuver.

A customized version of QGroundControl (QGC) is integrated into the system to enable effective mission monitoring and in-flight parameter control. Given that the UAV operates on the PX4 autopilot stack, which natively supports MAVLink and ROS 2 middleware, QGC serves as a compatible and extensible interface.

6. Experimental Setup

To validate the proposed UAV interception system, a series of experiments was conducted in both simulated and real-world conditions. The evaluation process was structured to isolate and test each subsystem under controlled conditions before integrating them into a complete interception loop.

Testing was carried out in two stages:

1. **HITL simulation** - assessing the complete interception control loop in Gazebo Classic with 2496 distinct test configurations, covering varying target speeds, approach angles, and visibility conditions.

2. **Real-world flight tests** - verifying system performance with actual UAVs in diverse environmental and operational scenarios.

Each stage was designed to provide both quantitative and qualitative metrics, allowing for the iterative refinement of algorithms and parameters.

HITL simulation was used to evaluate the interception pipeline under controlled yet realistic conditions, leveraging a framework built in Gazebo Classic with the PX4 flight stack. This setup enabled the testing of the full vision-control loop with accurate flight dynamics, real-time camera streaming, and onboard computation identical to that in field deployment.

The simulation world consisted of a flat ground plane and a single SUV model acting as the moving target. The UAV model was a PX4-compatible quadrotor connected to a Pixhawk flight controller in HITL mode. The camera feed from Gazebo was streamed through an HDMI-to-USB converter to the onboard computer, mimicking real-world video capture latency. ROS 2 nodes ran on the onboard computer, processing frames in real time and issuing control commands back to PX4.

Scenario Parameters

A total of 2496 distinct scenarios were generated by varying the following parameters:

- **Target velocity (m/s):** [0, 15]
- **Initial UAV-to-target horizontal distance (m):** [2, 100]
- **Initial UAV yaw angle (°):** [-25.8, 25.8]
- **Target motion direction relative to the UAV:** {toward, away, perpendicular, diagonal relative to the UAV}
- **UAV altitude (m):** [18, 200]
- **Wind (m/s):** [0, 5]

Each scenario included a predefined bounding box for tracker initialization to ensure reproducibility. All scenarios required the target to be visible at the start.

Each scenario was evaluated using the following **success criteria**:

- The interception is successful if the UAV makes contact with the SUV within 30 seconds.
- The test is considered a failure if the UAV hits the ground or exceeds the time limit.
- After each outcome, the pipeline restarts with the next configuration.

Simulation Results

Performance across target speed categories is summarized in Table 2. The results indicate that the system achieved performance for stationary and low-speed targets.

Failures occurred primarily in high-speed scenarios (15m/s) when the SUV was moving directly toward the camera. In these cases, rapid apparent size growth and high optical flow made stabilization and trajectory prediction more challenging, occasionally causing late reactions or ground impact. A more detailed analysis of these experimental scenarios and their outcomes is presented in the main text of the thesis.

Table 2: Interception performance by target speed in HITL.

Target Speed (m/s)	Success (%)	Avg. Time (s)	Ground Impacts	Timeouts
0	100	8	0	0
5	98	13.1	154	3
10	93	16.4	392	141
15	78	21.3	799	893

HITL simulation behavior closely matched results from real flights⁴. The framework proved effective for iterative tuning of tracking, control, and mission logic before field deployment.

Real-world evaluation was conducted to validate the proposed autonomous interception system under practical operational conditions. The primary test platform was the Reptile X500 multirotor, with additional verification flights performed on Holybro S500 and Holybro X500 frames. Onboard processing was handled by NVIDIA Jetson Nano, Jetson Xavier NX, and Jetson Orin Nano modules, with live video input provided via both USB and CSI cameras.

Flights were performed across a wide range of environmental and mission configurations, including:

- Varying initial yaw, pitch, and roll orientations of the UAV.
- Stationary and moving targets.
- Both ground-based and airborne targets.
- Short and long-range engagements, with varying visibility and occlusion conditions.
- Light to moderate wind, with occasional gusts.

Table 3: Real-world results: interception success rate by target speed.

Target Speed (m/s)	Number of Scenarios	Success (%)	Avg. Time (s)	Ground Impacts	Timeouts
0	70	95.7	5	2	1
5	36	92.7	12	2	1
10	24	83.3	17	2	2
15	20	75	30	3	2

In total, approximately **150 flights** were conducted, encompassing a diverse mix of test cases. For each trial, system logs and onboard video recordings were collected for offline debugging, parameter tuning, and qualitative performance evaluation. A more detailed analysis of real-world experiments is presented in the main text of the thesis.

Results Overview

The system successfully intercepted moving ground targets (Table 3) in the majority of test flights. Overall behavior in real-world conditions closely matched trends observed in the HITL simulation stage.

⁴ Rafael Perez-Segui et al., "Bridging the Gap between Simulation and Real Autonomous UAV Flights in Industrial Applications," *Aerospace* 10, no. 9 (2023): 814, <https://doi.org/10.3390/aerospace10090814>

Environmental Factors

Wind gusts occasionally introduced disturbances into the control loop, causing minor deviations in vehicle motion. Also, direct sunlight striking the camera lens sometimes created glare or lens flares, which could temporarily reduce image clarity and affect visual tracking. Despite these influences, the overall system performance remained stable, and no critical mission failures occurred during the test campaign.

7. Discussion and Conclusion

In conclusion, the objectives and novel contributions outlined in this thesis were successfully realized:

- **A hardware-agnostic UAV interception system** was designed and implemented, demonstrating compatibility with multirotor platforms equipped with standard flight controllers and onboard computing systems.
- **A real-time hybrid tracking** pipeline was developed and deployed on embedded UAV hardware. The system operated reliably under onboard computational constraints and supported operator-defined arbitrary targets, enabling flexible and practical interception scenarios.
- **A PID control system** incorporating yaw and pitch guidance, independent roll and thrust control, and real-time in-flight parameter tuning via MAVLink was designed and validated.
- **A comprehensive evaluation system** was established that integrates HITL simulations with diverse target dynamics and real-world field experiments. The results confirmed the robustness, effectiveness, and real-world applicability of the proposed UAV interception system under challenging environmental conditions.

This research is planned to be extended into a swarm-based UAV interception project, coordinating multiple autonomous interceptors for simultaneous target engagement and improved operational efficiency. This work will build directly upon the single-UAV system developed in this thesis, representing the next phase of practical deployment and algorithmic development.

List of the author's publications

[1] V. Melkonyan, V. Sahakyan, L. Kirakosyan, O. Hovhannisyan. "**Comparison of Single Object Tracking Algorithms on Video Sequences Captured from UAV.**" Vestnik of the Russian-Armenian University (2022): 67-75.

Contribution: Configured SIFT, SiamTPN, and TransT object tracking algorithms, conducted UAV flight experiments to collect aerial video data, and developed a unified methodology for combining and comparing tracker performance under real-world conditions.

[2] V. Sahakyan, V. Melkonyan, G. Gharagozyan, A. Avetisyan. "**Enhancing Image Recognition with Pre-Defined Convolutional Layers Based on PDEs.**" Programming and Computer Software. Vol. 49. No. 3. 2023: 192-197.

Contribution: Designed and implemented a hyperbolic PDE-based convolutional neural network layer, integrated it into existing deep learning frameworks (ResNet), and performed training and evaluation on the CIFAR dataset to assess performance improvements.

[3] A. Sardaryan, V. Sahakyan, V. Melkonyan, S. Sargsyan. "**An Accurate Real-Time Object Tracking Method for Resource Constrained Devices.**" Proceedings of the Institute for System Programming of the RAS. Vol. 36, issue 3. 2024: 283-294.

Contribution: Conducted the comparative analysis of tracking algorithms, implemented benchmarking experiments, and evaluated their performance on limited-resource hardware.

[4] A. Fahradyan, T. Baghdasaryan, V. Melkonyan, V. Sahakyan, L. Kirakosyan, O. Hovhannisyan, S. Sargsyan, A. Darbinyan. "**Comparing and improving change detection methods.**" Vestnik of the Russian-Armenian University (2024): 37-43.

Contribution: Developed the evaluation pipeline and data augmentation procedures for comparative change detection experiments.

[5] V. Sahakyan, V. Melkonyan, S. Sargsyan. "**Локализация целей в реальном времени на БПЛА с лазерным дальномером на подвесе.**" Proceedings of the Institute for System Programming of the RAS. Vol. 37, issue 4, part 1. 2025: 189-198.

Contribution: Built the UAV, developed the main system architecture and code, and conducted flight tests.

[6] V. Melkonyan. "**Autonomous UAV Control Based on Camera Video.**" Vestnik of the Russian-Armenian University (2025): 17-25.

Contribution: Sole author; designed and implemented the autonomous control system, conducted experiments, and prepared the manuscript.

Ամփոփում
Վահագն Գևորգի Մելքոնյան
Թիրախին հետևող ԱԹՍ ավտոպիլոտի մշակում համակարգչային տեսողության կիրառմամբ

Տվյալ ատենախոսության նպատակն է մշակել ինքնավար ԱԹՍ հարվածողական համակարգ, որն ունակ է իրական ժամանակում հայտնաբերել, հետևել և խոցել օդում կամ գետնի վրա շարժվող և անշարժ թիրախների: Առաջարկվող համակարգը նախատեսված է ապահովելու բարձր ճշգրտություն, կառավարման կայունություն և օպերատորի հետ փոխազդեցություն դինամիկ միջավայրերում:

Հետազատության շրջանակներում իրականացվել են հետևյալ հիմնական աշխատանքները.

- **Վերլուծել և գնահատել ԱԹՍ-ների տեսողական հետևման և խոցող առկա համակարգերը՝** հաշվի առնելով հաշվարկային արդյունավետությունը, կայունությունը և սահմանափակ հաշվարկային ռեսուրսներ ունեցող հարթակներում ներդրման կիրառելիությունը:
- **Նախագծել և մշակել մոդուլային ճարտարապետություն,** որը միավորում է ընկալումը, հետևումը և կառավարումը մեկ միասնական համակարգում՝ ապահովելով իրական ժամանակում աշխատանք on-board հաշվարկային հարթակների վրա:
- **Նախագծել և մշակել օբյեկտի հետևման թեթև և կայուն մոդուլ,** որը ապահովում է on-board սարքավորումների վրա (մինչև 1,280 GFLOPS) իրական ժամանակում աշխատանք տարբեր միջավայրային պայմաններում և թիրախների վարքագծի փոփոխությունների դեպքում՝ ինչպես նախապես սահմանված օբյեկտների դասերի համար, այնպես էլ class-agnostic օբյեկտների համար, ինչի շնորհիվ էլ հնարավոր է դառնում առաքելության ընթացքում դինամիկ թիրախ ընտրելը:
- **Նախագծել և մշակել ինքնավար խոցող կայուն կառավարման ալգորիթմ,** որը ճշգրիտ ուղղորդում է ԱԹՍ-ն դեպի թիրախ՝ պահպանելով թռիչքի կայունությունը և ներառելով ուղղության գնահատում, թիրախին մոտեցման տրամաբանություն և ուղղիչ կառավարման մեխանիզմներ, որոնք ադապտացվում են տարբեր թիրախների արագություններին, մոտեցման անկյուններին և գործառնական միջավայրերին:
- **Նախագծել և մշակել իրական ժամանակում գործող գետնային կառավարման և մոնիթորինգի համակարգ՝** ընդլայնված Ground Control Station (GCS) միջերեսով, որը ներառում է իրական ժամանակում վիդեո ցուցադրում, ինտուիտիվ թիրախի ընտրություն, թռիչքի ընթացքում պարամետրերի կարգավորում, օպերատորի արագ միջամտության հնարավորություններ և հստակ տեսողական հետադարձ կապ:
- **Նախագծել և իրականացնել ավտոմատացված Hardware-in-the-Loop (HITL) սիմուլյացիա՝** ԱԹՍ-ի տեսողական հետևման, ուղղորդման և կառավարման ալգորիթմների համակարգային փորձարկման և

վավերացման համար՝ տարբեր թիրախային վարքագծերի, միջավայրային պայմանների և առաքելության սցենարների դեպքում:

Աշխատանքի գիտական նորույթը բնութագրվում է հետևյալ հիմնական արդյունքներով.

- **Մշակվել է խոցող ԱԹՍ-ի ճարտարապետություն,** որը կարող է կիրառվել բազմառոտորային ԱԹՍ հարթակների վրա՝ համալրված թռիչքային կարգավորիչներով և onboard հաշվարկային համակարգերով:
- **Մշակվել է իրական ժամանակում գործող հիբրիդային հետևման համակարգ,** որը նախատեսված է ԱԹՍ-ի ներկառուցված հաշվարկային սարքավորումների համար և հնարավորություն է տալիս հետևել օպերատորի կողմից ընտրված կամայական թիրախի:
- **Մշակվել է ՀԻԴ կառավարման համակարգ՝** roll և thrust կառավարելու համար, ինչպես նաև իրականացվել է yaw և pitch անկյունների անկախ հաշվարկ՝ թռիչքի ընթացքում իրական ժամանակում պարամետրերի կարգավորման հնարավորությամբ:
- **Մշակվել է գնահատման համակարգ,** որը միավորում է HITL սիմուլյացիաները տարբեր թիրախների դինամիկայով և իրական դաշտային փորձարկումները:

Заключение

Мелконян Ваагн Геворкович

Разработка автопилота БПЛА для сопровождения целей с применением компьютерного зрения

Целью данной диссертации является разработка автономной поражающей системы БПЛА, способной в реальном времени обнаруживать, сопровождать и поражать движущиеся и неподвижные цели как в воздухе, так и на земле. Предлагаемая система предназначена для обеспечения высокой точности, устойчивости управления и взаимодействия с оператором в динамических условиях.

В рамках работы реализованы следующие основные направления:

- **Провести анализ и оценку существующих систем визуального сопровождения и перехвата БПЛА** с точки зрения вычислительной эффективности, устойчивости и возможности их развертывания на платформах с ограниченными вычислительными ресурсами.
- **Спроектировать и разработать модульную архитектуру,** объединяющую восприятие, сопровождение и управление в единую систему, способную работать в реальном времени на бортовых вычислительных платформах.
- **Спроектировать и разработать легковесный и устойчивый модуль визуального сопровождения,** обеспечивающий стабильную работу в реальном времени на бортовом оборудовании (с вычислительной мощностью менее 1,280 GFLOPS) при различных условиях окружающей

среды и поведении целей, поддерживающий как заранее определенные классы объектов, так и class-agnostic сопровождение для динамического выбора цели во время миссии.

- **Спроектировать и разработать устойчивый алгоритм управления автономным перехватом**, который точно направляет БПЛА к движущимся целям, сохраняя устойчивость полета и включая оценку направления, логику сближения с целью и корректирующее управление, адаптируемое к различным скоростям целей, углам сближения и условиям эксплуатации.
- **Спроектировать и разработать систему управления и мониторинга в реальном времени** с расширенным интерфейсом Ground Control Station (GCS), включающим видео в реальном времени, интуитивный выбор цели, настройку параметров во время полета, возможность быстрого вмешательства оператора и наглядную визуальную обратную связь.
- **Спроектировать и реализовать автоматизированную среду моделирования Hardware-in-the-Loop (HITL)** для систематического тестирования и валидации алгоритмов визуального сопровождения, наведения и управления БПЛА при различных сценариях поведения целей, условиях окружающей среды и типах миссий.

Следующие результаты характеризуют **научную новизну работы**:

- **Разработана архитектура поражающего БПЛА**, которая может быть реализована на мультикоптерных платформах, оснащенных полетными контроллерами и бортовыми вычислительными системами.
- **Разработан гибридная система отслеживания в реальном времени**, предназначенный для встроенного оборудования БПЛА и способный отслеживать произвольные цели, заданные оператором.
- **Разработана система управления ПИД** для управления креном и тягой, а также независимого расчета углов рыскания и тангажа с возможностью корректировки параметров в реальном времени во время полета.
- **Разработана система оценки**, объединяющая HITL-симуляции с различной динамикой целей и реальные полевые эксперименты.

