Design of an Image-Based Visual Servoing System for Autonomous Quadcopter Interception

Thesis Report

by

Clemente van der Aa

to obtain the degree of Master of Science
at the Delft University of Technology

to be defended publicly on December 19, 2024

Thesis committee:	Dr. J. Alonso-Mora
	Dr. C. Pek
Company supervisor:	Ir. R. Vermeer
Place:	Faculty of Mechanical Engineering, Delft
Project duration:	April, 2024 – December, 2024
Student number:	4690451
Company:	Avalor AI

Contents

Nomenclature

List of Abbreviations

Acronym	Definition
UAV	Unmanned Aerial Vehicle
IBVS	Image-Based Visual Servoing
IMU	Inertial Measurement Unit
FOV	Field of View
PID	Proportional-Integral-Derivative
SITL	Simulation-in-the-Loop
CEP	Circular Error Probable
VTOL	Vertical Take-Off and Landing
PBVS	Position-Based Visual Servoing
YOLOv8n	You Only Look Once (version 8 Nano)
CSRT	Discriminative Correlation Filter with Channel and Spatial Reliability
Kalman Filter	A mathematical method for sensor fusion and state estimation
YOLO	You Only Look Once (object detection algorithm)
ROS	Robot Operating System
PX4	PX4 Autopilot
Ardupilot	ArduPilot
OpenCV	Open Source Computer Vision Library
KCF	Kernelized Correlation Filters
MOOSE	Minimum Output Sum of Squared Error
Gazebo	Gazebo-classic Simulator
MAVLink	Micro Air Vehicle Link
MAVROS	MAVLink Extendable Communication Node for ROS
RTSP	Return to Starting Position
USB	Universal Serial Bus
RC	Remote Control
LOS	Line of Sight
CV	Constant Velocity
CA	Constant Acceleration
SM	Sinusoidal Maneuver
RPi 4B	Raspberry Pi 4 Model B
Jetson Nano	Nvidia Jetson Nano
Jetson Xavier NX	Nvidia Jetson Xavier NX
RL	Reinforcement Learning
DR	Domain Randomization
TL	Transfer Learning
MobileNet	MobileNet (a lightweight convolutional neural network architecture)

List of Symbols

Symbol	Description	Unit
r	Cartesian distance between pursuer and target in the world frame	m
x_t, y_t, z_t	Positions of the target in the world frame	m
x_p, y_p, z_p	Positions of the pursuer in the world frame	m
$\mathbf{e}$	Image error vector representing displacement from image center	
u, v	Horizontal and vertical pixel coordinates relative to image center	
$\dot{r}$	Time derivative of the Cartesian distance r	m/s
T	Time of interception	s
$\mathbf{v}_t, \mathbf{v}_p$	Velocity vectors of the target and pursuer	m/s
m	Mass of the quadcopter	kg
g	Gravitational constant	$\mathrm{m/s^2}$
$\mathbf{n}_z$	Unit vector in the inertial frame along the z-axis	
R_{BI}	Transformation matrix from body to inertial frame	
ϕ, θ, ψ	Roll, pitch, and yaw angles respectively	rad
F_i	Upward thrust produced by motor i	N
τ	Torque	$\mathrm{N \cdot m}$
J_x, J_y, J_z	Moments of inertia about the x, y, and z axes	$\mathrm{kg \cdot m^2}$
l	Distance from the rotors to the center of mass of the quadcopter	m
u_1, u_2, u_3, u_4	Control inputs defined for thrust and torques	
$\ddot{x}, \ddot{y}, \ddot{z}$	Second derivatives of positions (accelerations)	$\mathrm{m/s^2}$
$\dot{\phi}, \dot{\theta}, \dot{\psi}$	Time derivatives of roll, pitch, and yaw angles	rad/s
$\mathbf{P}$	Point in the camera frame	m
f	Focal length of the camera	m
$\dot{\mathbf{r}}_\mathbf{B}$	Spatial velocity vector in the body frame	
$\mathbf{J}_p$	Jacobian matrix relating camera velocity to image shifts	
$K_p^{\mathrm{yaw}}, K_i^{\mathrm{yaw}}, K_d^{\mathrm{yaw}}$	PID gains for yaw rate control	
ω_{yaw}	Yaw rate control input	rad/s
$K_p^{\mathrm{v}}, K_i^{\mathrm{v}}, K_d^{\mathrm{v}}$	PID gains for vertical velocity control	
v_z	Vertical velocity control input	m/s
$K_p^{\mathrm{x}}, K_i^{\mathrm{x}}, K_d^{\mathrm{x}}$	PID gains for forward velocity control	
$v_x^{\mathrm{target}}(t)$	Target forward velocity	m/s
$a_{\mathrm{desired}}(t)$	Desired acceleration for forward velocity	$\mathrm{m/s^2}$
$a_{\mathrm{limited}}(t)$	Limited acceleration after clipping	$\mathrm{m/s^2}$
$v_x(t + \Delta t)$	Forward velocity at the next time step	m/s
e_{d}	Distance error for interception	m
d_{hit}	User-defined parameter to ensure non-zero hitting velocity	m
$d_{\mathrm{estimated}}$	Estimated distance between drone and target	m
x	Normalized bounding box width	
T_{virtual}	Transformation matrix for the virtual plane projection	
e_u, e_v	Image errors along the u-axis and v-axis	
$e_u^{\mathrm{virtual}}, e_v^{\mathrm{virtual}}$	Projected image errors on the virtual plane	

Design of an Image-Based Visual Servoing System for Autonomous Quadcopter Interception

Abstract

Unmanned aerial vehicles (UAVs), particularly quadcopters, are increasingly employed in diverse applications due to their manoeuvrability and affordability. However, their susceptibility to GPS jamming and the need for skilled operators limit their operational resilience. This paper presents a new design of an Image-Based Visual Servoing (IBVS) control system for autonomous quadcopter interception, utilizing only a monocular camera and an Inertial Measurement Unit (IMU). The system features a Multi-Axis PID Controller with acceleration limiting and a virtual plane projection method to decouple pitch and vertical motions, thereby enhancing interception accuracy and maintaining target visibility within the drone's field of view (FOV). Furthermore, a perception module is designed to run Yolo-v8n and CSRT in parallel, followed by a Kalman filter, to deliver an accurate and robust representation of the target within the image, operating at 18 frames per second in Simulation-in-the-Loop (SITL) environments. Comprehensive SITL experiments and comparative analyses against recent IBVS algorithms demonstrate that the proposed proposed system achieves a 19% reduction in circular error probable (CEP) for static targets and superior performance in diverse scenarios involving moving targets. These findings validate the effectiveness of the proposed IBVS approach, offering a reliable and scalable solution for autonomous UAV interception in GPS-denied environments with potential applications in security, surveillance, and conflict zones.

Index Terms - Quadcopters, Interception, Autonomous Navigation, Vision-Based Control, Image-Based Visual Servoing, Perception

I Introduction

A Context

The proliferation of unmanned aerial vehicles (UAVs), particularly quadcopters, has transformed industries ranging from logistics to defence due to their vertical takeoff and landing (VTOL) capabilities, high manoeuvrability, and relatively low cost [1]. However, these benefits also introduce risks, as unauthorized drones increasingly intrude into restricted airspaces such as airports, military zones, and critical infrastructure [2]. These threats range from negligent amateur operators to deliberate incursions with malicious intent. Existing countermeasures, including electronic jamming systems and kinetic solutions like lasers or turrets, face significant limitations, including collateral damage, regulatory restrictions, and high costs [3].

In this context, quadcopters have emerged as a promising platform for interception due to their unique ability to match the speed and agility of rogue drones. Unlike stationary or ground-based systems, interceptor drones can operate within the same aerial domain as their targets, offering longer range and greater precision while minimizing unintended disruptions to lawful air traffic. Furthermore, their deployment is cost-effective compared to traditional military-grade systems [3], presenting the potential to serve as a scalable solution for securing sensitive airspaces [4].

Autonomy is critical for the effectiveness of quadcopter-based interception systems. Reliance on skilled operators limits scalability and response times. Furthermore, the radio signals emitted during manual operation makes these systems detectable and susceptible to targeting [5]. Autonomous systems alleviate this constraint and enhance resilience to electronic warfare, particularly GPS jamming, which remains a prevalent threat [6]. However, achieving autonomy must balance functionality with cost constraints. Advanced sensors, such as LiDAR or radar, significantly increase system costs, eroding the economic viability of small UAVs. Thus, developing interception solutions that operate with minimal sensor setups, specifically a monocular camera and an inertial measurement unit (IMU), is essential for maintaining affordability and operational efficiency [7].

B Image-Based Visual Servoing for Interception

Vision-based quadcopter interception systems face significant challenges in precise interception of non-cooperative flight targets. During interception, the observable field of view (FOV) is susceptible to the attitude of the quadcopter, making it challenging to maintain the 2D visibility required for IBVS. Interception accuracy is further affected by target manoeuvres, image processing delay, quadcopter dynamics delay, and guidance strategy. Traditional interception methods often rely on the assumption that both the target and pursuing drone's states are known or can be accurately predicted within controlled environments [8]. However, in real-world scenarios, targets may be passive, uncooperative, or actively adversarial, exhibiting unpredictable trajectories and unknown geometries. Moreover, external sensory data can be unreliable or unavailable in dynamic and often contested settings.

These challenges have driven interest in Image-Based Visual Servoing (IBVS), a technique that leverages visual data directly within the drone's control system. Unlike conventional methods that require precise state estimation in Cartesian coordinates, IBVS operates in image space, utilizing the error between current and desired image features to generate control commands [9]. Practically, this means that image coordinates from object detections are used as inputs for the interception control algorithm, producing motor commands that guide the drone toward the target. The approach reduces dependence on precise state estimation, enhancing robustness to calibration errors. It also improves computational efficiency, which is crucial for real-time operation on low-cost, resource-constrained platforms [10], [11]. IBVS's reliance on minimal sensory input makes it especially suitable for low-cost drones with limited onboard resources, thereby its principle is used for the design of the interception system of this thesis.

C Design Objective

The design objective of this thesis is formulated as follows:

Design Objective

To develop a low-cost IBVS control system to enable a quadcopter to intercept a target using only a monocular camera and an IMU.

This objective drives the development of an interception system capable of achieving high-speed, accurate interception while maintaining computational efficiency on resource-constrained hardware. The system must perform under various conditions, including dynamic target trajectories and challenging field-of-view (FOV) constraints, to validate its practical viability.

D Design Requirements

To achieve the design objective, the following requirements are defined along three dimensions:

(1) **Task**: The system must intercept adversary drones with non-zero hitting velocity. It assumes the target is already in the FOV. Search strategies or take-down methods, such as grippers or net guns, are beyond the scope of this thesis.
(2) **Hardware**: The quadcopter platform must be a small, lightweight drone with an X-shaped rotor configuration. Sensors include a rigidly mounted forward-facing monocular camera and an IMU. A companion computer must handle onboard computations for autonomy.
(3) **Performance**: The quadcopter must achieve a target interception success rate of at least 90% in controlled experiments. The system must operate with a frame rate of at least 20 fps and maintain a control latency below 50 ms to enable stable, agile flight. It should consistently intercept targets moving along curved trajectories at speeds of up to 10 m/s in various environmental conditions, including limited lighting and moderate wind disturbances (up to 5 m/s).

These requirements set clear boundaries for the design while emphasizing the need for simplicity and cost-effectiveness, ensuring the solution remains practical for real-world applications.

E Contributions and Approach

This thesis makes the following contributions:

(1) **Control System**: Development of a multi-axis PID control system incorporating acceleration limiting to ensure smooth and agile interception performance.
(2) **Virtual Plane Mechanism**: Introduction of a virtual plane mechanism to decouple pitch and vertical motion, enhancing field-of-view maintenance and improving control stability during high-speed maneuvers.
(3) **Perception Module**: Implementation of a perception module that combines lightweight object detection, tracking, and state estimation using a Kalman filter for reliable visual feedback under real-time constraints.

The contributions of this thesis were validated through a series of structured experiments designed to test and refine the proposed system. Firstly, a benchmark comparison was conducted to evaluate the performance of the system against existing methods in both static and dynamic interception scenarios. These tests analyzed metrics such as interception accuracy, response time, and system stability, showcasing the improvements achieved by the proposed approach. Next, an ablation study specifically examined the impact of the virtual plane mechanism. By comparing interception scenarios with and without the virtual plane, the study quantified its effectiveness in enhancing control stability and maintaining the target within the field of view during high-speed manoeuvres. Finally, the system was fully integrated into a real quadcopter platform, and preliminary flight tests were conducted. Although comprehensive testing of the IBVS algorithm was not completed within the scope of this thesis, first steps in bridging the sim-to-real gap were done, preparing the setup for extensive real-world test flights in future research.

By integrating these contributions and systematically validating them through rigorous experiments, this thesis demonstrates the feasibility of low-cost, camera-based interception systems that do not rely on GPS or other external aids. The findings advance the applicability of IBVS for autonomous quadcopters operating in contested or resource-constrained environments.

F Report Structure

The thesis is structured as follows: section **2** provides an overview of **related work**, examining alternative visual servo algorithms and their respective considerations. Section **3** presents the **problem statement**, defining the specific challenges and objectives that guided the development of the IBVS system. Section **4** outlines the **methodology** and design decisions made during the system's development. Section 5 describes the **system architecture**, specifying the hardware and software configurations employed. Section **6** presents the **results** from benchmark comparison, ablation study and real-world flight tests. Finally, Section **8** concludes the thesis with a **conclusion** of key findings and their implications for future research.

II Related Work

This section provides an overview of visual servoing methods employed in quadcopter interception tasks, categorized into three primary approaches identified in the literature: localization-based, learning-based, and image-based methods. Image-based methods are further subdivided into model-based and model-free approaches. The goal is to explain the trade-offs in using the methods in different settings and why in this paper the IBVS model-free method was adopted. The categories and their pipelines are illustrated in Fig. 1.

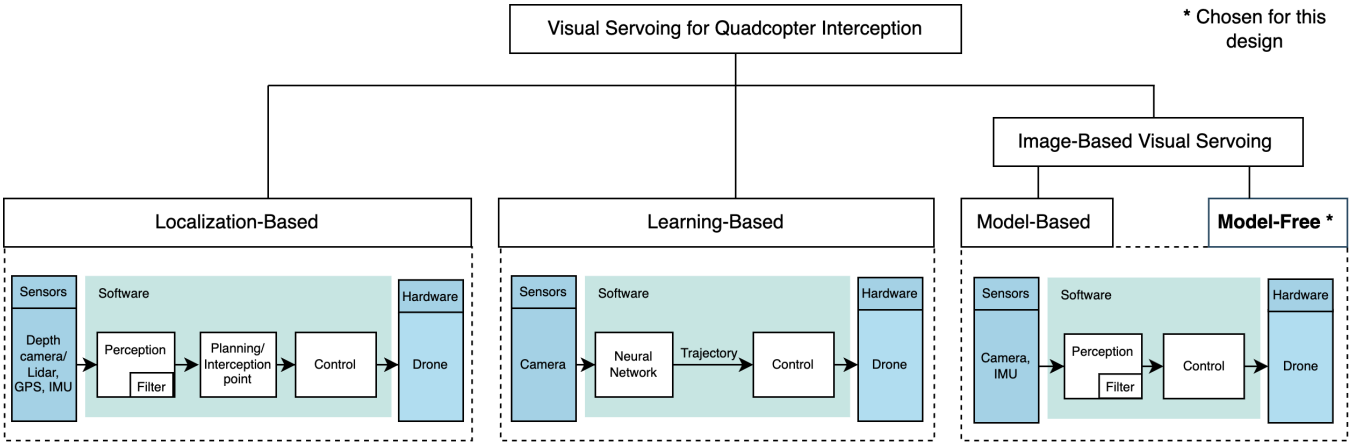

Figure 1: Different types of Visual Servoing algorithms and their pipeline

This figure shows the different types of visual servoing methods for quadcopter interception and how they are categorized. The general pipeline is visualized below each method. This pipeline is the same for model-based IBVS and model-free IBVS hence their pipelines are visualized in the same block.

A Localization-Based Methods

Localization-based methods, also referred to as Position-Based Visual Servoing (PBVS), concentrate on determining the target's position and orientation within Cartesian space utilizing visual information. By accurately estimating the target's three-dimensional position, PBVS facilitates the calculation of precise interception trajectories, which is critical for the successful interception by quadcopters [9]. For instance, in [12], a nonlinear Model Predictive Controller is implemented to intercept a target drone based on PBVS. This controller computes optimal control actions by considering both current and future trajectory points within its prediction horizon, as well as the UAV's constraints, to guide it towards the target. Another PBVS approach for quadcopter interception presented by [13] involves the design of an attitude rate loop controller, which aims to align the line-of-sight vector with the UAV's velocity vector to achieve effective target locking and interception. In [14], the target positions are estimated in three dimensions, and the target's trajectory is approximated using a Bernoulli lemniscate, from which an interception point on a straight segment of the target's path is selected. Additionally, [15] implements a double closed-loop PID controller that utilizes the three-dimensional estimated states of the target to track ground or aerial targets.

In summary, a primary advantage of PBVS is its capability to predict the trajectory and compute interception points. Nevertheless, the effectiveness of PBVS depends on accurate depth estimation and reliable camera calibration. Moreover, all the aforementioned methods incorporate additional sensors such as radar or depth/stereo cameras.

B Learning-Based Methods

Learning-based approaches involve utilizing neural networks to predict control commands directly from visual inputs, bypassing traditional perception and control modules [16]. Techniques such as reinforcement learning and supervised learning have demonstrated robustness against sensor noise and adaptability in diverse environments [17]. For instance, drones equipped with neural networks trained to autonomously follow trajectories can adapt to dynamic changes in real-time, providing exceptional flexibility for interception tasks [18].

Despite their potential, the practical challenges of real-world implementation are substantial. Learning-based methods require extensive training data, which is time-consuming and could be limited to their trained scenarios [19]. Additionally, learning-based approaches face notable limitations beyond computational demands, specifically the sim-to-real gap and lack of generalizability [20]. The sim-to-real gap refers to the difficulty in applying models trained in simulated environments directly to real-world scenarios, as simulations often fail to capture all the nuances of real-world dynamics. Furthermore, the generalizability of these models remains a challenge, as they typically require vast datasets that encompass diverse conditions to perform reliably across varying environments [21].

Due to these constraints, while promising, learning-based methods lack practicality for real-time, low-cost interception applications, where adaptability and robustness are paramount.

C Image-Based Visual Servoing

In contrast to PBVS, Image-Based Visual Servoing (IBVS) leverages visual data directly within the drone's control system, utilizing the error between current and desired image features to generate control commands [9]. IBVS methods are preferred in this paper due to PBVS's susceptibility to calibration errors and its reliance on highly accurate perception modules, which can be challenging to maintain in dynamic and resource-constrained environments. IBVS can be further categorized based on whether the control strategy incorporates the dynamical model of the quadcopter. Approaches that integrate the dynamical model are referred to as model-based, while those that do not are classified as model-free.

2.3.1 Model-Based IBVS Model-based IBVS methods explicitly incorporate the quadcopter's dynamic model and image-plane kinematics into the control design, enabling predictive and agile responses to target motion. Such approaches typically generate low-level commands (e.g., thrust and angular rates) by leveraging known system dynamics. For instance, Liu et al. [22] present a framework ensuring stability via Lyapunov-based analysis. Yang et al. [23] employ separate controllers for different axes, improving accuracy by directly accounting for motion dynamics in the control loop. Zhang et al. [11] propose gain-switching strategies to handle disturbances, ensuring rapid target tracking. More recent work by Yang et al. [24] and [13] integrate a designed line-of-sight vector with the UAV's model for high-speed interception, achieving stable tracking at velocities up to 10 m/s. Although these methods offer strong performance and formal stability guarantees, they demand tuning low-level control, and potentially greater computational resources.

2.3.2 Model-Free IBVS In contrast, model-free IBVS methods abstract away the drone's internal dynamics, relying instead on pre-existing flight controllers to achieve stability and translate image-plane errors into high-level commands. These approaches often use proportional or rule-based controllers. For example, Barisic et al. [25] utilize nonlinear scaling of image errors to command yaw and pitch adjustments, while Lee et al. [10] combine YOLO-based detection with a Proportional scheme to maintain the target within a predefined "forwarding zone." Other works like Pestana et al. [26] and Wyder et al. [27] implement straightforward error-to-command mappings and minimal onboard computations, enabling real-time tracking without complex modelling. While these model-free approaches offer easier integration and lower computational demand, their reliance on generic low-level controllers may limit responsiveness and robustness in aggressive or high-speed interception scenarios.

The choice between model-based and model-free IBVS methods involves a trade-off between control precision and computational complexity. Model-based approaches provide precise control and stability guarantees but require detailed system modeling and higher computational resources, which may not be feasible for real-time applications on resource-constrained platforms. On the other hand, model-free methods offer simplicity, computational efficiency, and adaptability, albeit with potential limitations in handling aggressive manoeuvres. Given the requirements of low-cost, real-time quadcopter interception tasks, where computational efficiency and adaptability are paramount, this paper adopts a model-free IBVS approach. This decision leverages the ease of implementation and lower computational demands of model-free methods, making them more suitable for the intended application context.

III Problem Statement

This thesis aims to develop an IBVS control algorithm for a quadcopter drone, referred to as the pursuer, to intercept another quadcopter, the target. The proposed algorithm uses monocular camera input for target perception and is designed to handle a variety of initial positions and velocities. Additionally, it addresses the challenge of intercepting targets following unknown and dynamic trajectories, a problem often referred to as *pursuit-evasion* [28]. Interception is defined as the pursuer making physical contact with the target at a non-zero impact velocity. The research assumes that the target is already within the FOV of the camera of the pursuer, removing the need for target search. Similarly, the physical means of capture, such as nets or grippers, are beyond the scope of this work.

A Mathematical Problem Formulation

The problem can be mathematically expressed as follows. The pursuer must minimize two primary quantities:

(1) The Cartesian distance between the pursuer and the target in the world frame, r, defined as:

$$r = \sqrt{(x_t - x_p)^2 + (y_t - y_p)^2 + (z_t - z_p)^2} \tag{1}$$

where (x_t, y_t, z_t) and (x_p, y_p, z_p) are the positions of the target and pursuer, respectively.

(2) The image error, $\mathbf{e}$, which represents the displacement of the target from the image center in the image frame:

$$\mathbf{e} = \begin{bmatrix} u \\ v \end{bmatrix} \tag{2}$$

where u and v denote the horizontal and vertical pixel coordinates of the target relative to the image center.

Control Objective: Minimize r while ensuring $\mathbf{e}$ remains within the normalized range $[-1, 1]$, thereby keeping the target within the camera's FOV, and achieve a non-zero rate of approach $\dot{r}$ at interception:

$$\mathbf{e} \in [-1, 1], \quad \lim_{t \to T} r = 0, \quad \dot{r} < 0, \tag{3}$$

where T is the time of interception, and $\dot{r}$ is the time derivative of r, given by:

$$\dot{r} = \frac{\mathbf{r} \cdot (\mathbf{v}_t - \mathbf{v}_p)}{\|\mathbf{r}\|}, \tag{4}$$

with $\mathbf{v}_t$ and $\mathbf{v}_p$ representing the velocity vectors of the target and pursuer, respectively.

B Constraints and Challenges

The interception must be achieved under the following constraints:

- *Dynamic Constraints:* The quadcopter is underactuated, meaning its translational and rotational dynamics are tightly coupled.
- *Perception Constraints:* The monocular camera provides a 2D projection of the 3D scene, introducing challenges in estimating the relative depth and motion of the target. Different environmental conditions, including variable lighting, background clutter, and the target's unpredictable geometry, further complicate accurate perception
- *Computational Constraints:* The control algorithm must be computationally efficient to run on limited onboard resources.

IV Methodology

A Modeling

4.1.1 Quadcopter Model The quadrotor's dynamics and camera model establish the foundation for the IBVS controller. The quadrotor dynamics describe how inputs influence movement, while the

camera model relates motion to the target's image position. These models are essential to understanding how the drone and camera interact in pursuit scenarios.

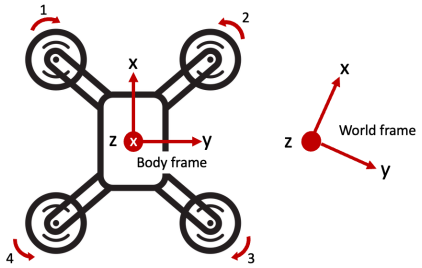

Figure 2: Quadcopter model

The quadcopter, shown in Fig. 2, has four fixed-pitch rotors that generate lift and torque through speed variations. Two rotor pairs, rotating in opposite directions, counteract the torque of each other to stabilize the system. Each motor produces an upward thrust (F) and a torque (τ), contributing to control over three principal axes: roll (ϕ), pitch (θ), and yaw (ψ). These dynamics yield coupled equations of motion with four control inputs but six degrees of freedom, creating inherent challenges in precise control.

The translational dynamics of the quadcopter are given by:

$$\begin{pmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{pmatrix} = \frac{1}{m} \sum_{i=1}^{4} F_i R_{BI} \mathbf{n}_z - g\mathbf{n}_z, \tag{5}$$

where m is the mass, g the gravitational constant, $\mathbf{n}_z$ the unit vector in the inertial frame, and R_{BI} the transformation matrix from body to inertial frame.

Rotational dynamics describe the response of the quadcopter to control inputs:

$$\begin{aligned} \phi &= \frac{l(F_2 - F_4)}{J_x}, \\ \theta &= \frac{l(F_1 - F_3)}{J_y}, \\ \dot{\psi} &= \frac{(F_1 - F_2 + F_3 - F_4)}{J_z}, \end{aligned} \tag{6}$$

where J_x, J_y, and J_z are the moments of inertia, and l is the distance from the rotors to the centre of mass of the quadcopter.

To simplify these equations, we define the control inputs as follows:

$$\begin{aligned} u_1 &= \frac{F_1 + F_2 + F_3 + F_4}{m}, \quad u_2 = \frac{(F_2 - F_4)}{J_x}, \\ u_3 &= \frac{(F_1 - F_3)}{J_y}, \quad u_4 = \frac{(F_1 - F_2 + F_3 - F_4)}{J_z}. \end{aligned} \tag{7}$$

Rewriting the equations of motion in terms of these inputs:

$$\begin{aligned} \ddot{x} &= u_1(\cos\phi \sin\theta \cos\psi + \sin\phi \sin\psi), \\ \ddot{y} &= u_1(\cos\phi \sin\theta \sin\psi - \sin\phi \cos\psi), \\ \ddot{z} &= u_1(\cos\phi \cos\theta) \\ \dot{\phi} &= u_2 l, \\ \dot{\theta} &= u_3 l, \\ \dot{\psi} &= u_4. \end{aligned} \tag{8}$$

These equations demonstrate the coupled dynamics. The inputs to control roll, pitch, yaw, and thrust affect both the translational and rotational behaviour of the drone. This coupling requires carefully coordinated control strategies to achieve smooth and stable flight while pursuing a dynamic target.

4.1.2 Image Kinematics The camera model maps the 3D motion of the quadcopter into 2D image coordinates, enabling IBVS to guide the quadcopter's movement. This relationship allows for the conversion of physical quadcopter dynamics into image-plane adjustments to minimize the error signal $e(t)$ in the IBVS feedback loop.

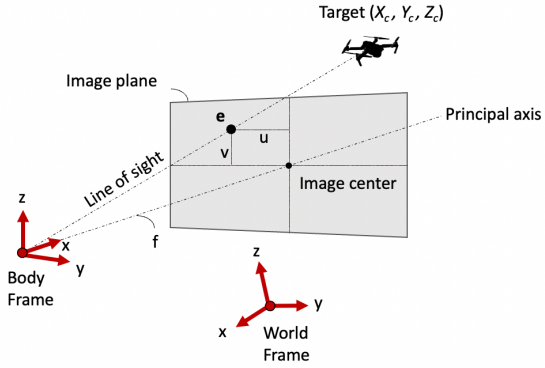

Figure 3: Pinhole Camera Model

Using the pinhole camera model (Fig. 3), a point $\mathbf{P} = \begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix}$ in the camera frame is projected onto the 2D image plane as:

$$\mathbf{e} = \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} \frac{f x_c}{z_c} \\ \frac{f y_c}{z_c} \end{pmatrix} \tag{9}$$

where f is the focal length of the camera. This projection provides the image coordinates $\mathbf{e}$, crucial for guiding the quadcopter toward a target within the image frame.

To understand how the velocity of the camera influences image motion, we define the spatial velocity vector in the body frame $\dot{\mathbf{r}}_\mathbf{B}$ as:

$$\dot{\mathbf{r}}_\mathbf{B} = \begin{pmatrix} v_x, v_y, v_z, \omega_x, \omega_y, \omega_z \end{pmatrix}^T \tag{10}$$

To connect the velocity of the quadcopter to changes in image coordinates $\dot{\mathbf{e}}$, we use the Jacobian $\mathbf{J}_p$:

$$\mathbf{J}_p = \begin{pmatrix} \frac{f}{z_c} & 0 & -\frac{u}{z_c} & -\frac{uv}{f} & \frac{f^2+u^2}{f} & -v \\ 0 & \frac{f}{z_c} & -\frac{v}{z_c} & -\frac{f^2+v^2}{f} & \frac{uv}{f} & u \end{pmatrix} \tag{11}$$

This matrix relates the camera's spatial velocity $\dot{\mathbf{r}}$ to $\dot{\mathbf{e}}$:

$$\dot{\mathbf{e}} = \mathbf{J}_p \dot{\mathbf{r}} \tag{12}$$

where $\dot{\mathbf{r}}$ includes translational and rotational velocities. $\mathbf{J}_p$ thus connects the movement of the quadcopter to image shifts, facilitating control adjustments that maintain the target within view by minimizing $e(t)$. This mapping is critical for developing control laws that directly influence the flight path of the quadcopter based on image feedback.

B Module Overview

This section describes the proposed interception system's architecture, detailing its key modules and their interactions. The design employs an IBVS approach, where image coordinates derived from object detections are used as inputs to the control algorithm. The system integrates modules for perception, state estimation, and control, ensuring seamless operation in dynamic interception scenarios. Figure 4 illustrates the structure of the designed system.

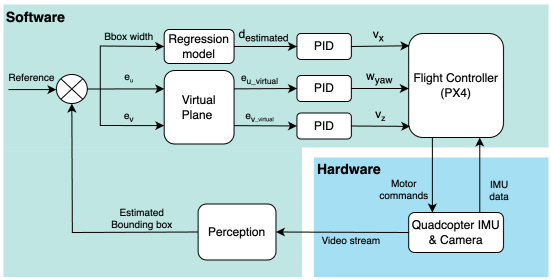

Figure 4: IBVS control loop of designed system

Control scheme of designed system. Design decisions for the controller and perception are covered in the subsequent subsections.

The process initiates with a comparison between the required feature— centred in the image to keep the target in view—and the detected feature, representing the target's actual position in the image. This comparison yields a feature error, which is then fed into the controller. The controller consists of two components: high-level and low-level. The high-level component generates position or velocity set points, while the low-level component maintains attitude stability. In this way motor commands are generated to minimize the image feature error and relative distance, effectively guiding the quadcopter toward its desired state—in this case, adhering to Eq. 3. Stability is crucial, as oscillations in flight can amplify perturbations in the image space, potentially leading to even more oscillatory behaviour.

In the following subsections, the considerations and decisions underlying the design of this system are discussed in detail.

C Perception

The perception module is an important component of the interception system, serving as the interface between the quadcopter's sensory inputs and its control mechanisms. In IBVS, the primary objective of the perception module is to accurately and efficiently generate image features of the target. This must be achieved at a rate compatible with the control loop while operating under the constraints of limited onboard computational resources. High levels of random errors can destabilize the control algorithm, whereas significant systematic errors may lead the quadcopter astray, potentially resulting in the loss of target visibility or failed interceptions. To address these challenges, the perception module in this design comprises three interconnected components: object detection, tracking, and filtering, visualized in **Fig. 5**.

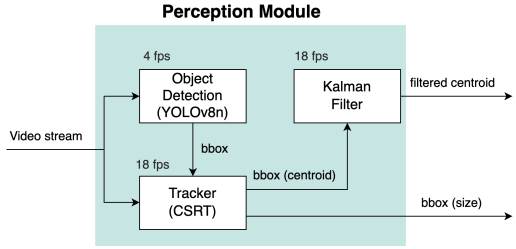

Figure 5: Perception pipeline

parallel execution of detection and tracking, the reinitialization mechanism, and the application of the Kalman filter for state estimation

4.3.1 ***Object detection*** For the object detection the YOLOv8 Nano model [29] is used. The YOLOv8 Nano operates by dividing the input image into a grid and simultaneously predicting bounding boxes and class probabilities for each grid cell. This end-to-end approach enables rapid processing, essential for maintaining high frame rates (18 fps) necessary for effective IBVS control.

In the comparative study by Lee et al. [10], various state-of-the-art object detection models were evaluated on their performance with quadcopter imagery. The study concluded that the YOLO Tiny variant offered the optimal compromise between speed and accuracy. Building on this, YOLOv8 Nano, the latest lightweight version in the YOLO series, improves upon earlier models such as YOLOv4 Tiny by incorporating architectural enhancements that further optimize performance under limited computational resources [30] [31]. Consequently, YOLOv8 Nano was selected for this design to ensure efficient and reliable object detection.

The model was trained using a dataset from Zheng et al. [32], which comprises over 13,000 images of a flying target UAV. These images encompass a diverse array of practical scenarios, including varying background scenes, viewing angles, relative distances, flying altitudes, and lighting conditions. The training process utilized standard hyperparameters over 100 epochs, ensuring robust learning and generalization across different environmental contexts.

4.3.2 ***Tracking*** While object detection effectively identifies targets within individual frames, it is inherently limited by its frame-by-frame processing nature, which may result in intermittent detection delays or missed frames. A tracking algorithm is employed to provide a continuous and high-frame-rate signal of the target's image features. In this design, the Discriminative Correlation Filter with Channel and Spatial Reliability (DCF-CSR), commonly referred to as CSRT [33], is implemented.

CSRT is integrated within the OpenCV library and leverages a spatial reliability map to dynamically adjust the filter's support region based on the spatial configuration of the target within the tracking frame. This adaptability enhances the tracker's robustness against occlusions and background clutter. Comparative studies have demonstrated that CSRT outperforms other tracking algorithms such as Kernelized Correlation Filters (KCF), Minimum Output Sum of Squared Error (MOOSE), and Mean-Shift in minimizing tracking failures and maintaining speed efficiency [34], [35].

4.3.3 ***Filtering*** Despite the combined efficacy of YOLOv8 Nano and CSRT, the system remains vulnerable to false positives, missed detections, and tracking inaccuracies induced by environmental noise or rapid target movements. To address these issues, a Kalman filter is incorporated to process the image centroid data, thereby smoothing out detection failures and predicting the target's subsequent position in the image plane.

The Kalman filter operates as a recursive estimator, utilizing a series of measurements observed over time to estimate the unknown state of a dynamic system [36]. In this context, the state comprises the target's position and velocity within the image plane. The choice of the Kalman filter is motivated by its optimality properties under Gaussian noise assumptions and its computational efficiency, which aligns with the real-time processing requirements of the IBVS control loop.

The equations governing the Kalman filter's operation, the parameters utilized and the tuning process are detailed in app. A.

4.3.4 Perception Integration To achieve reliable and efficient target localization, the perception module integrates object detection, tracking, and filtering. The YOLOv8 Nano detector and CSRT tracker operate in parallel, leveraging their respective strengths to maintain continuous and accurate tracking of the target. Specifically, YOLOv8 Nano performs object detection at a lower frequency, providing periodic bounding box updates that reinitialize the CSRT tracker. This reinitialization mitigates drift ensuring that the tracker remains accurately aligned with the target between detection intervals. Concurrently, the CSRT tracker maintains high-frame-rate tracking, offering smooth and real-time localization based on the most recent detections.

To further enhance the reliability of the perception system, a Kalman filter is employed to process the centroid data derived from both YOLOv8 Nano and CSRT. The Kalman filter smooths out transient errors and predicts the future position of the target, providing a stable and predictive estimate of the location of the target. This filtered centroid information is then utilized by the IBVS control system to generate precise control commands, ensuring robust and accurate interception.

The integration process is captured in Algorithm 1, which outlines the parallel execution of detection and tracking, the reinitialization mechanism, and the application of the Kalman filter for state estimation.

Algorithm 1 Perception Module Integration

```
while system is active do
    Execute YOLOv8 Nano detection at low frequency
    Execute CSRT tracking continuously at high frame rate
    if YOLOv8 Nano detects a new bounding box then
        Reinitialize CSRT tracker with the new bounding box
    end if
    Retrieve centroid from CSRT tracker
    Retrieve centroid from YOLOv8 Nano (if available)
    Apply Kalman filter to the centroid data
    Output smoothed and predictive target position
end while
```

This integrated approach ensures that the perception module delivers high-fidelity image features to the IBVS control system. By balancing the computational load between YOLOv8 Nano and CSRT, and enhancing the data with Kalman filtering, the system maintains target visibility and achieves accurate interception. The parallel execution and periodic reinitialization effectively reduce common perception-related issues such as false positives, missed detections, and tracking drift, thereby providing the overall efficacy and robustness of the interception system.

D Control Design

A multi-axis PID controller with an acceleration limiting strategy is proposed that combines individual PID controllers for yaw rate, vertical velocity, and forward velocity. By incorporating acceleration constraints into the forward velocity controller, this method ensures smooth and physically feasible motion across all controlled axes.

The image errors e_u and e_v are defined relative to the image centre, measuring the displacement of the target from the centre along the u-axis and v-axis, respectively. These errors are minimized by adjusting the yaw rate and vertical velocity as follows:

Yaw Rate Control: The yaw rate control input ω_{yaw} is calculated based on the error e_u along the u-axis:

$$\omega_{\text{yaw}} = K_p^{\text{yaw}} \cdot e_u + K_i^{\text{yaw}} \int e_u \, dt + K_d^{\text{yaw}} \frac{de_u}{dt} \tag{13}$$

Vertical Velocity Control: The vertical velocity control input v_z is derived from the error e_v along the v-axis:

$$v_z = K_p^{\text{v}} \cdot e_v + K_i^{\text{v}} \int e_v \, dt + K_d^{\text{v}} \frac{de_v}{dt} \tag{14}$$

Forward Velocity Control with Acceleration Limiting: To control forward velocity v_x, the distance error e_d is used. Acceleration clipping is applied to prevent large pitch changes, which could cause the target to move rapidly through the image, potentially leading to tracking issues. It is defined as:

$$v_x^{\text{target}}(t) = K_p^{\text{x}} \cdot e_d(t) + K_i^{\text{x}} \int_0^t e_d(\tau)\, d\tau + K_d^{\text{x}} \cdot \frac{de_d(t)}{dt} \tag{15}$$

$$a_{\text{desired}}(t) = \frac{v_x^{\text{target}}(t) - v_x(t)}{\Delta t} \tag{16}$$

$$a_{\text{limited}}(t) = \text{clip}\left(a_{\text{desired}}(t),\ -a_{\max},\ a_{\max}\right) \tag{17}$$

$$v_x(t + \Delta t) = v_x(t) + a_{\text{limited}}(t) \cdot \Delta t \tag{18}$$

Here the distance error e_d is crucial for interception, and it is defined as follows:

$$e_{\text{d}} = d_{\text{estimated}} + d_{\text{hit}} \tag{19}$$

Here, d_{hit} is a user-defined parameter that ensures the quadcopter hits the target with a non-zero hitting velocity. The $d_{\text{estimated}}$ term represents the estimated distance between the drone and the target, derived from visual features of bounding boxes in the image plane. The normalized bounding box width, computed as:

$$\text{Normalized Bounding Box Width} = \frac{\text{pixel width}}{\text{image width}} \tag{20}$$

was selected as the feature for distance estimation. This choice was made after comparing it to other features, such as the normalized bounding box size. The normalized width exhibited a smoother gradient across distances, leading to more stable and reliable estimations, especially under 10 meters where precision is critical. To map the normalized bounding box width to the actual distance, a degree-4 polynomial regression model was used:

$$d_{\text{estimated}}(x) = 1.538\times10^5 x^4 - 7.185\times10^4 x^3 + 1.201\times10^4 x^2 - 874.1x + 27.18 \tag{21}$$

where $d_{\text{estimated}}(x)$ represents the estimated distance as a function of x, the normalized bounding box width. This model ensures accuracy with a standard deviation of 2 meters for distances under 10 meters, supporting controlled deceleration during interception. Further details on the regression model development and validation are provided in Appendix B

4.4.1 Control Design Considerations The model-free multi-axis PID Control with acceleration limiting simplifies the control strategy by abstracting the relationship between system dynamics and image kinematics, which are difficult to encapsulate accurately within a model-based method. This makes it adaptable to unpredictable target movements and enhances robustness. Furthermore, this approach is computationally efficient, reducing overhead and enabling faster real-time responses compared to computationally intensive model-based solutions. Additionally, its simplicity facilitates easy tuning and implementation, and by generating high-level commands, it is scalable to other multirotor platforms with similar flight controllers like PX4 or Ardupilot, enhancing applicability and ease of integration across different hardware configurations.

On the other hand, its simple three degrees of freedom control outputs may not produce the agile movements needed for aggressive target trajectories. Furthermore, the lack of underlying models makes it difficult for this method to predict future movements in the image space of the target. Additionally, in this system where variables are highly interdependent, the model-free PID controllers might not effectively handle the coupling, whereas model-based controllers can incorporate these relationships into the control law. To address the coupling of variables, a virtual plane mechanism (discussed in the following section) is introduced. In this thesis experiments are conducted to validate if the benefits outweigh the drawbacks.

E Virtual Plane Projection

The dynamics of a quadcopter are inherently coupled, meaning that control inputs in one axis often impact others. In this setup, coupling becomes challenging due to a forward velocity command requires the quadcopter to pitch, which unintentionally alters the vertical error and generates an undesired vertical velocity command. Similarly, when executing a sharp turn, the quadcopter rolls, affecting both horizontal and vertical image errors and thus resulting in additional unwanted control commands. To address this issue, a virtual plane is proposed. This virtual plane is positioned at a focal distance in front of the camera, matching the distance of the original image plane. It is also perpendicular to the ground plane and aligned horizontally with the camera's yaw heading, as shown in Figure 6.

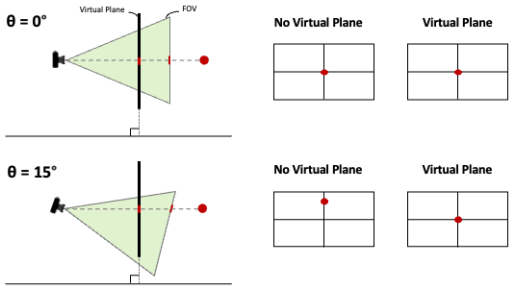

Figure 6: Virtual Plane [37]

The virtual image plane remains parallel to the inertial frame, allowing pitch and roll motions without affecting the image errors in an unwanted way.

We derive the formulas for the projected image errors e_u^{virtual} and e_v^{virtual} on the virtual plane in terms of the original image errors e_u and e_v and the pitch (θ) and roll (ϕ) angles. By defining a

transformation matrix that adjusts only for the pitch and roll angles, we ensure that the virtual plane remains aligned with the UAV's yaw heading but is unaffected by pitch or roll. This transformation can be expressed as follows:

$$T_{\text{virtual}} = \begin{pmatrix} 1 & \sin(\phi)\tan(\theta) & -\cos(\phi)\tan(\theta) \\ 0 & \cos(\phi) & \sin(\phi) \\ 0 & -\sin(\phi)/\cos(\theta) & \cos(\phi)/\cos(\theta) \end{pmatrix} \tag{22}$$

To find e_u^{virtual} and e_v^{virtual}, we apply the transformation matrix to the original image errors e_u and e_v:

$$\begin{pmatrix} e_u^{\text{virtual}} \\ e_v^{\text{virtual}} \end{pmatrix} = T_{\text{virtual}} \cdot \begin{pmatrix} e_u \\ e_v \\ f \end{pmatrix} \tag{23}$$

Expanding the matrix multiplication, the projected image errors e_u^{virtual} and e_v^{virtual} are given by:

$$e_u^{\text{virtual}} = e_u + e_v \sin(\phi)\tan(\theta) - f\cos(\phi)\tan(\theta) \tag{24}$$

$$e_v^{\text{virtual}} = e_v \cos(\phi) + f\sin(\phi) \tag{25}$$

The virtual plane projection offers several expected advantages. Decoupling pitch motion from the vertical image coordinate, it minimizes false vertical errors, thereby improving controller performance. Additionally, the control system benefits from reduced oscillations, as it avoids unnecessary corrections for pitch-induced errors, resulting in smoother motion and more stable control. Simplifying the control inputs also enables more predictable behaviour, as the decoupling allows the controller to focus on actual deviations.

However, a potential drawback is the limited field of view (FoV) in cases of significant pitch angles. If the quadcopter pitches too steeply, it risks losing the target from view due to the narrower effective FoV of the virtual plane. Experiments in Section 6 will investigate whether the benefits of this approach outweigh the limitations.

F Parameter Tuning of PID Controllers

The tuning of the three PID controllers was achieved through a systematic manual approach. To facilitate efficient parameter adjustments and high testing throughput, parameters were managed centrally via ROS and dynamically reconfigured in real-time through an interactive slider-based interface. This configuration allowed for the real-time tuning of parameters in the simulation environment, optimizing the feedback loop. The use of a virtual plane setup enabled the decoupling of the three PID control channels, allowing each channel to be tuned independently. By incrementally adjusting the PID parameters and analyzing system behaviour and error metrics, the parameter values presented in Table 1 were selected for use in the Software-in-the-Loop (SITL) experiments.

Table 1: PID Controller Parameters

Axis	K_p	K_i	K_d	$a_{\max}$ (m/s^2)
Horizontal	2.5	0.1	1.2	-
Vertical	4.0	0.1	1.7	-
Distance	1.5	-	0.4	2.5

V System Architecture

This section discusses the simulation and hardware setup employed in developing and testing of the quadcopter interception system.

A Software Setup

The software framework for the simulation environment integrates several components to model, control, and analyze the interception process. An overview of the key tools and their roles follows:

- **Gazebo:** The dynamics and visualization of the quadcopter drone are modelled in Gazebo-classic. The *iris* model is used as the quadcopter model due to its comparable dimensions to the Holybro X500 V2, used in real-world test flights. Since the proposed algorithm leverages the pre-tuned PX4 low-level controller and utilizes high-level velocity and yaw rate commands, no significant differences are expected between the simulation and real-world flights in terms of PID tuning. The sim-to-real gap is expected to lie primarily in the perception module.
- **PX4:** PX4 acts as the simulated flight controller and mirrors the behavior of the physical flight controller. It uses the MAVLink communication protocol to exchange data and control commands, ensuring compatibility across simulation and hardware platforms.
- **MAVROS:** MAVROS serves as middleware, bridging MAVLink messages with the Robot Operating System (ROS), allowing seamless communication between the PX4 flight controller and ROS nodes.
- **ROS Noetic:** ROS Noetic is the overarching communication framework that integrates all components. It facilitates modular communication using topics and services. ROS plugins enable Gazebo and PX4 to publish and subscribe to sensor data and control commands.

These nodes operate within the ROS ecosystem, utilizing its publish-subscribe architecture for real-time coordination.

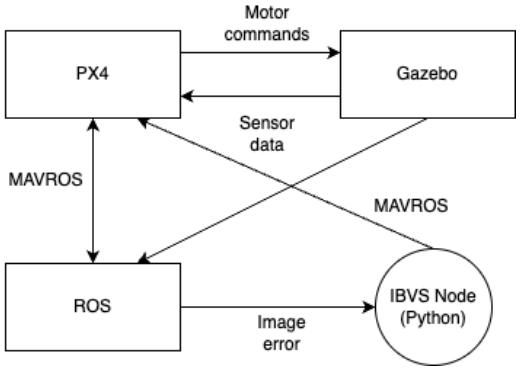

Figure 7: Software architecture and data flow in the simulation environment.

The integration of Gazebo, PX4, and ROS facilitates realistic simulations and modular development.

B Hardware Setup

The hardware setup mirrors the simulated environment to minimize the gap between virtual tests and real-world implementations. The following components form the physical test platform:

- **Quadcopter Platform:** The Holybro X500 V2 serves as the base frame, offering a stable and reliable structure for testing.
- **Flight Controller:** The Pixhawk 6C is employed, equipped with an Inertial Measurement Unit (IMU) and running PX4 flight control software.
- **Camera:** A USB camera with a resolution of 640x480 pixels captures real-time visual data for the IBVS algorithm.
- **Onboard Computer:** A Raspberry Pi 4B executes all computational tasks, including perception and control, enabling full autonomy onboard the quadcopter. The quadcopter runs ROS Noetic as the master node, while a laptop connected via telemetry acts as the slave node, primarily for monitoring and analysis. Central parameter management enables real-time dynamic reconfiguration of parameters, such as PID values, from the laptop.
- **Communication Devices:** A telemetry radio establishes a link between the quadcopter and the laptop. Additionally, an RC joystick provides a manual override mechanism for safety-critical operations.

Figure 8: Quadcopter test platform with integrated hardware components.

The Holybro X500 V2 equipped with a Pixhawk flight controller, Raspberry Pi, camera, and telemetry module.

C Safety Measures

Safety is a critical consideration during the preparation and operation of the quadcopter. In addition to pre-flight checks and in-flight monitoring, several safety mechanisms have been integrated into the system to ensure robust and reliable operation:

- **Geofence:** The PX4 geofence feature ensures the quadcopter operates within pre-configured boundaries. If the quadcopter breaches the geofence, experiences battery or other system failures, or receives a manual command from the laptop, the PX4 built-in Return to Starting Position (RTSP) is triggered.
- **RC Override:** The RC controller can override the offboard (autonomous) mode at any time. This allows the operator to safely land the quadcopter during a malfunction.

The hardware-software integration ensures a smooth transition between simulation and physical testing. ROS enables consistent communication protocols and data structures across environments, bridging the gap between virtual and real-world deployments.

VI Results

This section presents the outcomes of the experiments conducted, evaluating the proposed IBVS algorithm across various scenarios. The metrics and the benchmarking are discussed first to establish a foundation for the analysis.

A Metrics

To evaluate the performance of the proposed IBVS algorithm, several metrics were employed to quantify specific aspects of the system's effectiveness and robustness. The significance of these metrics and their application in the experiments are outlined below:

- **Interception Error**: This metric measures the Euclidean distance between the drone and the target at the moment of interception. A lower interception error indicates higher precision in guiding the drone to the target. It is particularly important in scenarios where the accuracy of interception directly impacts mission success. This metric was assessed using scatter plots, box plots, and the Circular Error Probable (CEP), defining the radius enclosing 50% of interception points.
- **Interception Time**: The time taken for the pursuer to reach the target from its initial position is used to evaluate the algorithm's speed and agility. A lower interception time signifies a more efficient trajectory and higher speed, being critical for fast-moving targets or time-sensitive missions.
- **Field-of-View (FOV) Maintenance**: This is evaluated by measuring *normal velocity* and *normal acceleration*, representing the components of the pursuer's movement orthogonal to the line of sight (LOS) to the target. Low values for these metrics indicate that the pursuer maintains a trajectory closely aligned with the LOS, keeping the target centred in the image frame and minimizing lateral deviations.
- **Trajectory Smoothness**: Oscillations in the pursuer's path are undesirable as they can destabilize the system and increase interception time. Normal acceleration profiles are analyzed to assess the smoothness of the trajectory and the stability of control.

B Benchmarking

To assess the performance of the proposed IBVS algorithm, it is benchmarked against model-based approaches in the literature. Yan et al. [24], a recent study published in September 2024, introduced an IBVS framework that employs a low-level controller to align

the drone's velocity vector with the line-of-sight (LOS) to the target. Their method demonstrated superior precision in intercepting quadcopters, utilizing only a monocular camera and an Inertial Measurement Unit (IMU), which aligns with our sensor constraints. Consequently, Yan et al.'s approach was selected as the primary benchmark. The studies it was compared to, by Jana et al. [38] and Yang et al. [13], are also included in the comparative analysis.

C Multi-Angle Static Target Interception Experiment

In this subsection, we replicate the experiment conducted by Yan et al. [24] to evaluate the performance of our proposed interception algorithm compared to their results. We conducted fifty static target interception experiments with targets located at various positions. The quadcopter initiates from the position $(0, 0, 10)$ meters in the world frame. The target positions however, as depicted in Fig. 9, are randomly generated within a range of 15 to 35 meters from the starting point of the pursuer.

As observed in Fig. 9(a), the trajectory shapes differ between the two methods. In Yan et al. [24], the trajectories begin together and diverge more as they approach the targets. In contrast, our proposed algorithm causes the divergence to occur earlier, resulting in interception angles that are more parallel to the ground plane.

Figures 9(b) and 9(c) illustrate that our proposed IBVS method achieves a lower circular error probable compared to the method in Yan et al. [24], with errors of 0.072 meters and 0.089 meters, respectively. This represents a reduction of 19%, indicating that our algorithm enhances the accuracy of static target interception.

D Moving Target Interception Experiment: Three Scenarios

This experiment evaluates the interception accuracy of the proposed algorithm across various target manoeuvring scenarios. Three manoeuvring models were tested: constant velocity (CV), constant acceleration (CA), and sinusoidal manoeuvre (SM). The parameters used were consistent with those defined at the beginning of this section. Table 2 depicts the initial relative positions of the pursuer and target, along with the target's movement in each scenario. The results for the CV, CA, and SM scenarios are depicted in Figures 10, 11, and 12, respectively.

Table 2: Experiment Parameters for Different Scenarios: Constant Velocity (CV), Constant Acceleration (CA), and Sinusoidal Maneuver (SM)

	CV	CA	SM
Start Position Pursuer (m)	$(0, 0, 10)$	$(0, 0, 10)$	$(0, 0, 10)$
Start Position Target (m)	$(0, 25, 1)$	$(-8, 15, 8)$	$(0, 30, 10)$
Movement Target	$\mathbf{v} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$	$\mathbf{a} = \begin{pmatrix} 0.8 \\ 0 \\ 0.2 \end{pmatrix}$	$\mathbf{p} = \begin{pmatrix} 2\sin\left(\frac{2\pi}{14}t\right) \\ 3t \\ 0 \end{pmatrix}$

6.4.1 Constant Velocity (CV) Scenario In the CV scenario, our proposed algorithm demonstrates a shorter interception time compared to the other methods, as shown in Fig. 10(a). Additionally, the normal acceleration exhibits a lower peak and overall smaller values (Fig. 10(c)), indicating a better ability to keep the target within the field of view (FOV) while approaching, even with a reduced interception time. Despite an interception error of 0.16 meters, which is slightly larger than that reported by Yan et al. [24], the error remains within the acceptable margin for hitting the target (the target is 0.4 meters wide), as illustrated in Fig. 10(d).

6.4.2 Constant Acceleration (CA) Scenario In the CA scenario, as shown in Fig. 11(a), the trajectory of our proposed algorithm initially overshoots in the y-direction but subsequently corrects itself to intercept the target. The methods proposed by Yan et al. [24] and others seem to better anticipate the target's path, moving more directly toward the interception point. Although the normal velocity of our algorithm remains lower than that of Yan et al. [24] throughout the interception (Fig. 11(b)), the normal acceleration experiences a spike around 3 seconds (Fig. 11(c)), corresponding to the sharp turn required to align with the trajectory of the target. Nevertheless, the interception time of 4.42 seconds is comparable to that of Yan et al. [24], and our algorithm achieves a slightly higher precision with an interception error of 0.41 meters compared to 0.45 meters, as shown in Fig. 11(d).

6.4.3 Sinusoidal Manoeuvre (SM) Scenario In the SM scenario, our proposed algorithm maintains normal acceleration and velocity below 0.5 m/s^2 and 0.5 m/s, respectively (Figs. 12(b) and 12(c)), significantly lower than the other algorithms. This indicates that, in the absence of vertical translation of the target, the algorithm is superior in keeping the target within its FOV. Although the interception error is slightly higher than that of Yan et al. [24], it remains within an acceptable range, affirming the algorithm's ability to intercept accurately, as depicted in Fig. 12(d).

E Virtual Plane Experiment

The objective of this experiment is to assess the effectiveness of the Virtual Plane technique in improving interception efficiency. To this end, we compare the interception trajectories and performance with and without the Virtual Plane under identical conditions. The simulation setup remains consistent, with the pursuer starting from $(0, 0, 10)$ meters and the target beginning at $(0, 10, 10)$ meters, moving this time at a higher velocity of $\mathbf{v} = (1, 8, 1)$ meters per second. This increased target speed intensifies the pitch angle required for pursuit, accentuating the impact of the Virtual Plane on interception dynamics.

In Fig. 13 (a), the trajectory results demonstrate that the pursuer utilizing the Virtual Plane (blue line) follows a more direct path toward the target. In contrast, the pursuer without the Virtual Plane (green line) adopts a less efficient trajectory, attempting to intercept the target from above. This divergence is attributed to the lack of a Virtual Plane, which causes pitch adjustments to amplify vertical error, triggering the vertical controller to issue compensatory commands that lead to overshooting. By incorporating the Virtual Plane, this drawback is mitigated, enabling the pursuer to intercept the target more efficiently and successfully, whereas the non-Virtual Plane approach results in a missed interception in this

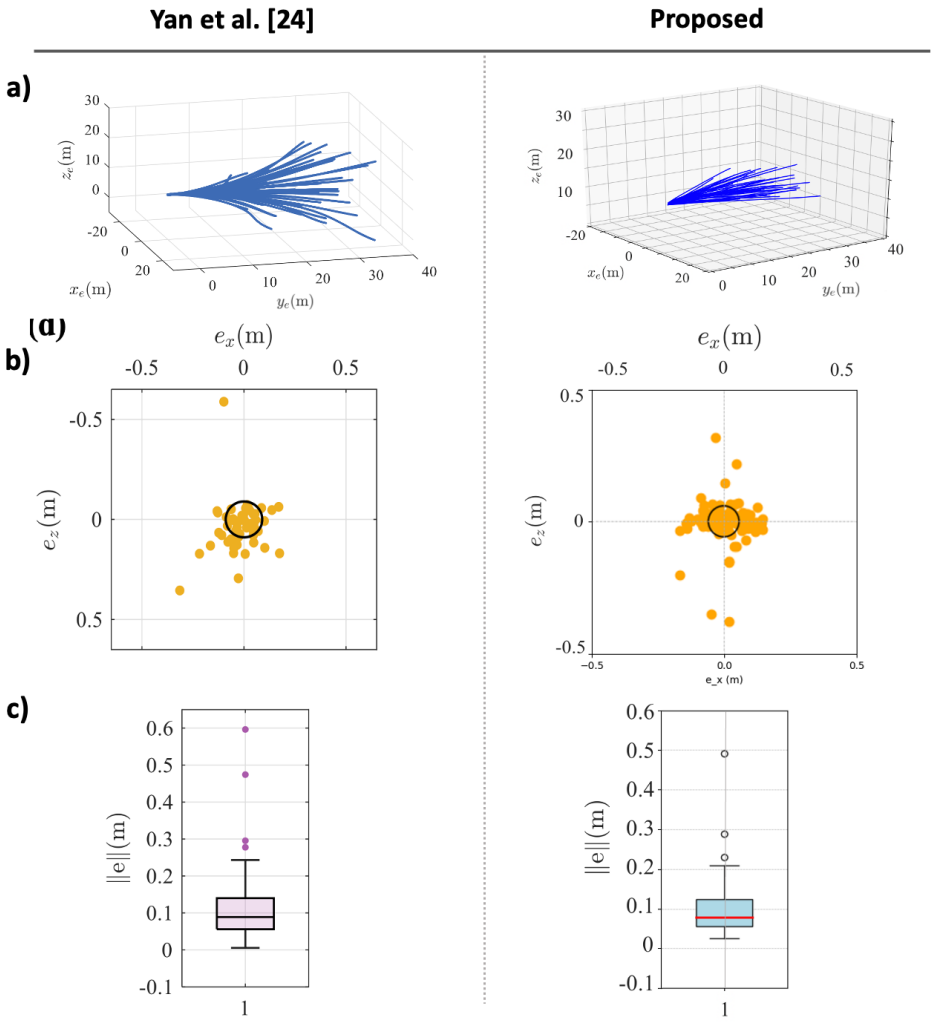

Figure 9: Results for intercepting targets from 50 different angles

(a) *Pursuer trajectories for 50 interceptions.* ***(b)*** *Scatterplot of interception errors for the 50 interceptions.* ***(c)*** *Boxplot of interception errors for the 50 interceptions.*

scenario.

The inefficiency of the non-Virtual Plane method is further highlighted in Figs. 13(b) and 13(c), where the normal velocity and acceleration profiles exhibit higher magnitudes and oscillatory behaviour. This instability arises from conflicting commands between the forward and vertical controllers. The Virtual Plane reduces these interactions, resulting in a more stable trajectory with lower normal acceleration and velocity variations.

In conclusion, the Virtual Plane technique yields a more efficient interception path, reduces interception time, and enhances flight stability, even at elevated target speeds (up to 8–12 m/s in this scenario).

Table 3: Benchmark comparison results for interception experiments under different scenarios

Scenario	Metric	Yan et al. [24]	Proposed
Multi-Angle Static	Circular Error Probable (m)	0.089	**0.072**
Constant Velocity (CV)	Interception Time (s)	6.5	**4.4**
	FOV Maintenance	Medium	Medium
	Interception Error (m)	**0.09**	0.16
Constant Acceleration (CA)	Interception Time (s)	4.4	4.4
	FOV Maintenance	Medium	Low
	Interception Error (m)	0.45	**0.41**
Sinusoidal Maneuver (SM)	Interception Time (s)	14.5	**10.8**
	FOV Maintenance	Medium	High
	Interception Error (m)	**0.04**	0.11

F Results Overview

The results are summarized in Table 3, showcasing performance across different scenarios. The proposed algorithm outperforms the

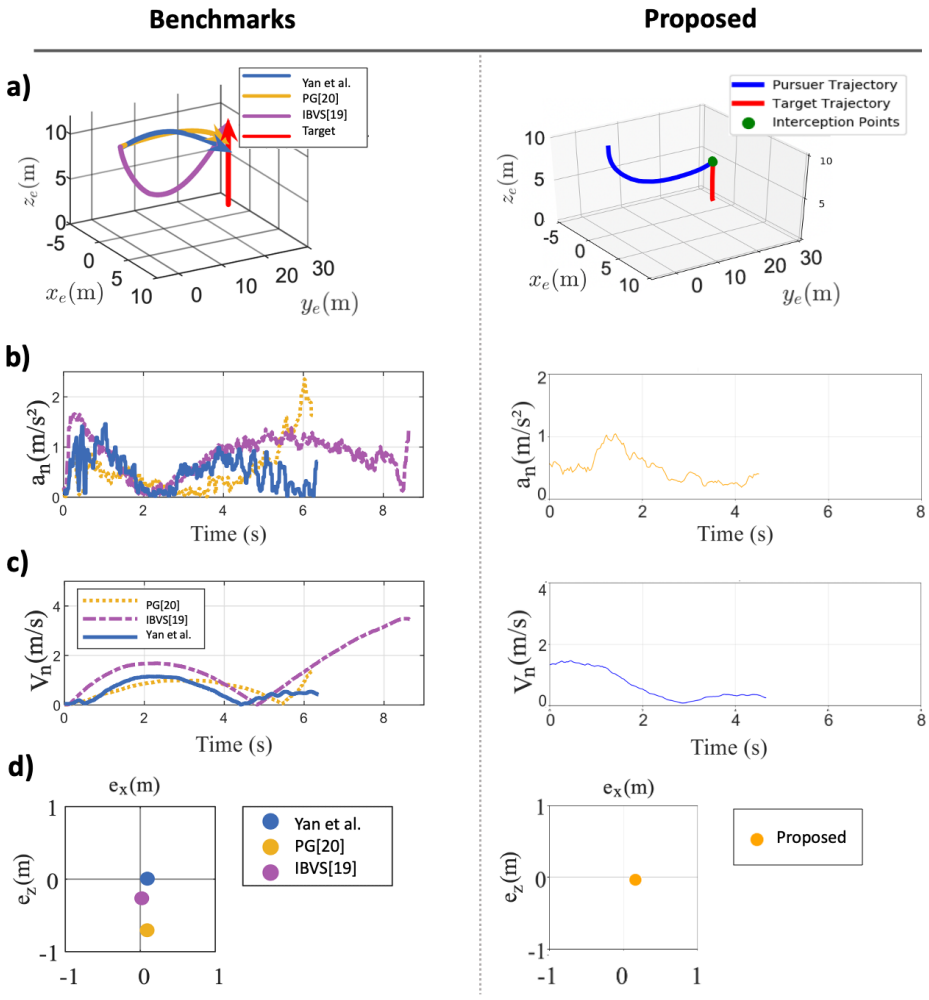

Figure 10: Results for intercepting a target with constant velocity

(a) *Pursuer and target trajectories.* ***(b)*** *Normal velocity of the pursuer during interception.* ***(c)*** *Normal acceleration of the pursuer during interception.* ***(d)*** *Interception errors. The benchmark papers are: Yan et al.: [24], PG: [38], IBVS: [13]*

benchmark in circular error probable (CEP) for static targets (0.072 m vs. 0.089 m) and demonstrates shorter interception times for constant velocity (CV) and sinusoidal maneuver (SM) scenarios. It achieves lower interception errors in the constant acceleration (CA) scenario, albeit with variations in field-of-view (FOV) maintenance. These results highlight the proposed algorithm's effectiveness in enhancing accuracy and efficiency.

VII Discussion

The results of this research highlight a series of trade-offs and challenges inherent to designing an effective, low-cost, autonomous quadcopter interception system. By delving into the underlying causes of these results, this discussion provides insights into the observed performance and the implications for future development.

A Perception Module Performance

The designed perception module, a critical component of the proposed system, demonstrated significant limitations in real-world tests, particularly in its frame rate when deployed on the Raspberry Pi 4b. This shortfall can be attributed to the high computational demands of Yolov8n, a state-of-the-art object detection model. While Yolov8n's accuracy is important in maintaining reliable tracking and interception, its processing requirements far exceed the capabilities of resource-constrained hardware like the Raspberry Pi. As a result, the frame rate dropped to levels insufficient for high-speed interception or scenarios involving rapid manoeuvres. This low frame rate creates a lag in feedback, impairing the system's ability to respond effectively to dynamic target movements and leading to missed interception opportunities or reduced precision.

The decision to employ the Raspberry Pi reflects a broader cost-efficiency trade-off in UAV design. While it enables the use of affordable, lightweight platforms, the computational limitations

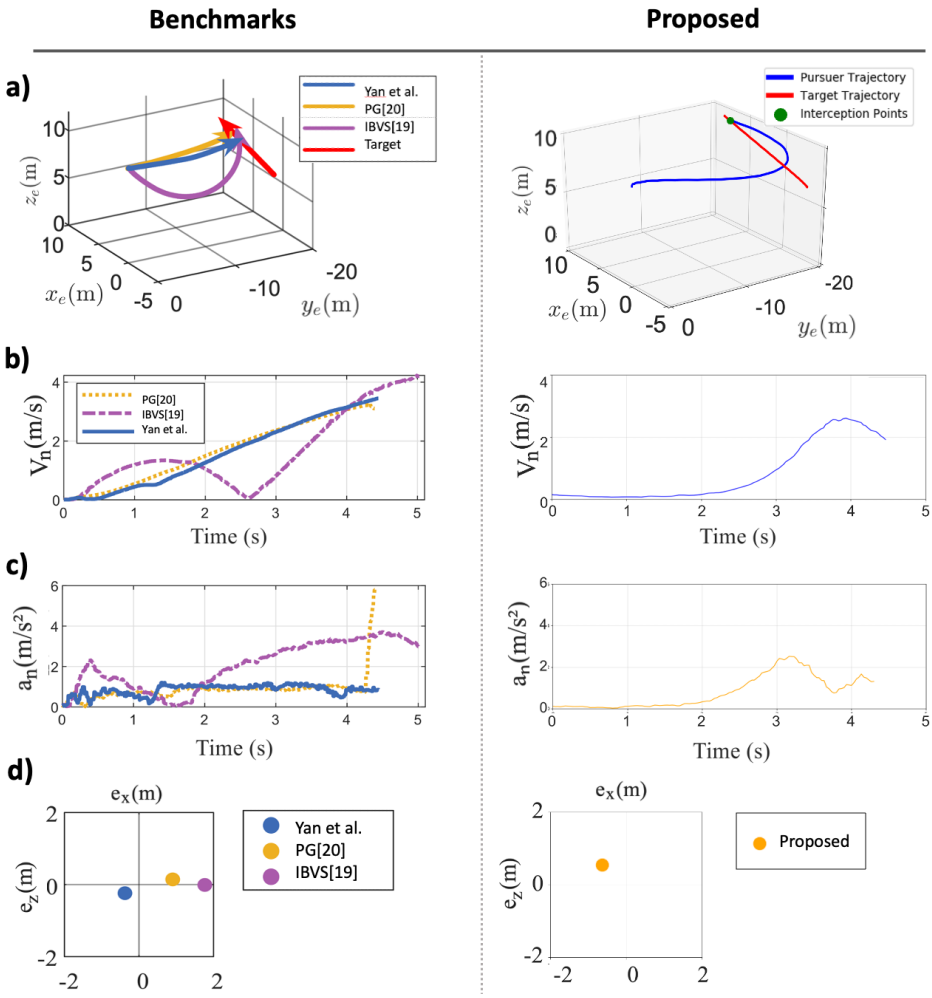

Figure 11: Results for intercepting a target with constant acceleration

(a) *Pursuer and target trajectories.* ***(b)*** *Normal velocity of the pursuer during interception.* ***(c)*** *Normal acceleration of the pursuer during interception.* ***(d)*** *Interception errors. The benchmark papers are: Yan et al.: [24], PG: [38], IBVS: [13]*

underscore the need for alternative approaches. For instance, leveraging less computationally intensive object detection models, such as MobileNet variants, or integrating a more powerful processing unit, like Nvidia's Jetson Nano or Xavier NX, could address the frame rate bottleneck. However, these alternatives introduce new challenges, including increased costs, higher power consumption, and potential reductions in battery life, which may compromise the drone's operational endurance.

B Virtual Plane Projection and Decoupling Dynamics

The implementation of the virtual plane mechanism emerged as a significant contributor to the system's improved control stability. By decoupling pitch and vertical motion, the virtual plane reduced the cascading effects of coupled dynamics. For example, in scenarios without the virtual plane, forward velocity commands inadvertently induced vertical errors, necessitating compensatory corrections that destabilized the system. The decoupling effect not only reduced these oscillations but also allowed for smoother trajectories and more efficient interception paths.

However, this stability came at the cost of a reduced effective FOV. The virtual plane's fixed alignment relative to the UAV, while effective for decoupling, constrains the system's ability to track targets during steep pitch angles. This limitation becomes particularly evident in scenarios with fast-moving or erratically manoeuvring targets, where maintaining FOV is critical. While the results validate the virtual plane's efficacy in simplifying control dynamics, future iterations could explore adaptive virtual planes. Such a system might dynamically adjust the plane's alignment based on the drone's attitude, balancing the benefits of decoupling with a broader FOV.

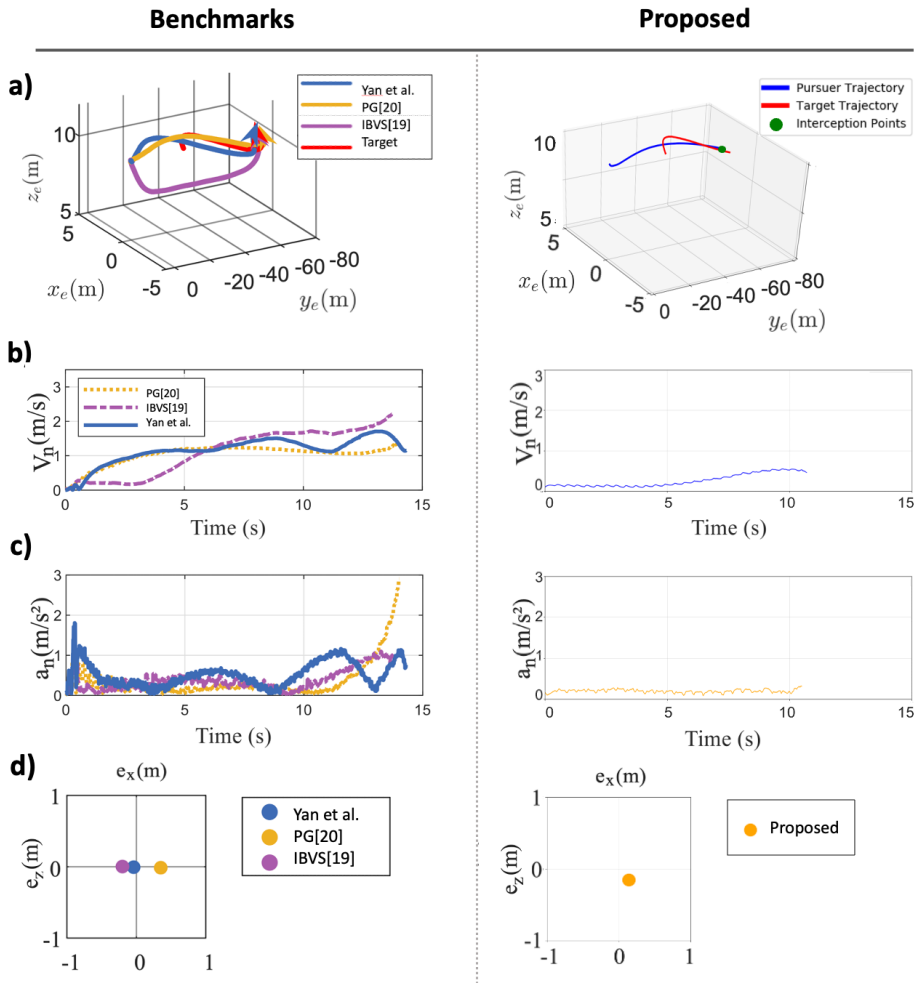

Figure 12: Results for intercepting a target with sinusoidal manoeuvre

(a) *Pursuer and target trajectories.* ***(b)*** *Normal velocity of the pursuer during interception.* ***(c)*** *Normal acceleration of the pursuer during interception.* ***(d)*** *Interception errors. The benchmark papers are: Yan et al.: [24], PG: [38], IBVS: [13]*

C Model-Free PID Controller: Simplicity and Limitations

The model-free PID controller demonstrated significant strengths, particularly in its simplicity and computational efficiency. These characteristics allowed the system to outperform benchmarks in terms of response time and stability during scenarios with less abrupt target movements. By reducing computational overhead, the controller facilitated rapid adjustments to maintain the target within the FOV, a critical factor in successful interception.

However, this simplicity also introduced notable limitations. The lack of predictive capabilities in the model-free PID design explains the overshooting observed during sharp lateral turns. Unlike model-based approaches, which leverage predictive algorithms to anticipate future target trajectories, the PID controller reacts solely to the current error, making it ill-suited for highly dynamic scenarios. This highlights a fundamental trade-off: while model-free control reduces complexity and computational demand, it sacrifices the ability to handle aggressive target manoeuvres effectively. Exploring hybrid control architectures that combine the responsiveness of model-free PID with the trajectory prediction of model-based approaches could offer a pathway to overcoming this limitation.

D Recommendations and Future Directions

To address the perception module's limitations, future research should explore lightweight object detection models optimized for real-time performance on low-power hardware. One promising avenue could involve deploying feature-based tracking algorithms that rely on computationally inexpensive heuristics rather than deep learning models. Alternatively, hybrid architectures combining lightweight object detection with intermittent updates from computationally intensive models could balance accuracy and speed.

In terms of control, incorporating predictive capabilities into the system represents a critical next step. Reinforcement learning models trained in simulation environments could enable the UAV

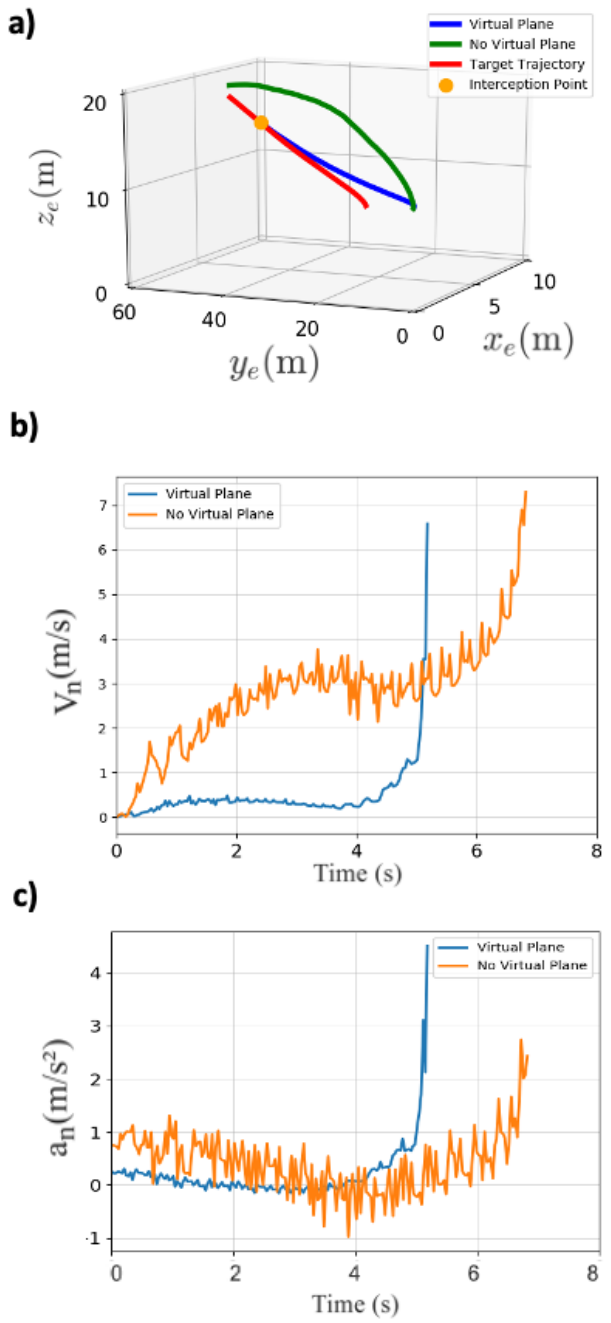

Figure 13: Intercepting a moving target with and without Virtual Plane

(a) *Pursuer and target trajectories.* ***(b)*** *Normal velocity of the pursuer during interception.* ***(c)*** *Normal acceleration of the pursuer during interception.*

to anticipate target trajectories, reducing overshooting and improving performance in dynamic scenarios. While the sim-to-real gap remains a challenge for learning-based approaches, advances in domain randomization and transfer learning may help bridge this gap, enabling real-world deployment.

Finally, the virtual plane mechanism could be refined by integrating adaptive capabilities that dynamically adjust the plane's orientation based on the drone's attitude and target position. While this would increase system complexity, it could enhance FOV maintenance, particularly during steep pitch manoeuvres.

VIII Conclusion

In this thesis, an IBVS control system was designed and evaluated to enable a low-cost quadcopter to intercept a target autonomously using only a monocular camera and an IMU for sensing. This work addressed a critical need for interception control systems resilient to GPS-denied environments, suitable for resource-constrained UAV platforms. Two core innovations, an Integrated Multi-Axis PID Control with Acceleration Limiting and a novel virtual plane projection, were developed to improve interception accuracy and FOV maintenance.

The first contribution is the control method, where the multi-axis PID control with acceleration limiting strategy combines independent PID controllers for yaw rate, vertical velocity, and forward motion. Benchmarking against Yan et al.'s model-based IBVS algorithm [24], this proposed control approach demonstrated a 19% reduction in circular error probable (CEP) for static target interception, validating its improvement in accuracy. Moreover, in scenarios requiring sustained target visibility, the proposed control method showed lower normal velocities and accelerations, indicating superior FOV maintenance with equivalent or reduced interception times. This outcome highlights the advantage of the model-free approach: reduced complexity and computational demand, while maintaining a competitive level of interception precision and FOV stability relative to more complex model-based methods.

The second contribution is the virtual plane projection technique, where image errors are mapped to a virtual plane orthogonal to the ground at the height of the quadcopter. This approach effectively decouples pitch and vertical motion, further stabilizing target tracking within the FOV. Simulation results affirmed the efficacy of this design, as ablation studies across different scenarios demonstrated reduced oscillatory behavior in image errors, contributing to greater stability. Moreover, using the virtual plane improved interception performance, with reductions in both interception error and time by notable percentages in tested scenarios, underscoring its practical value in dynamic target engagements.

In conclusion, this thesis demonstrates a feasible, resource-efficient IBVS approach for autonomous UAV interception that is both accurate and robust. The findings advance the field of autonomous UAV interception by providing a reliable method that meets design requirements while ensuring operational resilience. Future research may extend this work by optimizing the perception module or exploring advanced obstacle avoidance techniques. The presented system offers a scalable, practical solution for real-world UAV interception, with promising applications in security, surveillance, and beyond.

Code Availability

The source code used for this study is available at the following repository: https://github.com/cagvanderaa/thesis_avalor.git.

Conflict of Interest

The author conducted this research in collaboration with Avalor AI. The company did not influence the analysis or interpretation of the research findings. The author declares that there are no other conflicts of interest regarding the writing of this paper.

I Appendix: Mathematical Formulation of the Kalman Filter

The Kalman Filter is integrated into the perception module to enhance the accuracy and robustness of target state estimation by smoothing out detection noise and predicting future positions. Below, we outline the specific parameters and configurations used in our implementation.

A State Vector and Model

The state vector $\mathbf{x}_k$ at time step k is defined as:

$$\mathbf{x}_k = \begin{bmatrix} u_k \\ v_k \\ \dot{u}_k \\ \dot{v}_k \end{bmatrix}, \tag{26}$$

where u_k and v_k represent the target's position coordinates in the image plane, and $\dot{u}_k$ and $\dot{v}_k$ denote their respective velocities.

B System Dynamics

Assuming a constant velocity model, the state transition matrix $\mathbf{F}$ and the observation matrix $\mathbf{H}$ are defined as:

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{27}$$

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}. \tag{28}$$

Here, Δt is the time step between consecutive measurements, set to match the control loop frequency.

C Covariance Matrices

The performance of the Kalman Filter is significantly influenced by the choice of the process noise covariance matrix $\mathbf{Q}$ and the measurement noise covariance matrix $\mathbf{R}$. These matrices were empirically determined based on the system's dynamics and sensor characteristics.

A.3.1 Process Noise Covariance (Q)

$$\mathbf{Q} = \begin{bmatrix} 0.1 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0.5 \end{bmatrix}. \tag{29}$$

The diagonal elements of $\mathbf{Q}$ represent the uncertainty in the acceleration (process noise) of the target's motion.

A.3.2 Measurement Noise Covariance (R)

$$\mathbf{R} = \begin{bmatrix} 12.5 & 0 \\ 0 & 12.5 \end{bmatrix}. \tag{30}$$

The measurement noise covariance $\mathbf{R}$ accounts for the uncertainty in the centroid measurements obtained from the detection and tracking modules.

D Initial Conditions

Initialization of the Kalman Filter is crucial for convergence. The initial state estimate $\mathbf{x}_0$ and the initial estimate covariance $\mathbf{P}_0$ are set as follows:

$$\mathbf{x}_0 = \begin{bmatrix} u_0 \\ v_0 \\ 0 \\ 0 \end{bmatrix}, \tag{31}$$

$$\mathbf{P}_0 = \begin{bmatrix} 1000 & 0 & 0 & 0 \\ 0 & 1000 & 0 & 0 \\ 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 100 \end{bmatrix}. \tag{32}$$

Here, a high initial covariance reflects significant uncertainty in the initial state estimate.

E Filter Operation

The Kalman Filter operates in a recursive manner, alternating between prediction and update steps:

(1) **Prediction:**

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{F}\hat{\mathbf{x}}_{k-1|k-1}, \tag{33}$$

$$\mathbf{P}_{k|k-1} = \mathbf{F}\mathbf{P}_{k-1|k-1}\mathbf{F}^\top + \mathbf{Q}. \tag{34}$$

(2) **Update:**

$$\mathbf{y}_k = \mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_{k|k-1}, \tag{35}$$

$$\mathbf{S}_k = \mathbf{H}\mathbf{P}_{k|k-1}\mathbf{H}^\top + \mathbf{R}, \tag{36}$$

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{H}^\top\mathbf{S}_k^{-1}, \tag{37}$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\mathbf{y}_k, \tag{38}$$

$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k\mathbf{H})\mathbf{P}_{k|k-1}. \tag{39}$$

These steps are iteratively performed at each time step as new measurements $\mathbf{z}_k$ become available from the perception module.

F Parameter Tuning

The covariance matrices $\mathbf{Q}$ and $\mathbf{R}$ were manually tuned to balance the trust between the prediction model and the measurements. This tuning was guided by:

- **Process Noise ($\mathbf{Q}$):** Adjusted to reflect the expected variability in the target's motion. Higher values allow the filter to adapt more quickly to changes.
- **Measurement Noise ($\mathbf{R}$):** Set based on the reliability of the centroid measurements. Lower values increase the filter's reliance on measurements.

Empirical testing and visual inspection within the simulation environment ensured that the chosen parameters provided stable and accurate tracking under various conditions.

II Appendix: Distance Estimation

In this section, we explain the method used to establish distance estimation for the target drone. The objective is to develop a model that maps features extracted from the bounding box in image space to the corresponding distance in the world frame, representing the separation between the camera and the target. This distance serves as an input for the forward velocity controller. Given the limited forward acceleration, precise distance estimation is not critical for distances greater than 10 meters, as the forward velocity will reach a limit to adhere to the maximum acceleration. For distances under 10 meters, however, the model is required to estimate the distance with an accuracy of approximately 2 meters to facilitate controlled deceleration, thereby enhancing interception precision. The controller minimizes the distance error, e_{distance}, defined as $d_{\text{measured}} + d_{\text{hit}}$, where d_{hit} is set to 5 meters to ensure a non-zero impact velocity for distance estimations with a standard deviation of 2 meters.

A dataset of 90 annotated bounding box data points was collected in Gazebo simulations. The bounding box annotations were recorded from various distances and angles, noting the corresponding distances between the pursuer and target. Data collection began at a distance of 1 meter, where bounding boxes were recorded from three different angles. This process was repeated in increments of one meter up to a maximum distance of 15 meters. Using this dataset, we constructed two plots (Fig. 14) to compare distance estimation models based on two different bounding box features: normalized bounding box size and normalized bounding box width. In Fig. 14 **a**, the normalized bounding box size, calculated as:

$$\text{Normalized Bounding Box Size} = \frac{\text{pixel width}}{\text{image width}} \times \frac{\text{pixel height}}{\text{image height}} \tag{40}$$

is plotted against distance. Fig. 14 **b** plots the normalized bounding box width, calculated as

$$\text{Normalized Bounding Box Width} = \frac{\text{pixel width}}{\text{image width}} \tag{41}$$

against distance. For both cases, polynomial regression models of degrees 3, 4, and 5 were fitted to the data.

Upon examining the plots, we observe that the model based on normalized bounding box width (**Fig. 14 b**) demonstrates a superior spread of data points across different distances compared to the model based on bounding box size (**Fig. 14 a**). The normalized bounding box width exhibits a smoother gradient, particularly at distances below 10 meters, which provides a more gradual and distinguishable relationship between bounding box feature and distance. This characteristic reduces the steepness of the curve, facilitating a more stable and accurate regression model for distance estimation. In contrast, the bounding box size model presents more abrupt changes in data distribution, which could lead to higher variance in estimated distances, particularly at closer ranges where accuracy is most critical.

Given these observations, the normalized bounding box width was selected as the preferred feature for distance estimation. The

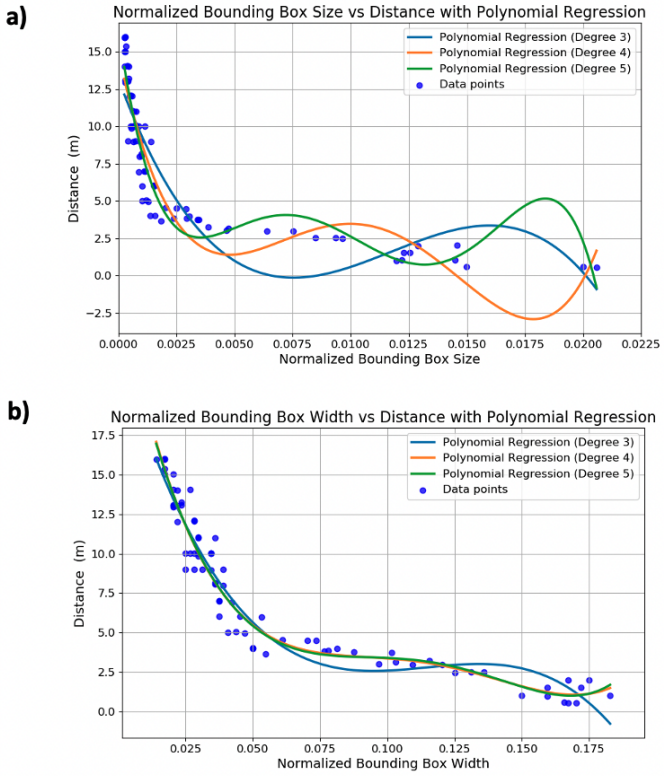

Figure 14: Empirical relationship between bounding box width and size with distance - Polynomial Regression
(a) *Normalized Bounding Box Size vs. Distance with Polynomial Regression.* ***(b)*** *Normalized Bounding Box Width vs. Distance with Polynomial Regression.*

degree-4 polynomial regression model for this feature yielded the best balance between simplicity and fit, described by:

$$d(x) = 1.538\times10^5 x^4 - 7.185\times10^4 x^3 + 1.201\times10^4 x^2 - 874.1x + 27.18 \tag{42}$$

where $d(x)$ represents the estimated distance as a function of the normalized bounding box width x. This model captures the nonlinear relationship between the bounding box width and distance, providing a reliable approximation across the range of observed distances.

References

[1] F. Nex, C. Armenakis, M. Cramer, *et al.*, "UAV in the advent of the twenties: Where we stand and what is next," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 184, pp. 215–242, Feb. 1, 2022, ISSN: 0924-2716. DOI: 10.1016/j.isprsjprs.2021.12.006. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0924271621003282 (visited on 05/24/2024).

[2] G. E. M. Abro, S. A. B. M. Zulkifli, R. J. Masood, V. S. Asirvadam, and A. Laouiti, "Comprehensive review of UAV detection, security, and communication advancements to prevent threats," *Drones*, vol. 6, no. 10, p. 284, Oct. 2022, Number: 10 Publisher: Multidisciplinary Digital Publishing Institute, ISSN: 2504-446X. DOI: 10.3390/drones6100284. [Online]. Available: https://www.mdpi.com/2504-446X/6/10/284 (visited on 05/24/2024).

[3] Kerry Chávez, Ori Swed. "Full article: Emulating underdogs: Tactical drones in the russia-ukraine war." (Jan. 8, 2024), [Online]. Available: https://www.tandfonline.com/doi/full/10.1080/13523260.2023.2257964? (visited on 10/01/2024).

[4] D. Kunertova, "The war in ukraine shows the game-changing effect of drones depends on the game," *Bulletin of the Atomic Scientists*, vol. 79, no. 2, pp. 95–102, Mar. 4, 2023, Publisher: Routledge _eprint: https://doi.org/10.1080/00963402.2023.2178180, ISSN: 0096-3402. DOI: 10.1080/00963402.2023.2178180. [Online]. Available: https://doi.org/10.1080/00963402.2023.2178180 (visited on 10/01/2024).

[5] D. Henriksen and J. Bronk, *The Air War in Ukraine: The First Year of Conflict*. Taylor & Francis, Aug. 1, 2024, 253 pp., Google-Books-ID: Gd8NEQAAQBAJ, ISBN: 978-1-04-009890-5.

[6] D. Kunertova, "Learning from the ukrainian battlefield: Tomorrow's drone warfare, today's innovation challenge," ETH Zurich, Aug. 2024, Artwork Size: 27 p. Medium: application/pdf, 27 p. DOI: 10.3929/ETHZ-B-000690448. [Online]. Available: http://hdl.handle.net/20.500.11850/690448 (visited on 09/27/2024).

[7] L.-G. Oprean, "Artillery and drone action issues in the war in ukraine," *Scientific Bulletin*, vol. 28, no. 1, pp. 73–78, Jun. 1, 2023, ISSN: 2451-3148. DOI: 10.2478/bsaft-2023-0008. [Online]. Available: https://www.sciendo.com/article/10.2478/bsaft-2023-0008 (visited on 10/01/2024).

[8] B. Zhu, J. Xu, A. Hanif Bin Zaini, and L. Xie, "A three-dimensional integrated guidance law for rotary UAV interception," in *2016 12th IEEE International Conference on Control and Automation (ICCA)*, 5, Jun. 2016, pp. 726–731. DOI: 10.1109/ICCA.2016.7505364. (visited on 04/05/2024).

[9] F. Chaumette and S. Hutchinson, "Visual servo control, part i: Basic approaches," *IEEE Robot. Autom. Mag.*, vol. 13, Jan. 1, 2006.

[10] Z. W. Lee, W. H. Chin, and H. W. Ho, "Air-to-air micro air vehicle interceptor with an embedded mechanism and deep learning," *Aerospace Science and Technology*, vol. 135, p. 108 192, Apr. 1, 2023, 7, ISSN: 1270-9638. DOI: 10.1016/j.ast.2023.108192. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1270963823000895 (visited on 05/02/2024).

[11] S. Zhang, X. Zhao, and B. Zhou, "Robust vision-based control of a rotorcraft UAV for uncooperative target tracking," *Sensors*, vol. 20, no. 12, p. 3474, Jan. 2020, 7, ISSN: 1424-8220. DOI: 10.3390/s20123474. [Online]. Available: https://www.mdpi.com/1424-8220/20/12/3474 (visited on 04/04/2024).

[12] A. Rodriguez-Ramos, A. A.-F. H. Bavle, J. Rodriguez-Vazquez, *et al.*, *Autonomous aerial robot for high-speed search and intercept applications*, 7, Dec. 10, 2021. [Online]. Available: http://arxiv.org/abs/2112.05465 (visited on 05/02/2024).

[13] K. Yang, C. Bai, Z. She, and Q. Quan, "High-speed interception multicopter control by image-based visual servoing," *IEEE Transactions on Control Systems Technology*, pp. 1–17, 2024, ISSN: 1063-6536, 1558-0865, 2374-0159. DOI: 10.1109/TCST.2024.3451293. arXiv: 2404.08296[cs]. [Online]. Available: http://arxiv.org/abs/2404.08296 (visited on 10/02/2024).

[14] A. Barišić, F. Petric, and S. Bogdan, "Brain over brawn: Using a stereo camera to detect, track, and intercept a faster UAV by reconstructing the intruder's trajectory," *Field Robotics*, vol. 2, no. 1, pp. 222–240, Mar. 10, 2022, 5, ISSN: 27713989. DOI: 10.55417/fr.2022009. [Online]. Available: https://fieldrobotics.net/Field_Robotics/Volume_2_files/Vol2_09.pdf (visited on 05/21/2024).

[15] S. Wu, R. Li, Y. Shi, and Q. Liu, "Vision-based target detection and tracking system for a quadcopter," *IEEE Access*, vol. 9, pp. 62 043–62 054, 2021, 7, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2021.3074413. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9409082 (visited on 04/04/2024).

[16] J. Wu, Z. Jin, A. Liu, L. Yu, and F. Yang, "A survey of learning-based control of robotic visual servoing systems," *Journal of the Franklin Institute*, vol. 359, no. 1, pp. 556–577, Jan. 1, 2022, ISSN: 0016-0032. DOI: 10.1016/j.jfranklin.2021.11.009. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0016003221006621 (visited on 12/04/2024).

[17] Z. Tan and M. Karaköse, "A new approach for drone tracking with drone using proximal policy optimization based distributed deep reinforcement learning," *SoftwareX*, vol. 23, p. 101 497, Jul. 1, 2023, ISSN: 2352-7110. DOI: 10.1016/j.softx.2023.101497. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352711023001930 (visited on 10/02/2024).

[18] A. Loquercio, E. Kaufmann, R. Ranftl, A. Dosovitskiy, V. Koltun, and D. Scaramuzza, "Deep drone racing: From simulation to reality with domain randomization," *IEEE Transactions on Robotics*, vol. 36, no. 1, pp. 1–14, Feb. 2020, ISSN: 1552-3098, 1941-0468. DOI: 10.1109/TRO.2019.2942989. [Online]. Available: https://ieeexplore.ieee.org/document/8877728/ (visited on 10/02/2024).

[19] Y. Zhang, Y. Hu, Y. Song, D. Zou, and W. Lin, *Back to newton's laws: Learning vision-based agile flight via differentiable physics*, Jul. 15, 2024. arXiv: 2407.10648[cs]. [Online]. Available: http://arxiv.org/abs/2407.10648 (visited on 10/02/2024).

[20] L. Lamberti, E. Cereda, G. Abbate, *et al.*, *A sim-to-real deep learning-based framework for autonomous nano-drone racing*, Dec. 14, 2023. arXiv: 2312.08991. [Online]. Available: http://arxiv.org/abs/2312.08991 (visited on 10/12/2024).

[21] D. Zhang, A. Loquercio, J. Tang, T.-H. Wang, J. Malik, and M. W. Mueller, *A learning-based quadcopter controller with extreme adaptation*, Sep. 19, 2024. arXiv: 2409.12949. [Online]. Available: http://arxiv.org/abs/2409.12949 (visited on 10/12/2024).

[22] Y. Liu, X. Li, T. Wang, Y. Zhang, and P. Mei, "Quantitative stability of quadrotor unmanned aerial vehicles," *Nonlinear Dynamics*, vol. 87, no. 3, pp. 1819–1833, Feb. 1, 2017, ISSN: 1573-269X. DOI: 10.1007/s11071-016-3155-9. [Online]. Available: https://doi.org/10.1007/s11071-016-3155-9 (visited on 10/02/2024).

[23] K. Yang and Q. Quan, "An autonomous intercept drone with image-based visual servo," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 8, May 2020, pp. 2230–2236. DOI: 10.1109/ICRA40945.2020.9197539. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9197539?casa_token=9L4sBEJm0N8AAAAA:3ruz2Pvbtn7ePPiD4YreDu-Ldd6lletVfHkdnb3m3041FWpqmZuOtZN7u51Rbmg-ZCqb27g (visited on 04/08/2024).

[24] H. Yan, K. Yang, Y. Cheng, Z. Wang, and D. Li, *Precise interception flight targets by image-based visual servoing of multicopter*, Sep. 26, 2024. arXiv: 2409.17497. [Online]. Available: http://arxiv.org/abs/2409.17497 (visited on 11/11/2024).

[25] A. Barisic, M. Car, and S. Bogdan, "Vision-based system for a real-time detection and following of UAV," in *2019 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED UAS)*, 8, Nov. 2019, pp. 156–159. DOI: 10.1109/REDUAS47371.2019.8999675. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8999675/metrics#metrics (visited on 04/04/2024).

[26] J. Pestana, J. L. Sanchez-Lopez, S. Saripalli, and P. Campoy, *Computer vision based general object following for GPS-denied multirotor unmanned vehicles*. Jun. 1, 2016, 1886 pp., 7, ISBN: 978-1-4799-3274-0.

[27] P. M. Wyder, Y.-S. Chen, A. J. Lasrado, *et al.*, "Autonomous drone hunter operating by deep learning and all-onboard computations in GPS-denied environments," *PLOS ONE*, vol. 14, no. 11, e0225092, Nov. 18, 2019, 8, ISSN: 1932-6203. DOI: 10.1371/journal.pone.0225092. [Online]. Available: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0225092 (visited on 05/16/2024).

[28] T. H. Chung, G. A. Hollinger, and V. Isler, "Search and pursuit-evasion in mobile robotics," *Autonomous Robots*, vol. 31, no. 4, pp. 299–316, Nov. 1, 2011, ISSN: 1573-7527. DOI: 10.1007/s10514-011-9241-4. [Online]. Available: https://doi.org/10.1007/s10514-011-9241-4 (visited on 05/08/2024).

[29] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, *You only look once: Unified, real-time object detection*, May 9, 2016. DOI: 10.48550/arXiv.1506.02640. arXiv: 1506.02640. [Online]. Available: http://arxiv.org/abs/1506.02640 (visited on 11/26/2024).

[30] J. Terven, D.-M. Córdova-Esparza, and J.-A. Romero-González, "A comprehensive review of YOLO architectures in computer vision: From YOLOv1 to YOLOv8 and YOLO-NAS," *Machine Learning and Knowledge Extraction*, vol. 5, no. 4, pp. 1680–1716, Dec. 2023, Number: 4 Publisher: Multidisciplinary Digital Publishing Institute, ISSN: 2504-4990. DOI: 10.3390/make5040083. [Online]. Available: https://www.mdpi.com/2504-4990/5/4/83 (visited on 11/26/2024).

[31] M. Sohan, T. Sai Ram, and C. V. Rami Reddy, "A review on YOLOv8 and its advancements," in *Data Intelligence and Cognitive Informatics*, I. J. Jacob, S. Piramuthu, and P. Falkowski-Gilski, Eds., Singapore: Springer Nature, 2024, pp. 529–545, ISBN: 978-981-9979-62-2. DOI: 10.1007/978-981-99-7962-2_39.

[32] Y. Zheng, Z. Chen, D. Lv, Z. Li, Z. Lan, and S. Zhao, "Air-to-air visual detection of micro-UAVs: An experimental evaluation of deep learning," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1020–1027, Apr. 2021, Conference Name: IEEE Robotics and Automation Letters, ISSN: 2377-3766. DOI: 10.1109/LRA.2021.3056059. [Online]. Available: https://ieeexplore.ieee.org/document/9343737 (visited on 11/26/2024).

[33] A. Lukezic, T. Vojir, L. C. Zajc, J. Matas, and M. Kristan, "Discriminative correlation filter with channel and spatial reliability," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI: IEEE, Jul. 2017, pp. 4847–4856, ISBN: 978-1-5386-0457-1. DOI: 10.1109/CVPR.2017.515. [Online]. Available: http://ieeexplore.ieee.org/document/8099998/ (visited on 11/26/2024).

[34] A. A. AlMansoori, I. Swamidoss, S. Sayadi, and A. Almarzooqi, "Analysis of different tracking algorithms applied on thermal infrared imagery for maritime surveillance systems," in *Artificial Intelligence and Machine Learning in Defense Applications II*, vol. 11543, SPIE, Sep. 20, 2020, pp. 30–40. DOI: 10.1117/12.2574793. [Online]. Available: https://www.spiedigitallibrary.org/conference-proceedings-of-spie/11543/1154308/Analysis-of-different-tracking-algorithms-applied-on-thermal-infrared-imagery/10.1117/12.2574793.full (visited on 11/26/2024).

[35] K. Ullah, I. Ahmed, M. Ahmad, and I. Khan, "Comparison of person tracking algorithms using overhead view implemented in OpenCV," in *2019 9th Annual Information Technology, Electromechanical Engineering and Microelectronics Conference (IEMECON)*, Mar. 2019, pp. 284–289. DOI: 10.1109/IEMECONX.2019.8877025. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8877025 (visited on 11/26/2024).

[36] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, Mar. 1, 1960, ISSN: 0021-9223. DOI: 10.1115/1.3662552. [Online]. Available: https://asmedigitalcollection.asme.org/fluidsengineering/article/82/1/35/397706/A-New-Approach-to-Linear-Filtering-and-Prediction (visited on 11/26/2024).

[37] H. Jabbari Asl, G. Oriolo, and H. Bolandi, "An adaptive scheme for image-based visual servoing of an underactuated uav," *International Journal of Robotics and Automation*, vol. 29, Jan. 1, 2014, 7. DOI: 10.2316/Journal.206.2014.1.206-3942.

[38] Shuvrangshu Jana, Open the ORCID record for Shuvrangshu Jana[Opens in a new window], *et al.* "Interception of an aerial manoeuvring target using monocular vision | robotica | cambridge core." (), [Online]. Available: https://www.cambridge.org/core/journals/robotica/article/interception-of-an-aerial-manoeuvring-target-using-monocular-vision/45EC80975FCD5727049599E55622B446 (visited on 12/06/2024).