

Utilizing a Quadcopter For Air-to-Air Interception of a Loitering Muniton

Thesis Report

Yamac Birol



Utilizing a Quadcopter For Air-to-Air Interception of a Loitering Munition

Thesis Report

Thesis report

by

Yamac Birol

to obtain the degree of Master of Science
at the Delft University of Technology
to be defended publicly on November 27, 2024 at 09:00

Thesis committee:

Chair:	Dr. Erik-jan van Kampen
Supervisors:	Dr. Ewoud J.J. Smeur
External examiner:	Dr. Frits de Prenter
Place:	Faculty of Aerospace Engineering, Delft
Project Duration:	December, 2023 - November, 2024
Student number:	5017300

An electronic version of this thesis is available at <https://repository.tudelft.nl/>.



Copyright © Yamac Birol, 2024
All rights reserved.

Preface

This report concludes my 11 months of preparation and dedicated effort for my thesis, marking the end of my 5-year journey at TU Delft. Over the course of this thesis, I have gained remarkable insights, deepened my knowledge in UAV technology—particularly in the counter-UAV domain—and honed my engineering judgment for the future. Beyond technical growth, this experience has allowed me to discover myself, pushing my limits to the fullest. These invaluable lessons and memories will stay with me throughout my life, and I am proud to say that completing this thesis has helped me grow into the best version of myself.

I am deeply grateful to those who have supported and guided me throughout this journey. First and foremost, I extend my heartfelt thanks to Dr. Ewoud Smeur, whose mentorship pushed me to reach my potential. He imparted not only technical knowledge but also critical engineering skills, demonstrating immense patience and understanding. Your encouragement and guidance have been instrumental in my growth, and I sincerely thank you for everything. I am also profoundly grateful to Erik van der Horst for sharing his expertise and invaluable insights into hardware systems. His unwavering support, especially during challenging moments in the lab, was not only a source of solutions but also a great learning experience. Those hands-on sessions were transformative and irreplaceable.

I would also like to thank my friends at TU Delft, who stood by me throughout this journey. Their companionship and encouragement kept me motivated and determined to persevere through the challenges.

Lastly, I am eternally grateful to my parents, who have supported me every step of the way. They have stood by my decisions, offering invaluable advice and unwavering encouragement. Their love, guidance, and belief in me have been my greatest sources of inspiration and motivation. I owe my achievements to their endless support and leadership.

To everyone who has been part of this journey—thank you. This accomplishment would not have been possible without you.

*Yamac Birol
Delft, Tuesday 19th November, 2024*

Contents

1	General Introduction	1
1.1	Research Questions	1
1.2	Structure of the Report	2
I	Preliminary Analysis	3
2	Literature Review	4
2.1	Executive Summary	4
2.2	Introduction	6
2.3	Detection and Tracking of a Non-cooperative UAV	6
2.4	Interception Guidance	11
2.5	Control	18
2.6	Revised Research Questions	20
II	Scientific Article	21
3	Control & Simulation Msc Thesis Paper Template	22
III	Additional Results	35
4	More Results	36
4.1	Hardware and Software	36
4.2	Cyberzoo Experiments	37
4.3	Visibility	39
IV	Closure	41
5	Conclusions & Recommendations	42
5.1	Conclusions.	42
5.2	Recommendations	43
	References	46

Nomenclature

List of Abbreviations

AAGD	Average Absolute Gray Difference	PID	Proportional Integral Derivative
AP	Average Precision	PN	Proportional Navigation
CNN	Convolutional Neural Network	PPN	Pure Proportional Navigation
DOF	Degree of Freedom	R-CNN	Recurrent Convolutional Neural Network
FPN	Feature Pyramid Network	RANSAC	Random Sample Consensus
GST	Generalized Structure Tensor	RNN	Recurrent Neural Network
GTPN	Generalized True Proportional Navigation	SIFT	Scale Invariant Feature Transform
HOG	Histogram of Gradients	SSD	Single Shot Detector
INDI	Incremental Nonlinear Dynamic Inversion	SVM	Unmanned Aerial System
LCM	Local Contrast Method	TLLCM	Tri-layer Local Contrast Measure
LOS	Line of Sight	TPN	True Proportional Navigation
mAP	Mean Average Precision	TVC	Thrust Vectoring Control
MOG	Mixture of Gaussians	UAS	Unmanned Aerial System
NDI	Nonlinear Dynamic Inversion	UAV	Unmanned Aerial Vehicle
PI	Proportional Integral	YOLO	You Look Only Once

General Introduction

Drones are having a lot of attention from both academia and industry for their ability to complete visual tasks efficiently such as surveillance and inspection, obstacle avoidance, goods/medical delivery. However, this comes with several consequences such as misuse of commercial drones in restricted air spaces as well as their use on the battlefield as kamikaze drones. Examples for these concerns could include the commercial misuse of a drone around the Gatwick Airport in London which caused severe delays [1], the use of drones to deliver bombs by ISIS [2], the attempted attack on the Venezuelan president by the use of 2 drones carrying explosives [3] and, more relevant to this research, the use of Iranian Shahed kamikaze drones by Russia in the ongoing Russian-Ukrainian war [4] as well as the Houthis [5]. This research focuses on loitering munitions, which refer to unmanned aerial vehicles (UAVs) designed to loiter in an area for extended periods, seeking targets autonomously or under remote control. These systems are classified as non-cooperative targets, as they operate independently and often unpredictably, posing significant challenges for interception and defense. At the moment, there are several ways of countering these threats such as jammer systems, net carrying UAS or nets launched from ground [6], GPS spoofing [7], a cheaper UAS controlled by a human dropping onto a more expensive UAS [4]. These methods are either too complex to be generalized or have other implications to systems in the same environment. Thus, neutralizing the loitering munition by slamming into it with a cheaper drone is deemed more feasible. Nevertheless, in order to counter these drones, detecting them is essential which can be done in several ways, including, analysing the radio frequency waves used by the communication between the drone and its operator, acoustic sensors sensing the inherent sound of the drone mostly originating from the propeller sound and the vibrations of the drone, optical sensors which include cameras to detect target drones and, use of radar for detection [8]. This research will mostly study the air to air scenarios where a UAS carrying a radar and a camera will be used to detect, track and intercept another target UAS, more specifically a loitering munition.

1.1. Research Questions

Methods for air-to-air detection, tracking and interception of a non-maneuvering UAV is detailed in the Literature Review (Chapter 2). In this analysis it can be followed that there are several methods for detecting, tracking and intercepting a loitering munition, nevertheless a complete method for autonomous detection, tracking and interception from start to interception is not detailed in academia or open-source. In order to come up with a solution for air-to-air interception of a loitering munition, following research objective and questions are detailed:

Research Objective

The main aim of the research is to design an algorithm that will run onboard a quadcopter which will detect and intercept a loitering kamikaze drone using an initial location from a ground sensor, with relatively high range, and an onboard camera as well as a distance value to the kamikaze drone (simulating an onboard radar) which both have relatively short range.

Research Question Main

How to design a real time algorithm to detect, track and intercept loitering munition onboard a quadcopter using infrared camera(s) and target distance information?

Research Question 1

How to implement the most efficient computer vision algorithm in terms of performance and speed for air to air detection of small drones with an infrared camera?

Research Question 2

How to extract position, velocity and acceleration information of a target drone using infrared camera and assuming a known distance to target drone?

Research Question 3

What is an optimal interception guidance law for intercepting UAS targets with the use of another UAS?

To keep the research in a reasonable time frame, several assumptions are made. These include:

- There is a ground system which detects the target drone from a further range and provides the interceptor drone with an initial location of the target drone.
- There is an onboard radar which provides the distance of the interceptor to the target drone. This radar is simulated by feeding the interceptor with the 3D distance between the target and the interceptor in real-time.

1.2. Structure of the Report

The structure of this thesis is as follows:

- Chapter 2 gives the state of the art methods in this research, comparison of these methods and the background theory behind the chosen method.
- Chapter II is a standalone document detailing the related work, methods used in the research, experiments to validate the methods and results and conclusions from the research.
- Chapter 4 details the additional results and work that were not put into the scientific article but still compliments the research.
- Chapter 5 presents the final conclusions and recommendations for the research.

Part I

Preliminary Analysis

Literature Review

2.1. Executive Summary

As drones become increasingly widespread and user-friendly, their presence around critical infrastructure for inspection and surveillance purposes is now commonplace. However, the accessibility of drone technology also introduces risks, as misuse of drones for harmful or hostile purposes becomes more frequent. This risk is especially apparent in ongoing conflicts, such as in Ukraine and Palestine, where commercial drones are being weaponized. To address these threats, this research proposes a counter-drone system utilizing a quadcopter designed to detect, track, and intercept loitering munitions. By providing an effective and responsive defense mechanism against adversary drones, this approach contributes significantly to advancing safer skies and protecting sensitive assets. In this review the literature is investigated to identify improvements on the current air-to-air interception methods.

Preliminaries

Air-to-air interception of adversary UAV requires a combination of advanced detection, tracking, and guidance systems. Effective interception involves predicting the target's motion, planning the interceptor's trajectory, and executing maneuvers to neutralize the threat.

The process typically involves multiple stages, starting with the detection of the loitering munition. Once detected, the target must be tracked to maintain its position relative to the interceptor drone. As the interceptor approaches the target, it must be guided using a control law to adjust its trajectory, ensuring a successful interception. Various guidance laws are employed to optimize the interception, considering the dynamics and/or path and motion of both the interceptor and the target. The interception of maneuvering targets also requires real-time adjustments to the flight path of the interceptor based on dynamic feedback, integrating trajectory generation methods that respect the system's physical limitations. These methods aim to minimize the terminal miss distance and ensure that the interceptor can engage the target effectively, whether in a head-on or chasing geometry. Thus, the research is divided into 2 main aspects:

- Methods for detecting and tracking a UAV
- Guidance laws for air-to-air interception of a UAV

The UAV Detection and Tracking Algorithm

The detection and tracking of small Unmanned Aerial Systems (UAS) using monocular RGB cameras is a key area of research in this study, focusing on air-to-air detection applications. Existing methods for detecting small drones fall into two categories: conventional and deep learning approaches. Conventional methods are often based on prior knowledge of the object of interest, utilizing techniques such as background subtraction, local contrast, and noise estimation. These methods excel in controlled settings but face limitations in complex environments, where their reliance on predefined object characteristics constrains adaptability [9], [10].

In contrast, deep learning approaches, particularly convolutional neural networks (CNNs), offer more advanced detection capabilities through model-based learning. One-stage detectors like YOLO and SSD balance detection accuracy and speed, while two-stage detectors, including R-CNN and its derivatives,

achieve higher accuracy at the cost of increased computational demands [11], [12]. The trade-off between appearance-based, motion-based, and hybrid detectors also influences detection performance. Appearance-based detectors focus on visual features, with methods that range from global models, which analyze the entire image, to local models that focus on predefined regions of interest [10]. Motion-based detectors, alternatively, identify moving objects through temporal changes, using techniques such as frame differencing and optical flow, though these methods risk detecting irrelevant objects due to camera motion or other moving background elements [13]. Hybrid models combine appearance and motion-based approaches to enhance real-time detection capabilities [10].

For tracking, both deep learning and optical flow techniques are widely used. Optical flow provides real-time tracking by analyzing pixel movement, with sparse optical flow often used to reduce processing requirements. Kalman filters are applied to maintain tracking accuracy during detection lapses, enabling continuous estimation of the target's position and velocity [13].

Interception Guidance

The interception guidance for drones can be achieved using both simple guidance laws and more complex trajectory generation algorithms. A basic approach involves keeping the target in the center of the field of view and approaching it with constant divergence. This method has been demonstrated in applications like landing towards a stationary target but is less applicable for moving targets. More advanced trajectory generation techniques, often used in homing missile guidance, have been explored extensively in the literature. These techniques, including Proportional Navigation (PN) and its variants, are widely regarded as optimal for minimizing terminal miss distance when intercepting non-maneuvering targets [14], [15], [16], [17].

PN and its derivatives, such as Pure Proportional Navigation (PPN), True Proportional Navigation (TPN), and Generalized True Proportional Navigation (GTPN), adjust the interceptor's lateral acceleration based on the rate of change in the line of sight. PPN has been shown to provide effective interception behavior for simple target maneuvers. However, PN requires specific conditions, such as the interceptor closing in on the target and having an acceleration greater than or equal to that of the target [14], [17].

The methods proposed by Hehn et al. [18], [19] and Mellinger et al. [20] provide foundational insights into trajectory generation and control for interception maneuvers. Hehn et al. propose an approach that focuses on real-time, feasible trajectory planning that respects dynamic constraints, employing Pontryagin's minimum principle and a bang-bang control strategy to ensure time-optimal solutions. This method guarantees that the quadcopter reaches a designated interception point while minimizing overshoot and excessive deceleration. In contrast, Mellinger et al. propose a work that introduces a sequence of trajectory segments and dynamic mode switching between controllers for various flight phases, allowing the quadcopter to perform precise, aggressive maneuvers. These controllers are adjusted in real time to adapt to changes in the environment, making this approach highly suitable for interception tasks. Also explored is the Model Predictive Control (MPC) as proposed by Ahmet et al. [21] for air-to-air interception which incorporates terminal constraints to ensure engagement at a desired impact angle, prioritizing visual alignment within the camera frame. By modifying the objective function to minimize maneuvering requirements near the end of the trajectory and tailoring the prediction horizon to vehicle limits, the approach enhances interception feasibility and robustness against processing delays.

Benchmark Testing and Future Research

The detection, tracking and interception guidance algorithms are developed in Matlab and tested first in a simulation. After satisfactory results from the simulation are achieved, series of flight tests are conducted on a quadcopter to test the real time applicability of the algorithm. For the onboard implementation of the algorithms, C and Python programming languages are used for their efficiency. For the flight test all the available facilities are used including the Cyberzoo in TU Delft campus for indoor testing and the Unmanned Valley in Valkenburg for outdoor testing.

2.2. Introduction

In this research an onboard interception guidance algorithm for intercepting loitering kamikaze drones is developed. The scope of the algorithm includes detecting, tracking and intercepting the kamikaze drone using onboard and ground based sensors. The onboard sensors include a camera and a radar. This research will focus on the data from the camera while the radar data is not actually collected onboard and is simulated by feeding the distance value to the interceptor drone. The ground based sensor is assumed to only give an initial location of the kamikaze drone. The main aim of the research is to design an algorithm that is run onboard a quadcopter which detect and intercept a loitering kamikaze drone using an initial location from a ground sensor, with relatively high range, and an onboard camera as well as a distance value to the kamikaze drone (simulating an onboard radar) which both have relatively short range.

There are several challenges that are addressed in this research.

Firstly the target drone needs to be detected. The initial location of the loitering munition is provided by a ground based system which make use of infrared cameras and acoustic sensors and will have a much greater detection range. The detection of drones from ground based systems including various sensors has been studied widely by academia [22], [23], [24]. The approaches of ground based detection are then translated into air-to-air detection of target drones which requires a transition from static measurements to dynamic measurements of the target drone, making detection more challenging.

Air-to-air detection of small drones is a recent and relevant research area. As air to air detection require measurements onboard another drone, the sensor suit available is limited related to mass and power requirements of small drones and mostly optical sensors are used [8]. Also it is seen that mostly RGB cameras (making use of the visible light range) have been studied in the literature for the purpose of drone detection. The challenges associated with air-to-air detection via an optical sensor include the dynamic measurement of target drone (requiring ego-motion analysis), target drones covering only a few pixels in the image limiting the detection range, other moving objects which have similar size as a drone such as birds making false positives a problem [25]. Initially it was determined that infrared camera(s) would be used instead of RGB cameras as they provide more contrast between background and the target making detection easier for most cases. Nevertheless, after further investigation during the research the focus shifted to RGB cameras and the reason is further elaborated in the literature study.

Secondly, the target drone needs to be tracked and necessary parameters for the guidance need to be extracted from it such as the position, velocity and possibly the acceleration data. This is done, in the literature, as a part of the detection [26] as well as separately [13], nevertheless both approaches are relevant for this research.

Finally the detected and tracked target drone needs to be intercepted by the interceptor drone, which could be done by a wide pool of methodologies ranging from simple guidance laws to complicated trajectory generation algorithms depending on the available sensor suit. Intercepting a target drone with the use of an interceptor drone follows a similar logic to a homing missile intercepting a non-stationary target, especially the terminal stages of homing missiles are relevant as that is the typical range for interception in this study. Homing missiles use Proportional Navigation (PN) law in its terminal phase and a lot of research has been done in this area due to PN's optimality, with assumptions, in the sense of minimum miss distance as well as its ease of implementation [14]. Some literature has adapted these missile researches to drones which have similar trajectories but different actuation [6].

All of these areas were studied extensively in the literature in the context of detecting, tracking and intercepting target UAS but mostly separately. This research aims to combine the most efficient ways of detection, tracking and interception of a target UAS into one research and help pave the way to a more complete air-to-air counter drone system.

2.3. Detection and Tracking of a Non-cooperative UAV

The previous sections highlighted the growing importance of unmanned aerial vehicles (UAVs) in both civil and defense aerospace sectors, as well as the critical need for effective countermeasures to prevent attacks and mitigate risks posed by adversarial or unauthorized UAVs. This chapter introduces the theoretical foundation for detecting and tracking a non-cooperative UAV. To provide context, an overview of the extensive research conducted in this field will be presented, summarizing key methodologies from relevant

studies and publications. The discussion will cover various detection and tracking approaches, detailing the techniques commonly used.

2.3.1. Detection Strategy

Detecting loitering kamikaze drones is crucial for implementing effective countermeasures. Several detection methods are available, each leveraging different characteristics of UAVs. One approach involves analyzing radio frequency (RF) signals to intercept communications between the drone and its operator, allowing early detection and tracking [8]. Acoustic sensors provide another solution by capturing the distinctive sounds produced by drone propellers and structural vibrations, which can be useful in detecting low-flying or stealthy UAVs [27]. Optical sensors, including various types of cameras, enable visual detection and identification, allowing for precise tracking of target drones based on their visual signatures. Additionally, radar systems offer robust detection capabilities by capturing the drone's movement and position, even under challenging visibility conditions [8]. Advanced methods may integrate multiple sensor types—such as combining RF analysis with optical or radar detection—to improve detection accuracy and reliability in complex environments.

For accurate distance estimation, radar is an effective tool due to its capacity to measure range precisely by calculating the time delay between signal transmission and reflection. This ability makes radar highly reliable for detecting an object's position in space. However, radar systems often struggle with accurately determining the orientation or angle of an object, especially in single-snapshot measurements where angle accuracy depends on the antenna array size and signal processing techniques used. Radar systems like millimeter-wave radar can provide accurate range data but may have limitations in angular accuracy due to the resolution of the radar's direction-of-arrival (DoA) estimation [28]. Many research efforts focus on improving angular resolution, often by employing advanced algorithms or multiple snapshots, but these solutions still fall short in dynamic or noisy environments [29]. Consequently, radar is usually combined with other sensing methods—such as optical or infrared sensors—to enhance orientation accuracy and achieve a more comprehensive understanding of an object's movement and position [29].

Combining a camera and radar system offers an optimal solution for accurately determining the location and velocity of a target drone. In this setup, the camera excels in providing a precise estimate of the target drone's orientation relative to the interceptor drone, thanks to its high angular resolution and detailed visual feedback. Meanwhile, the radar complements this by delivering an accurate distance measurement, allowing for reliable range tracking even in conditions with low visibility or challenging lighting. This combination of camera-based orientation with radar-based distance estimation enables a robust and precise tracking system, improving the interceptor drone's ability to pursue and engage with the target.

2.3.2. Choice of Electromagnetic Wavelength

The choice of detection method has been narrowed down to using cameras, rather than alternatives such as acoustic sensors or radio frequency detectors. However, it remains essential to select the optimal electromagnetic wavelength for imaging, as several camera options are available that operate across different bands. Based on Follansbee et al. [30] study, the following points outline the performance characteristics of various wavelengths for drone detection:

- **Visible (RGB) (0.4–0.7 μm):** Widely available sensors that offer high diffraction-limited resolution, low size, weight, and power (SWaP) requirements, and a small pixel pitch. This results in high image quality and photon availability, making RGB sensors effective under clear lighting conditions. RGB also maintains a consistent signal-to-noise ratio (SNR) with range, making it reliable for recognition and identification. However, object-sky contrast can be a limitation, especially in low-contrast lighting environments.
- **Near Infrared (NIR) (0.7–1 μm):** Shares advantages with the visible band, including high resolution, low SWaP, and small pixel pitch. NIR can improve contrast slightly in low-light conditions, enhancing its utility in some environments, although it provides minimal improvement over RGB under clear skies. SNR remains constant with distance, similar to RGB, which makes it suitable for short- to medium-range detection.
- **Short-Wave Infrared (SWIR) (1–1.7 μm):** Excels in degraded visual environments, such as haze, smoke, or fog, due to increased atmospheric transmission. SWIR also enhances target-sky contrast, providing greater detection range under low-visibility conditions. However, SWIR sensors often have

larger pixel pitches, resulting in lower image resolution, and may be limited by object-sky contrast at longer distances.

- **Extended Short-Wave Infrared (eSWIR) (2–2.5 μm):** Provides the highest atmospheric transmission among the bands studied, which improves target-sky contrast and diffraction-limited resolution over SWIR. This capability makes eSWIR suitable for highly degraded environments where visibility is a challenge. However, eSWIR's large pixel pitch limits detection resolution, and SNR declines significantly with distance, reducing its effectiveness for long-range detection.

After reviewing the strengths and limitations of each spectral band, RGB (visible light) presents a balanced solution for drone detection in typical operational environments. RGB cameras offer reliable performance with high resolution, consistency in SNR over varying distances, and availability in a cost-effective format. While SWIR and eSWIR provide advantages in extreme visibility conditions, RGB's versatility across diverse lighting and atmospheric conditions makes it the optimal choice for general drone detection applications.

2.3.3. Conventional Methods For Drone Detection From a Camera

The conventional methods for small target detection, including those used in drone detection, often rely heavily on prior knowledge of the target and the environment, which can limit their adaptability in dynamic or complex settings [12]. These methods, designed around specific assumptions, may not generalize well to unknown environments or targets [10].

Background subtraction techniques, such as the top-hat algorithm [31], max-mean, and max-median algorithms [32], are widely used to identify small targets by comparing consecutive frames and isolating the moving object from the static background. While they can be effective in environments where the background is relatively stable, they struggle in dynamic or cluttered scenes, especially under varying lighting conditions or with objects that blend into their surroundings. More advanced methods, such as the average absolute gray difference (AAGD) [33] and generalized structure tensor (GST)-based methods [34], attempt to enhance contrast and detect small targets by suppressing the background or analyzing local object textures, respectively. However, these methods still rely on prior assumptions, which makes them less robust in unfamiliar scenarios.

To overcome some of these limitations, noise estimation techniques treat the small target as a special form of noise and separate it from the background using noise models. The Mixture of Gaussians (MoG) model [35], for instance, models the small target as a sparse component of the background noise, allowing for target detection in complex environments. Despite being more flexible than traditional background subtraction methods, noise estimation still requires accurate modeling of both the target and the environment, making it difficult to apply in highly variable conditions.

Local contrast-based methods, such as the Local Contrast Measure (LCM)[36] and tri-layer local contrast method (TLLCM), enhance small targets by measuring pixel contrasts within localized regions of the image using sliding windows. These methods rely on adaptive thresholds to distinguish targets from the background, but their performance can degrade in highly variable environments where local contrasts may not be easily distinguishable. Furthermore, they often require fine-tuning for each specific setting, limiting their generalization.

While these conventional methods have shown promise in controlled environments, they struggle to maintain consistent performance in more complex or dynamic real-world conditions. Their reliance on expert prior knowledge about the target and environment means they are not easily adaptable to new or unseen scenarios. As a result, their ability to detect small targets, such as drones, beyond the assumptions of the prior knowledge is often limited. A data-driven approach, which learns from large datasets of various targets and environments, has the potential to overcome these challenges by providing greater generalization and adaptability, enabling the detection of unknown targets under more diverse conditions.

2.3.4. Deep Learning Methods For Detection From a Camera

Neural networks, particularly deep learning models, have shown significant promise in the detection and classification of drones in various environments. Traditional methods for drone detection, such as object feature extraction (e.g., Histogram of Oriented Gradients (HOG), Scale-Invariant Feature Transform (SIFT)) followed by classification through machine learning algorithms like Support Vector Machines (SVM) or

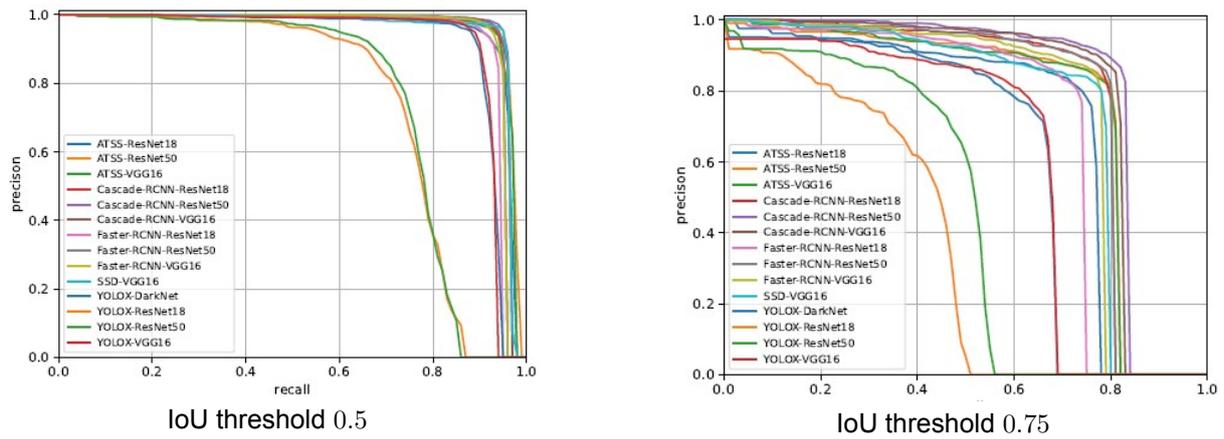


Figure 2.1: Precision comparison of several CNNs for drone detection [26]

AdaBoost, often fall short when handling the complexity and variability of UAV detection [37]. These conventional methods rely heavily on precise feature engineering, which can be computationally expensive and may struggle to adapt to new or unknown scenarios. In contrast, deep learning approaches, such as Convolutional Neural Networks (CNNs), have been applied to UAV detection with remarkable success. By directly learning relevant features from raw pixel data, CNNs can automatically adapt to various object appearances, backgrounds, and lighting conditions, making them more robust and versatile for detecting drones in dynamic environments.

Comparison of Models

CNNs give promising accuracy and precision results for drone detection. Nevertheless, there are a significant amount of CNNs to choose from with each having advantages and disadvantages. Based on the paper by Ye Zheng et al. [9] several CNNs are compared in order to come up with an optimal algorithm for target drone detection.

Single-Shot Detector (SSD)

As a one-stage detector, SSD is recognized for its high computational efficiency, particularly in its SSD512 variant, which offers stable and reliable performance. Given its relatively lightweight design, SSD is a good choice for applications with limited computational resources, such as real-time drone detection on mobile platforms.

RetinaNet

Another one-stage detector, RetinaNet achieves a balance between computational efficiency and accuracy similar to SSD. Its use of a focal loss function addresses the imbalance between foreground and background, making it effective for drone detection in cluttered environments. RetinaNet is a strong choice for resource-constrained applications, providing stable detection performance with relatively low computational demand.

YOLOv3

Known for its speed, YOLOv3 is the fastest algorithm in this comparison, providing real-time detection capabilities. This makes it highly suitable for fast-moving drone detection scenarios where quick response times are crucial. YOLOv3 outperforms models like RefineDet and Faster R-CNN in terms of speed, making it ideal for applications requiring both accuracy and low latency [38].

RefineDet

RefineDet, another one-stage detector, aims to enhance detection accuracy by refining object boundaries at an intermediate stage. While faster than some two-stage detectors, it does not match YOLOv3 in speed. This architecture can be a good compromise for applications prioritizing accuracy over speed, though it may not be ideal for real-time drone tracking in highly dynamic environments.

Faster R-CNN

As a two-stage detector, Faster R-CNN separates the tasks of region proposal and object detection, leading to higher accuracy but lower speed compared to one-stage detectors. It is suitable for applications with ample computational resources and less critical speed requirements, such as in surveillance-based drone detection where precise localization is necessary.

Feature Pyramid Network (FPN)

Another two-stage model, FPN improves the detection of objects at different scales by using a pyramid structure to extract features at multiple resolutions. For drone detection, where target sizes can vary significantly with distance, FPN's scale-invariant approach enhances accuracy but requires more computational power than one-stage models, making it suitable for settings with sufficient resources.

Cascade R-CNN

Cascade R-CNN, also a two-stage model, is noted for achieving high average precision (AP), making it an excellent choice in applications where accuracy is paramount. Its stable and superior performance across various backgrounds and target scales makes it suitable for drone detection in complex and varied environments. However, its computational demands are higher, limiting its use in real-time applications on mobile platforms.

Grid R-CNN

The Grid R-CNN is one of the most accurate models, achieving the highest AP among the models compared. It excels in detecting small targets, like drones, even in complex backgrounds and with varying scales, which is ideal for challenging drone detection tasks. However, it is the most time-consuming model due to its intricate grid-based refinement mechanism, best suited for high-precision applications where speed is less critical, such as post-event analysis or drone identification in static images.

Main Takeaways

- For real-time drone detection with limited computational resources, **YOLOv3** is typically the best option due to its speed and reliable performance.
- For applications with a moderate computational budget and a focus on accuracy, **SSD512** or **RetinaNet** provide a balance between speed and stability.
- For high-accuracy, resource-intensive applications, **Cascade R-CNN** or **Grid R-CNN** offer superior performance but at the cost of higher computational demands, making them suitable for offline processing or high-powered systems.

Several research has been conducted to achieve a good balance between detection speed and accuracy. Selected methods are presented in this study. A prominent research has been conducted by Hanqing Guo et al. [10] where a combined global-local detection method leverages both motion and appearance-based classifiers to enhance accuracy and adaptability. The method begins with a global detector, where a YOLOv5s model, optimized for high-confidence, appearance-based detection, scans the full image. This global approach is complemented by a motion-based module that activates if the appearance detector does not confidently detect a target. Motion compensation in this module relies on grid-based keypoints to estimate homography, with frame alignment achieved through a pyramidal Lucas-Kanade method and RANSAC for outlier rejection. Moving objects are then segmented via frame differencing, followed by post-processing with morphological operations and connected component analysis to filter out non-target regions.

When the global detector locates a target, a local detector refines the search, focusing on a cropped area around the target to reduce computational load. This local detector utilizes a low-confidence threshold for appearance-based detection, while a motion-based classifier assesses target movement through velocity and angle variations. A Kalman filter estimates the target's position for an adaptive search region, predicting future frames to keep the target centered. A detector switcher manages transitions between global and local detectors: switching from global to local upon initial detection and reverting if successive local detections fail, leveraging the assumption that the target has not moved substantially.

This method is particularly robust in challenging conditions, using adaptive search regions and classifier switching to handle dynamic scenes, thus offering a balance between accuracy and computational

efficiency.

Another noteworthy method for drone detection is presented by Jie Zhao et al. [11] where The YOLO-ViT model is integrated with the MobileViT network as its backbone, leveraging both CNNs and transformers to enhance feature extraction for drone detection. This combination generates multi-scale feature maps with detailed information by capturing both local and global features within an image, yet operates with fewer parameters and a simplified training process. The architecture is built with a C3-PANet neck network, a multi-scale feature fusion structure that includes key components like the SPPCSPC module, CARAFE upsampling, C3 layers, and MP downsampling. This configuration aggregates information across different layers in a bottom-up manner, effectively merging fine details from feature maps and enhancing the model's detection accuracy for small, complex targets.

Despite MobileViT's lightweight design, which could reduce model accuracy, the C3-PANet architecture optimizes the feature propagation path and perceptual field, enhancing small-target detection while mitigating information loss. K-Means++ clustering further improves performance by refining anchor box sizes to better accommodate multi-scale and occluded small targets typical in aerial UAV imagery. The clustering of anchor boxes ensures appropriate sizing, minimizing the risk of target loss due to overly large anchors.

This approach not only preserves accuracy but also achieves computational efficiency. Experimental results indicate that YOLO-ViT reduces parameters and computational demands by 52.6% and 69.3%, respectively, compared to the YOLOv7 model, while yielding a modest increase of 0.9% in mean average precision (mAP) and a 0.5% enhancement specifically in vehicle target detection accuracy.

2.4. Interception Guidance

The detection and tracking algorithm is complimented by the interception guidance methods where the tracked target drone is intercepted in mid-air. There are several methods to accomplish this feat. Several of the promising methods are detailed in this section.

2.4.1. Proportional Navigation

Proportional Navigation (PN) has been a dominant guidance law in missile interception for decades, valued for its simplicity, practicality, and robustness across a wide range of applications, including surface-to-air, air-to-air, and air-to-surface missiles, as well as space rendezvous missions [14]. PN's fundamental appeal lies in its reliance on the rate of change of the line of sight (LOS) between the pursuer and the target, allowing the interceptor to apply a control force proportional to this rate, with the goal of minimizing the terminal miss distance. Its relatively low demand for complex sensor inputs and the ease of implementation make PN an ideal choice for many tactical systems, especially when rapid and accurate interception is required.

The concept of PN encompasses two primary variants: True Proportional Navigation (TPN) and Pure Proportional Navigation (PPN). These differ primarily in how the navigation constant is defined and referenced. In TPN, the interceptor's guidance is referenced directly to the LOS vector, meaning that the interceptor's maneuver is based on the geometry of the target's motion relative to the interceptor. In contrast, PPN uses the interceptor's velocity vector as the reference frame, leading to a guidance law that is based on the velocity of the interceptor and the rate of change of the LOS. While TPN tends to be more analytically tractable—especially for non-maneuvering targets, PPN is often seen as more “natural” in practical applications, especially when the target is actively maneuvering or when minimal information about the target's motion is available [17].

The application of PN, and specifically PPN, in the interception of a quadcopter presents a unique set of challenges and opportunities. Quadcopter targets are typically small, highly agile, and capable of performing rapid, unpredictable maneuvers, which complicates interception dynamics. Despite these challenges, the inherent simplicity and adaptability of PN, especially PPN, make it an attractive choice for the guidance of interceptors designed to neutralize such targets [6]. By applying the principles of PN, the interceptor can continuously adjust its trajectory based on the observed relative motion of the target, ensuring that it remains on an optimal intercept course despite the target's evasive actions.

In this context, both TPN and PPN have their respective advantages and limitations. True Proportional Navigation offers a more direct and mathematically elegant approach, particularly when the target's motion

can be considered in a more predictable manner. However, TPN's reliance on LOS-based acceleration and deceleration introduces practical implementation challenges, particularly for real-time control systems and for targets that exhibit rapid, erratic motion [14]. On the other hand, Pure Proportional Navigation is often more robust and easier to implement in practical systems, as it requires fewer sensor inputs and tends to be less sensitive to variations in the engagement geometry. It can also be more forgiving in terms of initial conditions, which is critical in real-world missile intercepts where uncertainty in target position, speed, and trajectory is common [17].

This section will explore the application of PN, specifically, PPN in the interception of a quadcopter. The section discusses the adaptation of this guidance law to the unique dynamics of quadcopters, how both TPN and PPN qualitatively compare with each other. By delving into the practical considerations of implementing PN for this novel type of target, it is aimed to highlight the strengths and limitations of each variant in achieving successful interception. Through this analysis, it is demonstrated how, despite the complex nature of intercepting agile aerial targets like quadcopters, Proportional Navigation remains a highly viable and effective guidance law.

Derivation of Proportional Navigation Law

The derivation of the PN guidance law is essential to understand how the method accomplishes the necessary interceptor maneuvers for air-to-air interception. It also sheds light into how the interception conditions and assumptions are put together as well as which scenarios are fitting for an interception using the PN laws. The derivation is based on the papers published by Palumbo et al. [14] and Shukla et al. [17]. Consider an interception geometry as seen in Figure 2.2, where $\mathbf{r}_{i,w}$ and $\mathbf{r}_{t,w}$ are the position vectors

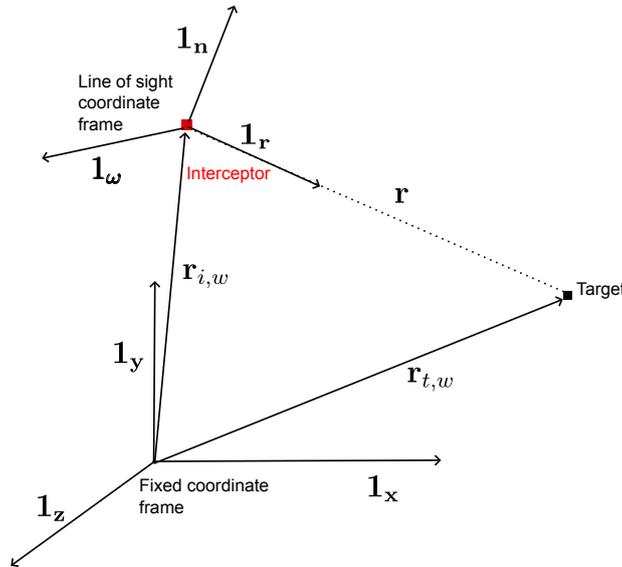


Figure 2.2: An arbitrary interception geometry and the corresponding coordinate frame definitions.

of the interceptor and target in the world coordinate frame respectively. The relative position vector \mathbf{r}_r is defined as shown in Equation 2.1:

$$\mathbf{r}_r = \mathbf{r}_{t,w} - \mathbf{r}_{i,w}. \quad (2.1)$$

In Figure 2.2, the line of sight (LOS) coordinate system originating from the interceptor is defined where the $\mathbf{1}_r$ is the unit vector parallel to the relative position vector \mathbf{r}_r , $\mathbf{1}_n$ is the unit vector aligned with direction of the change in line of sight and $\mathbf{1}_w$ is defined as $\mathbf{1}_w = \mathbf{1}_r \times \mathbf{1}_n$. Subsequently the relative position vector can also be calculated as $\mathbf{r}_r = R\mathbf{1}_r$. R is defined as the Euclidean distance of the line of sight, $\|\mathbf{r}_r\|$. To calculate the relative velocity, \mathbf{v}_r , the relative position is differentiated with respect to the fixed coordinate frame to obtain the following relative velocity expression in Equation 2.2:

$$\mathbf{v}_r = \dot{R}\mathbf{1}_r + R\frac{\delta}{\delta t}\mathbf{1}_r. \quad (2.2)$$

In Equation 2.2 the relative velocity is seen to have two parts: (1) a change in the line of sight line due to a change in distance and (2) a change in direction due to change in the rate of line of sight unit vector

($\mathbf{1}_r$). As mentioned earlier, the vector \mathbf{n} is defined along the change in the rate of line of sight line with the following expression in Equation 2.3:

$$\mathbf{n} = \frac{\delta}{\delta t} \mathbf{1}_r. \quad (2.3)$$

The unit vector along the vector n is defined as $\mathbf{1}_n = \frac{\mathbf{n}}{\|\mathbf{n}\|}$. The third and final unit vector to complete the right handed line of sight coordinate frame the $\mathbf{1}_\omega$ is defined, as mentioned earlier, as the cross product between the former two unit vectors, perpendicular to both.

A significant variable in the derivation is the angular velocity of the line of sight coordinate system with respect to the inertial reference frame φ and is given by the following expression in Equation 2.4:

$$\dot{\varphi} = \dot{\phi}_r \mathbf{1}_r + \dot{\phi}_n \mathbf{1}_n + \dot{\phi}_\omega \mathbf{1}_\omega \quad (2.4)$$

where the components of the rotation of the line of sight coordinate frame are given by Equation 2.5:

$$\begin{aligned} \dot{\phi}_r &= \dot{\varphi} \mathbf{1}_r, \\ \dot{\phi}_n &= \dot{\varphi} \mathbf{1}_n, \\ \dot{\phi}_\omega &= \dot{\varphi} \mathbf{1}_\omega. \end{aligned} \quad (2.5)$$

Following the angular velocity of the line of sight coordinate frame, the n vector can be calculated as Equation 2.6:

$$\mathbf{n} = \frac{d}{dt} \mathbf{1}_r + \dot{\varphi} \times \mathbf{1}_r. \quad (2.6)$$

. In this equation, the $\frac{d}{dt}$ represents the time derivative with respect to a rotating coordinate frame. It is known that the line of sight unit vector is constant and hence, the vector can be calculated as $\mathbf{n} = \dot{\varphi} \times \mathbf{1}_r$. Consequently the unit vector has the definition: $\mathbf{1}_r = \frac{\dot{\varphi} \times \mathbf{1}_r}{\|\dot{\varphi} \times \mathbf{1}_r\|}$. On top of this, the relative velocity formula is derived as the following expression in Equation 2.7:

$$\mathbf{v}_r = \dot{R} \mathbf{1}_r + R(\dot{\varphi} \times \mathbf{1}_r). \quad (2.7)$$

In order to obtain the relative acceleration expression, the relative velocity is differentiated as shown:

$$\frac{\delta}{\delta t} \mathbf{v}_r = \ddot{R} \mathbf{1}_r + 2\dot{R}(\dot{\varphi} \times \mathbf{1}_r) + R(\ddot{\varphi} \times \mathbf{1}_r) + R[\dot{\varphi} \times (\dot{\varphi} \times \mathbf{1}_r)]. \quad (2.8)$$

Looking at the individual components in Equation 2.8, the cross product terms are expanded upon to achieve the desired relative acceleration in terms of the angular velocity components of the line of sight coordinate frame with respect to the inertial frame. Using Equation 2.4 and $\mathbf{1}_r = [100]^T$, the term $\dot{\varphi} \times \mathbf{1}_r$ is developed resulting in the following equation:

$$\dot{\varphi} \times \mathbf{1}_r = \det \begin{vmatrix} \mathbf{1}_r & \mathbf{1}_n & \mathbf{1}_\omega \\ \dot{\phi}_r & \dot{\phi}_n & \dot{\phi}_\omega \\ 1 & 0 & 0 \end{vmatrix} = \dot{\phi}_\omega \mathbf{1}_n - \dot{\phi}_n \mathbf{1}_\omega \quad (2.9)$$

It is known from Equation 2.3 that the direction of n can't have a component along $\mathbf{1}_\omega$ and hence, the component $\dot{\phi}_n \mathbf{1}_\omega$ is zero. Following a similar derivation the other cross product components in Equation 2.8 are expanded upon and result in the following expressions:

$$\begin{aligned} \dot{\varphi} \times \mathbf{1}_r &= \det \begin{vmatrix} \mathbf{1}_r & \mathbf{1}_n & \mathbf{1}_\omega \\ \dot{\phi}_r & \dot{\phi}_n & \dot{\phi}_\omega \\ 1 & 0 & 0 \end{vmatrix} \\ &= \dot{\phi}_\omega \mathbf{1}_n - \dot{\phi}_n \mathbf{1}_\omega, \\ \ddot{\varphi} \times \mathbf{1}_r &= \ddot{\phi}_\omega \mathbf{1}_n. \end{aligned} \quad (2.10)$$

Putting in Equations 2.9 and 2.10 in Equation 2.8, the relative acceleration in terms of the components of the angular velocity of the line of sight coordinate system is derived as follows:

$$\frac{\delta}{\delta t} \mathbf{v}_r = \mathbf{a}_{t,w} - \mathbf{a}_{i,w} = (\ddot{R} - R\dot{\phi}_\omega^2) \mathbf{1}_r + (2\dot{R}\dot{\phi}_\omega + R\ddot{\phi}_\omega) \mathbf{1}_n + (R\dot{\phi}_\omega\dot{\phi}_r) \mathbf{1}_\omega. \quad (2.11)$$

In Equation 2.11 the individual components in the line of sight coordinate frame which influence the desired relative acceleration can be observed. Looking at the first component along the line of sight line ($\mathbf{1}_r$) several conditions for interception can be devised such as:

- The line of sight rate $\dot{\phi}_\omega$ should go to zero.
- The interceptor should be able to achieve a higher acceleration than the target along the line of sight line ($\mathbf{a}_{i,w} \mathbf{1}_r \geq \mathbf{a}_{t,w} \mathbf{1}_r$).
- The initial rate of change in the range should be negative ($\dot{R}(0) < 0$). This way the range R will decrease linearly ($(\mathbf{a}_{t,w} - \mathbf{a}_{i,w}) \mathbf{1}_r = 0$).

In order for the interceptor to intercept the target it should accelerate such that the line of sight rate goes to zero. Observing the second component of Equation 2.11, first the closing in velocity is defined as $v_c = -\dot{R}$. From the conditions it is known that the closing in velocity has to be positive ($v_c > 0$). Treating the closing in velocity and range as constant, the Laplace transform of the second component is taken as follows:

$$(\mathbf{a}_{t,w}(s) - \mathbf{a}_{i,w}(s)) \mathbf{1}_n = (sR - 2v_c) \dot{\phi}_\omega(s). \quad (2.12)$$

where s is the Laplace variable. Defining the interceptor acceleration to be perpendicular to the line of sight line as $\mathbf{a}_{i,w} \mathbf{1}_n(s) = \Lambda \dot{\phi}_\omega(s)$, the transfer function from the acceleration of the target to the line of sight rate is given as following:

$$\frac{\dot{\phi}_\omega(s)}{\mathbf{a}_{t,w} \mathbf{1}_n} = \frac{1}{sR - 2v_c + \Lambda}. \quad (2.13)$$

As seen in Equation 2.13, to guarantee a stable system $\Lambda > 2v_c$ condition should be satisfied. Hence the True Proportional Navigation law is derived as:

$$\mathbf{a}_{i,w_r} = Nv_c \dot{\phi}_\omega, N > 2. \quad (2.14)$$

Similarly the Pure Proportional Navigation law is given in the following expression satisfying the aforementioned conditions:

$$\mathbf{a}_{i,w_r} = N\dot{\phi} \times \mathbf{v}_{i,w}. \quad (2.15)$$

2.4.2. Comparison TPN vs PPN

The qualitative analysis of the TPN vs PPN is given by Shukla et al. [17]. The main takeaways can be followed below.

Proportional Navigation (PN) is a widely employed guidance law in missile defense systems, precision intercept technologies, and space applications due to its simplicity, effectiveness, and robustness in maneuvering targets. The two principal variants of PN—PPN and TPN—differ in their mathematical formulation, practical implementation, and overall effectiveness in a variety of scenarios. While both methods aim to minimize the miss distance between the pursuer and the target, they each exhibit distinct advantages and limitations. This section explores and contrasts the two guidance laws, comparing them on the basis of their theoretical foundation, implementation challenges, trajectory behavior, control efficiency, and overall robustness in real-world applications. It is important to understand the differences between the two methods and how they perform in certain situations. For this purpose Shukla et al. has compared the two methods and the findings and main takeaways are detailed below.

Theoretical Foundations and Mathematical Formulation

The key distinction between PPN and TPN lies in how the guidance commands are applied relative to the pursuer's velocity vector and the line of sight (LOS) between the pursuer and the target.

PPN: In Pure Proportional Navigation, the interceptor's acceleration is applied perpendicular to its velocity vector. This means that the interceptor's guidance law is based on the angular rate of the target relative to the interceptor, without introducing any longitudinal (forward) acceleration or deceleration components. As a result, the interceptor's forward velocity remains constant throughout the engagement. The guiding principle of PPN is relatively straightforward: it keeps the pursuer's trajectory oriented towards the target by adjusting its lateral acceleration in proportion to the rate of change of the LOS angle.

TPN: In contrast, True Proportional Navigation defines the pursuer's acceleration in terms of the LOS angle between the pursuer and the target. This results in a more complex maneuvering requirement, as the acceleration has components both along and perpendicular to the pursuer's velocity vector. The forward acceleration component in TPN, which is proportional to the sine of the angle between the pursuer's velocity and the LOS, leads to a change in the pursuer's speed throughout the interception. This forward velocity variation requires additional considerations, particularly in practical applications where such acceleration might not be feasible using control surfaces and/or small motors.

Implementation Challenges and Control Effort

One of the most critical factors in comparing PPN and TPN is how the guidance laws are implemented and the control effort required to execute them.

PPN Implementation: PPN is generally considered easier to implement, especially in interceptors relying on control surfaces and/or motors. Since the acceleration is purely lateral to the pursuer's velocity vector, the system requires less complexity in terms of control surface deflections or thruster management. Additionally, PPN does not require any longitudinal acceleration or deceleration, which makes it more practical for real-time control in dynamic environments. The absence of forward acceleration simplifies the system design and improves robustness by reducing the number of required actuators and systems in the missile.

TPN Implementation: In True Proportional Navigation, however, the need to apply acceleration both along and perpendicular to the velocity vector creates significant implementation challenges, particularly for systems relying on control surfaces. The forward velocity component that must be controlled can be problematic for interceptors using conventional aerodynamic surfaces, as controlling both the direction and speed of the interceptor simultaneously is difficult and often inefficient. TPN may require more sophisticated systems, such as reaction thrusters, which are commonly used in space applications but are cumbersome and less efficient in atmospheric engagements. The additional forward velocity variations introduce complexity in autopilot design, and the control surfaces must manage more variables, making the system more prone to errors and inefficiencies.

Forward Velocity and Acceleration

One of the most important differences between PPN and TPN is the way they handle the forward velocity of the pursuer.

PPN and Constant Forward Velocity: In PPN, since the commanded acceleration is applied purely in the lateral direction, the pursuer's forward velocity remains constant throughout the interception. This is an advantageous property for interceptors, as maintaining a constant speed minimizes the variability in the aerodynamic characteristics and simplifies the control system's task. The steady forward velocity also means that the interceptor's trajectory is easier to predict and adjust, leading to higher reliability and fewer control errors.

TPN and Forward Velocity Variations: In TPN, the commanded acceleration introduces both lateral and longitudinal components. The longitudinal acceleration causes the pursuer's forward velocity to fluctuate during the engagement, which can have detrimental effects. In particular, when the geometry of the engagement deviates from a direct tail-chase or collision course, TPN requires significant acceleration and deceleration to adjust the pursuer's velocity. These speed fluctuations can lead to problems in interceptor trajectory prediction, as they introduce non-linearities into the system and result in more complex, and potentially unstable, flight paths. Furthermore, the required acceleration/deceleration

increases dramatically as the engagement geometry moves further from a direct line-of-sight approach, making TPN particularly inefficient in these scenarios.

Control Efficiency and Practicality

PPN and Control Efficiency: From a control-efficiency standpoint, PPN consistently requires less control effort than TPN. Since the pursuer's speed remains constant and only lateral adjustments are required, the control effort is minimized. The absence of longitudinal acceleration/deceleration means that the control surfaces or thrusters only need to handle lateral acceleration, making the system more energy-efficient. Moreover, PPN has fewer restrictions on initial conditions, allowing for greater flexibility in the intercept scenario. For most engagement geometries, PPN results in a more efficient use of control resources and provides a higher level of robustness to variations in target behavior or environmental conditions.

TPN and Control Inefficiency: TPN, by contrast, is less efficient in its control effort due to the longitudinal velocity variations and the resulting need for acceleration/deceleration in both the lateral and forward directions. As demonstrated in numerical simulations in [17], the extra control effort required for TPN increases significantly for large departures from tail-chase or collision course conditions. In an air-to-air interception scenario, this inefficiency could manifest as wasted fuel or power, and could potentially lead to errors in trajectory corrections or missed intercepts. The necessity for sophisticated control mechanisms to manage both lateral and longitudinal components of acceleration reduces the overall efficiency of the TPN law, especially in atmospheric engagements.

Trajectory Behavior and Robustness

PPN and Trajectory Stability: The trajectory behavior of a missile under PPN is typically more stable and predictable. Since the forward velocity is constant, the missile follows a more straightforward path towards the target, with the primary challenge being to maintain a constant lateral acceleration based on the changing LOS angle. This simplicity in trajectory design makes PPN an inherently more robust choice for a wide range of engagement conditions, especially when dealing with targets that are highly maneuverable.

TPN and Unbounded Trajectory Variations: The trajectory behavior under TPN, however, can become erratic and unpredictable, particularly when the engagement geometry deviates from a near-collision course. The need to accelerate and decelerate along the forward direction introduces the possibility of unbounded acceleration in some cases, which can result in significant instability. This is especially problematic for targets that are highly maneuverable or when there is uncertainty in the initial engagement geometry. Additionally, TPN imposes strict limitations on the initial conditions, such as the requirement for the LOS rate to remain bounded and for the relative velocity between the pursuer and target to be within certain limits. These constraints make TPN less robust in real-world applications, where initial conditions may not always align perfectly with the assumptions of the guidance law.

Practical Considerations

While both PPN and TPN provide effective interception strategies, **PPN** is generally the superior choice for most practical applications. The main reasons for this are its simplicity, reduced control effort, and robustness in a variety of engagement scenarios. PPN's constant forward velocity and lateral-only acceleration allow for easier implementation in interceptors which have lower acceleration capabilities, resulting in more efficient use of resources and a more stable trajectory. TPN, while analytically attractive in certain cases, is less practical due to its complex trajectory behavior, requirement for forward velocity variation, and the additional control effort needed to manage both longitudinal and lateral components of acceleration.

From a practical standpoint, PPN is more adaptable, requiring fewer restrictions on initial conditions and providing a more efficient and robust solution in real-world missile interception systems. On the other hand, TPN may find limited application in scenarios where reaction thrusters are used, such as in space-based systems, but it is generally less suitable for conventional, aerodynamically controlled missiles and quadcopters which have small motors. Thus, for the majority of interceptor systems, PPN remains the more practical, efficient, and robust guidance law.

2.4.3. Modern Interception Guidance Methods

Interception guidance for air-to-air interception by quadcopters has gained significant attention in recent years. To achieve precise, rapid, and feasible interception maneuvers, various trajectory generation and control methods have been developed. This literature review explores three primary works—those of Markus Hehn et al. [18], [19] and Daniel Mellinger et al. [20] each offering unique contributions to trajectory generation and control strategies applicable to quadcopter interception maneuvers.

Feasible Trajectory Generation and Control

Markus Hehn et al. work on *Quadcopter Trajectory Generation and Control* [18] focuses on generating feasible trajectories that respect both dynamic and input constraints of a quadcopter. The main goal is to create a trajectory generation algorithm that can plan paths from any initial state to a target position, allowing for online updates at a frequency of approximately 50 Hz. Key points of the approach include:

- Trajectories are designed to bring the quadcopter to a target as quickly as possible while ensuring dynamic feasibility.
- An implicit feedback control law is integrated by replanning the trajectory at each controller update.
- The method considers the jerk of each translational degree of freedom (DoF) as planning inputs, limiting both jerk and acceleration to avoid control saturation.

In terms of dynamic modeling, direct control over the quadcopter's rotational rates $(\omega_x, \omega_y, \omega_z)$ and mass-normalized thrust a is assumed. While rotational dynamics are largely ignored, the model limits control inputs based on feasibility conditions. The trajectory planning itself focuses on decoupled jerk trajectories, which are time-optimal for each DoF, subject to feasibility checks.

Real-Time Trajectory Generation for Interception Maneuvers

In a complementary work, Hehn et al. explores real-time trajectory generation in the context of interception maneuvers [19]. This work emphasizes real-time, lightweight trajectory computation that allows the quadcopter to reach a specified interception point (x, y, z) at a designated time t . Highlights of this approach include:

- The trajectory generation algorithm plans from arbitrary initial states and guarantees feasibility with respect to defined constraints.
- By using Pontryagin's minimum principle, necessary conditions for optimal input trajectories are derived, utilizing a bang-bang control approach for time-optimal solutions.
- The planned trajectory emphasizes reaching rest after interception to avoid overshoot and excessive deceleration.

Hehn et al. addresses state constraints through a direct adjoining approach, where the Hamiltonian function is augmented by state constraints with a focus on minimizing final time. The trajectory generation approach is adaptable, allowing the quadcopter to cross a specified position at a precise time while respecting dynamic limitations.

Precise Aggressive Maneuvers

Daniel Mellinger et al. work on *Trajectory Generation and Control for Aggressive Maneuvers with Quadrotors* [20] takes a different approach, focusing on designing dynamically feasible trajectories for aggressive maneuvers. Mellinger et al. introduces a family of trajectory controllers that achieve precise positioning in a desired state within state space. The main aspects of the approach include:

- Development of a sequence of trajectory segments, each associated with a controller defined by a goal state in the quadrotor's state space.
- The design of three distinct controllers—attitude control, hover control, and 3D path following—which are sequentially applied to achieve complex maneuvers.
- Real-time refinement of controllers through iterative trials, adjusting feedforward parameters to compensate for dynamic model errors and actuator limitations.

In trajectory generation, Mellinger et al. [20] emphasizes mode switching between controllers, which can be triggered by either time or events. This flexible controller sequencing allows the quadrotor to perform complex interception maneuvers effectively by transitioning through various control modes as needed.

The trajectory generation and control methods explored in these works provide a foundational understanding of air-to-air interception maneuvers for quadcopters. Hehn et al. [18] contributions offer a structured approach for real-time, feasible trajectory planning, with attention to dynamic constraints and optimality. Mellinger et al. [20] focus on aggressive maneuvers and mode switching contributes to effective interception strategies that can adapt to dynamic environments. Together, these approaches serve as a basis for advanced interception guidance methods, forming a robust foundation for future developments in this field.

2.5. Control

The control methodology used to follow the references created by the guidance law is also significant for the interception. Two promising method for control are discussed in this section.

2.5.1. Thrust Vectoring Control (TVC)

The Thrust Vectoring Control (TVC) is demonstrated by Marina et al. [6] is capable of following PN guidance law for successful interception. This section is based on the derivation and results obtained by Marina et al. [6]. TVC is a control strategy in which the required thrust vector is initially determined, and subsequently, control actions are defined based on this vector. In TVC, external forces and disturbances, such as drag, gravity, and wind, can be accounted for directly. This integration allows for faster disturbance rejection, as these forces are introduced through feedforward control. Consequently, the performance of the TVC is highly dependent on the accuracy of the estimates for these external forces and disturbances.

For a UAV, which possesses four controllable degrees of freedom (attitude angles yaw, pitch, roll, and thrust), only three are necessary to define the thrust vector: roll, pitch, and thrust magnitude. The yaw angle can be controlled independently, which can enhance the UAV's capability to track a moving target.

The commanded thrust vector, as defined in TVC, can be formulated as [6]:

$$T_{\text{cmd}} = k_p \circ e_p + k_{pi} \circ \int_0^{\Delta t} e_p dt + k_v \circ e_v + k_a \circ e_a + mg + ma_{\text{cmd}} + F_D, \quad (2.16)$$

where k_p , k_{pi} , k_v , and k_a are vector gains associated with position, velocity, and acceleration errors, given by e_p , e_v , and e_a . The element-wise multiplication \circ allows each axis to have different gain values due to the varying dynamics along the z -axis compared to the x and y axes. The error terms are defined as

$$e_p = p_{\text{cmd}} - p, \quad e_v = v_{\text{cmd}} - v, \quad e_a = a_{\text{cmd}} - a, \quad (2.17)$$

where p_{cmd} , v_{cmd} , and a_{cmd} denote the commanded position, velocity, and acceleration, typically derived from the guidance system. Equation (3.27) includes both feedback-related terms (the first four) and feedforward terms, all expressed in the inertial frame.

When wind velocity is ignored, the drag force F_D is described by

$$F_D = TR A_{\text{bla}} R^T v + k_{\text{par}} \circ k_v \|v\|, \quad (2.18)$$

where A_{bla} represents the Blade Flapping Coefficients matrix, and k_{par} is the Parasitic Drag Coefficients vector. The first term dominates at low speeds, while the second term becomes significant for speeds exceeding 10 m/s, which is relevant for this mission.

To convert the commanded thrust vector into roll, pitch, and thrust magnitude, the UAV's desired orientation must be determined.

This requires taking into account the independent commanded yaw angle ψ_{cmd} . The intermediate orientation vector along the x axis, denoted as x_{cmd}^ψ , can be expressed as:

$$x_{\text{cmd}}^\psi = \begin{bmatrix} \cos(\psi_{\text{cmd}}) \\ \sin(\psi_{\text{cmd}}) \\ 0 \end{bmatrix}. \quad (2.19)$$

The desired z -axis orientation in the body-fixed frame, z_{cmd}^B , is then defined by normalizing the commanded thrust vector:

$$z_{\text{cmd}}^B = \frac{T_{\text{cmd}}}{\|T_{\text{cmd}}\|}. \quad (2.20)$$

Using the orthogonality property of the reference frames, the y axis of the desired orientation, y_{cmd}^B , can be obtained as

$$y_{\text{cmd}}^B = \frac{z_{\text{cmd}}^B \times x_{\text{cmd}}^\psi}{\|z_{\text{cmd}}^B \times x_{\text{cmd}}^\psi\|}. \quad (2.21)$$

The x_{cmd}^B vector can then be calculated as

$$x_{\text{cmd}}^B = y_{\text{cmd}}^B \times z_{\text{cmd}}^B. \quad (2.22)$$

Using these three orthogonal vectors, the commanded rotation matrix R_{cmd} is given by

$$R_{\text{cmd}} = \begin{bmatrix} x_{\text{cmd}}^B & y_{\text{cmd}}^B & z_{\text{cmd}}^B \end{bmatrix}. \quad (2.23)$$

Based on Equation (3.34) and by equating terms in the Rotation Matrix defined in Section 3.6, the required roll and pitch angles can be extracted as:

$$\begin{bmatrix} \phi_{\text{cmd}} \\ \theta_{\text{cmd}} \end{bmatrix} = \begin{bmatrix} \tan^{-1} \left(\frac{R_{32}}{R_{33}} \right) \\ \tan^{-1} \left(\frac{-R_{31}}{\sqrt{R_{32}^2 + R_{33}^2}} \right) \end{bmatrix}. \quad (2.24)$$

The thrust magnitude T_{cmd} can be computed by projecting the commanded thrust vector onto the body-fixed z axis:

$$T_{\text{cmd}} = T_{\text{cmd}} \cdot z_{\text{cmd}}^B. \quad (2.25)$$

These resulting values, ϕ_{cmd} , θ_{cmd} , and T_{cmd} , are then sent to the Flight Controller, enabling the UAV to follow the specified trajectory.

2.5.2. Nonlinear Dynamic Inversion (NDI)

Nonlinear Dynamic Inversion (NDI), first applied to aircraft control by Snell et al. [39] in 1992, was introduced to manage the complex dynamics of supermaneuverable fighter aircraft, which experience significant nonlinear behavior, especially at high angles of attack. NDI derives a nonlinear transformation from the system model, which relates a set of pseudo-control inputs, denoted by ν (typically, desired time derivatives of output vectors), directly to actuator commands. By assuming actuator dynamics are negligible, this transformation allows the system to be treated as a linearized, decoupled system where ν links directly to output derivatives. Consequently, proportional-integral (PI) or proportional-integral-derivative (PID) controllers can generate ν with a single gain set to achieve consistent linear response irrespective of the system state [40].

The general form of a control-affine nonlinear system is represented by:

$$\dot{x} = f(x) + G(x)u, \quad y = h(x) \quad (2.26)$$

where $x \in \mathbb{R}^m$ denotes the system state, $y \in \mathbb{R}^p$ the outputs to be controlled, $f(x)$ the system's natural evolution, $h(x)$ the output function, $G(x)$ the input matrix dependent on state, and $u \in \mathbb{R}^n$ the control inputs.

In cases where the system has no internal dynamics, the inputs relate to output derivatives by differentiating y with respect to time. For some elements, u may not appear after a single differentiation, necessitating further derivation until an algebraic relationship between u and the time derivative of each output y_i is achieved. Assuming this relationship is found after one differentiation for all outputs, we define the Lie derivatives $L_f h(x) = \nabla h(x)f(x)$ and $L_G h(x) = \nabla h(x)G(x)$, yielding:

$$\dot{y} = L_f h(x) + L_G h(x)u \quad (2.27)$$

With a non-zero, square, and invertible $L_G h(x)$ matrix, the system dynamics can be decoupled and inverted using the control input u as follows:

$$u = [L_G h(x)]^{-1}(\nu - L_f h(x)) \quad (2.28)$$

where $\nu = \dot{y}$ represents the pseudo control input. This method, however, relies heavily on accurate models of $L_f h(x)$ and $L_G h(x)$; inaccuracies can reduce control effectiveness [39].

2.5.3. Incremental Nonlinear Dynamic Inversion (INDI)

An incremental variant of NDI (INDI) addresses some NDI limitations by focusing on control input increments, Δu , instead of the absolute control input u [41] [40]. INDI was initially motivated by its ability to manage non-affine nonlinear actuator characteristics, but Sieberling et al. [40] demonstrated that INDI could operate effectively by neglecting the aerodynamic model of the platform and instead measuring the current achieved pseudo-control input, $\dot{y} = \dot{\omega}$ (for rotational accelerations in inner-loop control).

The INDI control law for rotational rates ω and accelerations $\dot{\omega}$ takes the form:

$$\dot{\omega} = \dot{\omega}_0 + \left. \frac{\partial(JM_c)}{\partial u} \right|_{\omega_0, u_0} \Delta u \quad (2.29)$$

where J represents the inertia matrix, and M_c is the control effectiveness model. This approach minimizes the need for a full aerodynamic model, but does require accurate current values of ω_0 and $\dot{\omega}_0$ [40].

To address challenges in estimating rotational accelerations, methods such as optimal filtering [40], sensor processing with low-pass filters [42] have been applied. Adaptive implementations [42] further reduce reliance on predetermined actuator effectiveness, which can vary under different flight conditions.

2.6. Revised Research Questions

The research objective and questions were initially determined. Nevertheless as the thesis progressed, the main question and objective remained the same while sub-questions were subject to adjustments. To reach the research objective several steps need to be taken. A significant step is to identify the research gaps. While doing so, it is known that new and unexplored areas of interest may arise.

The detection and tracking of small drones from a camera have become a prominent area for research. A lot of effort has been put on detecting drones from an image in different lighting and background conditions. Nevertheless, to the authors knowledge the radar and the camera has not been combined to detect and track a small UAV. On top of this air-to-air interception of non-cooperative UAVs have been studied by the literature extensively and has shown promising results. The research gap this thesis tries to address is combining both approaches in a simple manner for utilizing a quadcopter for air-to-air interception of a non-cooperative UAV. With the research the theoretical methods are simplified and implemented to achieve a robust framework and generalized for common application in real-life conditions. Although it is known that private companies such as Auterion has a complete software solution that can intercept a loitering munition from a camera, an open source solution has not been presented.

On top of this, even though the infrared camera was determined to be the first choice for the visual feedback method, the devised setup was not sufficient for a detection. This was because the setup consisted of a Raspberry Pi NoIR camera with a $940nm$ infrared bandpass filter above it provided too short of a wavelength for using the benefits of the infrared spectrum such as the high contrast between the drone and background. The longer wavelength infrared cameras were deemed too expensive for this research and thus, the visual spectrum cameras replaced the infrared cameras.

Considering these the revised questions are as follows:

- How to design a real time algorithm to detect, track and intercept loitering munition onboard a quadcopter using an RGB camera and target distance information?
- How to implement the most efficient computer vision algorithm in terms of performance and speed for air to air detection of small drones with an RGB camera and a radar?
- How to extract position, velocity and acceleration information of a target drone using RGB camera and assuming a known distance to target drone?
- What is an optimal interception guidance law for intercepting UAS targets with the use of another UAS?

Part II

Scientific Article

Utilizing a quadcopter for air-to-air interception of a loitering munition

Yamac Birol* and Ewoud J.J. Smeur†

*Aerospace Engineering, Technical University of Delft, Delft, The Netherlands

†Aerospace Engineering, Technical University of Delft, Delft, The Netherlands

Abstract—The increasing use of drones for both civilian and military applications has raised substantial safety and security concerns, including risks of misuse in restricted airspaces and on the battlefield. Loitering kamikaze drones, in particular, represent a significant threat, necessitating effective countermeasures. This paper proposes an onboard interception guidance algorithm designed to detect, track, and intercept such drones, relying on data from a camera and simulated radar input. The system is initiated using the target’s initial location provided by ground-based sensors, while the onboard system processes camera data and simulated radar-derived distance values. The key challenge is ensuring successful interception within the limited operational range of these sensors. Several detection algorithms are implemented and compared, and the chosen detection method is integrated with the OpenCV CSRT tracker. A linear Kalman filter is employed to estimate the target’s position and velocity, effectively handling intermittent missed detections. The proposed solution is evaluated by intercepting a simulated target UAV with a real interceptor drone in TU Delft Cyberzoo, feeding targets position and velocity to the interceptor, alongside validating the detection and tracking algorithms.

I. INTRODUCTION

Drones have collected significant attention from both academia and industry due to their efficiency in visual tasks such as surveillance, inspection, obstacle avoidance, and the delivery of goods or medical supplies. However, this increased usage raises concerns, including the misuse of commercial drones in restricted airspaces, exemplified by the incident around Gatwick Airport in London, which caused severe delays [29], as well as their use as kamikaze drones in conflict zones. Notable examples include the use of drones to deliver bombs by ISIS [17], the attempted attack on the Venezuelan president using two explosive-carrying drones [2], and the ongoing deployment of Iranian Shahed kamikaze drones by Russia in the Russian-Ukrainian war [3], along with their use by the Houthis [9].

To counter these threats, several methods have emerged, including jamming systems, nets launched from the ground [19], GPS spoofing [25], and less expensive UAS intercepting more costly drones [3]. Nevertheless the solutions are not fully autonomous and rely on a human operator. Effective detection is crucial for these countermeasures and can be achieved through various means, including analyzing radio frequency communications, acoustic sensors that capture the

inherent sound of drones, optical sensors like cameras, and radar detection [27].

This research aims to develop an onboard interception guidance algorithm for detecting, tracking, and intercepting loitering kamikaze drones. The algorithm will utilize onboard camera data and simulated radar inputs, with ground-based sensors providing initial location information for the target drone. One of the primary goals of this work is to enhance the drone’s capabilities towards achieving full autonomy, reducing reliance on human operators.

II. RELATED WORK

The detection and tracking of small unmanned aerial systems (UAS) using monocular RGB cameras has become a significant research topic. This includes detection scenarios ranging from ground-to-air, air-to-ground, and air-to-air [7, 36]. Similarly, infrared cameras have been employed for small target detection in both air-to-ground and ground-to-air applications [18, 40, 6]. Detection methods can broadly be categorized into two types: conventional and deep learning-based approaches.

Conventional methods typically rely on prior knowledge of the object of interest. These methods often focus on background subtraction to isolate small targets. Examples include the top-hat algorithm [38], the max-mean algorithm [32], and more sophisticated background suppression techniques such as the average gray difference algorithm [1] and generalized structure tensor-based methods [10]. Other methods include noise estimation, where the target is treated as noise and detected using specialized noise models [11]. Local contrast methods, which search local regions using sliding windows, are also prevalent; these methods include pixel-contrast approaches and adaptive thresholding techniques [13]. However, conventional methods tend to underperform in complex backgrounds or varying lighting conditions due to their reliance on prior knowledge and are often only applicable in specific scenarios [6].

Deep learning approaches, particularly those based on convolutional neural networks (CNNs), have achieved state-of-the-art performance in UAV detection. Algorithms such as Faster R-CNN and SSD (Single Shot Multibox Detector) have produced strong detection results, balancing speed and accuracy [39, 37, 41]. Among these, Grid R-CNN and Cascaded

R-CNN offer superior accuracy in terms of Average Precision (AP), but at the cost of higher computational demand [41]. The SSD algorithm, by contrast, offers good computational efficiency and strong performance across different target scales, making it suitable for small object detection [41]. YOLO-based detectors are particularly notable for their high speed; YOLOv3 is recognized for being the fastest, while YOLOv5 achieves a balance between speed and accuracy, and YOLOX provides better performance compared to YOLOv3 with faster processing than YOLOv5 [41, 37, 40].

An important trade-off in detection is between appearance-based, motion-based, and hybrid detectors. Appearance-based detectors, which may use global or local methods, can be computationally intensive when using global approaches, while local methods may be inefficient without prior knowledge of the target's location [6]. Hybrid methods combine global and local features, either by training CNNs to utilize both or by dynamically switching between global and local detectors in real time [14]. A prominent example is the local patch network combined with global attention, which includes a supervised attention module and an LPNet for extracting local features [6]. Motion-based detectors, which focus on detecting moving objects by analyzing motion features, also play a significant role. Methods range from background subtraction techniques [20] to spatio-temporal approaches [24]. Hybrid approaches, which incorporate both appearance and motion features, are employed to optimize detection performance and speed for real-time applications [14].

Tracking of small UAS, another key area of research, has been approached using both deep learning and optical flow techniques. Deep learning-based methods include convolutional Siamese networks, Transformer-based methods like TransT, and advanced trackers like LTMU, which employs a meta-updater for improved performance [39]. Optical flow approaches, both dense and sparse, track features across frames. Sparse optical flow methods such as Kalman trackers have been particularly effective in estimating the relative velocity and acceleration of a target using a "constant velocity" model [39], [20].

Finally, the interception guidance can be achieved by a simple guidance law as well as more complex trajectory generation algorithms. An interception guidance law could be as simple as trying to keep the detected and tracked object in the center of the field of view and approach it with constant divergence. Trajectory generation algorithms in the context of homing missile guidance as well as UAS guidance have been studied extensively in the literature. In general homing missiles which intercept manoeuvring targets use Proportional Navigation (PN) and its derivatives Pure Proportional Navigation (PPN), True Proportional Navigation (TPN), Generalised True Proportional Navigation (GTPN), Ideal Proportional Navigation (IPN), Augmented Proportional Navigation (APN) [35]. PN in general is accounted as an optimal guidance strategy which minimizes the terminal miss distance [26]. The PN approaches try to create a guidance law such that the lateral acceleration of the interceptor is

proportional to the line of sight rate and perpendicular to the interceptor velocity direction (PPN, IPN) or to the line of sight direction (TPN, GTPN), GTPN has a fixed angle to the normal direction of the line of sight [26], [30], [8]. PPN seem to show good interception behavior for simple maneuvers such as circling or an abrupt change of direction [22]. However, PN also has several assumptions such as the change in magnitude of the line of sight (closing in speed) should be negative, so the interceptor should be approaching towards the target and in further distances should be in a near collision course [26] [8]. Also the interceptor acceleration should be higher or equal than the target acceleration along the line of sight [26]. Proportional Navigation is also adapted to use in a UAS for intercepting maneuvering targets [22]. Also used is the Pontryagin's Minimum Principle to find an optimal trajectory which adheres to UAS constraints [15], [16]. Model predictive control is used with impact angle constraints for keeping the target drone at the camera FOV in [5] however, the included terminal velocity constraint limits interception to head-on or chasing geometries. In [23] trajectories are generated by the sequential composition of different controllers in a hybrid mode where changes to underlying parameters and set points in the controllers allow different trajectories, nevertheless requires a-priori knowledge of the interception location.

III. METHODS

A. Assumptions

The scope of the research necessitates several assumptions for the design. These are as follows:

- A ground based radar provides an initial position and speed estimate of the target drone. In a realistic scenario, the interceptor drone is activated when the loitering munition is initially detected. Hence as an initial condition, the first detected position and speed is fed to the interceptor.
- The interceptor drone is equipped with an onboard radar that is providing the distance to the target drone. Due to the scope of the research, integrating a radar is infeasible. Nevertheless, a radar is effective at determining the range to a target, but it is less capable of accurately indicating the target's precise direction relative to the radar. Thus, just the range value is provided to the interceptor drone by calculating and sending it from the ground control station.
- The target drone remains mostly non-maneuvering, with its acceleration kept to a minimum. Only minor variations in the flight path are introduced to prevent the position from being extrapolated based solely on the initial position and velocity.

B. UAV Detection and Position Tracking Algorithm

It has been demonstrated that a target drone is most discernible within the visible and near-infrared wavelengths [18]. This enhanced visibility is attributed to the significant contrast between the drone and the background in these spectral regions. Initially, the target drone is expected to occupy a small pixel area, approximately 10 to 20 pixels, making high contrast

imperative for the initial detection and tracking processes. The interception trajectory is designed to ensure that the interceptor approaches the target from a lower altitude, utilizing the sky as the background. This trajectory serves a dual purpose: it enhances the contrast between the target drone and the background while also simplifying the background complexity, thereby reducing the likelihood of occlusion during detection and tracking.

Given that the camera is not mounted on a gimbal and is fixed at a specific angle relative to the interceptor’s body reference system, these considerations are critical to maximizing the target drone’s visibility within the camera’s field of view (FOV) throughout the interception process. Following the implementation of the detection and tracking provisions, an appearance-based detection method is deemed adequate for interception purposes. Several methodologies have been investigated, including a YOLOv7-based detector, a Tiny YOLOv3-based detector, and a blob detector. These approaches were selected for their high accuracy and low computational demand. The detection and tracking algorithms are executed on a Raspberry Pi 4, which has limited computational capacity; thus, convolutional neural network (CNN)-based detectors were determined to be too computationally intensive for onboard processing. Nonetheless, some preliminary results from these detectors are presented in this study.

The CNN models (YOLOv7 and Tiny YOLOv3) are trained on datasets collected using the Raspberry Pi NoIR camera with IMX219 color calibration for RGB video recording on the Parrot Bebop2. The images feature the black Sonicmodell AR Wing Pro and Parrot Disco, used as target drones in the Unmanned Valley Valkenburg. The dataset consists of target drones in various orientations and attitudes, captured at distances ranging from 50 to 100 meters.

The parameters of the YOLO models during training affect the performance during inference. The YOLOv7 is trained using an input image of 640x480 pixels with 3 color channels. The rest of the essential training parameters for the YOLOv7 is given in table I.

The input to the Tiny YOLOv3 model consists of an image frame resized to 416x416 pixels, with 3 color channels representing the RGB spectrum. The original frame undergoes a downsampling process to fit the 416x416 pixel requirement, maintaining the necessary aspect ratio and spatial integrity to ensure accurate detection, nevertheless decreases the accuracy with respect to an image with 640x480 pixels. The remaining training parameters are detailed in Table II.

The blob detection method exploits the fact that the target drone is much darker than the background (sky) and hence employs adaptive blob detection to identify a dark drone against a lighter background. The approach begins by converting the video frame to grayscale, simplifying the data and reducing computational overhead. To account for variations in lighting, an adaptive thresholding technique is applied, with the threshold determined dynamically based on the mean pixel intensity of each frame. This adjustment enhances robustness in varying light conditions, critical for outdoor environments

Parameter	Value
Initial Learning Rate (lr0)	0.01
Final Learning Rate (lrf)	0.1
Momentum	0.937
Weight Decay	0.0005
Warmup Epochs	3.0
Warmup Momentum	0.8
Warmup Bias Learning Rate	0.1
Box Loss Gain (box)	0.05
Class Loss Gain (cls)	0.3
Object Loss Gain (obj)	0.7
IoU Training Threshold (iou_t)	0.20
Anchor Threshold (anchor_t)	4.0
HSV Hue Augmentation (hsv_h)	0.015
HSV Saturation Augmentation (hsv_s)	0.7
HSV Value Augmentation (hsv_v)	0.4
Image Translation (translate)	0.2
Image Scale (scale)	0.9
Image Flip Left-Right (fliplr)	0.5
Mosaic	1.0
Mixup	0.15
Loss OTA	1

TABLE I
YOLOV7 TRAINING PARAMETERS

Parameter	Value
Momentum	0.9
Weight Decay	0.0005
Angle	0
Saturation	1.5
Exposure	1.5
Hue	0.1
Learning Rate	0.001
Burn-in	1000
Max Batches	500200
Policy	Steps
Steps	400000, 450000
Scales	0.1, 0.1

TABLE II
TINY YOLOV3 TRAINING PARAMETERS

[28]. Following this, a Gaussian blur is employed to mitigate image noise, improving the precision of contour extraction, which is conducted using the OpenCV implementation of Suzuki-Abe algorithm[33]. The method computes the bounding box around the largest detected contour, capturing the object’s location. By dynamically adjusting to background exposure, the technique ensures efficient and accurate blob detection, particularly in high-contrast scenarios. This algorithm is computationally lightweight, making it highly suitable for real-time object detection on edge devices.

The detection algorithm results can be seen in Figure 1. The YOLOv7 and Tiny YOLOv3 detection capabilities are found to be similar in terms of accuracy where the detection is made within a confidence range of 60% – 97% and an average confidence of 75% with the Tiny YOLOv3 having half the inference time. YOLOv7 runs at 0.2 FPS while the Tiny YOLOv3 runs at 0.4 FPS on the Raspberry Pi 4B. The YOLO algorithms result a higher confidence when the target drone is viewed from below Figure 1d outlining the fixedwing shape while result in a lower confidence when target drone is viewed from the front or back seen as a line Figure 1c. The

blob detector has the worst accuracy out of the detectors but runs at 8 FPS on the Raspberry Pi 4B.



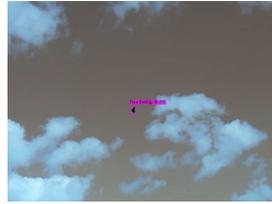
(a) Blob detector detection where the target drone is at 0° roll and pitch attitude



(b) Blob detector detection where the target drone is making a turn



(c) Tiny YOLOv3 detection where the target drone is at 0° roll and pitch attitude



(d) Tiny YOLOv3 detection where the target drone is making a turn

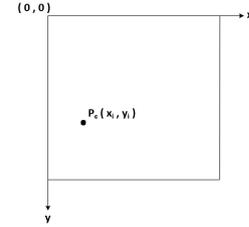
Fig. 1. Detection capabilities of different algorithms. The images captured onboard Parrot Bebop2 in flight and hovering. The target drone is the black Sonicmodell AR Wing Pro. (A) and (B) show blob detection, while (C) and (D) show Tiny YOLOv3 detection.

On top of the detection algorithm, the target drone is tracked in every consecutive frame for velocity estimate. This is achieved by a built in OpenCV tracker, The CSRT tracker, which is based on the Discriminative Correlation Filter with Channel and Spatial Reliability (CSR-DCF) algorithm, works by integrating both channel reliability and spatial reliability into the tracking process. The spatial reliability map adjusts the filter to focus on the most suitable regions of the object, which helps track non-rectangular objects more accurately and reduces drift during occlusion. The channel reliability scores are used to weigh filter responses, enhancing the tracking accuracy. This allows the CSRT tracker to handle complex situations like occlusions, illumination changes, and deformations [21]. The CSRT tracker runs at 10 FPS on the Raspberry Pi 4B.

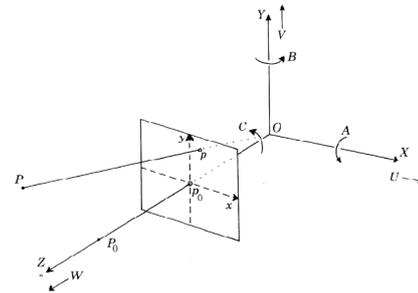
Due to computational constraints, the detection algorithm is executed every n frames, with n being experimentally determined for each algorithm. Between detections, interim tracking is performed using the OpenCV CSRT (Discriminative Correlation Filter with Channel and Spatial Reliability) tracker. More computationally intensive algorithms, such as the CNNs, require larger n values. Specifically, for the CNNs, $n = 50$ and $n = 100$ are tested, while for the lighter blob detector, smaller values of $n = 10$ and $n = 20$ are used. This design allows for object tracking between detection intervals, with

the detection algorithm supplying an updated bounding box around the target drone. The necessity for periodic re-detection stems from the dynamic size and appearance of the drone as the interceptor moves closer to the target, requiring frequent refreshment of the tracked object. The detection algorithm is executed on frames L_i where i is a multiple of n (i.e., $i = k \cdot n$ with $k \in \mathbf{Z}^+$). In each detection frame a bounding box is generated around the detected drone using either CNN based detector or the blob detector, and the pixels within this bounding box serve as input to the CSRT tracker. To enhance the tracker's accuracy, the video frame undergoes grayscale conversion, which simplifies the image by removing color information and reducing dimensionality, thereby focusing on intensity contrasts [12]. Subsequently, thresholding is applied to increase the contrast between the drone and its background, dynamically adjusting the threshold based on pixel intensity to handle varying illumination conditions [28]. A Gaussian blur is then applied to reduce noise and further enhance the drone's visibility by smoothing intensity variations in the background [4]. These preprocessing steps ensure that the tracker maintains robust performance, even in challenging dynamic visual conditions, allowing reliable target tracking throughout the visible frames.

C. Perspective Projection



(a) Pixel (image) coordinate frame



(b) Camera coordinate frame

Fig. 2. Coordinate frame definitions

After the target is detected in the L_i^{th} frame, the centroid of the bounding box is tracked as the position of the target. The detected or tracked centroid is in the image coordinates

as depicted in Figure 2a. The position of the target in world coordinate frame is required for interception. Equation 1 demonstrates the transformation from world frame to pixel frame. The position in the world coordinate frame is calculated using the extrinsic and intrinsic matrices of the camera. Matrix $[\mathbf{R}|\mathbf{t}]$ is the extrinsic matrix and matrix \mathbf{A} is the intrinsic matrix.

$$\mathbf{P}_p = \mathbf{A}[\mathbf{R}|\mathbf{t}]\mathbf{P}_w \quad (1)$$

The extrinsic matrix is used to transform coordinates from the world coordinate frame, which is an arbitrary right-handed Cartesian coordinate system, first into the body-fixed coordinate frame, and subsequently into the camera coordinate frame, illustrated in Figure 2b, where the Z-axis is oriented outward from the camera. The intrinsic matrix represent the transformation from the camera frame to pixel frame. For this the 3D point in the camera frame needs to be mapped on to the 2D pixel space. Perspective projection is used to determine the intrinsic matrix. Homogeneous coordinates are used in perspective projections, depicted in Equation 2, to unify rotation, translation, and depth scaling in a single matrix operation, simplifying transformations between 3D world coordinates and 2D image coordinates. They also handle the perspective distortion by representing depth, ensuring accurate projection of objects at varying distances from the camera [34].

$$\mathbf{P}_w = \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} \quad \mathbf{P}_b = \begin{bmatrix} x_b \\ y_b \\ z_b \\ 1 \end{bmatrix} \quad \mathbf{P}_c = \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} \quad \mathbf{P}_p = \begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} \quad (2)$$

The relationship between the world and body coordinate frames is described by a rotation matrix, \mathbf{R} , and a translation vector, \mathbf{t} . The translation vector represents the position of the world coordinate frame's origin relative to the body coordinate frame, accounting for the positional difference between the two origins. This vector can be computed by taking the negative of the position components of the interceptor drone in the world coordinate frame and then applying the rotation matrix (\mathbf{R}) that describes the transformation from the world frame to the camera frame. The rotation matrix is a 3×3 matrix, while the translation vector is a 1×3 column vector. However, the interception guidance algorithm requires the relative position of the target with respect to the interceptor, the translation vector becomes redundant in the detection and tracking algorithm.

$$\mathbf{P}_b = \begin{bmatrix} \mathbf{R}_{wb} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \mathbf{P}_w \quad (3)$$

The camera coordinate frame is obtained through two sequential rotations from the body coordinate frame: the first is a 90-degree rotation about the Z-axis of the body, followed by a second rotation of $90 + \zeta$ degrees around the intermediate X-axis, where ζ represents the angle between the camera's Z-axis and the body's X-axis. The relationship between a point

in the world coordinate frame (p_w) and in the body coordinate frame (p_b) is given by Equation 3 and Equation 4.

$$\mathbf{p}_c = [\mathbf{R}_{bc} \quad \mathbf{0}] \mathbf{p}_b \quad (4)$$

Equation 5 demonstrates the transformation from the camera coordinates p_c to pixel coordinates p_p . The intrinsic matrix \mathbf{A} is composed of the focal length and the optical center as can be seen in Equation 6. Optical center (c_x, c_y) is the orthogonal projection of the camera center to the image plane. The focal length is the distance of the image plane to the optical center. As the image plane can be a rectangle instead of a square necessitating two focal lengths, f_x and f_y , in the two directions of the image plane, x and y respectively.

$$\mathbf{p}_p = \mathbf{A} \mathbf{p}_c \quad (5)$$

$$\mathbf{A} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (6)$$

In this study the pixel frame to camera coordinate frame transformation is used, the equations to achieve this transformation is found inverting Equation 5 and are demonstrated in Equation 7 and Equation 8.

$$\frac{x_c}{z_c} = \frac{u_p - c_x}{f_x} \quad (7)$$

$$\frac{y_c}{z_c} = \frac{v_p - c_y}{f_y} \quad (8)$$

As the camera frame position is calculated as a fraction of the absolute depth, the radar data becomes significant in the accuracy of the whole detection and tracking algorithm.

D. Velocity Tracking Algorithm

The tracked centroids of the bounding box detecting the target drone across frames are used to calculate the velocity of the target drone. Equation 10 and Equation 11, where $x_i = \frac{x_c}{z_c}$ and $y_i = \frac{y_c}{z_c}$, give the velocity of the target drone for the x and y axis in the pixel (image) frame respectively. The notation for the translation and rotation are taken from Figure 2b and are in the camera frame.

As the optical flow vectors calculated are caused by relative motion of the interceptor and target, the translation vector $T_{c,r} = [U, V, W]$ is also relative and unknown. The rotation vector $R_c = [A, B, C]$ is known only for the interceptor drone which hosts the camera and is received from the gyro measurements of the interceptor. R_c is related to the Euler angles via Equation 4. The notation for both the rotation and translation vector in the camera frame can be seen in Figure 2b. The absolute scale z_c is obtained using the absolute distance from radar and the x_c and y_c camera coordinates from the previous prediction assuming the movement of the target is small and hence negligible across consecutive frames (Equation 9).

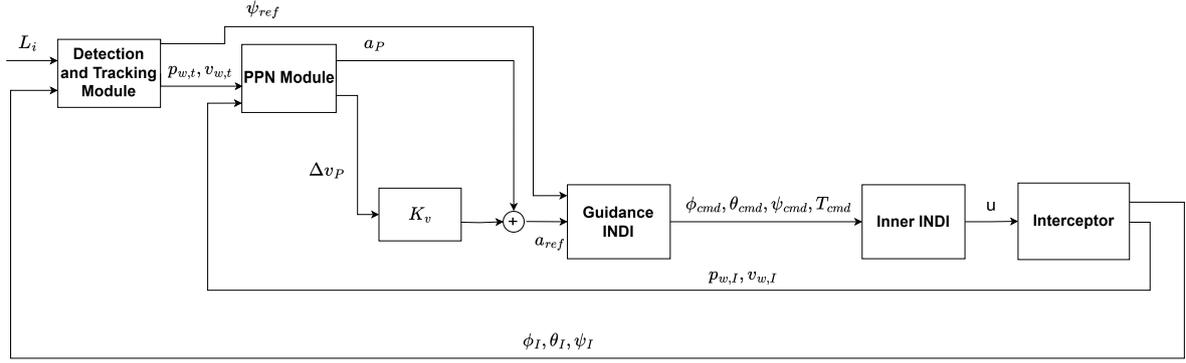


Fig. 3. Block diagram of the interception algorithm

$$z_{c,k+1} = \sqrt{\|r_k\|^2 - x_{c,k}^2 - y_{c,k}^2} \quad (9)$$

$$u_p = -\frac{U}{z_c} + x_i \frac{W}{z_c} + Ax_i y_i - Bx_i^2 - B + Cy_i \quad (10)$$

$$v_p = -\frac{V}{z_c} + y_i \frac{W}{z_c} - Cx_i + Ay_i^2 + A - Bx_i y_i \quad (11)$$

Equations 10 and 11 give the velocity of the target drone in the pixel frame where u_p is the velocity along the x axis and v_p is the velocity along y axis shown in Figure 2a. Rearranging equations 10 and 11 for calculating velocities in camera frame results in equations 12 and 13. The W which is the velocity along the z axis in the camera coordinate frame is calculated by differentiating the z_c position.

$$U = -\frac{W \cdot x_c}{z_c} + \frac{A \cdot x_c \cdot y_c}{z_c} - \frac{B \cdot x_c^2}{z_c} - B \cdot z_c + C \cdot Y - u_i \cdot z_c \quad (12)$$

$$V = -\frac{W \cdot y_c}{z_c} - \frac{B \cdot x_c \cdot y_c}{z_c} - \frac{A \cdot y_c^2}{z_c} + A \cdot z_c - C \cdot x_c - v_i \cdot z_c \quad (13)$$

The target velocity vector in the camera coordinate frame can be transformed to the world coordinate frame using a similar equation as Equation 3 and Equation 4. For this transformation going from the camera coordinate frame to world coordinate frame the \mathbf{R} is the inverse of the world to camera frame rotation matrix and the translation vector \mathbf{t} is replaced with the velocity of the interceptor drone v_I in the world coordinate frame. As detailed for the position, the interception guidance requires the target velocity relative to the interceptor and hence the velocity vector becomes redundant in the coordinate transformation.

E. Kalman Tracking

The state vector $\mathbf{x}_k = [\mathbf{p}_{w,t}, \mathbf{v}_{w,t}, \mathbf{a}_{w,t}]$ for time k is composed of the position vector $\mathbf{p}_{w,t} = [x_{w,t}, y_{w,t}, z_{w,t}]$, velocity vector $\mathbf{v}_{w,t} = [\dot{x}_{w,t}, \dot{y}_{w,t}, \dot{z}_{w,t}]$ and acceleration vector

$\mathbf{a}_{w,t} = [\ddot{x}_{w,t}, \ddot{y}_{w,t}, \ddot{z}_{w,t}]$. Also defined is the output vector $\mathbf{o}_n = [\mathbf{p}_{w,t}, \mathbf{v}_{w,t}]$ which are the target states of interest for the interception guidance. It is important to note that the position and velocity of the target drone are in world coordinate system but are relative to the interceptor drone.

$$\mathbf{x}_k = \mathbf{F}\mathbf{x}_{k-1} + \epsilon_n \quad (14)$$

$$\mathbf{o}_k = \mathbf{H}\mathbf{x}_k + \nu_k \quad (15)$$

Equation 14 shows the state equation where constant acceleration is assumed and Equation 15 shows the output equation where k is the time index. In the equations F is the state transition matrix and H is the output matrix, $\epsilon_n \approx N(0, \sigma_\epsilon^2)I$ is the modelling error and $\nu_n \approx N(0, \sigma_\nu^2)$ is the measurement error. I is the identity matrix of appropriate size and dt is the time step.

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & 0 & dt & 0 & 0 & 0.5 \cdot dt^2 & 0 & 0 \\ 0 & 1 & 0 & 0 & dt & 0 & 0 & 0.5 \cdot dt^2 & 0 \\ 0 & 0 & 1 & 0 & 0 & dt & 0 & 0 & 0.5 \cdot dt^2 \\ 0 & 0 & 0 & 1 & 0 & 0 & dt & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & dt & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & dt \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (16)$$

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \quad (17)$$

Also the covariance matrix \mathbf{P} initially has the form 18 with

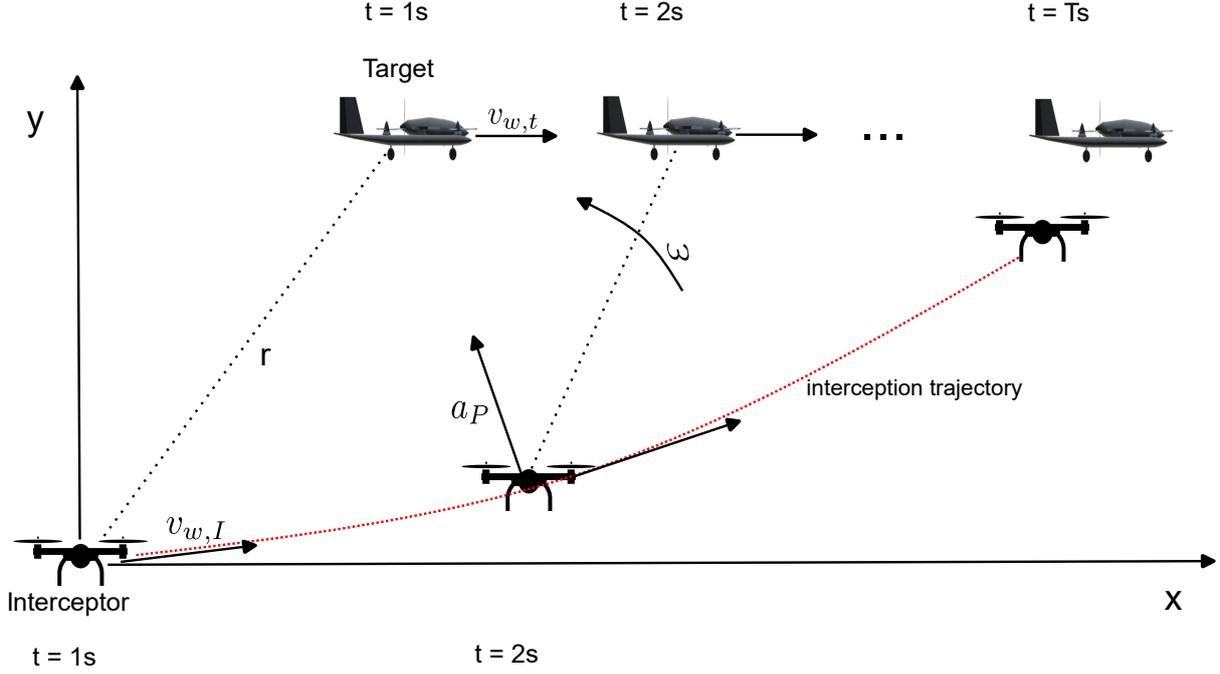


Fig. 4. An example interception trajectory and resulting scene geometry. a_P can be observed to be orthogonal to ω and $v_{w,t}$. T is the time of interception.

large values of K_p and K_v .

$$\mathbf{P} = \begin{bmatrix} K_p & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & K_p & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & K_p & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & K_v & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & K_v & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & K_v & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & K_v & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & K_v & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & K_v \end{bmatrix} \quad (18)$$

The \mathbf{Q} is the matrix which represent the process noise statistics. The cross-covariance components (σ_{pv}^2 , σ_{pa}^2 and σ_{va}^2) are assumed zero.

$$\mathbf{Q} = \begin{bmatrix} \sigma_p^2 & 0 & 0 & \sigma_{pv}^2 & 0 & 0 & \sigma_{pa}^2 & 0 & 0 \\ 0 & \sigma_p^2 & 0 & 0 & \sigma_{pv}^2 & 0 & 0 & \sigma_p^2 & 0 \\ 0 & 0 & \sigma_p^2 & 0 & 0 & \sigma_{pv}^2 & 0 & 0 & \sigma_{pa}^2 \\ \sigma_{pv}^2 & 0 & 0 & \sigma_v^2 & 0 & 0 & \sigma_{va}^2 & 0 & 0 \\ 0 & \sigma_{pv}^2 & 0 & 0 & \sigma_v^2 & 0 & 0 & \sigma_{va}^2 & 0 \\ 0 & 0 & \sigma_{pv}^2 & 0 & 0 & \sigma_v^2 & 0 & 0 & \sigma_{va}^2 \\ \sigma_{pa}^2 & 0 & 0 & \sigma_{va}^2 & 0 & 0 & \sigma_a^2 & 0 & 0 \\ 0 & \sigma_{pa}^2 & 0 & 0 & \sigma_{va}^2 & 0 & 0 & \sigma_a^2 & 0 \\ 0 & 0 & \sigma_{pa}^2 & 0 & 0 & \sigma_{va}^2 & 0 & 0 & \sigma_a^2 \end{bmatrix} \quad (19)$$

The \mathbf{R} is the matrix which represent the measurement noise statistics. As the same measurement tools, camera and radar, are used for all measurements, the covariance is assumed to be the same across components.

$$\mathbf{R} = \begin{bmatrix} \sigma_v^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_v^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_v^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_v^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_v^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_v^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \sigma_v^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma_v^2 & 0 \end{bmatrix} \quad (20)$$

The Kalman filter devised follows the linear Kalman filter structure and has the following steps:

Step 1: Prediction state (Equation 21)

$$\mathbf{x}_{k+1,k} = \mathbf{F} \cdot \mathbf{x}_{k,k} \quad (21)$$

Step 2: Prediction of the covariance matrix (Equation 22)

$$\mathbf{P}_{k+1,k} = \mathbf{F} \cdot \mathbf{P}_{k,k} \cdot \mathbf{F}^T + \mathbf{Q} \quad (22)$$

Step 3: Innovation of the state (Equation 23)

$$\mathbf{o}_{k+1} = \mathbf{z}_{k+1} - \mathbf{H} \cdot \mathbf{x}_{k+1,k} \quad (23)$$

Step 4: Innovation of the covariance (Equation 24)

$$\mathbf{S}_{k+1} = \mathbf{H} \cdot \mathbf{P}_{k+1,k} \cdot \mathbf{H}^T + \mathbf{R} \quad (24)$$

Step 5: Kalman gain (Equation 25)

$$\mathbf{K}_{k+1} = \mathbf{P}_{k+1,k} \cdot \mathbf{H}^T \cdot \mathbf{S}_{k+1}^{-1} \quad (25)$$

Step 6: Updated optimal state estimation (Equation 26)

$$\mathbf{x}_{k+1,k+1} = \mathbf{x}_{k+1,k} + \mathbf{K}_{k+1} \cdot \mathbf{o}_{k+1} \quad (26)$$

Step 7: Updated covariance (Equation 27)

$$\mathbf{P}_{k+1,k+1} = (\mathbf{I} - \mathbf{K}_{k+1} \cdot \mathbf{H}) \cdot \mathbf{P}_{k+1,k} \quad (27)$$

When a measurement is available, all the steps of the linear Kalman filter are performed. However, in the case of intermittent missed detections, the prediction state from Equation 21 is used as the best estimate, and only Step 1 (prediction) and Step 2 (error covariance update) are executed.

F. Interception Guidance Algorithm

The Pure Proportional Navigation (PPN) guidance commands an acceleration perpendicular to the interceptors velocity and the line of sight rate [30]. This acceleration generates a turn which is proportional to the angular velocity of the line of sight line ω . The goal is to accelerate such that $\omega = 0[\text{rad/s}]$ which leads to a collision triangle as illustrated in Figure 4 if the following assumptions hold:

- The interceptor acceleration is greater than or equal to the target acceleration along the line of sight (los) line.
- The initial change in the line of sight range should be negative ($\dot{r} < 0[m]$).

The collision triangle, under the assumptions mentioned above, leads to interception. To abide by these assumptions, the interception guidance algorithm initially commences after the detection and tracking modules produce a first relative position and velocity estimate. Subsequently, an initial velocity and acceleration reference is produced for the interceptor. This ensures that the line-of-sight range decreases throughout the interception.

The PPN acceleration a_P is calculated using Equation 28, where $\mathbf{v}_{w,I}$ is the interceptor velocity in world coordinate frame and N_P is the navigation gain and is analytically proven to decrease the r to 0m over time when $N_P \leq -3$ [26].

$$\mathbf{a}_P = N_P \mathbf{v}_{w,I} \times \omega \quad (28)$$

The line of sight rate ω is computed using Equation 29, where relative velocity $\mathbf{v}_r = \mathbf{v}_{w,I} - \mathbf{v}_{w,t}$.

$$\omega = \frac{\mathbf{r} \times \mathbf{v}_r}{\mathbf{r} \cdot \mathbf{r}^T} \quad (29)$$

In order to better track the desired velocity, especially when there may be rapid change in velocity commands such as during the interception of a circular moving target, also a velocity reference v_P is generated using the calculated PPN acceleration. Employing Equation 30 the PPN velocity reference is calculated. The initial velocity is calculated using Equation 31. The first part of Equation 31 accounts for the direction of the line of sight line and is scaled to increase the velocity along the line and the second part adds the target velocity ensuring the velocity of the interceptor is greater than the target along the line of sight line. The velocity direction is found by integrating the PPN acceleration and the velocity magnitude V_P is computed as $V_P = \|\mathbf{v}_{P_0}\|$. The velocity reference in combination with the PPN acceleration is used to

create one acceleration reference for the guidance controller, which can be seen in Figure 3.

$$\mathbf{v}_{P_k} = V_P \frac{\mathbf{v}_{P_{k-1}} + dt \cdot \mathbf{a}_{P_{k-1}}}{\|\mathbf{v}_{P_{k-1}} + dt \cdot \mathbf{a}_{P_{k-1}}\|} \quad (30)$$

$$\mathbf{v}_{P_0} = b \frac{r_0}{\|r_0\|} + \mathbf{v}_{w,t} \quad (31)$$

The calculated reference values are then fed to the Incremental Non-Linear Dynamic Inversion (INDI) guidance controller. This controller receives the necessary acceleration a_{ref} and yaw angle ψ_{ref} calculated using 32 and output commanded pitch angle θ_{cmd} , roll angle ϕ_{cmd} , yaw angle ψ_{cmd} and thrust magnitude T_{cmd} .

$$\psi_{ref} = \arctan(y_{w,t}/x_{w,t}) \quad (32)$$

The inner INDI, implemented as detailed in [31], receives the commanded Euler angles and thrust magnitude and outputs the necessary angular velocity of the motors $u = [\omega_{p_1}, \omega_{p_2}, \omega_{p_3}, \omega_{p_4}]$. The overall structure of the interception algorithm can be seen in Figure 3. Here the Detection and Tracking Module receives the frame L_i and attitude angles and calculates the necessary yaw angle ψ_{ref} , the state of the target \mathbf{o}_k which is the best state estimate from the Kalman filter. The PN Module receives the target states \mathbf{o}_k and interceptor states $\mathbf{p}_{w,I}$ and $\mathbf{v}_{w,I}$ and outputs reference velocity and acceleration.

G. Visibility

For accurate detection and tracking, the visibility of the target drone in the camera frame is maximized. The camera is mounted on top of the interceptor and angled positively with respect to the interceptor's body x-axis, meaning the camera is positioned to look above the interceptor. To maximize visibility in this configuration and ensure that the background remains primarily the sky, the interceptor approaches the target from below. The effect of the parameter b is explored in Equation 31, which is used to calculate the initial velocity components. b scales the velocity along the direction of the relative position; a larger b in a specific axis results in a greater initial velocity in that axis.

To ensure the interceptor approaches from below, the ratio between the relative positions in the x and y axes and the z axis is considered using Equation 33. The motivation behind the formula for C_b is to balance the interceptor's velocity across all axes to prevent it from reaching the target's z-axis position too quickly. The sum $x + y$ is divided by 2 to account for the fact that two independent distances (x and y) are being combined, while only one vertical distance z is being considered. Without this normalization, the horizontal component might dominate the ratio excessively, since $x + y$ typically adds more value than z , making the interceptor respond too aggressively to horizontal distances at the expense of maintaining a smooth vertical approach. If the ratio C_b is high, it means the interceptor needs to travel further in the x and y directions compared to the z-axis. Without adjusting for this, if the parameter b were kept the same across all axes,

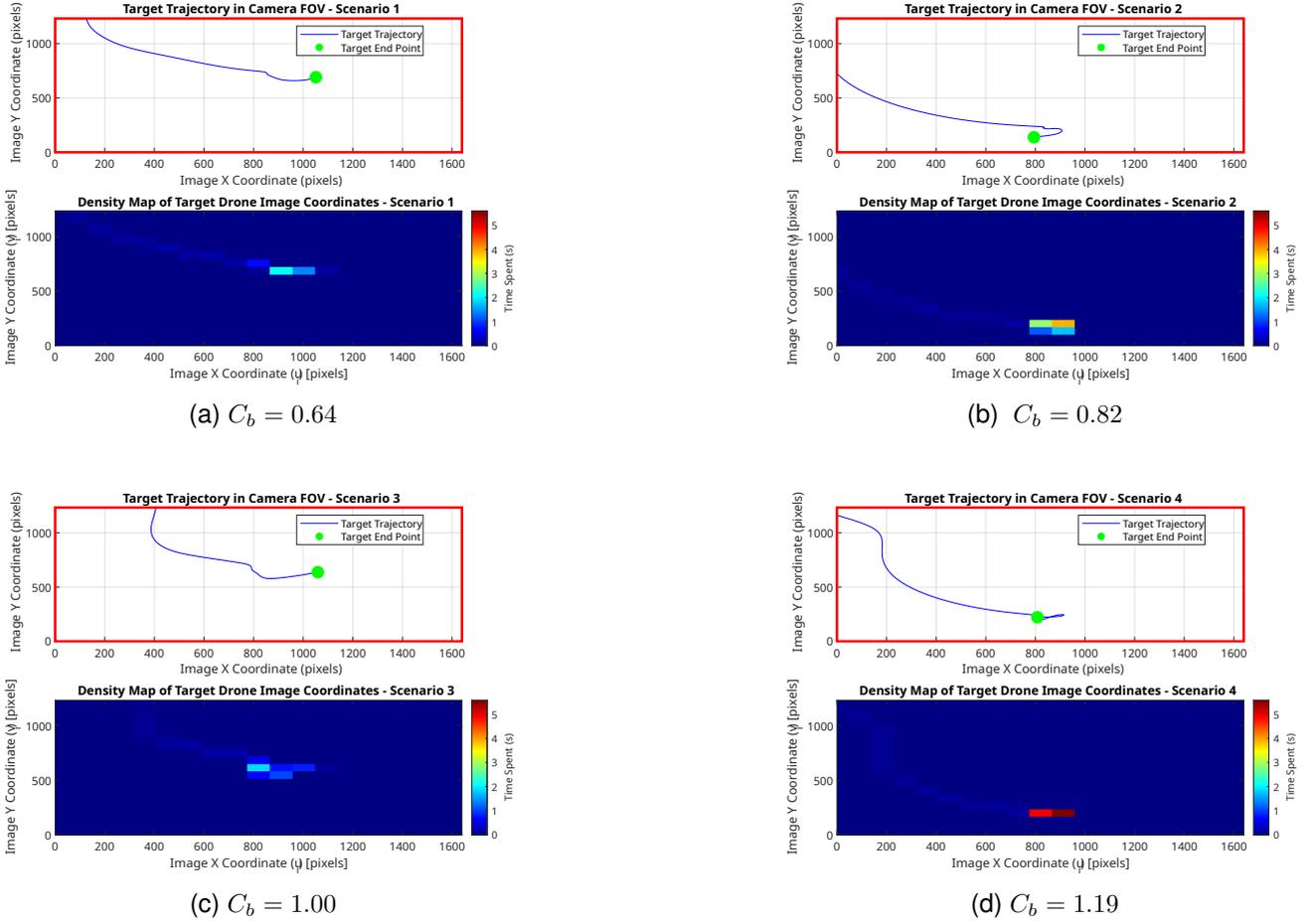


Fig. 5. Target drone trajectory in the camera frame and corresponding density maps

the interceptor would reach the target's z -axis position sooner than its $x - y$ position, potentially causing the target drone to move out of the camera's field of view.

By incorporating the ratio C_b , the initial velocity along the z -axis is reduced when the horizontal distance (x and y) is large, allowing the interceptor to maintain a consistent approach from below while keeping the target within the camera's field of view. This ensures smoother, more effective tracking and interception.

$$C_b = \frac{r_x + r_y}{2r_z} \quad (33)$$

In order to account for this effect, the z value of the b parameter, b_z , is scaled inversely with the ratio C_b . As the displacement in the x and y axes increases significantly relative to the z axis, the z value of the b parameter should decrease, resulting in a lower initial velocity in the z axis.

A MATLAB simulation of several interception scenarios, where the initial relative position satisfies $\|r_0\| \leq 150$ [m], was conducted. From the simulation results, an empirical formula for b_z was derived to ensure that the final desired positions in the x and y axes are not reached before the z

axis. The simulation also validated that detection and tracking are feasible, and it was used to determine the camera angle ζ that maximizes target drone visibility.

Figure 5 illustrates the simulation results with $\zeta = 80^\circ$. Each scenario shows different and increasing values of C_b , and the target drone remains visible in each case when the scaled b_z is applied. Additionally, the simulation results show that a camera angle of $\zeta = 80^\circ$ maximizes target drone visibility, with the target drone remaining in the camera frame for 83% of the total time across all interception scenarios. The decrease in visibility is primarily due to scenarios where the target drone is in close proximity in the $x-y$ plane while at a further distance in z axis. To address this, the values of the b parameters in the x and y axes should also be varied according to the C_b ratio.

Another factor affecting target drone visibility occurs during interceptions of circularly moving targets, where the interceptor follows a curved path, as shown in Figures 7a and 8c. Although this does not affect the overall interception result, it may cause the target to temporarily move out of the field of view. To mitigate this, the acceleration of the target drone can be feedforwarded to the Proportional Navigation (PN)

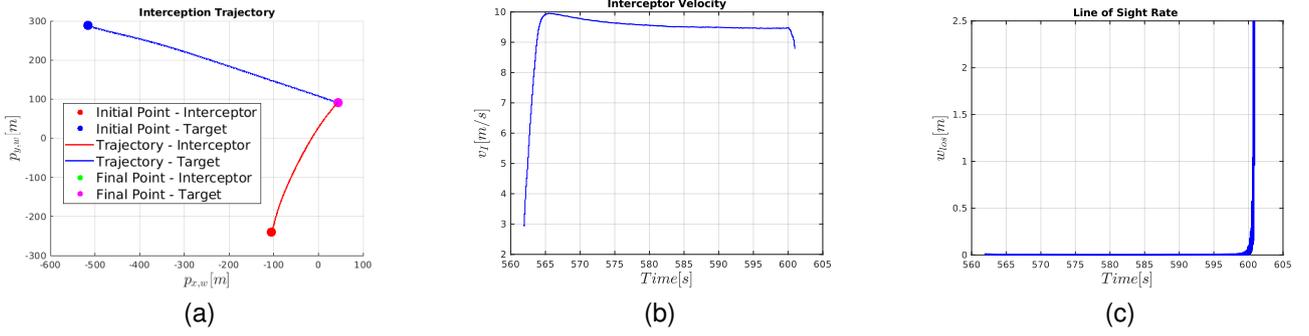


Fig. 6. Interception of a target moving in a straight line using JSBSim. From left to right, the figures show: (a) the 2D position of the target relative to the interceptor, with the closest distance being 0.48, [m]; (b) the interceptor’s velocity during the interception; and (c) the line-of-sight rate between the interceptor and the target.

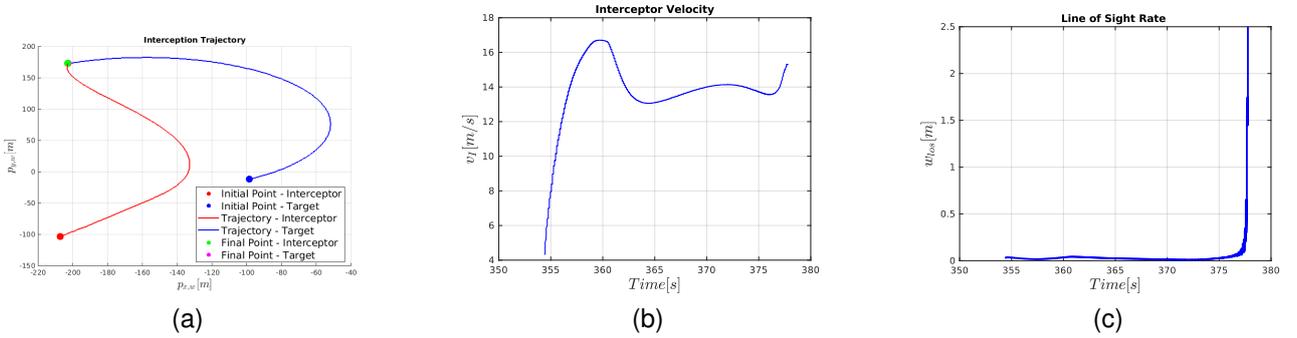


Fig. 7. Interception of a circular moving target using JSBSim. From left to right, the figures show: (a) the 2D position of the target relative to the interceptor, with the closest distance being 0.38, [m]; (b) the interceptor’s velocity during the interception; and (c) the line-of-sight rate between the interceptor and the target.

acceleration in Equation 28.

IV. EXPERIMENTAL RESULTS

To validate the interception guidance algorithm, several tests are conducted, including JSBSim simulations of realistic interception scenarios and a flight test at the TU Delft Cyberzoo using a Parrot Bebop2 quadcopter as the interceptor and a simulated target. For the experiments, the proportional navigation algorithm is implemented in the Paparazzi autopilot framework. In the JSBSim simulations, both the interceptor and target drones are simulated, with initial distances and velocities accurately reflecting real-world conditions. Despite the simulations, the velocity and positional accuracy were representative of a larger-scale outdoor experiment.

In the experiment in TU Delft Cyberzoo Lab, for each scenario, the interceptor started at a fixed altitude of 1.0[m], while the target remained at a fixed altitude of 5.0[m]. The interception was considered complete when the interceptor reached a distance of 0.5[m] from the target. During flight tests in the Cyberzoo, where the target is simulated in JSBSim, the position and velocity data from the simulated target were sent to the ground control station. This data is then transmitted to the interceptor, which is being tracked by the OptiTrack system within the lab’s environment. Several interception scenarios

are presented, one where the target is moving in a straight line, one where the target is circling at a fixed radius and the interceptor starts from the center of the circle and one where the target is circling and the interceptor starts outside the circle. The plots can be seen from Figure 8. In each scenario the interceptor is able to intercept the target in the given 0.5[m] threshold.

Although position error of the interceptor is not taken into account in the controller, in all interception scenarios, the interceptor’s position remains aligned with the required trajectory for successful interception. However, in scenarios where the target is circling such as Figures 7a, 8b, 8c, the initial trajectory prediction assumes the target moves without acceleration, leading to the interceptor following a curved path. This curve stems from the limitations of pure proportional navigation, which does not account for target acceleration. This suggests that a target executing a circular maneuver can potentially deceive the interceptor, depending on the initial conditions such as the target’s and interceptor’s velocity and position, as well as the radius of the target’s circular maneuver. These effects are shown in Figure 9, specifically Figure 9c, demonstrating the effect of the target circle radii on interception error, show that the interception error increases significantly when the interceptor is intercepting a circular

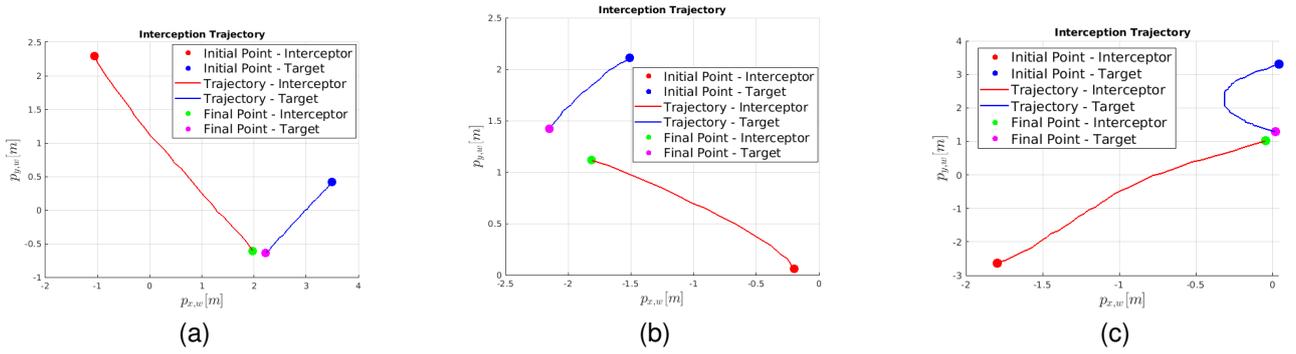


Fig. 8. 2D position plots of interception scenarios in TU Delft Cyberzoo where the interceptor is flown in real-life and the target is simulated using JSBSim. In all scenarios the closest 3D distance is between interceptor and target is $0.5[m]$. The plots, from left to right, depict: (a) a target moving in a straight line, (b) a target moving in a circular path with the interceptor starting inside the circle, and (c) a target moving in a circular path with the interceptor starting outside the circle.

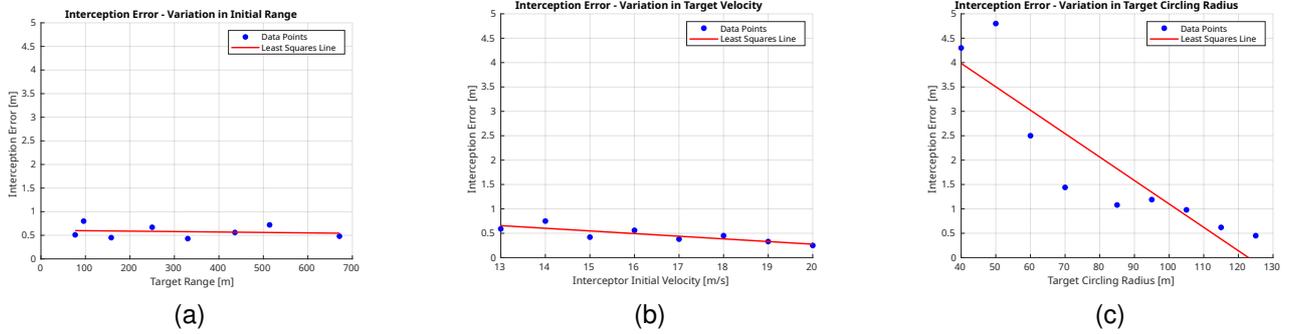


Fig. 9. Plots for closest 3D distance between the target and the interceptor given variations in, from left to right, initial range, target velocity and target circling radius. The data is collected from experiments where both drones are simulated in JSBSim.

moving target and tighter the turn is for the circle maneuver the larger the error due to increase in acceleration.

During the interceptions, the velocity remained fairly constant in the scenario where the target was moving in a straight line, demonstrated in Figure 6b. Nevertheless when the target is moving in circles, slight oscillations are apparent in the magnitude of the velocity, seen in Figure 7b. As the target is accelerating/decelerating outside the predicted envelope of the proportional navigation, the line of sight rate around the specific axis increases/decreases, in turn this increase/decrease in line of sight rate results in an increase/decrease in commanded acceleration in that axis. In a circling target scenario the acceleration outside the predicted envelope happens at a fixed period which causes the oscillation on the components of the velocity. As the commanded velocity by the PN is only changed in direction from the initial velocity and not in magnitude as seen in Equation 30, this is attributed to a change in acceleration commanded by the proportional navigation guidance.

In theory the line of sight should go to $0[rad/s]$ as the interception progresses. A similar trend can be observed in the interception scenarios. Nevertheless, in close range the line of sight rate seems to increase and oscillate. This can be

explained by looking at how the line of sight rate is calculated in Equation 29. The line of sight rate is inversely proportional to the line of sight range squared. Meaning in close proximity even a tiny change in the line of sight produce a high line of sight rate.

In Figure 9, the effects of varying parameters, such as the initial range between the interceptor and target, target velocity, and the radius of the circular trajectory, are explored. First, it is observed that a target moving in a circular path results in a larger interception error (averaging $1.00, m$) compared to a target moving in a straight line (averaging $0.5, m$), as shown in Figures 9a and 9b for the straight-line case, and in Figure 9c for the circular path case. Second, an increase in initial range or target velocity does not significantly affect interception performance, provided the interceptor has sufficient maximum velocity. Third, a smaller circling radius leads to greater interception error, due to the corresponding increase in the required acceleration.

V. CONCLUSION

This paper presents a comprehensive solution for the autonomous air-to-air interception of loitering kamikaze drones. Several detection algorithms, including YOLOv7, Tiny

YOLOv3, and a blob detector, were implemented and evaluated in terms of accuracy and computational efficiency. While YOLO-based models delivered superior detection accuracy, with an average confidence of 75%, the blob detector offered faster processing at 8 FPS on the Raspberry Pi 4B, making it better suited for real-time onboard execution. For target tracking, the Discriminative Correlation Filter with Channel and Spatial Reliability (CSRT) tracker was employed, enhanced by image preprocessing techniques such as adaptive thresholding and Gaussian blur, achieving approximately 10 FPS on the same platform. A linear Kalman filter was integrated to fuse data from the camera and radar, helping to mitigate the effects of missed detections.

The guidance strategy was based on PPN, adapted for quadcopters, and paired with an Incremental Nonlinear Controller to ensure the interceptor approached the target from below. Tests conducted in the TU Delft Cyberzoo and simulations showed the algorithm could successfully intercept both circling and straight-line targets, with an average interception error of 2.1 m for circling targets and 0.5 m for straight-line targets, when the target's position and velocity were provided. The study found that accounting for the target's acceleration would improve interception accuracy, particularly for circular movements, although this remains constrained by the onboard sensor's limitations in detecting acceleration.

A visibility study, which leveraged the decoupled yaw axis for optimal camera positioning, showed that the target remained within the camera's field of view for 83% of the interception duration, given an initial distance of up to 150 m between the interceptor and the target. The study concludes that intercepting a loitering UAV using a camera and radar is feasible but highlights several limitations. First, while convolutional neural networks (CNNs) offer better scalability for detection, further optimization is required for edge deployment to achieve a frame rate higher than 2 FPS. Second, intercepting targets with circular movement would benefit from better acceleration detection capabilities. Lastly, ensuring that the interceptor drone can move faster than the target drone is critical for accommodating a wider range of initial scene geometries and improving scalability.

In the future, the devised detection and tracking algorithm should be integrated with the interception guidance algorithm to test the autonomous interception of a non-cooperative target drone. On top of this an edge device with better computational abilities, such as a GPU, should be explored for deploying YOLO detection algorithms.

REFERENCES

- [1] Saeid Aghaziyarati, Saed Moradi, and Hasan Talebi. "Small infrared target detection using absolute average difference weighted by cumulative directional derivatives". In: *Infrared Physics Technology* 101 (June 2019). DOI: 10.1016/j.infrared.2019.06.003.
- [2] Al Jazeera. *Venezuela's Nicolas Maduro survives drone 'attack'*. <https://aljazeera.com/news/2018/8/5/venezuelas-nicolas-maduro-survives-drone-attack>.
- [3] BBC. *How are 'kamikaze' drones being used by Russia and Ukraine?* <https://www.bbc.com/news/world-62225830>.
- [4] John Canny. "A Computational Approach to Edge Detection". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 6 (Nov. 1986), pp. 679–698. DOI: 10.1109/TPAMI.1986.4767851.
- [5] Cetin A.T. and Koyuncu E. "Model Predictive Control-Based Guidance with Impact Angle Constraints for Visual Quadrotor Interception". In: *9th International Conference on Control, Decision and Information Technologies* (2023), pp. 2544–2549. DOI: 10.1109/codit58514.2023.10284484.
- [6] Fang Chen et al. "Local Patch Network With Global Attention for Infrared Small Target Detection". In: *IEEE Transactions on Aerospace and Electronic Systems* 58.5 (2022), pp. 3979–3991. DOI: 10.1109/TAES.2022.3159308.
- [7] Yueru Chen et al. "A Deep Learning Approach to Drone Monitoring". In: (Dec. 2017). DOI: 10.48550/arXiv.1712.00863.
- [8] Namhooon Cho and Youdan Kim. "Optimality of augmented ideal proportional navigation for maneuvering target interception". In: *IEEE Transactions on Aerospace and Electronic Systems* 52.2 (2016), pp. 948–954. DOI: 10.1109/TAES.2015.140432.
- [9] Dan Sabbagh. *Houthis call west's bluff with renewed Red Sea drone assault*. <https://www.theguardian.com/world/2024/jan/10/houthis-call-west-bluff-with-renewed-red-sea-drone-assault>.
- [10] Chenqiang Gao, Jinwen Tian, and Ping Wang. "Generalised-structure-tensor-based infrared small target detection". In: *Electronics Letters* 44 (Dec. 2008), pp. 1349–1351. DOI: 10.1049/el:20081781.
- [11] Chenqiang Gao et al. "Infrared Small-dim Target Detection Based on Markov Random Field Guided Noise Modeling". In: *Pattern Recognition* 76 (Nov. 2017). DOI: 10.1016/j.patcog.2017.11.016.
- [12] R.C. Gonzalez and R.E. Woods. *Digital Image Processing*. 4th ed. London: Pearson Education, 2018.
- [13] Jinhui Han, Saed Moradi, and Iman Faramarzi. "A Local Contrast Method for Infrared Small-Target Detection Utilizing a Tri-Layer Window". In: *IEEE Geoscience and Remote Sensing Letters* (Dec. 2019), pp. 1–5. DOI: 10.1109/LGRS.2019.2954578.
- [14] Yin Zhang Hanqing Guo Ye Zheng. "Global-Local MAV Detection under Challenging Conditions based on Appearance and Motion". In: *Cornell University* (Dec. 2023).
- [15] Markus Hehn and Raffaello D'Andrea. "Quadcopter Trajectory Generation and Control". In: *IFAC Proceedings Volumes* 44.1 (2011), pp. 1485–1491. DOI: 10.1109/SSST.2002.1027091.
- [16] Markus Hehn and Raffaello DAndrea. "Real-time trajectory generation for interception maneuvers with quadcopters". In: *2012 IEEE/RSJ International Con-*

- ference on Intelligent Robots and Systems*. 2012, pp. 4979–4984. DOI: 10.1109/IROS.2012.6386093.
- [17] Joby Warrick. *Use of weaponized drones by ISIS spurs terrorism fears*. https://www.washingtonpost.com/world/national-security/use-of-weaponized-drones-by-isis-spurs-terrorism-fears/2017/02/21/9d83d51e-f382-11e6-8d72-263470bf0401_story.html.
- [18] Lindsey Wiley Joshua Follansbee and Patrick Leslie. “Drone detection performance in the reflective bands: visible, near infrared, short wave infrared, and extended short wave infrared”. In: *Optical Engineering* 61 (2022), pp. 95–106.
- [19] Keith Wagstaff. *Tokyo Police to Deploy Net-Carrying Drone to Catch Rogue Drones*. <https://www.nbcnews.com/tech/tech-news/tokyo-police-deploy-net-carrying-drone-catch-rogue-drones-n478596>.
- [20] Jing Li et al. “Fast and Robust UAV to UAV Detection and Tracking From Video”. In: *IEEE Transactions on Emerging Topics in Computing* 10.3 (2022), pp. 1519–1531. DOI: 10.1109/TETC.2021.3104555.
- [21] Alan Lukezic, Tomas Vojir, and Jiri Matas. “A Discriminative Correlation Filter Tracker with Channel and Spatial Reliability”. In: *International Journal of Computer Vision* 126 (Jan. 2018), pp. 671–688. DOI: 10.1007/s11263-017-1061-3.
- [22] Endre Papp Marina Moreira and Rodrigo Ventura. “Interception of Non-Cooperative UAVs”. In: *IEEE International Symposium on Safety, Security, and Rescue Robotics* (2019).
- [23] Daniel Mellinger, Nathan Michael, and Vijay Kumar. “Trajectory Generation and Control for Precise Aggressive Maneuvers with Quadrotors”. In: *STAR* 79 (2012), pp. 361–373.
- [24] Sara Minaeian, Jian Liu, and Young-Jun Son. “Effective and Efficient Detection of Moving Targets From a UAV’s Camera”. In: *IEEE Transactions on Intelligent Transportation Systems* 19.2 (2018), pp. 497–506. DOI: 10.1109/TITS.2017.2782790.
- [25] Juhwan Noh et al. “Tractor Beam: Safe-Hijacking of Consumer Drones with Adaptive GPS Spoofing”. In: *ACM Trans. Priv. Secur.* 22.2 (Apr. 2019). ISSN: 2471-2566. DOI: 10.1145/3309735.
- [26] Neil Palumbo, Ross Blauwkamp, and Justin Lloyd. “Basic Principles of Homing Guidance”. In: *Johns Hopkins Apl Technical Digest* 29 (Jan. 2010), pp. 25–41.
- [27] Robin Radar. *10 Types of Counter Drone Technology To Detect And Stop Drones Today*. <https://www.robinradar.com/press/blog/10-counter-drone-technologies-to-detect-and-stop-drones-today>.
- [28] Payel Rot, Saurab Dutta, and Nilanjan Dey. “Adaptive thresholding: A comparative study”. In: *2014 International Conference on Control, Instrumentation, Communication and Computational Technologies* (July 2014). DOI: 10.1109/ICCICCT.2014.6993140.
- [29] Samira Shackle. *The mystery of the Gatwick drone*. <https://www.theguardian.com/uk-news/2020/dec/01/the-mystery-of-the-gatwick-drone>.
- [30] U.S. Shukla and P.R. Mahapatra. “The proportional navigation dilemma-pure or true?”. In: *IEEE Transactions on Aerospace and Electronic Systems* 26.2 (1990), pp. 382–392. DOI: 10.1109/7.53445.
- [31] Ewoud Smeur, Guido Croon, and Q.P. Chu. “Cascaded Incremental Nonlinear Dynamic Inversion Control for MAV Disturbance Rejection”. In: *Control Engineering Practice* 73C (Apr. 2018), pp. 79–90. DOI: 10.1016/j.conengprac.2018.01.003.
- [32] Meng Hwa Er Suyog D. Deshpande. “Max-mean and max-median filters for detection of small targets”. In: *Signal and Data Processing of Small Targets* (Oct. 1999). DOI: <https://doi-org.tudelft.idm.oclc.org/10.1117/12.364049>.
- [33] Satoshi Suzuki and be Keiichia. “Topological structural analysis of digitized binary images by border following”. In: *Computer Vision, Graphics, and Image Processing* 30 (1985), pp. 32–46. DOI: 10.1016/0734-189X(85)90016-7.
- [34] Carlo Tomasi. “Detection and Tracking of Point Features”. In: 1991.
- [35] C. Tournes and C.D. Johnson. “Augmented proportional navigation guidance using subspace-stabilization”. In: *Proceedings of the Thirty-Fourth Southeastern Symposium on System Theory (Cat. No.02EX540)* (2002), pp. 471–475. DOI: 10.1109/SSST.2002.1027091.
- [36] Viktor Walter, Matous Vrba, and Martin Saska. “On training datasets for machine learning-based visual relative localization of micro-scale UAV’s”. In: (May 2020). DOI: 10.1109/ICRA40945.2020.9196947.
- [37] Daniel Tan Wei Xun, Yoke Lin Lim, and Sutthiphong Srigrarom. “Drone detection using YOLOv3 with transfer learning on NVIDIA Jetson TX2”. In: *2021 Second International Symposium on Instrumentation, Control, Artificial Intelligence, and Robotics (ICA-SYMP)*. 2021, pp. 1–6. DOI: 10.1109/ICA-SYMP50206.2021.9358449.
- [38] Ming Zeng, Jianxun Li, and Zhang Peng. “The design of Top-Hat morphological filter and application to infrared target detection”. In: *Infrared Physics Technology* 48 (Apr. 2006), pp. 67–76. DOI: 10.1016/j.infrared.2005.04.006.
- [39] Jie Zhao et al. “Vision-Based Anti-UAV Detection and Tracking”. In: *IEEE Transactions on Intelligent Transportation Systems* 23.12 (2022), pp. 25323–25334. DOI: 10.1109/TITS.2022.3177627.
- [40] Xia Y Zhao X. “YOLO-ViT-Based Method for Unmanned Aerial Vehicle Infrared Vehicle Target Detection”. In: *Remote Sensing* 15 (2023).
- [41] Ye Zheng et al. “Air-to-Air Visual Detection of Micro-UAVs: An Experimental Evaluation of Deep Learning”. In: *IEEE Robotics and Automation Letters* 6.2 (2021), pp. 1020–1027. DOI: 10.1109/LRA.2021.3056059.

Part III

Additional Results

4

More Results

This section highlights additional work conducted during the research that was not included in the scientific article. It provides an overview of the UAVs used in the experiments, including their hardware configurations, as well as the intermediate software developed to integrate these components effectively. Furthermore, it elaborates on the Cyberzoo experiment results briefly mentioned in the scientific article, offering a more comprehensive analysis. Additionally, a block diagram illustrating the visibility simulation process is presented, along with supplementary visibility results to ensure completeness.

4.1. Hardware and Software

The ultimate goal of the thesis was to be able to demonstrate the interception in real-life with two drones. The drones can be seen in Figure 4.1.



SonicModell AR Wing Pro



Parrot Bebop2

Figure 4.1: Drones used in the experiments

The AR Wing Pro was received as a kit and was ready for manual flight. However, to facilitate the experiments, it was necessary to enable autonomous flight for the target drone. Initially, the AR Wing Pro was flown manually using the PaparazziUAV autopilot software with the attitude controller enabled in the

rotorcraft firmware. Subsequently, the drone was configured for autonomous operation using the fixed-wing firmware, where a PID controller was implemented and tested at the Unmanned Valley. Ultimately, the AR Wing Pro achieved autonomous flight patterns, including oval and circular trajectories, which were suitable for straight-line interception experiments. It was found that the AR Wing Pro had a minimum velocity of approximately $15m/s$ in no-wind conditions, a critical parameter since the Parrot Bebop2 has a maximum velocity of $18m/s$. This velocity difference was important for ensuring that the Proportional Navigation (PN) guidance method maintained an initial advantage in line-of-sight velocity for the interceptor.

The Parrot Bebop2 was chosen as the interceptor for the experiments due to its ease of implementation and the availability of existing modules for autonomous flight. As noted earlier, the Bebop2's lower maximum velocity compared to the AR Wing Pro limited the interception geometries, restricting most scenarios to head-on configurations. To enhance visibility, the camera's yaw axis was used to track the target. However, in high-velocity scenarios, motor saturation often rendered the yaw axis unavailable.

For image processing and camera operations, a Raspberry Pi 4B with 2 GB of RAM and a Raspberry Pi NoIR V2 camera were used. Although the initial plan was to utilize the camera's infrared mode, the RGB mode was ultimately employed due to the considerations discussed in the literature review (Chapter 2). This was achieved using the IMX219 calibration already available on the Raspberry Pi 4B.

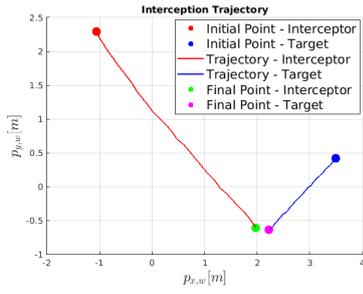
During the experiments, a significant challenge arose with the setup, as the Raspberry Pi camera and its wires had to be mounted on top of the Bebop2. This placement occasionally interfered with the Bebop2's GPS signals, causing interruptions in GPS-based navigation.



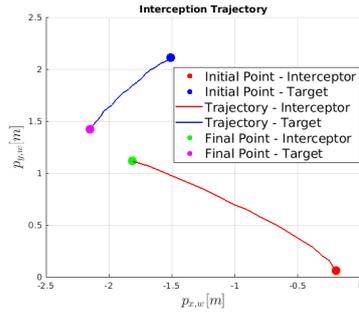
Figure 4.2: Autonomous flight of the AR Wing Pro

4.2. Cyberzoo Experiments

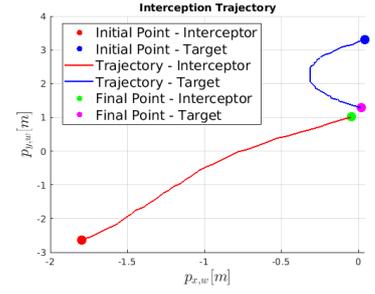
Cyberzoo experiments, as detailed in the scientific article (Chapter II), were conducted to test the real-life interception capabilities of the Proportional Navigation guidance. These tests, as can be seen from the figures below, the interception velocity was around $1 - 2m/s$. Even though this was not in the velocity range for the real experiment, it validated the PN algorithm in low speed regimes. It should be noted that, different from the planned outdoor experiment, the navigation used Optitrack data compared to the GPS.



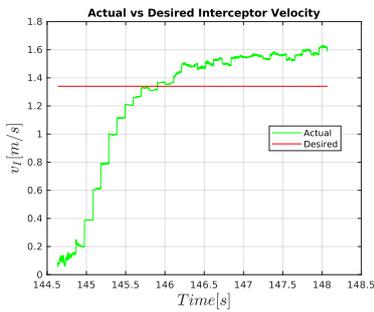
Position of the interceptor and target in 2D where target is moving in a straight line.



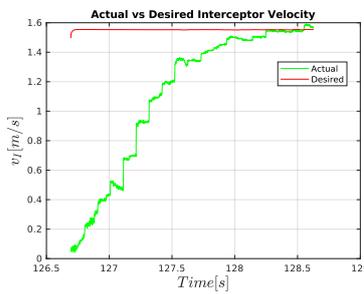
Position of the interceptor and target in 2D where target is moving in a circle and interceptor starts from the center of the circle.



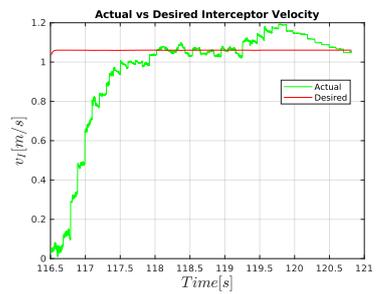
Position of the interceptor and target in 2D where target is moving in a circle and interceptor starts outside the circle.



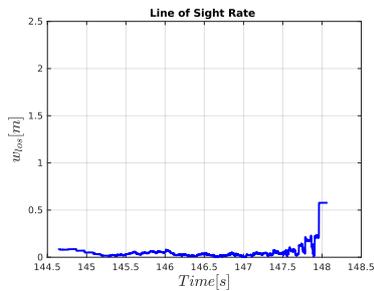
Magnitude of the reference and actual velocity of the interceptor where the target is moving in a straight line.



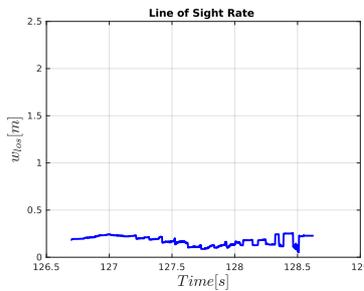
Magnitude of the reference and actual velocity of the interceptor where target is moving in a circle and interceptor starts from the center of the circle.



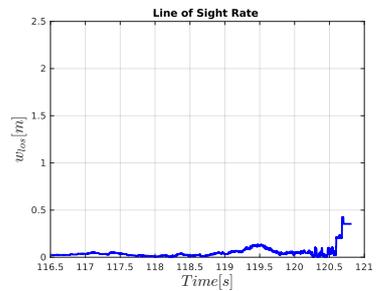
Magnitude of the reference and actual velocity of the interceptor where target is moving in a circle and interceptor starts outside the circle.



The line of sight rate where the target is moving in a straight line.



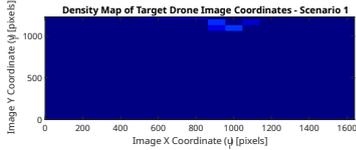
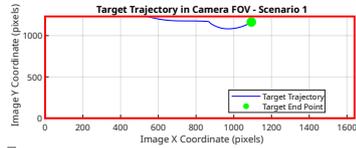
The line of sight range where target is moving in a circle and interceptor starts from the center of the circle.



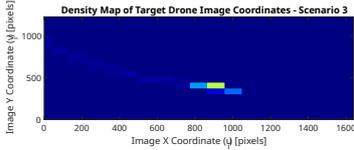
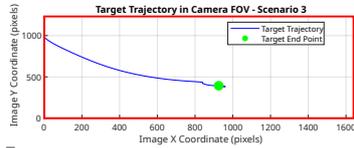
The line of sight range where target is moving in a circle and interceptor starts outside the circle.

4.3. Visibility

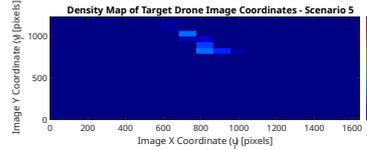
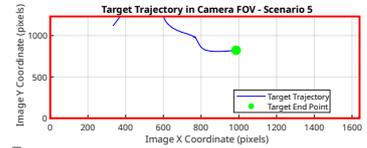
A simplified Matlab block diagram for the simulation for determining the visibility of the target drone in the field of view can be seen in Figure 4.3. Also additional visibility plots can be seen below. It should be noted that the yaw controller is working to point the camera towards the target. As the C_b increases the target drone is lower in the frame. This is due to the target drone being much further in the $x - y$ plane and when the interceptor pitches and rolls towards the target, the image appears on the lower half of the frame while when the target drone is much closer in the $x - y$ plane the target drone appears on the top half.



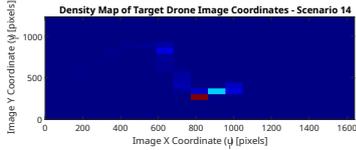
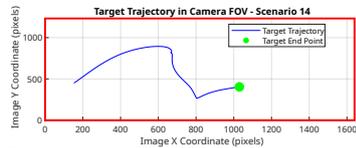
$C_b = 0.11$



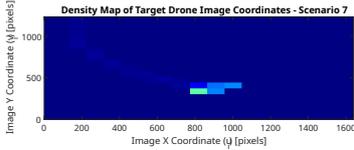
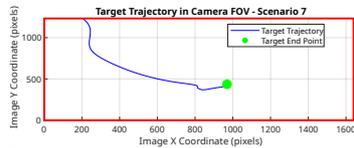
$C_b = 0.29$



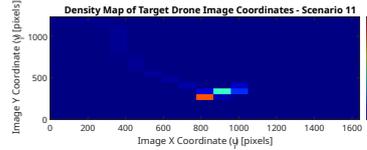
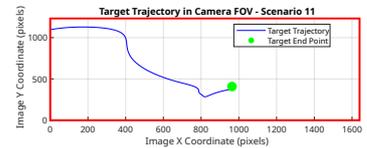
$C_b = 0.35$



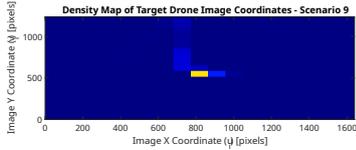
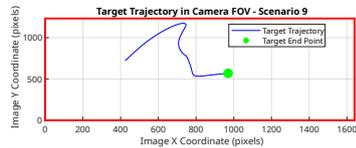
$C_b = 0.56$



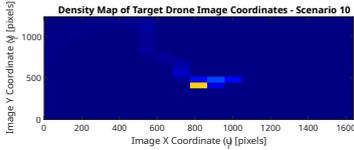
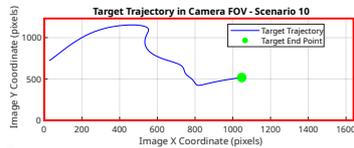
$C_b = 0.70$



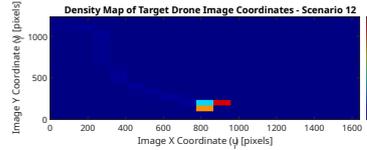
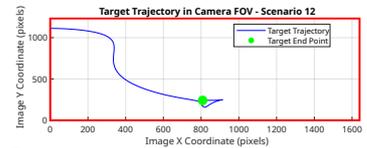
$C_b = 0.82$



$C_b = 0.94$



$C_b = 1.08$



$C_b = 1.28$

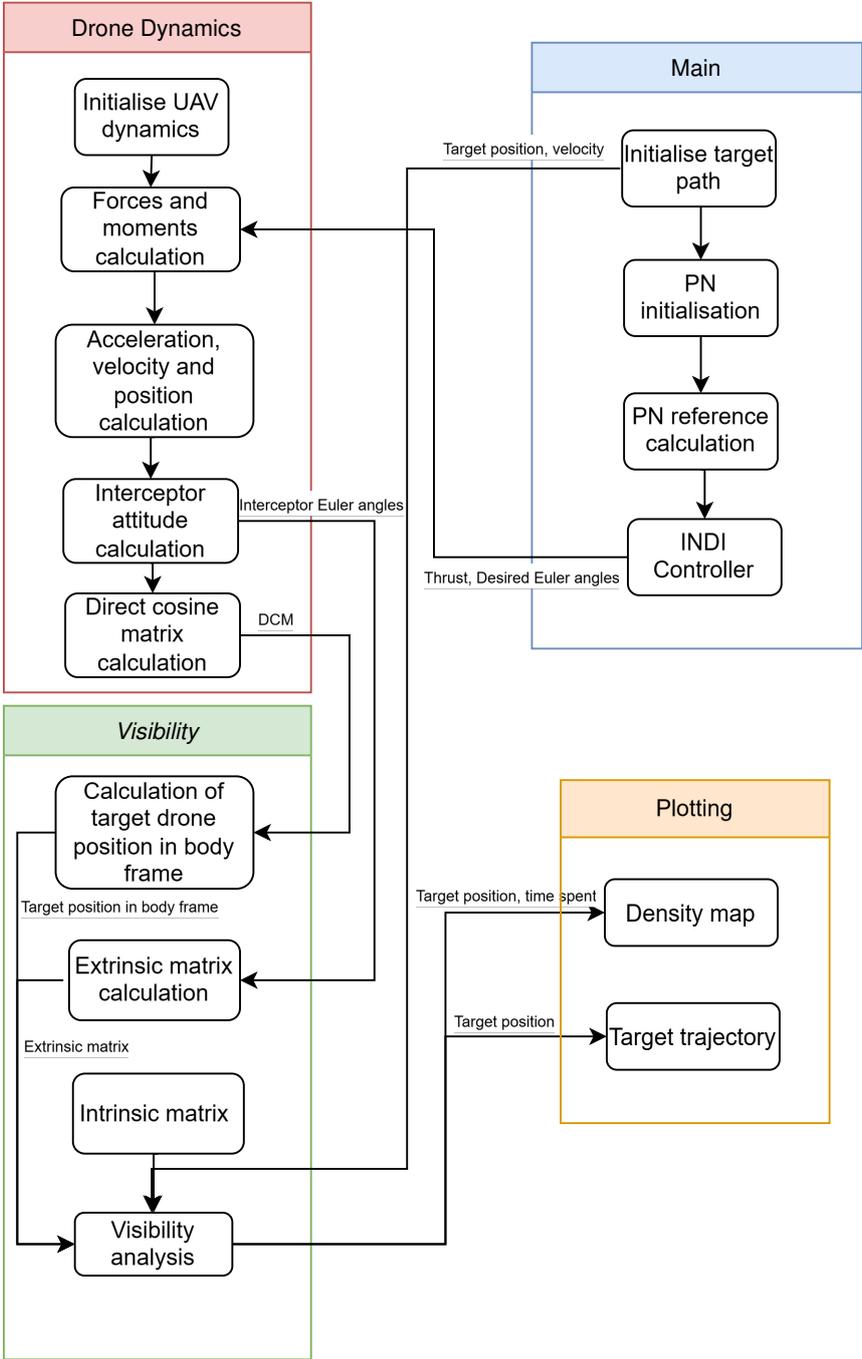


Figure 4.3: Caption describing the figure.

Part IV

Closure

Conclusions & Recommendations

5.1. Conclusions

The thesis demonstrates the effectiveness of a comprehensive framework for autonomous air-to-air interception of loitering kamikaze drones. Detection algorithms were evaluated, with YOLOv7 achieving superior accuracy (75% confidence), while the blob detector provided faster processing at 8 FPS, making it suitable for onboard execution. The tracking system, employing a CSRT tracker with preprocessing and Kalman filtering, achieved 10 FPS, ensuring robust performance against missed detections. Guidance tests showed consistent interception success, with an average error of 1.0 m for circling targets and 0.4 m for straight-line targets. A visibility study revealed the target remained within the camera's field of view for 83% of the interception duration for initial distances up to 150 m.

Here it is also explored whether the posed research questions were answered properly. The research questions posed in ?? are repeated below for convenience.

Research Question Main

How to design a real time algorithm to detect, track and intercept loitering munition onboard a quadcopter using RGB camera and target distance information?

Research Question 1

How to implement the most efficient computer vision algorithm in terms of performance and speed for air to air detection of small drones with an RGB camera?

Research Question 2

How to extract position, velocity and acceleration information of a target drone using infrared camera and assuming a known distance to target drone?

Research Question 3

What is an optimal interception guidance law for intercepting UAS targets with the use of another UAS?

The research presented in this thesis addresses the challenge of designing a real-time algorithm for detecting, tracking, and intercepting loitering munitions onboard a quadcopter using an RGB camera and target distance information. The proposed methodology encompasses a detailed comparison of state-of-the-art methods for detecting and tracking non-cooperative UAVs, evaluating their feasibility for real-time application. Additionally, the study presents interception guidance laws, testing their efficacy through simulations and real-world experiments involving a quadcopter interceptor targeting simulated

UAVs. This integrated approach bridges theoretical advancements and practical applications, contributing significantly to the domain of autonomous air-to-air interception.

Several computer vision algorithms were investigated, focusing on their speed and accuracy trade-offs. CNN-based methods, such as YOLOv7, demonstrated superior detection accuracy but faced limitations in computational speed when deployed on edge devices like the Raspberry Pi. Conventional methods, including blob detection, offered faster processing but at the cost of reduced detection accuracy. The study highlights that with advanced hardware, such as GPUs, CNN-based methods become the preferred choice due to their scalability and precision. This insight underlines the importance of hardware optimization in real-time aerial interception systems.

A robust method for extracting position and velocity data of the target drone was also developed using a linear Kalman filter. This approach provided a systematic framework for sensor fusion and target state estimation. However, due to the unavailability of real-world target drone data with accurately known position and velocity, the validation of this method remained a limitation of the study. Future work should focus on generating datasets that facilitate the evaluation of such algorithms under realistic conditions.

The thesis further delves into the comparison of interception guidance laws, identifying Proportional Navigation (PN) as an optimal strategy under specific assumptions and constraints. PN was validated through both literature and experimental results, showcasing its effectiveness in ensuring successful engagements while maintaining simplicity and computational efficiency. These findings emphasize the suitability of PN for UAV interception scenarios, particularly when paired with modern tracking and detection systems.

5.2. Recommendations

The research highlighted improvements on the already existing air-to-air interception methods. Nevertheless, still more improvements are in place. Several recommendations after this study are as follows:

- Several videos of the target drone from the interceptor drone should be made with known position and velocity data of the target drone at proper time stamps. This will enable the validation of the detection and tracking algorithm.
- A better hardware for image processing should be used such as a GPU for speeding up the CNN inference time. This will enable a more scalable detection algorithm to be employed
- The interception guidance should be tested in an outdoor setting with both drones present in real life which will enable the validation of the interception guidance at proper distance and velocity regimes.

References

- [1] Samira Shackle. *The mystery of the Gatwick drone*. <https://www.theguardian.com/uk-news/2020/dec/01/the-mystery-of-the-gatwick-drone>.
- [2] Joby Warrick. *Use of weaponized drones by ISIS spurs terrorism fears*. https://www.washingtonpost.com/world/national-security/use-of-weaponized-drones-by-isis-spurs-terrorism-fears/2017/02/21/9d83d51e-f382-11e6-8d72-263470bf0401_story.html.
- [3] Al Jazeera. *Venezuela's Nicolas Maduro survives drone 'attack'*. <https://aljazeera.com/news/2018/8/5/venezuelas-nicolas-maduro-survives-drone-attack>.
- [4] BBC. *How are 'kamikaze' drones being used by Russia and Ukraine?* <https://www.bbc.com/news/world-62225830>.
- [5] Dan Sabbagh. *Houthis call west's bluff with renewed Red Sea drone assault*. <https://www.theguardian.com/world/2024/jan/10/houthis-call-wests-bluff-with-renewed-red-sea-drone-assault>.
- [6] Endre Papp Marina Moreira et al. "Interception of Non-Cooperative UAVs". In: *IEEE International Symposium on Safety, Security, and Rescue Robotics* (2019).
- [7] Juhwan Noh et al. "Tractor Beam: Safe-Hijacking of Consumer Drones with Adaptive GPS Spoofing". In: *ACM Trans. Priv. Secur.* 22.2 (Apr. 2019). DOI: 10.1145/3309735.
- [8] Robin Radar. *10 Types of Counter Drone Technology To Detect And Stop Drones Today*. <https://www.robinradar.com/press/blog/10-counter-drone-technologies-to-detect-and-stop-drones-today>.
- [9] Ye Zheng et al. "Air-to-Air Visual Detection of Micro-UAVs: An Experimental Evaluation of Deep Learning". In: *IEEE Robotics and Automation Letters* 6.2 (2021), pp. 1020–1027. DOI: 10.1109/LRA.2021.3056059.
- [10] Yin Zhang Hanqing Guo Ye Zheng. "Global-Local MAV Detection under Challenging Conditions based on Appearance and Motion". In: *Cornell University* (Dec. 2023).
- [11] Xia Y Zhao X. "YOLO-ViT-Based Method for Unmanned Aerial Vehicle Infrared Vehicle Target Detection". In: *Remote Sensing* 15 (2023).
- [12] Fang Chen et al. "Local Patch Network With Global Attention for Infrared Small Target Detection". In: *IEEE Transactions on Aerospace and Electronic Systems* 58.5 (2022), pp. 3979–3991. DOI: 10.1109/TAES.2022.3159308.
- [13] Jing Li et al. "Fast and Robust UAV to UAV Detection and Tracking From Video". In: *IEEE Transactions on Emerging Topics in Computing* 10.3 (2022), pp. 1519–1531. DOI: 10.1109/TETC.2021.3104555.
- [14] Neil Palumbo et al. "Basic Principles of Homing Guidance". In: *Johns Hopkins Apl Technical Digest* 29 (Jan. 2010), pp. 25–41.
- [15] Namhoon Cho et al. "Optimality of augmented ideal proportional navigation for maneuvering target interception". In: *IEEE Transactions on Aerospace and Electronic Systems* 52.2 (2016), pp. 948–954. DOI: 10.1109/TAES.2015.140432.
- [16] C. Tournes et al. "Augmented proportional navigation guidance using subspace-stabilization". In: *Proceedings of the Thirty-Fourth Southeastern Symposium on System Theory (Cat. No.02EX540)* (2002), pp. 471–475. DOI: 10.1109/SSST.2002.1027091.
- [17] U.S. Shukla et al. "The proportional navigation dilemma-pure or true?" In: *IEEE Transactions on Aerospace and Electronic Systems* 26.2 (1990), pp. 382–392. DOI: 10.1109/7.53445.

- [18] Markus Hehn et al. "Quadrocopter Trajectory Generation and Control". In: *IFAC Proceedings Volumes* 44.1 (2011), pp. 1485–1491. DOI: 10.1109/SSST.2002.1027091.
- [19] Markus Hehn et al. "Real-time trajectory generation for interception maneuvers with quadrocopters". In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2012, pp. 4979–4984. DOI: 10.1109/IR0S.2012.6386093.
- [20] Daniel Mellinger et al. "Trajectory Generation and Control for Precise Aggressive Maneuvers with Quadrotors". In: *STAR 79* (2012), pp. 361–373.
- [21] Cetin A.T. and Koyuncu E. "Model Predictive Control-Based Guidance with Impact Angle Constraints for Visual Quadrotor Interception". In: *9th International Conference on Control, Decision and Information Technologies (2023)*, pp. 2544–2549. DOI: 10.1109/codit58514.2023.10284484.
- [22] Hongbo Sun et al. "Improving the Doppler Resolution of Ground-Based Surveillance Radar for Drone Detection". In: *IEEE Transactions on Aerospace and Electronic Systems* 55.6 (2019), pp. 3667–3673. DOI: 10.1109/TAES.2019.2895585.
- [23] Bilal Taha et al. "Machine Learning-Based Drone Detection and Classification: State-of-the-Art in Research". In: *IEEE Access* 7 (2019), pp. 138669–138682. DOI: 10.1109/ACCESS.2019.2942944.
- [24] Przemysław Flak. "Drone Detection Sensor With Continuous 2.4 GHz ISM Band Coverage Based on Cost-Effective SDR Platform". In: *IEEE Access* 9 (2021), pp. 114574–114586. DOI: 10.1109/ACCESS.2021.3104738.
- [25] Angelo Coluccia et al. "Drone vs. Bird Detection: Deep Learning Algorithms and Results from a Grand Challenge". In: *MDPI* 21.8 (2021). DOI: 10.3390/s21082824.
- [26] Jie Zhao et al. "Vision-Based Anti-UAV Detection and Tracking". In: *IEEE Transactions on Intelligent Transportation Systems* 23.12 (2022), pp. 25323–25334. DOI: 10.1109/TITS.2022.3177627.
- [27] Wijnker D et al. "Hear-and-avoid for unmanned air vehicles using convolutional neural networks". In: *International Journal of Micro Air Vehicles* 13 (2021). DOI: 10.1177/1756829321992137.
- [28] Jens Reimann et al. "Highly Accurate Radar Cross-Section and Transfer Function Measurement of a Digital Calibration Transponder without Known Reference—Part II: Uncertainty Estimation and Validation". In: *Remote Sensing* 15.8 (2023). DOI: 10.3390/rs15082148.
- [29] Hong Liu et al. "Single-Snapshot Direction of Arrival Estimation for Vehicle-Mounted Millimeter-Wave Radar via Fast Deterministic Maximum Likelihood Algorithm". In: *World Electric Vehicle Journal* 15.7 (2024). DOI: 10.3390/wevj15070321.
- [30] Lindsey Wiley Joshua Follansbee et al. "Drone detection performance in the reflective bands: visible, near infrared, short wave infrared, and extended short wave infrared". In: *Optical Engineering* 61 (2022), pp. 95–106.
- [31] Ming Zeng et al. "The design of Top-Hat morphological filter and application to infrared target detection". In: *Infrared Physics Technology* 48 (Apr. 2006), pp. 67–76. DOI: 10.1016/j.infrared.2005.04.006.
- [32] Meng Hwa Er Suyog D. Deshpande. "Max-mean and max-median filters for detection of small targets". In: *Signal and Data Processing of Small Targets* (Oct. 1999). DOI: <https://doi-org.tudelft.idm.oclc.org/10.1117/12.364049>.
- [33] Saeid Aghaziyarati et al. "Small infrared target detection using absolute average difference weighted by cumulative directional derivatives". In: *Infrared Physics Technology* 101 (June 2019). DOI: 10.1016/j.infrared.2019.06.003.
- [34] Chenqiang Gao et al. "Generalised-structure-tensor-based infrared small target detection". In: *Electronics Letters* 44 (Dec. 2008), pp. 1349–1351. DOI: 10.1049/el:20081781.
- [35] Chenqiang Gao et al. "Infrared Small-dim Target Detection Based on Markov Random Field Guided Noise Modeling". In: *Pattern Recognition* 76 (Nov. 2017). DOI: 10.1016/j.patcog.2017.11.016.

- [36] Jinhui Han et al. "A Local Contrast Method for Infrared Small-Target Detection Utilizing a Tri-Layer Window". In: *IEEE Geoscience and Remote Sensing Letters* (Dec. 2019), pp. 1–5. DOI: 10.1109/LGRS.2019.2954578.
- [37] Sara Minaeian et al. "Effective and Efficient Detection of Moving Targets From a UAV's Camera". In: *IEEE Transactions on Intelligent Transportation Systems* 19.2 (2018), pp. 497–506. DOI: 10.1109/TITS.2017.2782790.
- [38] Daniel Tan Wei Xun et al. "Drone detection using YOLOv3 with transfer learning on NVIDIA Jetson TX2". In: *2021 Second International Symposium on Instrumentation, Control, Artificial Intelligence, and Robotics (ICA-SYMP)*. 2021, pp. 1–6. DOI: 10.1109/ICA-SYMP50206.2021.9358449.
- [39] S. Antony Snell et al. "Nonlinear inversion flight control for a supermaneuverable aircraft". In: *Journal of Guidance, Control, and Dynamics* (1992), pp. 976–984. DOI: 10.2514/3.20932.
- [40] S. Sieberling et al. "Robust Flight Control Using Incremental Nonlinear Dynamic Inversion and Angular Acceleration Prediction". In: *Pediatric Critical Care Medicine - PEDIATR CRIT CARE MED* 33 (Nov. 2010). DOI: 10.2514/1.49978.
- [41] R. R. da Costa et al. "Reentry Flight Controller Design Using Nonlinear Dynamic Inversion". In: *Journal of Spacecraft and Rockets* 40 (2003), pp. 64–71. DOI: 10.2514/2.3916.
- [42] Ewoud Smeur et al. "Incremental Control and Guidance of Hybrid Aircraft Applied to a Tailsitter Unmanned Air Vehicle". In: *Journal of Guidance, Control, and Dynamics* (Feb. 2018). DOI: 10.2514/1.G004520.