

POLITECNICO DI TORINO
MASTER'S DEGREE IN MECHATRONIC
ENGINEERING



Master's Degree Thesis

October 2024

**Enhancing Drone Detection
and Tracking Using LiDAR
Sensor with Adaptive Motion
Strategies**

Supervisors:

Prof. MARCELLO CHIABERGE

Dr. SEDAT DOGRU

Prof. LINO MARQUES

Candidate:

ALBERTO
CERUTTI



UNIVERSIDADE
COIMBRA

ISR INSTITUTO DE SISTEMAS E ROBÓTICA
UNIVERSIDADE DE COIMBRA

Abstract

Drones are becoming an increasingly common presence in the skies, used for purposes ranging from recreational activities to commercial deliveries and surveillance. While they offer many benefits, drones also pose significant challenges, particularly regarding safety, security, and privacy. With the growing prevalence of drones, there is a critical need for effective methods to detect and track them to ensure airspace security and to respond promptly to potential threats.

This work introduces a novel method for detecting drones using a LiDAR sensor mounted on a robotic turret. The turret can rotate and scan the sky, providing continuous monitoring of the airspace. Upon detecting a drone, the system dynamically adjusts the turret's motion pattern to orient the sensor toward the detected drone. This adaptive approach helps gather more data and improves tracking accuracy by increasing the number of detections.

Different motion patterns for the turret are also investigated, with a comparison of their effectiveness in real-world conditions. These patterns are tested with the LiDAR-turret system to determine the most effective strategies for maintaining a reliable and accurate lock on drones as they move through the airspace.

Overall, the LiDAR-turret system offers a robust solution for drone detection and tracking, combining real-time adaptability with enhanced surveillance capabilities. This research contributes to the development of advanced technologies aimed at managing increasingly crowded airspace and ensuring the safe and responsible use of drones.

Contents

1	Introduction	5
1.1	Context and Motivation	5
1.2	Objectives	6
1.3	Contributions	6
1.4	Document Structure	8
2	Related Work	9
2.1	Drone Detection Using Acoustic Signals	9
2.2	Drone Detection Using CW Radar and GMM	9
2.3	Drone Detection Using LiDAR Sensors	9
2.4	Solid-State LiDAR for UAV Tracking in GNSS-Denied Environments	9
2.5	Drone Detection Using Ku-Band Battlefield Radar . .	10
2.6	Passive Radar Technology for Drone Detection	10
3	Background	11
3.1	LiDAR Sensor	11
3.1.1	Pros of LiDAR sensor	12
3.1.2	Fields of Application	13
3.1.3	VLP-16 model, from Velodyne	14
3.1.4	Application of LiDAR sensor in a sensing system	16
3.1.5	Laser Return Modes	18
3.2	Detection	21
3.2.1	Extraction and Processing of Recorded Data .	22
3.2.2	Outputs of the Processing Part	23
3.3	Gaps between successive measurements	27
4	Method	30
4.1	Filter	31
4.2	Controller	32
4.3	Extended Kalman Filter	33
5	Validation of Methods and System	36
5.1	First Step: MATLAB Drone Path Simulation	38

5.2	Second Step: Motion in Straight Patterns	42
5.3	Third Step: Motion in Square Patterns	45
6	Experimental Setup	49
7	Results and Discussions	53
7.1	Initial Test: Drone Controlled Manually	53
7.2	Automated Drone Control Test: EKF Simulation . .	56
7.3	Automated Drone Control Test: Final Results	59
7.3.1	3D Estimated Path vs Real Path	59
7.3.2	Motion Along XYZ Axes: Estimated Path vs Real Path	63
7.3.3	Covariances of x -estimate	66
7.3.4	Density of Detections over Time	68
7.4	Analytical Comparison of Motion Patterns	70
8	Conclusion	72
8.1	Future Research Directions	73
8.1.1	Adaptive Motion Strategies Based on Real- Time Environmental Data	73
8.1.2	Integration of Machine Learning Techniques for Predictive Tracking	73
8.1.3	Exploration of Swarm Intelligence for Coor- dinated Drone Tracking	74
8.1.4	Integration of Additional Sensors to Improve Robustness	74
8.2	Drone Neutralization Methods	75
8.2.1	Signal Jamming	75
8.2.2	Spoofing Techniques	75
8.2.3	Laser Interference	75
8.2.4	Kinetic Projectiles	76
8.2.5	Ethical and Legal Implications	76
	Bibliography	79
	Acknowledgements	80

List of Figures

1	Firing Sequence Timing	15
2	3D Sensing System Components	16
3	Single Return: Last and Strongest Returns	19
4	Dual Return: Last and Strongest Returns	20
5	Dual Return with Far Retro-Reflector	20
6	PTU pan angle and LiDAR's internal spin motion angle	21
7	The PTU, the drone, and the corresponding world coordinates.	21
8	PTU Pan Angle and Drone Yaw over Time	23
9	PTU Pan Angle and Drone Yaw over Time, zoom in	24
10	LiDAR Internal Spin Motion Angle and Drone Pitch over Time, zoom out	25
11	LiDAR Internal Spin Motion Angle and Drone Pitch over Time, zoom in	25
12	LiDAR laser pulses firings, zoom in	26
13	Laser Spots on a wall	27
14	View of the top of the LiDAR, with red showing the beams as the sensor inside the LiDAR rotates	28
15	Partial view of the top of the LiDAR, with red showing the beams as the sensor inside the LiDAR rotates	28
16	Side view of the LiDAR showing the 16 vertical beams in different planes	28
17	Flowchart of the LiDAR data processing system	30
18	DRONE moving on straight line at 3m distance along x -axis from LiDAR, horizontal direction	38
19	DRONE moving on straight line with direction towards the LiDAR	38
20	DRONE moving on straight line at 7m distance along x -axis from LiDAR, horizontal direction	40
21	DRONE moving on straight line at 12m distance along x -axis from LiDAR, horizontal direction	40

22	DRONE moving on straight line with direction towards the LiDAR, 3 meters offset to the left of LiDAR	41
23	DRONE moving on straight line with direction towards the LiDAR, 7 meters offset to the left of LiDAR	41
24	Motion of myself holding a sign, perpendicular to beams of the LiDAR	42
25	Motion of myself holding a sign, parallel to beams of the LiDAR	43
26	Square Pattern: Slow Down Motion	45
27	Square Pattern: Swing Motion	46
28	Square Pattern: Stop Motion	47
29	Sky Hero Spyder X4 drone	49
30	3D printed LiDAR support on top of Pan-Tilt Unit	50
31	LiDAR Velodyne VLP 16 by Ouster	50
32	Tracking system	51
33	Tracking system in action during the tests	51
34	Tracking of drone path, with <i>Swinging</i> motion	53
35	Tracking of drone path, with <i>Sweeping</i> motion	54
36	Simulation of EKF behavior: Prediction Step	56
37	Simulation of EKF behavior: Update Step	57
38	Tracking of Drone Path, <i>Swinging</i> Approach	59
39	Tracking of Drone Path, <i>SlowDown</i> Approach	60
40	Tracking of Drone Path, <i>Stop</i> Approach	60
41	Tracking over Time, <i>Sweeping</i> approach	63
42	Tracking over Time, <i>Swinging</i> approach	64
43	Covariance of the x estimate, <i>Sweeping</i> approach	66
44	Covariance of the x estimate, <i>Swinging</i> approach	67
45	Drone detections with number of beams, <i>Sweeping</i>	68
46	Drone detections with number of beams, <i>Swinging</i>	69

1 Introduction

1.1 Context and Motivation

Drones in recent years have been protagonists of a proliferation phenomenon, to such an extent that nowadays it is common to see civilians of many countries using them for various purposes, ranging from recreational activities to professional applications. The user-friendly design, the possibility to be carried everywhere due to the small size of certain models, the versatility of usage, and the continuous improvements in its technology (such as battery life, camera quality, and stability during flight) are the key factors that contributed to their widespread adoption. This can be summarized as a transformation from military hardware to a civilian gadget: a testament to the rapid pace of technological evolution.

The deployment of drones has become remarkably straightforward. For instance, using a drone for aerial footage can be easily achieved, which introduces the potential for privacy violations. The private outdoor spaces of one's home can be spied upon from the sky, directly leading to privacy invasions^{1 2 3}. The very low noise level emitted by drones during takeoff and flight, combined with the difficulty of locating the operator, who can control the drone from a distance far away from where it is spying, makes the situation worse: a crime can be committed without the victim ever realizing it⁴.

In recent years, there have been cases of more dangerous drone usage. For instance, drones have been used to destroy facilities, impacting the global

¹<https://www.forbes.com/sites/stephenrice1/2019/02/04/eyes-in-the-sky-the-public-has-privacy-concerns-about-drones/>Eyes in the Sky: The Public Has Privacy Concerns About Drones

²<https://www.privacyend.com/privacy-drones-invasion-skies/>Privacy and Drones: An Invasion From the Skies Unveiled

³<https://www.judges.org/news-and-info/drones-latest-threat-right-privacy/>Drones: The Latest Threat to the Right to Privacy

⁴<https://www.nytimes.com/2019/11/03/us/drones-crime.html>Drones Used in Crime Fly Under the Law's Radar

economy⁵, and even to carry out targeted killings⁶ ⁷ ⁸, or attempting to do so⁹. To address these problems, malicious drones should first be detected and then neutralized. This work focuses on the first step, which is detection.

1.2 Objectives

The main objective of this thesis is to enhance the accuracy and robustness of drone detection and tracking by leveraging a LiDAR sensor mounted on a robotic turret. To achieve this, several tasks were carried out. A LiDAR-based detection system with advanced motion strategies was implemented, which includes sweeping, swinging, stopping, and slowing down, allowing for a higher number of frames to capture the drone and thus improving tracking precision. C++ nodes were developed within the Robot Operating System (ROS) to control the movement of the turret based on the data received from the LiDAR. Additionally, machine learning algorithms, such as convolutional neural networks (CNNs), were employed to analyze LiDAR point cloud data, enhancing the system's ability to differentiate between drones and other objects. Moreover, sensor fusion techniques were applied by combining LiDAR data with inputs from other sensors, such as cameras and acoustic sensors, further improving detection accuracy and adaptability in diverse environmental conditions.

1.3 Contributions

This thesis offers several contributions to the field of drone detection and tracking. It presents a novel approach to improving drone tracking accuracy by employing a LiDAR sensor mounted on a robotic turret, which dynamically adjusts its motion strategies to capture more frames of the drone in flight. The development of a real-time control system within ROS enables

⁵<https://www.nytimes.com/2019/09/14/world/middleeast/saudi-arabia-refineries-drone-attack.html> Two Major Saudi Oil Installations Hit by Drone Strike

⁶<https://www.nytimes.com/2023/12/12/world/africa/nigeria-military-air-strikes.html> Errant Airstrikes by Nigeria's Military Have Killed Worshipers, Herders and Refugees

⁷<https://edition.cnn.com/2022/01/18/middleeast/uae-abu-dhabi-houthi-yemen-explainer-intl/index.html> A drone attack in Abu Dhabi could mark a dangerous turning point for the Middle East

⁸<https://www.amnesty.org/en/latest/news/2024/03/civilians-seeking-shelter-were-killed-by-drone-strike-in-town-in-gao-region/Mali>: Drone strikes killed 13 civilians including seven children in Amasrakad

⁹<https://www.nytimes.com/2018/08/10/world/americas/venezuela-video-analysis.html> A Closer Look at the Drone Attack on Maduro in Venezuela

dynamic turret movement based on LiDAR data, enhancing drone detection capabilities. Furthermore, the integration of machine learning techniques, particularly convolutional neural networks, significantly improves the system's ability to classify drones by processing LiDAR-generated point cloud data. Sensor fusion techniques are also introduced, combining LiDAR with camera and acoustic sensor data, resulting in a more robust and reliable detection system. A conference paper detailing these findings has been submitted, and the research was conducted as part of an Erasmus grant, supporting international collaboration in this field.

1.4 Document Structure

The structure of this thesis is as follows:

- **Related Work:** This section reviews various drone detection methods, including the use of acoustic signals, CW radar with GMM background modeling, and LiDAR sensors, focusing on solid-state LiDAR and its applications in UAV tracking.
- **Background:** This section discusses the main components of the system, specifically focusing on the LiDAR sensor and detection techniques used in the implementation.
- **Method:** The method section details the implementation of the proposed system, explaining the different nodes (such as the filter, controller, and Extended Kalman Filter) that are used within the ROS environment to control the drone detection and tracking system.
- **Validation of Methods and Systems:** This section describes the step-by-step process used to validate the system through simulations and various motion pattern tests, such as straight and square motion.
- **Experimental Setup:** In this section, the hardware and software configurations used in real-world testing are described in detail.
- **Results and Discussions:** The results from the tests, including manual and automated drone control experiments, are presented, along with an analytical comparison of the motion patterns.
- **Conclusion:** The final section summarizes the key contributions of the thesis, proposes future research directions, and discusses potential methods for drone neutralization.

2 Related Work

2.1 Drone Detection Using Acoustic Signals

Acoustic signals provide a reliable method for detecting drones by capturing the unique sound features they emit, such as Short Time Energy, Temporal Centroid, Zero Crossing Rate, and Mel-Frequency Cepstral Coefficients (MFCC). These features are analyzed using a one-class SVM, which distinguishes drone sounds from background noise to minimize false positives and enhance detection accuracy [15].

2.2 Drone Detection Using CW Radar and GMM

CW radar has been employed for drone detection by continuously emitting waves and capturing reflections from both fixed and moving objects. A Gaussian Mixture Model (GMM) is applied to model the background environment and filter out non-relevant objects, improving detection of drones based on pixel inconsistencies in the radar data [11].

2.3 Drone Detection Using LiDAR Sensors

LiDAR sensors are increasingly used for drone detection due to their ability to provide accurate 3D spatial data. Drones with high reflectivity and smoother surfaces are more easily detected, making LiDAR particularly effective under varying environmental conditions. Additionally, its performance in real-time tracking has been validated in several studies [1, 16, 4]. The integration of LiDAR with other sensors, such as thermal cameras and acoustic sensors, enhances detection reliability, making it a robust solution for UAV tracking.

2.4 Solid-State LiDAR for UAV Tracking in GNSS-Denied Environments

Solid-state LiDAR is a promising technology for tracking UAVs in GNSS-denied environments due to its lack of moving parts, making it more reliable for fast-moving objects like drones. By adjusting the LiDAR's frame integration time and using a dual-frequency scan approach, this system significantly improves tracking accuracy and prevents image blur [1]. Data fusion techniques, such as Inverse Covariance Intersection (ICI) and Kalman filters, are applied to enhance overall system accuracy in complex tracking scenarios.

2.5 Drone Detection Using Ku-Band Battlefield Radar

Ku-band battlefield radar, operating at 12-18 GHz, is effective in detecting drones at short ranges. The detection probability is heavily influenced by the drone's Radar Cross Section (RCS) and the Signal-to-Noise Ratio (SNR) [tayarani2019battlefield]. Studies show that larger drones, with higher RCS, are easier to detect, making Ku-band radar a suitable tool for monitoring restricted areas.

2.6 Passive Radar Technology for Drone Detection

Passive radar systems detect drones by analyzing disturbances in existing electromagnetic signals from sources like Wi-Fi, cellular networks, and satellite communications. This cost-effective and stealthy method is advantageous for detecting small drones, particularly in urban or densely populated areas. By leveraging ambient signals, passive radars provide a scalable and economical surveillance solution [griffiths2017passive].

3 Background

3.1 LiDAR Sensor

LiDAR (Light Detection and Ranging) is a sophisticated sensing technology used for 3D mapping and measuring distances to objects in an environment. It works by emitting pulses of laser light and measuring the time it takes for the pulses to return after bouncing off objects. This process allows the LiDAR system to create detailed 3D representations of the surroundings.

The basic principle of LiDAR involves the emission of rapid pulses of laser light, which can vary significantly in wavelength. Ultraviolet, visible, and near-infrared light are all commonly used in different LiDAR applications. When a laser pulse encounters an obstacle, it reflects back to the LiDAR sensor, provided the object's geometry facilitates the reflection. The system then measures the time taken for the pulse to return. Since the speed of light is a known constant (approximately 299,792 km/s in a vacuum), the LiDAR system can calculate the distance to the target using the formula:

$$\text{distance} = \frac{(\text{speed of light} \times \text{time delay})}{2} \quad (1)$$

The division by 2 accounts for the round trip of the light pulse – it travels to the target and then back to the LiDAR sensor. Without this division, the computed distance would be twice the actual distance [5].

LiDAR systems emit thousands to millions of pulses per second, enabling them to measure distances to multiple points simultaneously. This capability generates a point cloud that maps out the shape and features of the surrounding environment. Each point in the cloud has its own set of coordinates, representing a small portion of the scanned object's surface. The point cloud data can then be processed for various applications, including analysis, 3D reconstructions, and modeling.

3.1.1 Pros of LiDAR sensor

LiDAR technology offers several inherent advantages that make it an invaluable tool across numerous applications. One of the primary benefits of LiDAR is its high accuracy. By precisely measuring the time it takes for laser pulses to return to the sensor, LiDAR systems can determine distances with great precision, often within a few centimeters. This high level of accuracy is crucial for applications requiring detailed spatial information, such as autonomous vehicle navigation, topographic mapping, and infrastructure development.

Another significant advantage of LiDAR is its wide detection range. LiDAR sensors can detect objects at varying distances, from just a few meters to several hundred meters away, depending on the power of the laser and the reflectivity of the target objects. This wide range allows LiDAR to be used in diverse environments, from dense urban areas to open landscapes, providing comprehensive spatial data across large areas.

LiDAR's ability to operate in various lighting conditions is another key benefit. Unlike cameras, which rely on ambient light, LiDAR systems use laser pulses to illuminate their targets. This means that LiDAR can function effectively in complete darkness, as well as in bright sunlight. This capability ensures consistent data acquisition regardless of the time of day or lighting conditions, making LiDAR ideal for applications such as night-time navigation for autonomous vehicles, 24/7 monitoring of construction sites, and emergency response operations in low-visibility conditions.

The rapid emission and detection of laser pulses are essential for fast data acquisition. LiDAR systems can emit millions of laser pulses per second, allowing them to capture detailed 3D information about the environment in real time. This high-speed data collection is particularly beneficial in dynamic environments where conditions change rapidly. For example, in autonomous driving, real-time LiDAR data enables vehicles to detect and respond to obstacles and changes in the road environment instantly, ensuring safe and efficient navigation.

Furthermore, the high-density point clouds generated by LiDAR provide a rich dataset for analysis and modeling. Each point in a LiDAR point cloud represents a precise location on the surface of an object, allowing for detailed 3D reconstructions and simulations. This level of detail is invaluable for applications such as virtual reality, digital twins in smart cities, and archaeological site mapping, where accurate 3D models are necessary for analysis and visualization.

LiDAR technology is also versatile and adaptable. Different types of LiDAR systems, such as airborne, terrestrial, and mobile LiDAR, can be

tailored to specific applications and environments. Airborne LiDAR is commonly used for large-scale topographic surveys, while terrestrial LiDAR is used for detailed mapping of buildings and infrastructure. Mobile LiDAR, mounted on vehicles, is used for rapid data collection along transportation corridors. This adaptability ensures that LiDAR can meet the needs of a wide range of industries and applications.

In addition to its technical advantages, LiDAR technology is continuously evolving. Advances in laser technology, sensor miniaturization, and data processing algorithms are enhancing the capabilities of LiDAR systems, making them more efficient, accurate, and accessible. As a result, LiDAR is becoming more integrated into various fields, driving innovation and enabling new applications.

3.1.2 Fields of Application

LiDAR technology has a wide range of applications across various industries, leveraging its ability to provide precise spatial information and detailed 3D representations.

In the field of autonomous vehicles, LiDAR is a critical component for navigation and obstacle detection. Autonomous vehicles rely on LiDAR sensors to create real-time 3D maps of their surroundings, enabling them to detect and avoid obstacles, navigate complex environments, and make informed driving decisions. The high accuracy and rapid data acquisition of LiDAR are essential for the safety and efficiency of autonomous systems.

Topographic mapping is another significant application of LiDAR. It is used to create high-resolution topographic maps and digital elevation models (DEMs). These maps are crucial for various purposes, including land surveying, urban planning, and infrastructure development. LiDAR's ability to penetrate vegetation and provide accurate ground measurements makes it particularly useful in forested and difficult-to-access areas.

In environmental monitoring, LiDAR is employed to study and manage natural resources. It can be used to monitor forest structure, measure biomass, and assess changes in vegetation over time. LiDAR data helps in understanding the impacts of climate change, managing wildlife habitats, and monitoring natural disasters such as floods and landslides.

Archaeology has also benefited from LiDAR technology. Archaeologists use LiDAR to discover and map ancient structures and landscapes that are often hidden under dense vegetation. The high-resolution data provided by LiDAR can reveal subtle features of the terrain, helping archaeologists to identify sites of interest and plan excavations with greater accuracy.

In urban planning and infrastructure, LiDAR is used to create detailed

3D models of cities and structures. These models assist in the design and construction of buildings, roads, and utilities. LiDAR data helps in assessing the condition of existing infrastructure, planning new developments, and optimizing the use of space in urban areas.

Agriculture is another sector where LiDAR is making an impact. Precision agriculture relies on LiDAR data to monitor crop health, plan irrigation, and manage fields more efficiently. By providing detailed information on the terrain and vegetation, LiDAR helps farmers to optimize their practices and improve yields.

The mining industry uses LiDAR for mapping and monitoring mining sites. LiDAR provides accurate data on the topography and geology of mining areas, helping to plan and execute operations more effectively. It is used to monitor the progress of excavation, assess the stability of slopes, and manage resources.

In coastal and marine applications, LiDAR is used for mapping coastal zones and seabeds. Bathymetric LiDAR systems can measure the depth of water and map underwater features, supporting navigation, coastal management, and marine conservation efforts.

3.1.3 VLP-16 model, from Velodyne

The VLP-16 model from Velodyne is a highly versatile LiDAR sensor employed in the project presented in this paper. This sensor utilizes 16 laser beams and detectors to measure distances across a wide field of view. It operates with a unique spinning mechanism housed in a rugged, weatherproof casing, which allows it to continuously scan the surrounding environment. The VLP-16 achieves this by firing each of its lasers at an impressive rate of 18,000 times per second. Consequently, it generates detailed and accurate 3D point cloud data in real-time.

To understand the operation of the VLP-16 in greater detail, the firing sequence of its lasers is crucial. Each of the 16 lasers is activated and then recharged every 55.296 microseconds. Within this time frame, each laser fires sequentially, with a precise interval of 2.304 microseconds between successive firings. Following the completion of all 16 laser firings, there is a brief idle period of 18.43 microseconds before the next cycle begins. This pattern results in a complete cycle time of 55.296 microseconds for the lasers to fire, recharge, and repeat, ensuring a continuous and consistent data acquisition process.

The lasers employed in the VLP-16 are semiconductor laser diodes. These diodes are created by applying a current through a series of stacked p-n junctions, which stimulates photon emission. These photons are then focused

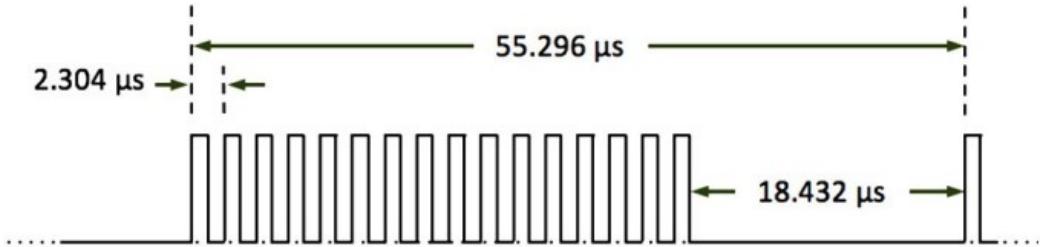


Figure 1: Firing Sequence Timing

into a narrow, coherent laser beam. The VLP-16 lasers are classified as Class 1 lasers, which adhere to strict safety standards. This classification ensures that even with direct exposure to the human eye, the lasers do not pose a significant risk, making them safe for use in a variety of environments and applications [2].

In addition to their safety, the design and operational characteristics of the VLP-16 make it suitable for a wide range of applications, from autonomous vehicles to robotics and surveying. The precise timing and firing mechanism contribute to the sensor's ability to provide high-resolution and accurate 3D mapping, which is essential for tasks requiring detailed environmental understanding. The robust construction ensures reliable performance under various environmental conditions, further enhancing its utility in both research and practical implementations.

3.1.4 Application of LiDAR sensor in a sensing system

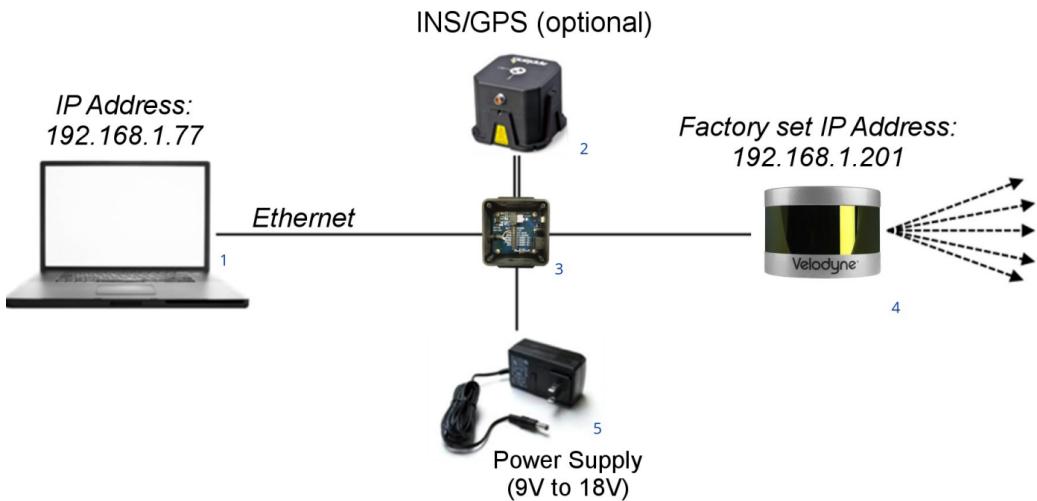


Figure 2: 3D Sensing System Components

Fig.2 can be useful to better understand how a sensing system that utilizes LiDAR technology is organized, showcasing the integration of multiple components to facilitate the acquisition and processing of 3D spatial data. This system is often used in applications such as autonomous vehicles, robotics, and environmental mapping, where precise spatial awareness is crucial.

The starting component is the laptop computer (labeled as 1 in the figure), which serves as the central processing hub. The laptop is configured with a specific IP address that allows it to communicate seamlessly with the Velodyne interface box, which is another critical component in the system. The IP address configuration is crucial; it ensures that the laptop and the interface box are on the same network subnet, enabling efficient data transfer and control of the LiDAR sensor. The laptop is equipped with specialized software, often provided by the LiDAR manufacturer, that is responsible for interfacing with the LiDAR, visualizing the data in real-time, and storing it for further analysis.

The Velodyne interface box (3 in the figure) acts as a bridge between the laptop and the LiDAR sensor. This device is essential for the communication and data transfer processes. It is connected to the laptop via an Ethernet cable, which allows the high-bandwidth data stream generated by the LiDAR sensor to be transmitted to the laptop for processing. The interface box also receives power through a DC power supply cable, ensuring that it operates continuously and reliably. Additionally, it is directly connected to

the LiDAR sensor, typically through a specialized cable. This connection not only facilitates data transfer but, depending on the system's design, may also provide power to the LiDAR sensor itself.

Furthermore, the interface box can be connected to an optional INS/GPS antenna and interface (2 in the figure). The inclusion of an INS (Inertial Navigation System) and GPS is particularly important in applications where precise positioning and orientation data are required. This setup is crucial for scenarios where the spatial data needs to be georeferenced, or where the movement of the system must be tracked accurately in real-time. The INS/GPS integration allows the system to maintain spatial awareness even when GPS signals are weak or unavailable, such as in urban canyons or indoor environments.

The LiDAR sensor (4 in the figure) is the core component of the system. LiDAR, which stands for Light Detection and Ranging, operates by emitting laser pulses and measuring the time it takes for these pulses to return after hitting objects in the environment. By analyzing the time of flight for each pulse, the sensor can calculate the distance to the object, thereby creating a 3D point cloud that represents the surrounding environment. This point cloud data is rich in detail and is essential for applications that require precise environmental mapping and object detection. The LiDAR sensor is capable of scanning in a 360-degree field of view, depending on the model, capturing comprehensive spatial data in real-time.

The process begins when the LiDAR sensor, powered either directly or through the interface box, emits its laser beams. As these beams hit objects, the reflected pulses are captured by the sensor. The data is then transmitted to the Velodyne interface box, which processes the signals and sends the relevant information to the laptop computer. The software on the laptop processes this incoming data, allowing for real-time visualization and analysis. If an INS/GPS system is integrated, the spatial data is also synchronized with positional information, ensuring that the point cloud is accurately referenced in space.

3.1.5 Laser Return Modes

The VLP-16 LiDAR sensor offers advanced capabilities through its laser return modes, which are crucial for interpreting data, particularly in environments with multiple reflective surfaces. The sensor supports three return modes: Strongest, Last, and Dual, each designed to cater to different application needs by selecting specific reflections from the laser pulses.

When a laser pulse is emitted from the VLP-16, it diverges gradually, increasing the likelihood of interacting with multiple objects along its path. For example, a pulse might hit a tree branch and the ground behind it, resulting in multiple returns. The sensor's return modes allow users to specify which of these reflections to record.

The Strongest Return Mode is designed to capture the reflection with the highest intensity, typically from the most reflective surface. This mode is particularly useful in scenarios where accurate detection of bright or retro-reflective surfaces is crucial. For instance, in urban mapping, the strongest return could help in accurately identifying traffic signs or license plates, which are designed to reflect more light back to the sensor.

In contrast, the Last Return Mode captures the final reflection in a sequence, which usually corresponds to the farthest object along the laser pulse's path. This mode is valuable in applications where the goal is to measure the distance to a surface that is behind an obstacle, such as the ground beneath vegetation. By recording the last return, the sensor can provide data that represents the farthest object, which is often critical in aerial surveys or forestry applications where ground elevation needs to be mapped even when there is an overlying canopy.

The Dual Return Mode offers an even more sophisticated approach by capturing both the strongest and the last returns, provided the two reflections are separated by at least one meter. This mode is especially useful in complex environments where multiple layers of objects are present, such as in forested areas where a pulse might first hit the upper canopy before reaching the ground. Dual returns enable the sensor to provide detailed information about both the canopy and the terrain beneath it, which can be essential for accurate tree height measurement or biomass estimation.

An important aspect of understanding these return modes is recognizing the relationship between laser power, reflectivity, and distance. Reflectivity is a measure of how much light is reflected back to the sensor, and it varies with the material and surface characteristics of the object. The VLP-16's calibrated reflectivity values, ranging from 0 to 255, allow the sensor to distinguish between diffuse reflectors like tree trunks (which scatter light) and retro-reflectors like road signs (which reflect light back directly). This differ-

entiation is vital for interpreting the data accurately, as it influences which return mode might be most appropriate for a given application.

In scenarios where multiple returns are possible, the choice of return mode directly impacts the data's interpretation. For instance, in Dual Return Mode, if the strongest return is also the last one, the sensor defaults to reporting the second-strongest return, ensuring that meaningful data is captured, even in complex reflective environments.

The flexibility of the VLP-16's return modes, combined with its calibrated reflectivity measurements, makes it a powerful tool for a wide range of applications, from urban mapping to environmental monitoring. Understanding and utilizing these modes effectively can significantly enhance the quality and accuracy of LiDAR data, providing deeper insights into the physical world.

To better understand the concepts listed above, Fig.3 4 5 are reported.

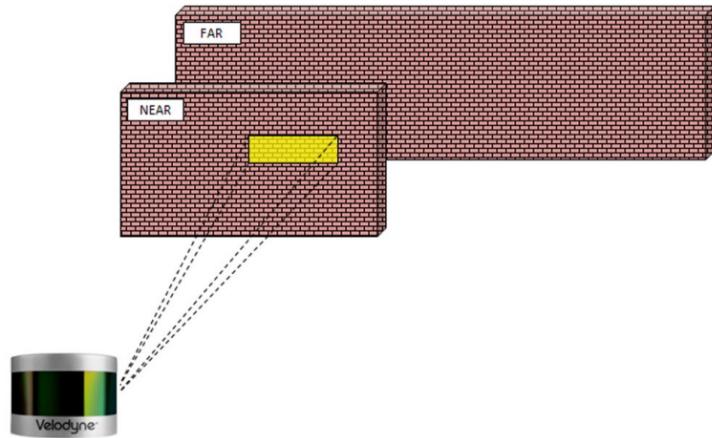


Figure 3: Single Return: Last and Strongest Returns

Fig.3 presents the case in which the laser pulses hit only one obstacle, and as consequence the related return is the strongest and last one.

Fig.4 shows a laser pulse hitting the near wall with most of its energy while the rest reaches the far wall, the Dualreturn mode allows to capture measurements from both surfaces. However, the sensor only records both returns if the two objects are separated by at least one meter.

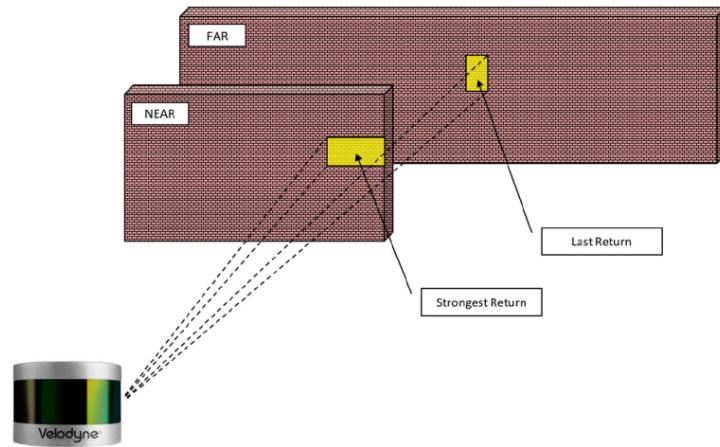


Figure 4: Dual Return: Last and Strongest Returns

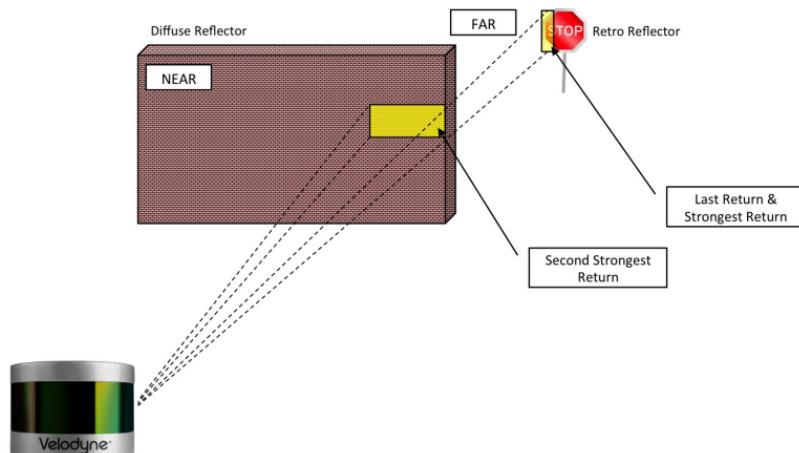


Figure 5: Dual Return with Far Retro-Reflector

In Fig.5 is possible to see that most of the beam hits the far wall, making it the strongest return. However, it's also possible that the far wall is distant enough that, despite reflecting most of the beam, the near wall still produces the strongest return.

3.2 Detection

The term 'detection' refers to the ability of the LiDAR sensor to collect at least one data point from a flying drone. It's crucial to highlight the conditions that must be met for a detection to occur and to ease the understanding of this concept, some figures can be helpful. For the sake of clarity, Fig.6 and Fig.7 show the physical meaning of two angles that are going to be mentioned many times in the current subsection:

- θ , which represents the rotation of the turret around the z-axis (pan motion of the PTU)
- ϕ , which refers to the spin motion internal to the LiDAR

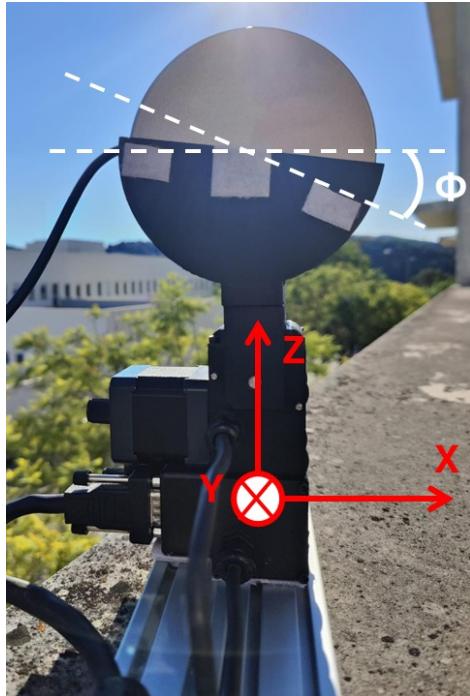


Figure 6: PTU pan angle and LiDAR's internal spin motion angle

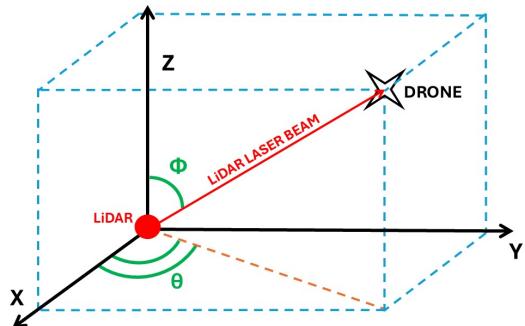


Figure 7: The PTU, the drone, and the corresponding world coordinates.

3.2.1 Extraction and Processing of Recorded Data

ArduPilot is an open-source autopilot system that provides comprehensive flight control for a wide range of unmanned vehicles, including drones, planes, cars and boats. It is widely utilized in both hobbyist and professional settings due to its flexibility, robustness, and extensive feature set. The system is designed to manage and automate the navigation and operation of these vehicles, enabling them to perform complex tasks with minimal human intervention. When an ArduPilot-controlled drone is in operation, it generates a wealth of data related to its flight performance and environmental interactions. This data includes information such as altitude, speed, orientation, GPS position, battery status, and sensor readings. The data is recorded by the flight controller in real-time and stored in log files. These logs are crucial for analyzing the vehicle's performance, diagnosing issues, and improving future operations. The recording process is accomplished through the flight controller's onboard logging system. As the drone flies, the flight controller continuously monitors and records various parameters from its sensors and systems. This data is then written to a log file stored on a memory card within the flight controller or transmitted to a ground station for real-time monitoring and logging.

To extract and analyze this data, the log files are downloaded from the flight controller to the computer, then loaded into MATLAB using the *Ardupilog* function, which reads the binary log data from the specified file path. This function creates a MATLAB object or structure containing all the recorded data from the drone's flight controller.

With the log file successfully loaded, the processing part starts by extracting arrays for longitude, latitude, and altitude from the GPS data within the log. These arrays provide crucial geographic and elevation information necessary for analyzing the drone's trajectory and performance. Additionally, the code calculates the precise timestamps for each GPS data point.

3.2.2 Outputs of the Processing Part

The outputs of the processing part are presented from Fig.8 to Fig.12. Fig.8 illustrates the evolution over time of both the PTU pan angle and the drone yaw angle: the red path is taken from the ArduPilot logs from the drone itself, while the blue saw tooth plot represents the *Sweeping* motion during a field test, in which the PTU speed was constant, as shown by the slopes of the lines that maintain the same value, only changing in sign.

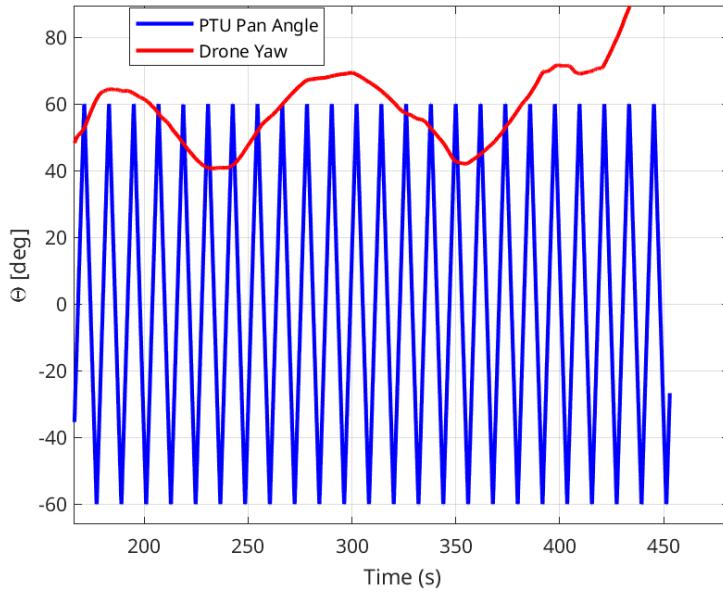


Figure 8: PTU Pan Angle and Drone Yaw over Time

We are particularly interested in the intersections between the blue and red plots, as these points represent the instances when the PTU, and thus the LiDAR mounted on top of it, is oriented towards the drone. Let's focus on one of these intersections, as shown in Fig.9, and let's consider the one that, as highlighted by the label in figure, occurs after 231.683 seconds from the beginning of the test.

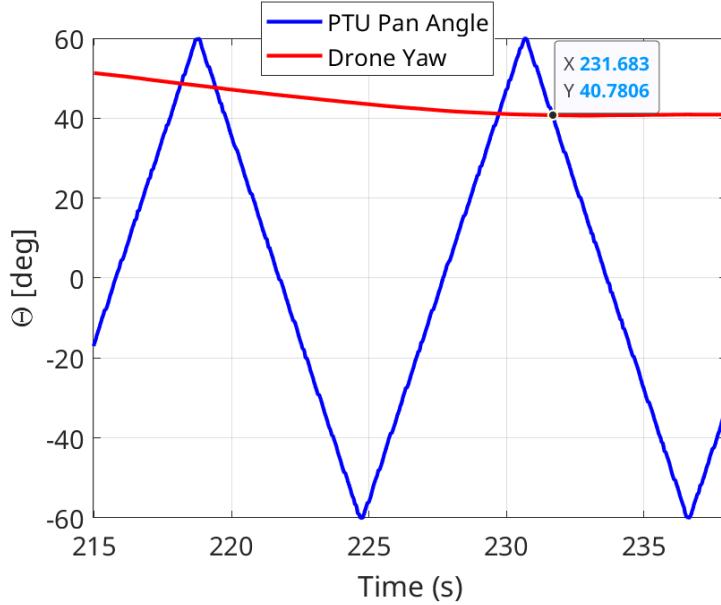


Figure 9: PTU Pan Angle and Drone Yaw over Time, zoom in

However, it is important to note that this is not sufficient to consider the highlighted point a detection. For a detection to occur, the angle covered by the internal spin motion of the LiDAR, related to the consecutive firing of the 16 laser pulses, must also match the drone's pitch angle. If this condition is not met, the situation would be the following: even if the PTU orients the LiDAR in the correct direction to see the drone at that precise time instant, if the LiDAR is not firing the laser beam at the right angle, it will not be possible to collect any data. Focusing on Fig.10, it shows the behavior over time of both the drone's pitch angle and the LiDAR's internal rotation angle. Notably, at the corresponding time of the intersection in Fig.9, there is also an intersection in Fig.11.

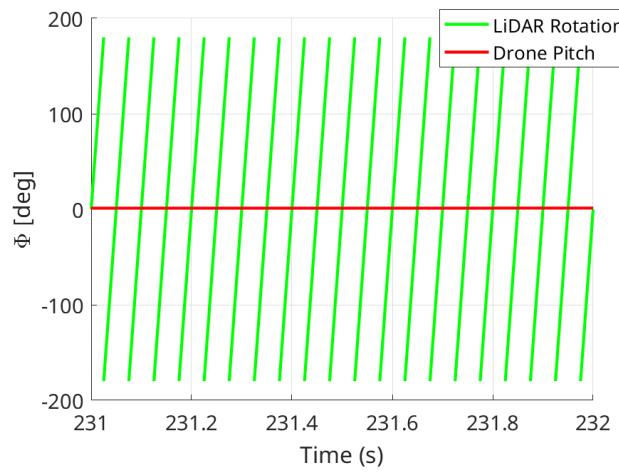


Figure 10: LiDAR Internal Spin Motion Angle and Drone Pitch over Time, zoom out

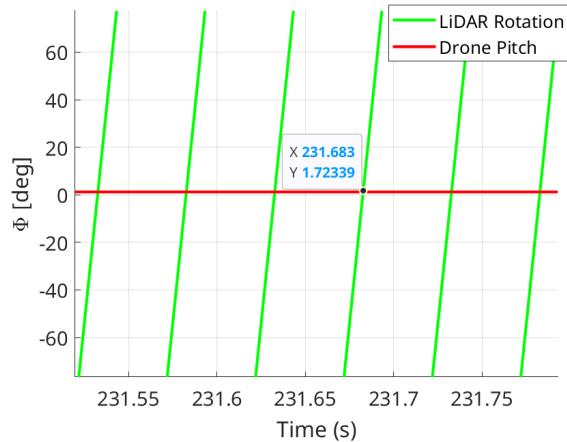


Figure 11: LiDAR Internal Spin Motion Angle and Drone Pitch over Time, zoom in

The red path, representing the drone pitch, appears flat both in Fig.10 and Fig.11 because it displays a very short time period, about 0.02 seconds.

At this point, it is still too early to conclude that a detection has occurred. One final step is needed: we must ensure that a laser pulse was fired and active at that specific time instant to confirm the drone's detection. This concept has been previously highlighted in Fig.1.

Otherwise, the PTU would orient the LiDAR in the exact direction, the internal spin motion of the sensor allows the laser pulse to hit the flying drone if fired, but if the beam is not fired at that instant, no data will be collected. In the case took in analysis, looking at Fig.12 we can conclude that a detection verifies at 231.683 seconds, since a laser pulse was indeed fired at that time.

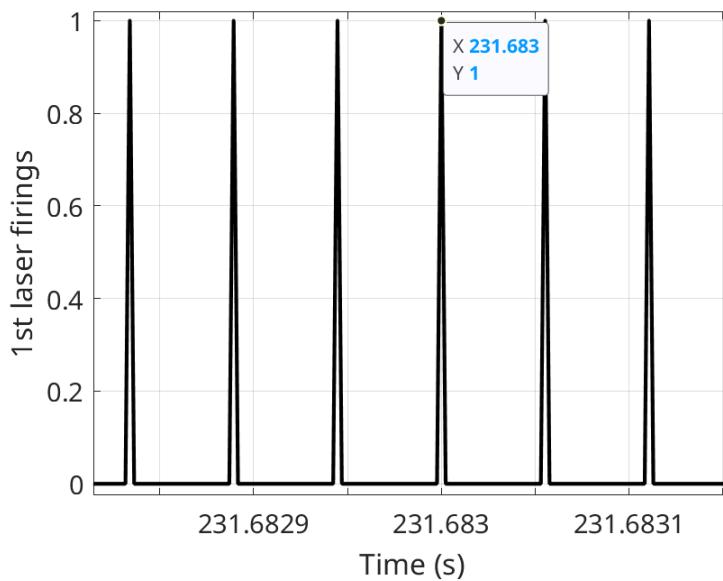


Figure 12: LiDAR laser pulses firings, zoom in

3.3 Gaps between successive measurements

Trying to visualize with an infrared camera how the fired lasers look like on an object surface, they appear as a pattern of spots, where each spot is a small rectangular area composed of three smaller bands of light, as shown in Fig.13. At the sensor's ring lens, these laser spots measure 9.5 mm tall by 12.7 mm wide: the more the object is far from the sensor, the longer will be the distance that the laser pulses have to cover and, due to their divergence, the measures of the laser spots will be larger.

Inside the VLP-16 sensor it's noticeable the presence of a vertical array of lasers that produce multiple laser scans, where each scan line, that can be seen as adjacent laser scans, consists of multiple laser spots.

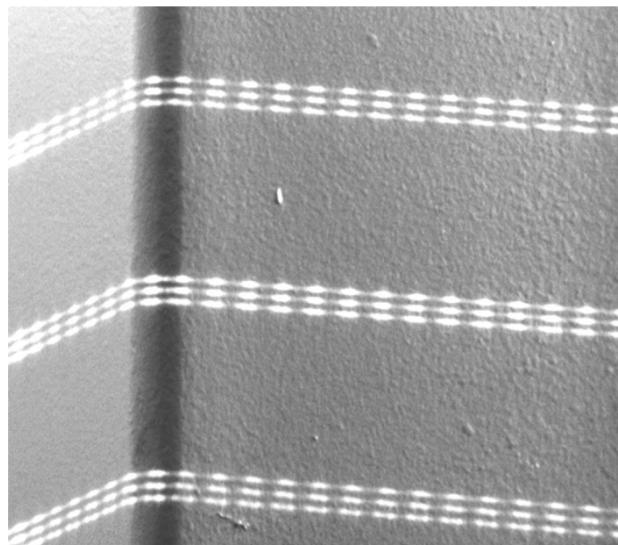


Figure 13: Laser Spots on a wall

The gap between these scan lines enlarges by increasing the distance of the target from the sensor and can be calculated using the formula:

$$\text{gap} = \text{distance} \times \tan(\text{vertical angle between scan lines}) \quad (2)$$

To better understand how the LiDAR emits lasers and how the gaps in the data emerge, refer to Fig.14. This figure presents a top view of the LiDAR, where the red beams indicate the laser emissions as the sensor rotates. Additionally, a comparison with Fig.16 reveals that a 3D LiDAR scans its environment along both vertical and horizontal axes, resulting in much denser measurements along the latter. Fig.15 is attached to make the gaps between the red beams of Fig.14 more visible.

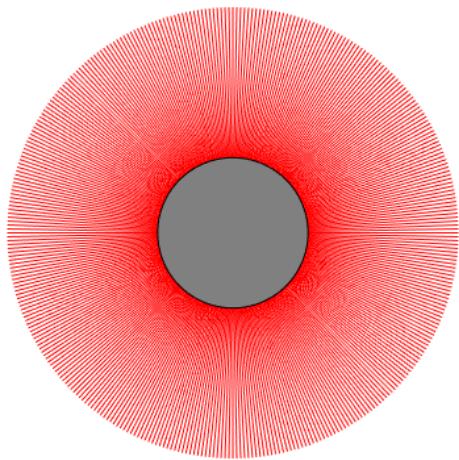


Figure 14: View of the top of the LiDAR, with red showing the beams as the sensor inside the LiDAR rotates

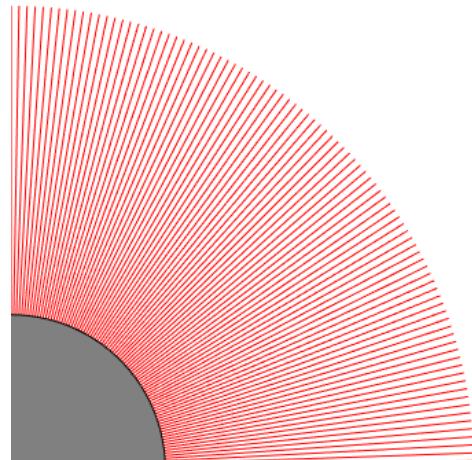


Figure 15: Partial view of the top of the LiDAR, with red showing the beams as the sensor inside the LiDAR rotates

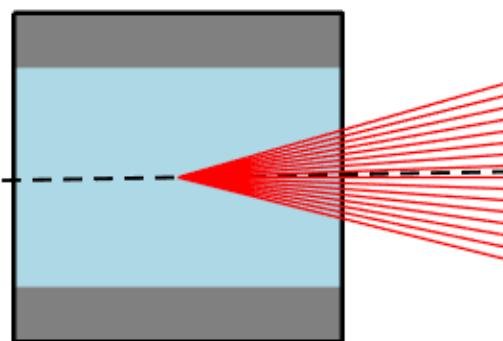


Figure 16: Side view of the LiDAR showing the 16 vertical beams in different planes

The issue of the presence of gaps between successive measurements, makes it more challenging to detect small targets at larger distances¹⁰. As consequence, detection and accurate tracking of a drone that is flying far away from the LiDAR becomes probabilistic, since the target potentially (and probably) can hide in the gaps.

To address this issue, the system that will be utilized aims at improving the number of detections by temporarily focusing the scanning action of the LiDAR on the target. This focusing can be seen as *SlowingDown*, *Swinging* around or *Stopping* to the heading of the target. These types of motion will be carried out thanks to the turret PTU-E46 Pan-Tilt Unit from FLIR Systems, though only the pan motion will be utilized in this case. The LiDAR mounted on the top of the turret is set to rotate around the z-axis (pan motion), accordingly to the selected type of motion of the turret: in this way, the expected goal is to gather a sufficient amount of data and, as consequence, to reduce the uncertainty that affects the drone position. To enhance the performance of the system is crucial to improve the detection probability: to achieve this, the key is to analyze how detection rates change by varying system parameters during simulations. Having concluded that part, field testing will be indispensable for validating these enhancements in a real-world scenario.

¹⁰https://docs.clearpathrobotics.com/assets/files/clearpath,robotics_023729 - TDS2 - 2c7454cf9f317be53ce1938dca7ddcf4.pdf LiDAR VLP - 16 User Manual (63 - 9243 Rev.E)

4 Method

To achieve our goal, the first stage revolves around the detection of the target of interest. This step is followed by tracking process, meant as the continuous estimation of the target's position, that will be optimized by dynamically modifying the type of motion to which the LiDAR is subjected, aimed at maximizing the number of times the drone remains inside the field of view of the LiDAR, ensuring it is visible for as long as possible. The current section will provide a detailed explanation of the working principles of each component, with a particular focus on the interactions between them. This includes the hardware components, ROS nodes, and the data exchanged between them. The implementation of the detection and identification mechanisms, alongside with the integration with the overall system, will be highlighted.

To facilitate the understanding of the components and their interactions, a block diagram is presented in Fig.17: it illustrates the sequence and relationship between the various elements of the drone tracking system.

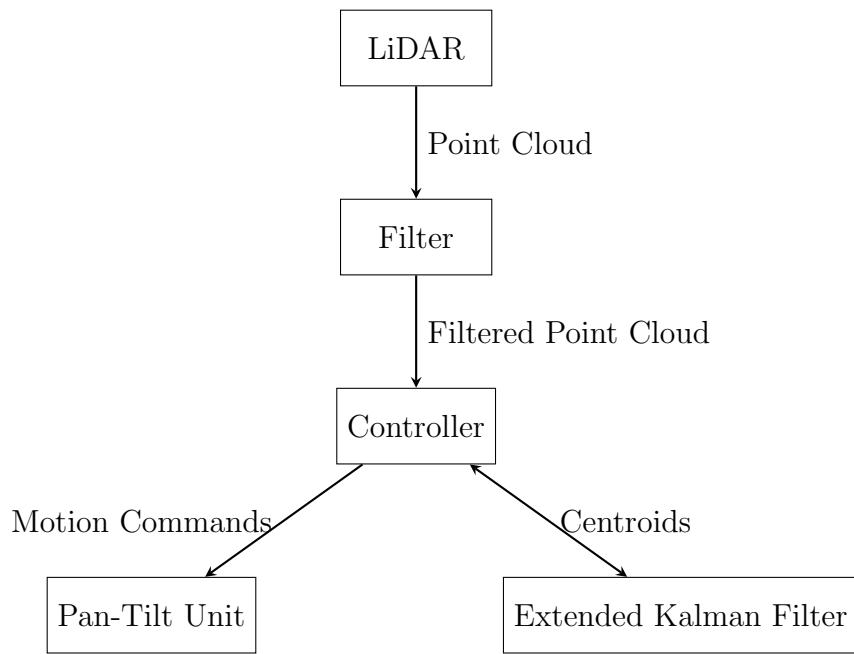


Figure 17: Flowchart of the LiDAR data processing system

Since the LiDAR and PTU turret have already been clarified, in the following subsections the exploration will delve into the ROS nodes and their interactions, detailing their respective roles and how they communicate to ease the system's functionality.

4.1 Filter

The C++ Filter node takes as input the point cloud published on the */velodyne* topic from the LiDAR and processes it through two consecutive filtering actions:

- XYZ filter
- 3D convex hull filter

The 3D convex hull is built within the filtered region from the first step (once the hull is defined, the node publishes it as a marker to allow real-time visualization in tools). The marker is populated with the points from the convex hull, making it easy to see the boundaries being used for filtering action. The dimensions of the hull and of the XYZ filter have been chosen by studying the length and width of the field through Google Earth, knowing the exact position in which the LiDAR would have been positioned. A crucial aspect in the field was verifying the computations to ensure that there were no trees, street lamps, or any other obstructions within the constructed hull. The importance of applying the XYZ filter at first is to avoid a large amount of computations: the working principle of the hull filter includes the comparison of each point of the published cloud with each surface of the 3D hull (in total 6, since it is a parallelepiped). In that way it has been possible to speed up the filtering process. Through the *CropHull* function of the Point Cloud Library (PCL) for C++, the pre-computed convex hull is used to extract points that lie within it. The filtered points are then converted back to a ROS message and published for further use and visualization on the */filteredpoints* topic. Additionally, the filtered points are saved to a CSV file, in which each point's coordinates, timestamp, ring index, and message sequence number are recorded, enabling analysis and tracking of the filtered data in a second moment.

4.2 Controller

For what regards the C++ Controller node, it's possible to choose between 4 different types of motion for the PTU:

- Swing behavior, the PTU oscillates between two angles determined by the initial position and predefined limits. The state transitions from *StartMotion* to *Motion*, and it keeps track of the number of swings in each direction. When the specified number of swings is reached, it switches to the *Sweeping* state, resetting the turret to its full range of motion
- Slow Down behavior, the PTU's speed is reduced by a factor, and the state transitions based on the angle difference from the start position. Once the angle exceeds a specified threshold, it returns to the *Sweeping* state with normal speed.
- Stop behavior, halts the turret motion for a predefined duration before resuming its *Sweeping* rotation. It sets a low speed to stop the PTU and uses a timer to measure the stop duration. Once the duration elapses, it resumes normal speed and transitions to the *Sweeping* state.

The node processes incoming point cloud data to detect clusters and calculate centroids, recording them to a CSV file. If significant movement is detected between consecutive centroids, it triggers a state change to start a new motion cycle. During the operation, the node records state changes, motion types, and relevant data points to a CSV file for logging and analysis. It's important to highlight that for each selected way of motion, it starts to be applied only when the received filtered point cloud from the filter node is not empty, otherwise the PTU goes on with its usual motion which is set to be the sweep between two angles that can be changed dynamically. This node communicates also with the EKF node, sending to it the centroids of the detected clusters.

4.3 Extended Kalman Filter

The node subscribes to a topic that provides centroid positions (on which they have been previously published by the controller node) and uses these to update the EKF state. It initializes the filter with an initial centroid position, and then alternates between Prediction and Update phases to refine the estimated state. The Prediction phase uses a state transition model to predict the next state based on the previous state and the elapsed time. In the Update phase, the filter incorporates the latest centroid measurement to correct the predicted state. The EKF state consists of the position and velocity of the centroid in 3D space, represented as a 6-dimensional vector. The node also publishes the filtered centroid positions on another topic and writes various data, including predictions, updates, and state information, to CSV files for further analysis. Additionally, it includes a mechanism to handle lost detections by checking the covariance matrix and elapsed time since the last detection. If the system detects that the object is lost, it reinitializes the filter when a new centroid is detected.

Going deeper in the EKF working principle, it's useful to report the formulas implemented by this filter:

- Prediction:

$$\begin{aligned} x_p &= F \cdot x; \\ x &= x_p; \\ P_p &= F \cdot P \cdot F^T + Q; \\ P &= P_p; \end{aligned}$$

- Update:

$$\begin{aligned} z &= \begin{bmatrix} x_{\text{centroid}} \\ y_{\text{centroid}} \\ z_{\text{centroid}} \end{bmatrix}; \\ e &= z - H \cdot x_p; \\ S &= H \cdot P_p \cdot H^T + R; \\ K &= P_p \cdot H^T \cdot S^{-1}; \\ x &= x_p + K \cdot e; \\ P &= (I - K \cdot H) \cdot P_p; \end{aligned}$$

The state vector x carries variables representing the system's State. The Covariance Matrix P quantifies uncertainty in these variables.

The State Transition Matrix F is useful to understand how the state evolves over time in a linear dynamic system. The Process Noise Covariance Matrix Q models uncertainty in the state transition process. The Measurement Vector z contains the XYZ coordinates of the centroid obtained in the processing phase implemented by the Controller node, and it's used to refine the state estimate x . The Measurement Matrix H maps the state vector x to z , establishing the relationship between state variables and measurements. The Measurement Noise Covariance Matrix R characterizes noise in the measurements z . The Identity Matrix I is employed in operations to maintain structure and update state estimates in the EKF algorithm. The subscript p indicates a predicted variable.

Here below the values assigned to the entries of the matrix are reported:

- State Transition Matrix F :

$$F = \begin{bmatrix} 1 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

- Process Noise Covariance Matrix Q :

$$Q = \begin{bmatrix} 0.01 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.01 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.01 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.01 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.01 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.01 \end{bmatrix}$$

- Measurement Matrix H :

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

- Measurement Noise Covariance Matrix R :

$$R = \begin{bmatrix} (1)^2 & 0 & 0 \\ 0 & (1)^2 & 0 \\ 0 & 0 & (0.2)^2 \end{bmatrix}$$

- Covariance Matrix P :

$$P = \begin{bmatrix} (1)^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & (1)^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & (0.2)^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & (4)^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & (4)^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & (4)^2 \end{bmatrix}$$

The values configured in the matrices are carefully chosen to model the dynamics of the drone and the sensor's characteristics. The State Transition Matrix F is designed to predict the next state based on the current state, incorporating a time step Δt of 0.05 seconds, derived from a system frequency of 20 Hz. This matrix includes both position and velocity states, with the diagonal elements maintaining the state values and the off-diagonal elements accounting for the influence of velocity on position changes.

The Process Noise Covariance Matrix Q represents the uncertainty in the system's model, reflecting inaccuracies in predictions. It features diagonal entries of 0.01, indicating a small, uniform variance for each state, which suggests a relatively low level of uncertainty in the model's predictions. This choice implies that the system dynamics are well understood and modeled.

The Measurement Matrix H maps the state vector to the measurements provided by the LiDAR. Since the LiDAR detects only the position of the drone, H has ones in the positions corresponding to the position states and zeros elsewhere. This setup ensures that the measurements directly update the position estimates without affecting the velocity components.

The Measurement Noise Covariance Matrix R captures the noise characteristics of the LiDAR measurements. It has variances of 1 for the x and y positions and 0.2 for the z position, indicating that the LiDAR is more precise in measuring the x and y positions compared to the z position. This reflects potential measurement noise and uncertainty associated with the sensor.

Finally, the Initial Covariance Matrix P represents the initial uncertainty in the state estimates. It is diagonal with variances of 1 for the position states and 16 (4^2) for the velocity states. This configuration suggests greater initial uncertainty in the velocity compared to the position. The relatively large initial variances for the velocity states allow the filter to accommodate significant initial uncertainty while progressively refining the state estimates as new measurements are obtained.

This approach ensures a robust and efficient drone tracking system with precise performance across scenarios. By comparing results from various motion patterns, we can identify the one yielding the highest drone detection rate, improving position tracking and achieving our primary objective.

5 Validation of Methods and System

In this section, the preliminary tests conducted to evaluate the performance of the LiDAR sensor and the supporting software will be described. The validation part was divided into three steps, each designed to assess different aspects of the system's functionality.

The first step 5.1 involved simulations carried out in MATLAB. Here, two different scenarios were simulated: one where the drone moved horizontally in front of the LiDAR sensor, and another where it moved vertically, directly towards the sensor. These simulations were crucial for gaining an initial understanding of the LiDAR's detection capabilities, specifically in terms of how the number of detections varied with the drone's position and movement relative to the sensor. This provided a foundational insight into the expected behavior of the system before moving on to real-world tests.

The second part of the testing was conducted in a real-world environment, focusing on verifying the correct operation of the C++ nodes running within the ROS (Robot Operating System) environment. During these tests, the LiDAR sensor was kept in a fixed position, meaning it was not rotating. The active nodes included the filter, the controller, and the Extended Kalman Filter (EKF). However, the node responsible for controlling the Pan-Tilt Unit (PTU), which would allow the LiDAR to rotate, was not activated at this stage. The goal of this phase was to assess the performance of the system under straightforward, controlled conditions where the sensor's position remained static. The testing took place on a terrace, providing an open and controlled environment. A critical aspect of the setup was the design of the convex hull used for filtering the LiDAR data. The convex hull was configured to start from a specific height, which was intentionally set to be above my own height. This setup ensured that the filter would only consider objects within a defined 3D space that was relevant for the system's intended application, filtering out irrelevant detections below this height. In the first test, I walked in a straight line towards the LiDAR and then back again, repeating this motion several times. This setup was designed to provide a basic evaluation of how effectively the EKF could track my movement and whether the system could accurately detect and localize my position as I moved closer to and further from the sensor. By keeping the sensor fixed, the test isolated the performance of the EKF and the other active nodes, allowing us to verify their functionality in a simple, controlled scenario. The second test introduced slightly more complexity by varying my movement paths. I walked along three different straight lines, each at a different distance from the LiDAR. This variation in trajectory and distance was intended to further challenge the EKF, testing its ability to maintain

accurate tracking and localization across different paths. The fixed position of the LiDAR ensured that any deviations in tracking accuracy could be attributed to the behavior of the EKF and related nodes, rather than to any movement of the sensor itself. To interact with the filtered 3D space defined by the convex hull, I used a large sign that I held in my hands, allowing me to reach into the area that the filter was designed to focus on. This approach was essential for verifying that the filtering mechanism worked as intended—filtering out irrelevant detections below the specified height and only processing data from objects within the target 3D space. Additionally, it allowed for testing the robustness of the system in distinguishing between relevant and irrelevant data, ensuring that the EKF and associated nodes could maintain accurate tracking and localization under these specific conditions. These tests were crucial for establishing a baseline performance of the system’s core components, ensuring that the filter, controller, and EKF were functioning correctly before moving on to more complex scenarios. The node responsible for the PTU motion was intentionally kept inactive during this phase and was only activated in subsequent tests, where the complexity of the system’s operation would be increased by introducing dynamic sensor motion and more intricate movement patterns.

The final phase of the validation process took place in a real-world environment but introduced a more complex scenario where the LiDAR sensor was mounted on a dynamically rotating turret. This test differed significantly from the previous ones because, unlike earlier scenarios where the LiDAR was stationary, the sensor was now in constant motion. The setup involved me walking in a square pattern around the LiDAR while the turret rotated continuously. This created a dynamic situation where both the target (myself) and the sensor were moving simultaneously. The test aimed to evaluate the system’s performance under realistic, challenging conditions, focusing on how accurately the LiDAR could detect and track my position while both the sensor and I were in motion. This was key to assessing its ability to maintain reliable detections despite constant changes in the field of view caused by turret rotation. In addition to tracking accuracy, this test also provided an opportunity to observe how well the Extended Kalman Filter (EKF) could integrate dynamically changing data. As the turret rotated and I moved along the square path, the EKF needed to process rapidly shifting inputs and maintain accurate localization and object detection. The constantly shifting sensor perspective while tracking a moving target provided a rigorous test for the EKF’s capabilities.

These tests, from simulations to real-world scenarios, ensured a thorough understanding of the system and validated the performance of both the LiDAR and its software.

5.1 First Step: MATLAB Drone Path Simulation

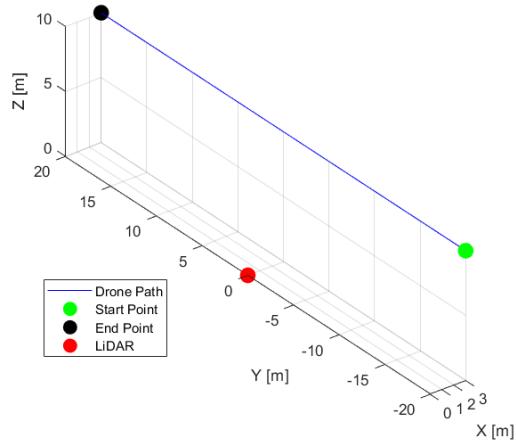


Figure 18: DRONE moving on straight line at 3m distance along x -axis from LiDAR, horizontal direction

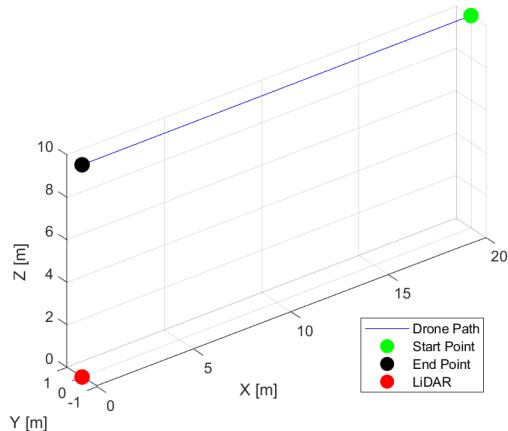


Figure 19: DRONE moving on straight line with direction towards the Li-DAR

In the MATLAB simulations, the way in which the drone's trajectory impacts the detection performance of a stationary LiDAR sensor has been explored, as illustrated in Figs. 18 and 19.

For the horizontal trajectory shown in Fig.18, the drone moves parallel to the LiDAR's x -axis, maintaining a consistent distance of 3 meters from the sensor. It starts at -20 meters along the y -axis and travels for 40 meters in the y -direction. In this scenario, the LiDAR's detection capability varies depending on the drone's position relative to its field of view. When the drone is centrally located along its path, it is within the LiDAR's effective detection range, leading to a higher number of detections. However, as the drone approaches the edges of its path, it moves out of the LiDAR's detection range due to the increasing distance from the sensor. Consequently, fewer detections are recorded as the drone moves towards the start and end points of its trajectory, where it is no longer within the sensor's view.

In contrast, the vertical trajectory depicted in Fig.19 involves the drone moving directly towards the LiDAR along the x -axis. The drone begins at a distance of 20 meters from the LiDAR and travels directly towards it. In this path, the drone remains within the LiDAR's field of view throughout its entire flight. This continuous presence within the sensor's detection range results in a significantly higher number of detections compared to the horizontal path. The entire trajectory keeps the drone within the effective range of the LiDAR, maximizing the number of detections.

What comes out from the simulation results is that the LiDAR detects a greater number of drone positions when the drone's path ensures it remains within the sensor's field of view for the duration of the flight. The vertical trajectory, where the drone moves directly towards the sensor, yields a higher number of detections because the drone remains continuously within the detection range. In contrast, the horizontal trajectory results in fewer detections due to the drone being outside the sensor's range during the outer portions of its path.

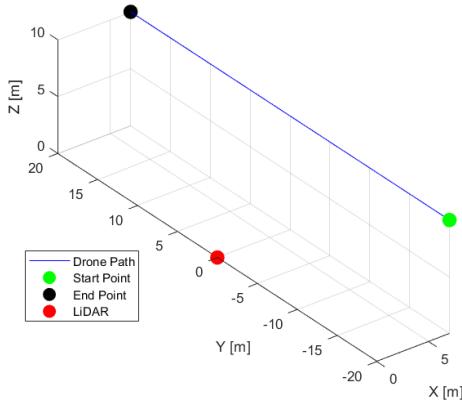


Figure 20: DRONE moving on straight line at 7m distance along x -axis from LiDAR, horizontal direction

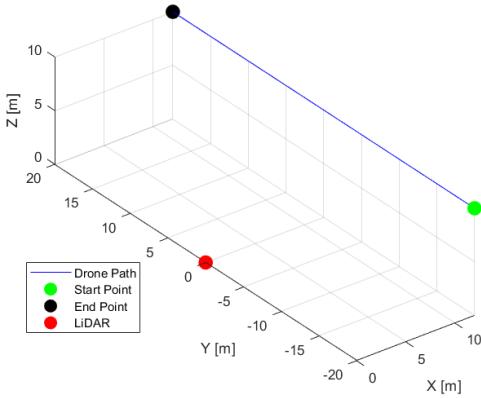


Figure 21: DRONE moving on straight line at 12m distance along x -axis from LiDAR, horizontal direction

In Figs. 20 and 21, the case of drone's horizontal motion have been further investigated: what comes out is that, as expected, as the distance over which the drone moves horizontally increases, the number of detections decreases. This occurs because the farther the drone travels from the sensor, the harder it becomes to detect.

The same concept has been applied to Figs.22 and 23. The result is the same: a lower amount of detections with respect to the case showed in Fig.19.

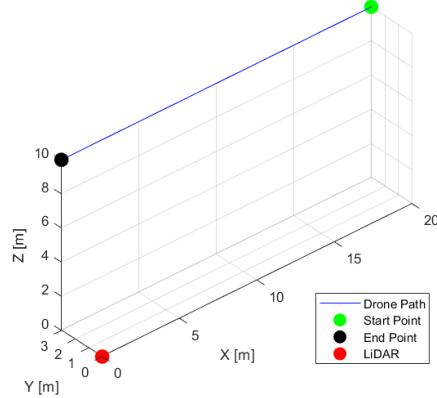


Figure 22: DRONE moving on straight line with direction towards the LiDAR, 3 meters offset to the left of LiDAR

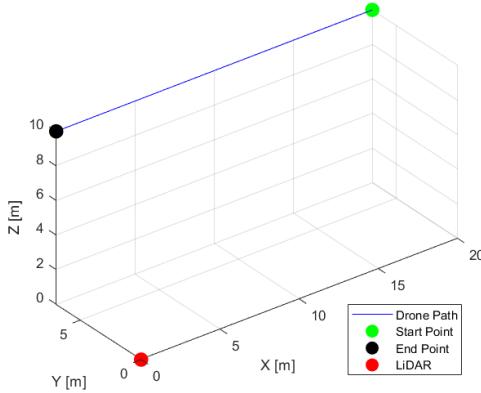


Figure 23: DRONE moving on straight line with direction towards the LiDAR, 7 meters offset to the left of LiDAR

It's important to note that for motion along the y -axis, the number of detections decreases more gradually compared to motion along the x -axis as the distance increases. This is because, in motion along the y -axis, the drone remains within the LiDAR's field of view for a larger portion of its flight path. In contrast, with motion along the x -axis, the drone quickly moves out of the LiDAR's field of view as it moves further away, leading to a more rapid decline in detection rates.

5.2 Second Step: Motion in Straight Patterns

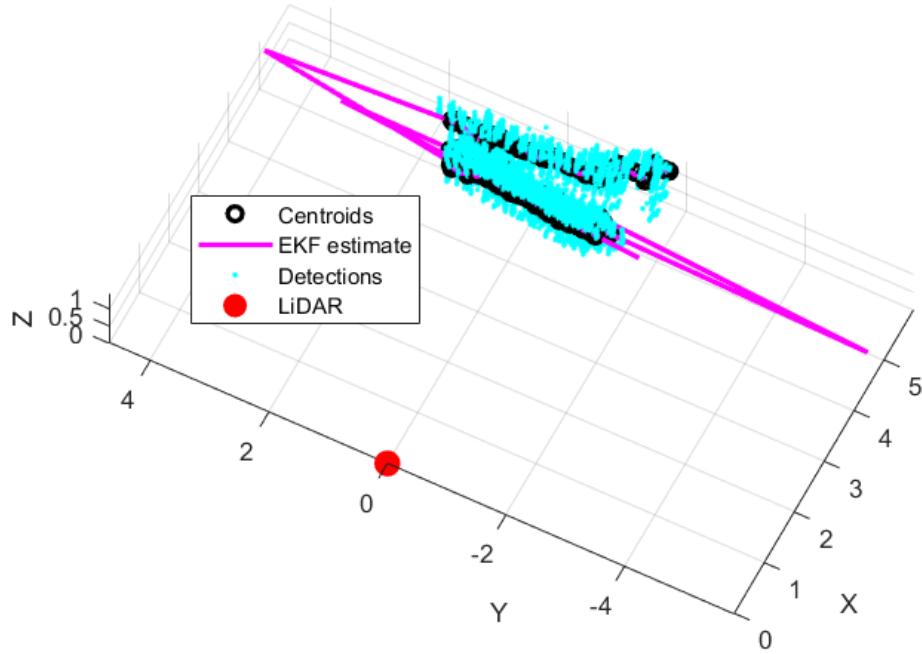


Figure 24: Motion of myself holding a sign, perpendicular to beams of the LiDAR

Fig.24 illustrates the motion of objects along three lines positioned between 5 and 6 meters from the LiDAR along the x-axis. The first line represents the furthest distance, while the other two lines are close to each other at a distance of 5 meters.

A significant concentration of detection points, shown in cyan, can be observed above and below the black rings in the figure. This distribution is expected because the cyan points result from the filtering process applied to the LiDAR's acquired data. If we imagine a rectangular object moving with its largest surface facing the LiDAR throughout its trajectory, it makes sense that the centroid of this surface should move along a straight line, as depicted in the figure.

Another observation is the range of data acquisition by the LiDAR, which spans approximately from -2 meters to +2 meters along the y-axis. Beyond this range, data acquisition ceases. Consequently, during intervals without new data, the Extended Kalman Filter (EKF) estimates the object's position,

which closely follows the actual path. This is evident in the estimates, which extend to more than 4 meters to the left and approximately -5 meters to the right at their extremes.

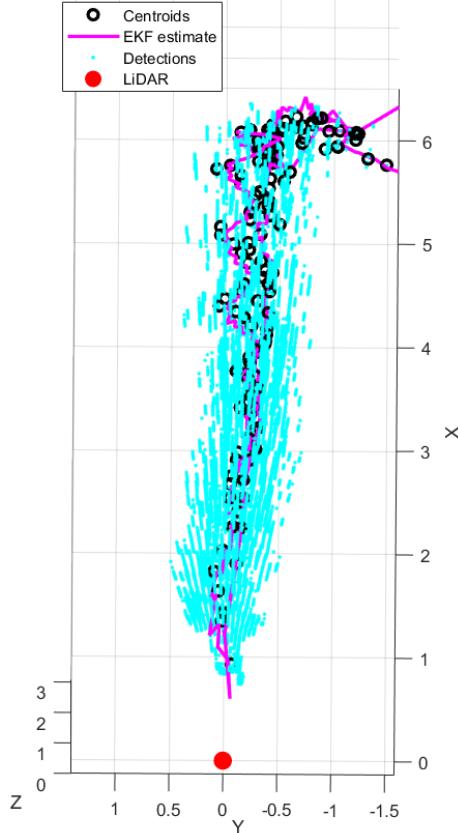


Figure 25: Motion of myself holding a sign, parallel to beams of the LiDAR

Fig.25 shows off another type of motion, where the starting point was located in the top-right corner of the image, and the endpoint was slightly above it. During this motion, I walked towards the LiDAR, maintaining the target within the sensor's field of view at all times, before reversing direction and moving away from it.

As observed, the density of detected points increases as I moved closer to the LiDAR. This increased density is especially evident when I reversed direction and began to move away from the sensor. This phenomenon is expected due to the LiDAR's higher resolution and sampling rate when objects are in closer proximity. Interestingly, the centroids of the detected points trace a path through the center of the point cloud, which aligns with our expectations.

The Extended Kalman Filter (EKF) also performed well in this scenario. The EKF effectively tracked the target's motion, largely due to the continuous update steps that refine the estimate based on the incoming measurements. Because the target remained within the LiDAR's visible range throughout the motion, the EKF was consistently supplied with new data, allowing it to correct and improve the state estimates. This continuous correction is crucial, as it helps to mitigate the effects of noise and sensor inaccuracies, ensuring that the estimated trajectory remains closely aligned with the true path of the target.

5.3 Third Step: Motion in Square Patterns

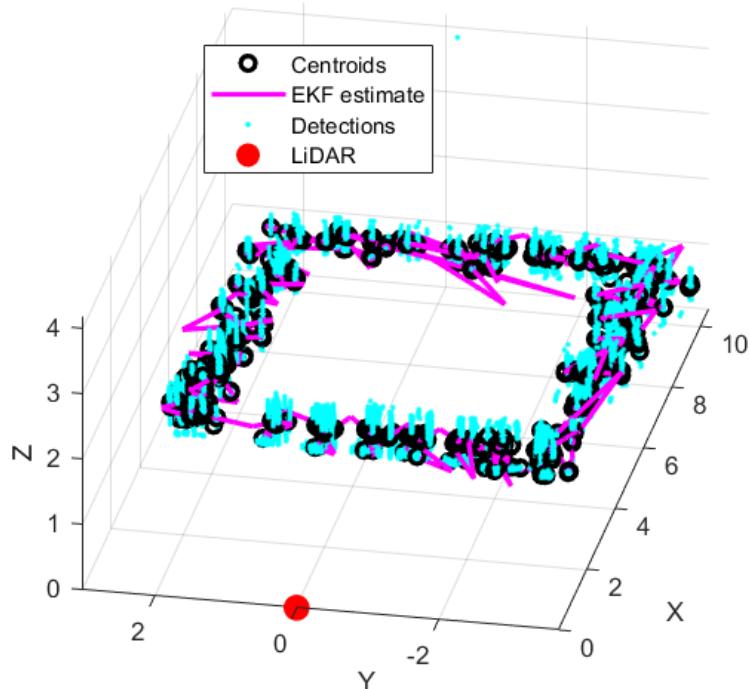


Figure 26: Square Pattern: Slow Down Motion

Fig.26 shows a 3D plot illustrating the system’s response when the dynamic motion is set to slow down. In this scenario, when the LiDAR, followed by the filter, identifies a cluster of points as a potential drone, the turret receives a command to reduce its rotational speed to one-quarter of its original rate for a brief period (specifically for a 10° angular span, as set for this test).

From the 3D plot, it is evident that the Extended Kalman Filter (EKF) occasionally estimates the position of the target with noticeable errors. This is likely due to the nature of the dynamic motion algorithm employed. Once the system identifies a target, it continues to move in a manner optimized for tracking objects moving at higher speeds or with rapid positional changes. However, in this test, the target (held in my hands) was moved at a very low speed. The discrepancy between the system’s response and the actual target movement suggests that this implementation may not be ideal for scenarios involving slow-moving objects. The EKF, under these conditions, struggles to maintain accurate tracking, leading to less precise estimations of the target’s position. This highlights a potential area for refinement in the

system's motion control strategy to better accommodate targets with varying movement characteristics.

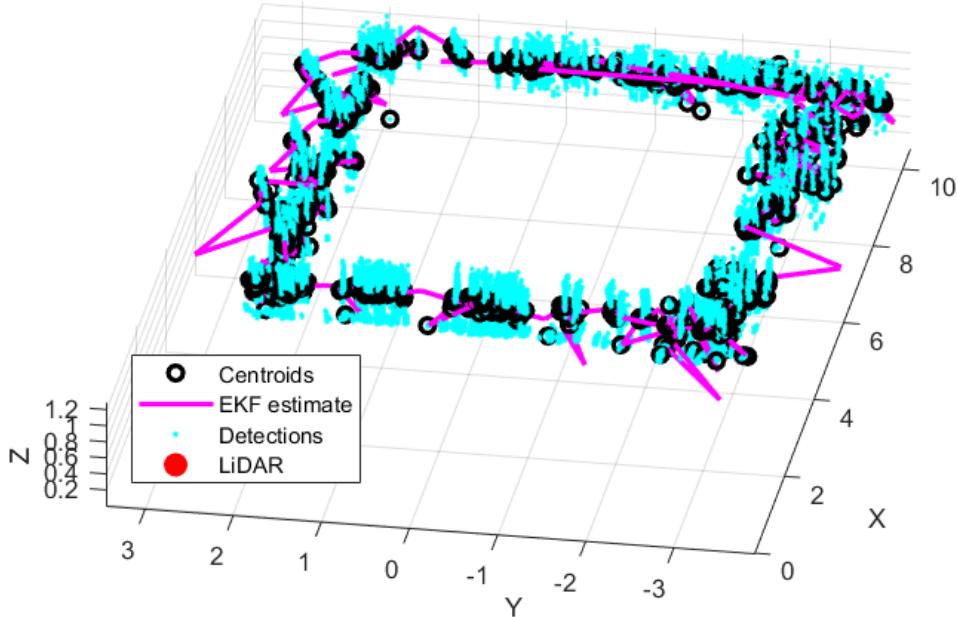


Figure 27: Square Pattern: Swing Motion

Fig.27 presents a 3D plot illustrating the system's response when operating under the swing dynamic motion configuration. In this mode, once the LiDAR and filtering process identify a cluster of points as a potential drone, the turret begins to oscillate within a narrow angular range of 10° (5° to the left and 5° to the right) centered towards the direction of the detected target.

The analysis of the 3D plot reveals that the Extended Kalman Filter (EKF) occasionally estimates the target's position with a slight margin of error. However, in this case, the swing motion proves to be advantageous. The oscillation towards a fixed direction allows the LiDAR to focus on and around the identified target point, which is particularly beneficial given that the target (which I was holding) was moved at a very slow speed. This controlled back-and-forth motion ensures that the sensor captures the largest possible amount of data around the target, improving the overall accuracy of the system's tracking ability in scenarios involving slow-moving objects.

In this context, the swing dynamic motion is highly effective. It enables

the system to concentrate its sensing efforts on a localized area, thereby maximizing data acquisition and refining the EKF's estimation process. This implementation is well-suited for capturing detailed information about slowly moving targets, ensuring that even minimal movements are accurately detected and tracked.

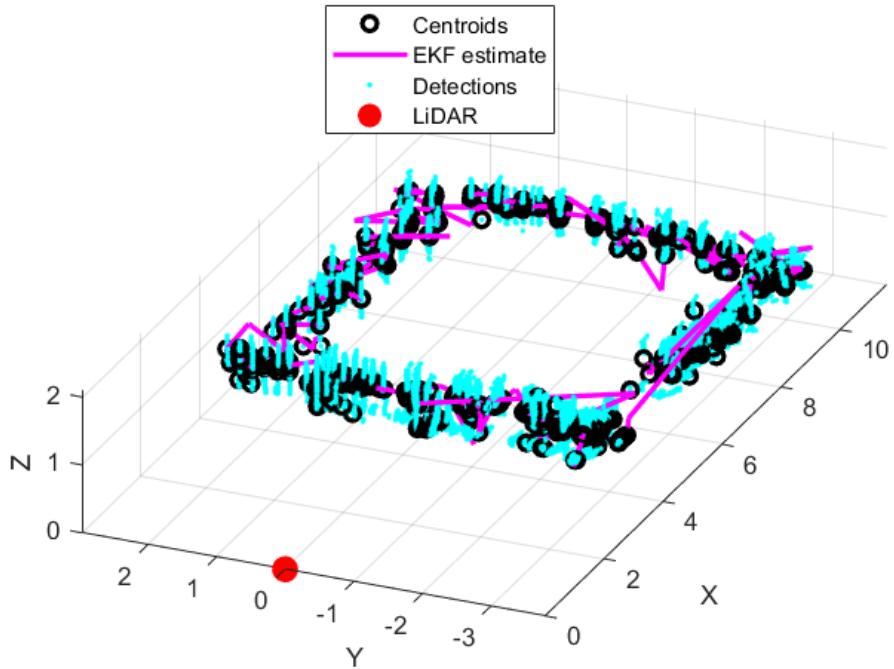


Figure 28: Square Pattern: Stop Motion

Fig.28 illustrates a 3D plot of the system's behavior when utilizing the stop dynamic motion configuration. In this mode, whenever the LiDAR and filtering process identify a cluster of points as a potential drone, the turret immediately halts its movement and remains stationary, pointing the LiDAR towards the target for a fixed duration of 5 seconds before resuming motion.

The plot shows that the Extended Kalman Filter (EKF) performs with exceptional accuracy under this configuration. The stop motion strategy is particularly effective in this scenario, where the target (held in my hands) was moving at a very low speed. By *Stopping* the turret and maintaining the LiDAR's focus on the identified target, the system maximizes the number of detections and data points collected during the stationary period. This prolonged focus allows the EKF to process more consistent and reliable data,

significantly improving its ability to accurately estimate the target's position.

Compared to the other dynamic motion strategies, the stop approach proves to be superior for tracking slow-moving targets. The turret's stationary phase ensures that the LiDAR can capture the highest possible amount of detailed information, reducing errors in target estimation. This method is particularly well-suited for scenarios where the target is moving slowly, as it enables the system to effectively 'lock on' to the target, providing the EKF with the best possible data to work with.

6 Experimental Setup

The target drone that has been used is a Sky Hero Spyder X4, presented in Fig. 29, with a 0.85-meter carbon fiber frame. It is powered by 400 rpm/V motors and a 6S 16000 mAh LiPo battery. The drone exhibits a Drotek DP0601 GNSS receiver with an external GNSS antenna, a Drotek RM3100 compass and two inertial measurement units (IMUs). The total weight of the drone is 5.5 kg, and it can operate for up to 45 minutes, allowing for extended flights and, as consequence, for recording a large amount of data.



Figure 29: Sky Hero Spyder X4 drone

For what regards the detection system, it is made up by three components:

- PTU-E46 Pan-Tilt Unit by FLIR Systems (Fig.30), which is engineered to deliver cost-effective, rapid, and precise positioning for cameras and other payloads. It supports up to 4 kg payload, achieves speeds higher than 300° per second, with a resolution of about 0.129°
- 3D printed support, designed to maintain the LiDAR in a fixed position while the PTU is rotating, paying particular attention to not obstruct the fired laser pulses

- LiDAR Velodyne VLP 16 by Ouster (Fig.31), it features a measurement range of up to 200 meters and generates approximately 600,000 points per second, with a 360° horizontal and 40° vertical field of view



Figure 30: 3D printed LiDAR support on top of Pan-Tilt Unit



Figure 31: LiDAR Velodyne VLP 16 by Ouster

In Fig.32 is possible to see the overall system that has been used to carry on the experiment, with all the three components put together.

It's important to underline that to compare the different motion patterns in order to determine which one is the best to enhance the detection and tracking performances, the drone was kept inside the convex hull during all data collection (to make the second filtering action possible). Moreover, the same drone path has been repeated the closer one to the other in order to have the consistent conditions, for more accurate further analysis. In last instance, in Fig.33 is reported the overall tracking system in action, in particular is represented the *Stopping* motion approach. In this captured frame, the sensor is resting towards the detected drone position.



Figure 32: Tracking system



Figure 33: Tracking system in action during the tests

Using the instrumentation described in this section, CSV files have been recorded for each of the proposed motion approaches based on field tests. In particular the collected data include:

- filtered points, the ones from the original Point Cloud captured by the LiDAR that lie within both the XYZ filtered area and the convex hull, published by the Filter node
- centroids, the Controller node clusters the filtered points, then computes the these centroids and publishes them
- state, this file recorded the time instants in which I have a change in the state of the system, i.e. when the selected type of motion is the *Swinging*, then this file records when there's the transition from standard *Sweeping* motion to the *Swinging* one, and the other way round
- Extended Kalman Filtered values, filled with time instants in which a prediction or an update occurs, saving for each one of them the current State Vector $x = [x, y, z, \dot{x}, \dot{y}, \dot{z}]$ and the elements on the diagonal of the Covariance Matrix P , so $[P(1, 1), P(2, 2), P(3, 3), P(4, 4), P(5, 5), P(6, 6)]$
- ROS bag file, which allows me the recording and playing back of ROS message data, in this case the ones coming from a whole drone flight
- ArduPilot logs, saved in a binary format, records detailed information about the drone's flight, such as GPS coordinates, IMU readings, battery status, altitude, speed, and flight modes

By processing all these quantities, it's possible to carry on analysis and to compare each approach with the others, allowing for the determination of the motion pattern that best fits our needs, as shown in Sec.7.

7 Results and Discussions

This section examines the various methods used for controlling the drone and evaluates their performance across different tests. It begins with the *Initial Test*, where the drone is manually flown to establish a baseline of its behavior. Next, in the *Automated Drone Control Test*, the drone's performance is assessed using an Extended Kalman Filter (EKF) in a simulated environment. Following this, the *Automated Drone Control Test* evaluates the system's performance in real-world conditions. Finally, the *Analytical Comparison of Motion Patterns* contrasts the movement patterns between manual and automated control, identifying the strengths and differences of each approach.

7.1 Initial Test: Drone Controlled Manually

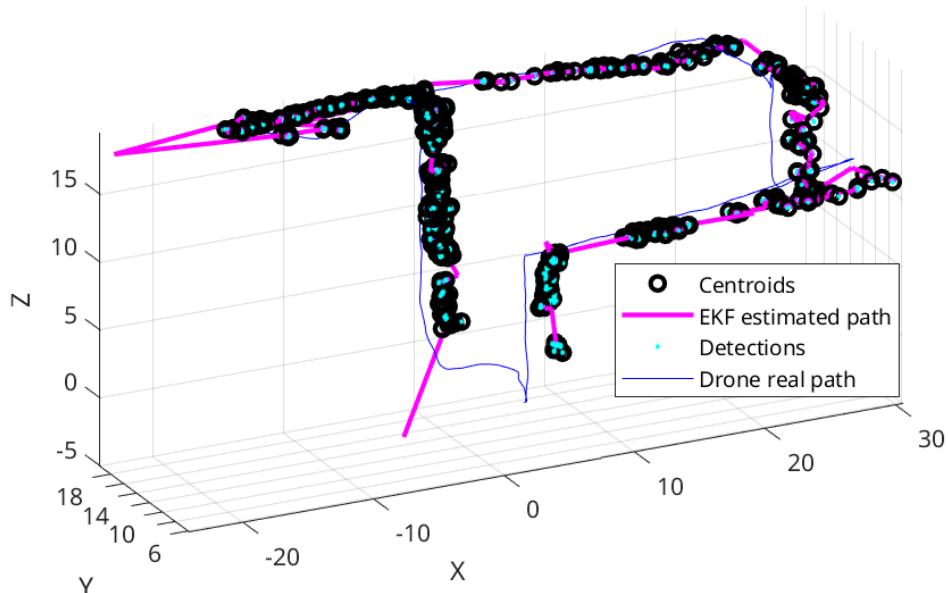


Figure 34: Tracking of drone path, with *Swinging* motion

Fig.34 has been built up to show in a clear way how the detection and tracking processes are going. The blue path is displaying the exact path

followed by the drone during the flight, the data from the ArduPilot logs have been used to trace it. This plot serves as a benchmark to evaluate how much are close the results from the implemented approach to the actual values. The cyan points represent all the points that are selected during the filtering step, while the black circles figure the centroids. It's possible to notice that the amount of filtered points exceeds the one of black circles, this is due to the fact that at a certain time instant the LiDAR captures n points of the drone. These points are then processed in the Controller node, where they are clustered and a single centroid is computed from the starting n points. Despite this, the difference is slight, since most of the times the captured points of the drone are few, especially when the drone reaches the leftmost and rightmost points within its path. The magenta plot represents the output of the Extended Kalman Filter. When detections are denser, the EKF provides a more accurate path because a higher number of received centroids leads to increase the frequency of update steps: each of them refines the state estimate by including new measurements, reducing uncertainty and improving the precision of the drone's trajectory. As expected, with more frequent updates, the EKF can more precisely track the drone's motion.

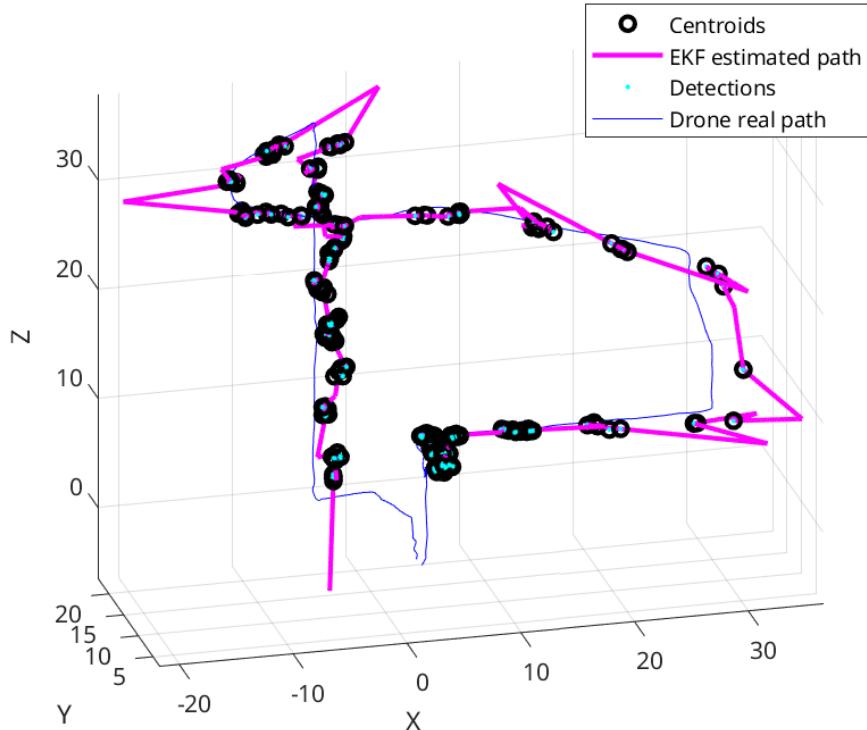


Figure 35: Tracking of drone path, with *Sweeping* motion

Comparing Fig.34 with Fig.35, it's clear the difference in the number of detections (represented by the filtered cyan points) and in the EKF curve accuracy with respect to the ArduPilot logs. For the *Swinging* scenario, it has been possible to collect a significantly larger amount of data, thanks to the LiDAR focusing on a limited angular section of space, around the centroid of a previously detected cluster. Instead of continuing with standard rotation, which would waste time by rotating towards empty space, the LiDAR concentrated on that specific area. This focus increased also the number of updates, leading to enhanced tracking accuracy of the flying drone. The *Swinging* approach not only maximizes the efficiency of the LiDAR but also demonstrates the advantage of adaptive scanning strategies in dynamic environments.

7.2 Automated Drone Control Test: EKF Simulation

To ensure more consistent data, the tests were repeated by changing the method of controlling the drone's motion from manual to automatic. This adjustment allowed for a uniform flight path across all tests, enabling more accurate comparisons when evaluating the results of different dynamic motion implementations.

Before conducting these tests with automated drone motion, the behavior of the Extended Kalman Filter (EKF) was simulated using MATLAB software. This simulation was crucial for providing a detailed analysis of how the Extended Kalman Filter would perform in various scenarios, ensuring accurate estimation of the drone's state (such as position, velocity, and orientation) despite the presence of noise and other uncertainties in the system. By thoroughly testing the Extended Kalman Filter in a controlled simulation environment, its performance was validated, and the parameters were fine-tuned before application to real-world drone flights, thus enhancing the reliability of subsequent tests. The output of these tests are shown in Figs.36 and 37.

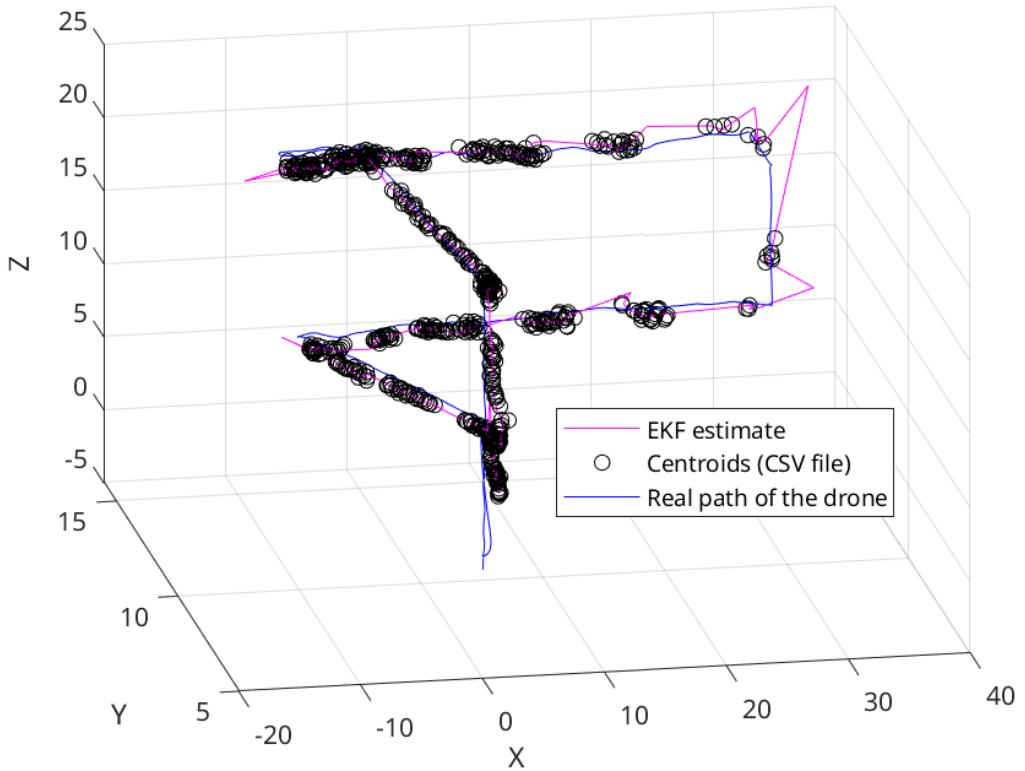


Figure 36: Simulation of EKF behavior: Prediction Step

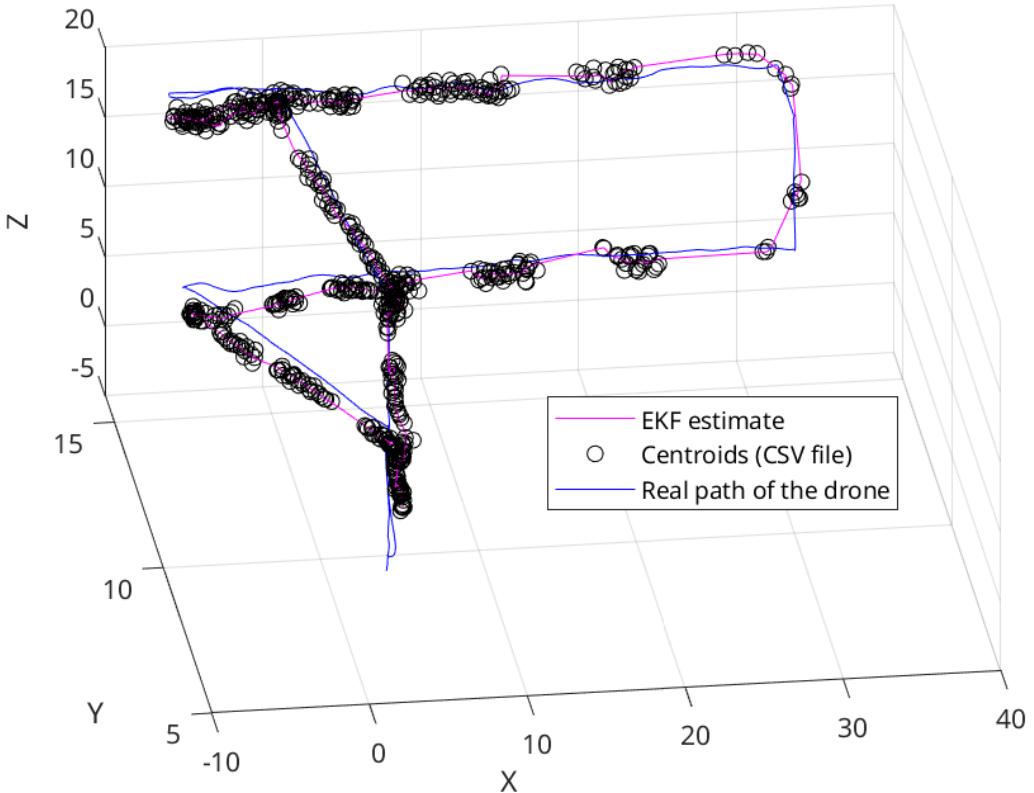


Figure 37: Simulation of EKF behavior: Update Step

Based on the figures, the first observation is that the blue plot is identical in both. This plot was obtained by having the drone follow a predefined path autonomously, with the LiDAR and turret activated simultaneously. The chosen adaptive motion strategy was a *Swinging* motion.

To extract and analyze this data, log files were downloaded from the flight controller and then loaded into MATLAB using the *Ardupilot* function. This function reads the binary log data from the specified file path and creates a MATLAB object or structure containing all the recorded data from the drone's flight controller. This process enabled the generation of the blue plot.

Regarding the centroids (represented by black circles), it is evident that their distribution along the estimated path is identical in both figures. This consistency is because the same recorded file from the field was used for both, as indicated in the legend, which mentions a CSV file.

The key difference lies in the magenta plot, which provides insight into how the EKF would have performed in a real-life application. To gain a deeper understanding of the filter's behavior, a separation was made be-

tween the update step and the prediction step. This separation allows for a more detailed analysis of the EKF's performance by distinguishing between the 'corrected' states (after sensor measurements are integrated) and the 'predicted' states (before corrections), enabling an assessment of the filter's accuracy and responsiveness under various conditions.

As expected, the update graph shows a path much closer to the blue one. This is because it includes only the points representing the position of the center of the captured drone frame (the 'corrected' positions). In a real application, these corrected positions would help bring the estimate closer to the actual drone path. Moreover, this plot contains significantly fewer points than the one shown in Fig.36, which displays a larger error compared to the update graph.

It is also noticeable that the error is greater when the drone is flying further from the LiDAR, while the estimate is more accurate when the drone is closer to the LiDAR.

7.3 Automated Drone Control Test: Final Results

7.3.1 3D Estimated Path vs Real Path

In this part, the figures illustrating the output of the three dynamic motion strategies are reported. From these, a first analysis can be carried on.

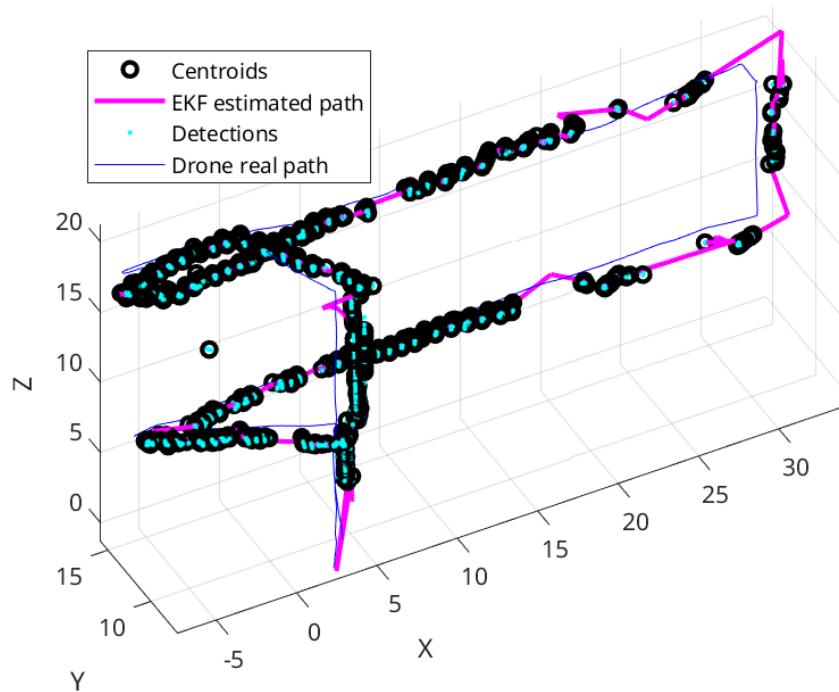


Figure 38: Tracking of Drone Path, *Swinging Approach*

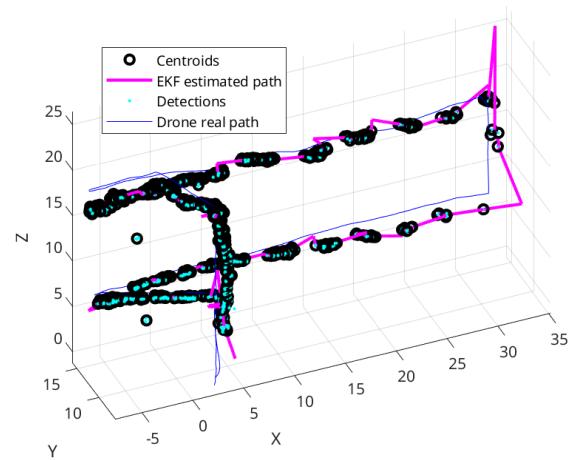


Figure 39: Tracking of Drone Path, *SlowDown* Approach

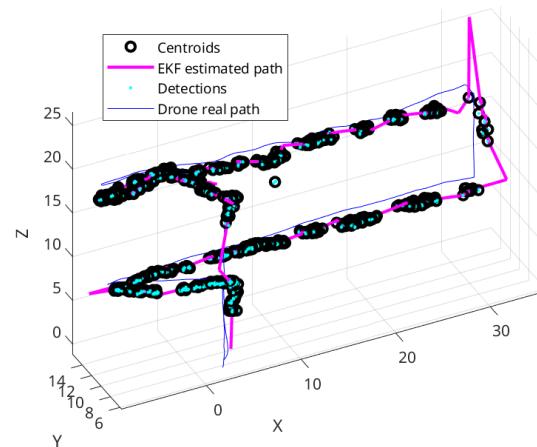


Figure 40: Tracking of Drone Path, *Stop* Approach

The three figures share a common characteristic: in the spatial region closer to the LiDAR, there is a higher density of detections and corresponding centroids (represented by the cyan points and black circles, respectively). However, as the distance from the LiDAR increases, particularly beyond 30 meters, the density of detected points associated with the drone diminishes noticeably.

This trend is also evident in the dynamic motion graphs, where the detections and centroids can be observed as forming distinct clusters. These clusters are separated by segments of the drone's path where no detections were captured. This pattern can be attributed to the mechanics of the dynamic motions. Specifically, when the LiDAR sensor captures even a single point from the drone, it tends to concentrate its scanning within a narrow angular range, rather than continuing its usual full rotation. This behavior results in an increased number of detections in that localized area but causes a reduction in the detection of points that fall outside this angular range.

Among the different dynamic motions, the *Stop* motion appears to produce the highest number of detections, particularly in the regions furthest from the LiDAR, as illustrated in Fig.40. This suggests that the *Stop* motion is effective at capturing more points over a larger distance.

However, when it comes to the accuracy of the Extended Kalman Filter (EKF) estimate, the *Stop* motion is not the optimal choice. As depicted in Fig.40, the EKF path (indicated by the magenta line) is less smooth, especially in the upper part of the path where the drone is farthest from the LiDAR sensor. Additionally, the presence of a peak on the right side of the path suggests that the filter was diverging at that point, likely due to insufficient updates during that period. A similar divergence is also evident in Fig.39 for the *SlowDown* motion.

In contrast, the *Swing* motion, as shown in Fig.38, does not exhibit this divergence. The EKF path in this case is smoother and more closely aligned with the actual path of the drone (represented by the blue line), indicating a more reliable estimation process.

In summary, as will be later detailed in Tab.1, the *Stop* motion is the best approach if the goal is to maximize the number of detections. However, the key focus should not simply be on the number of detections but rather on their spatial distribution and effectiveness in providing meaningful information. If a large number of detections are concentrated in a specific area, they may not contribute significantly to improving tracking accuracy. This is particularly true in scenarios where the drone is closer to the LiDAR system, as it is easier to identify and track in such areas. However, in regions where the drone is farther from the LiDAR, it becomes more difficult to detect, leading to a lower frequency of detections. These sparse detections

are, paradoxically, the most critical, as they provide the necessary data to accurately estimate the drone's position when it is hardest to track.

Therefore, having a large quantity of detections concentrated in a limited area can lead to an overestimation of tracking precision in that specific region, while leaving gaps in other areas where the drone's position is less certain. In contrast, a lower number of detections, if they are well-distributed along the drone's flight path, can result in a more balanced and accurate overall tracking performance. This balanced distribution helps in creating a smoother and more precise estimation of the drone's trajectory, thereby reducing potential errors and improving the reliability of the tracking system.

Moreover, it is essential to consider the overarching goal of this project, which is to enhance the tracking capabilities of the system in a way that prioritizes the protection of individuals' privacy and safety. The rationale behind this focus is that more accurate tracking allows for better control and management of drone operations, minimizing the risk of privacy infringements or potential harm to people. Given these objectives, the subsequent section will delve deeper into comparing the *Swinging* approach with the *Sweeping* method. This comparison will be conducted to assess how the *Swinging* approach, which has been identified as yielding superior results, improves upon the *Sweeping* method across various dimensions. This includes aspects such as detection distribution, tracking accuracy, and the overall effectiveness of the system in meeting the project's privacy and safety goals.

7.3.2 Motion Along XYZ Axes: Estimated Path vs Real Path

Figs.38, 39 and 40 illustrate the evolution of the drone position in a 3D environment, but they don't provide information about how the position is changing over time. In order to address this, Fig.41 and Fig.42 can be helpful by showing how the drone's position evolves along the XYZ coordinates with respect to time. In particular, the *Swinging* approach will be compared to the *Sweeping* approach. This comparison is intended to demonstrate how adopting a dynamic motion pattern significantly improves drone tracking across all three spatial directions (XYZ) compared to standard motion. The *Swinging* approach has been chosen for this comparison because, as highlighted in the previous subsection, it provides the most accurate estimate of the drone's motion in the sky and so the most precise tracking of it among the presented approaches.

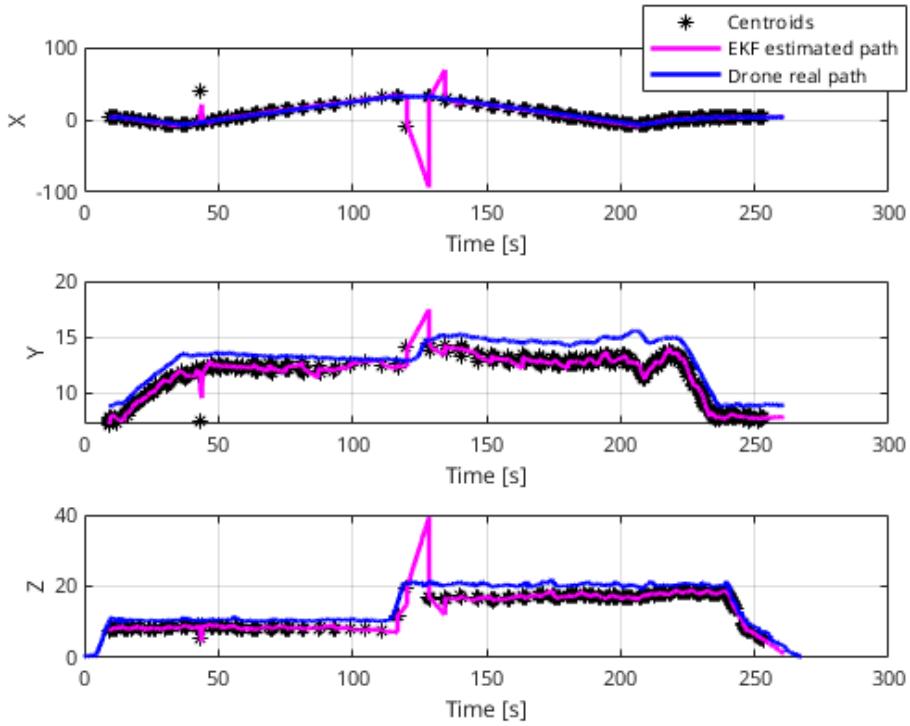


Figure 41: Tracking over Time, *Sweeping* approach

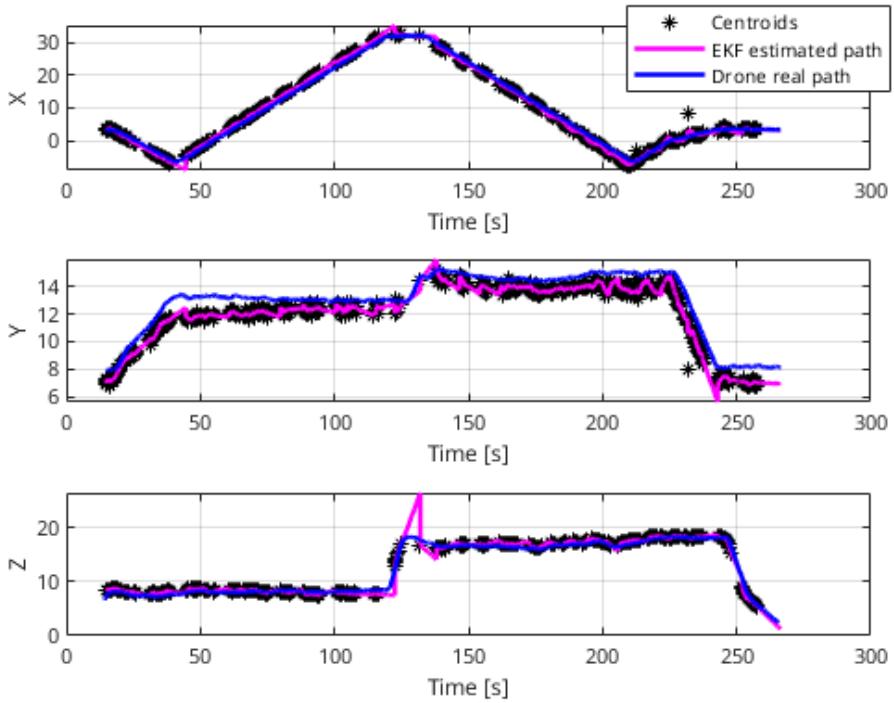


Figure 42: Tracking over Time, *Swinging* approach

Starting with the first subplot of the two figures, it is immediately evident that in the *Sweeping* graph, there is a significant error in the EKF (Extended Kalman Filter) estimate of the drone's position. This error occurs almost exactly at the peak of the blue curve, which represents the point where the drone is furthest from the LiDAR sensor, slightly more than 30 meters away. At these maximum distances, the EKF performs poorly due to the reduced number of detections. The drone, being less visible to the LiDAR at this distance, is harder to detect, leading to fewer captured frames and, consequently, a substantial increase in the prediction error of the filter.

In contrast, the *Swinging* approach shows a marked improvement. This method involves concentrating the rotation of the LiDAR around areas where detections are more likely, which mitigates the divergence seen in the *Sweeping* case. As a result, the EKF estimate in the *Swinging* scenario closely aligns with the actual path of the drone, following it with a high degree of precision and minimizing the estimation error even at the drone's maximum distance from the sensor.

Additionally, it is important to highlight that tracking the drone along the y -axis proves to be the most challenging. This axis exhibits a consis-

tent error between the captured drone frames and the actual drone motion pattern. This discrepancy can be attributed to the fact that the y -axis, often corresponding to lateral or side-to-side movement, is more difficult to track accurately due to potential occlusions or variations in detection angles, making it harder for the LiDAR to consistently capture accurate data.

Nonetheless, the advantages of the *Swinging* motion strategy are evident, particularly when compared to the traditional *Sweeping* approach. This is especially clear in the z -axis, which often represents the drone's altitude or vertical movement. In the *Sweeping* case, there is a noticeable error peak in both figures, occurring roughly at the midpoint of the drone's path—where the distance from the LiDAR sensor is greatest. This peak represents a significant error in the tracking system, highlighting the limitations of the *Sweeping* method in handling the drone's position when it is farthest from the sensor.

In the *Swinging* scenario, however, this problematic peak is either absent or significantly reduced. Along the y -axis, the peak disappears entirely, indicating that the *Swinging* strategy effectively eliminates the major tracking errors associated with the *Sweeping* approach. For the z -axis, the peak is reduced from 40 meters to 24 meters, demonstrating a substantial improvement in tracking accuracy.

These observations underscore the superiority of the *Swinging* approach over the traditional *Sweeping* method in various aspects of drone tracking. The *Swinging* strategy's ability to maintain a closer alignment between the EKF estimate and the drone's actual path, especially at maximum distances, leads to more reliable and accurate tracking. By reducing errors along both the y and z axes, the *Swinging* approach not only enhances the overall performance of the tracking system but also contributes to the project's broader goals of improving safety and protecting privacy through more precise drone monitoring.

7.3.3 Covariances of x -estimate

To better understand the accuracy of the estimated position, obtained through the implemented tracking approach, compared to the real position of the drone, and to see how the uncertainty in the drone position estimate is reduced when passing from a standard *Sweeping* motion to the *Swinging* one, the covariances of the first state variable are reported in the figures. Physically, since first state variable corresponds to the x -coordinate of the drone's position, then $P(1,1)$ indicates the uncertainty in the estimate of x -coordinate of the drone. A higher value of $P(1,1)$ means a greater uncertainty in the x -position estimate, while a lower value indicates a higher confidence in the estimate.

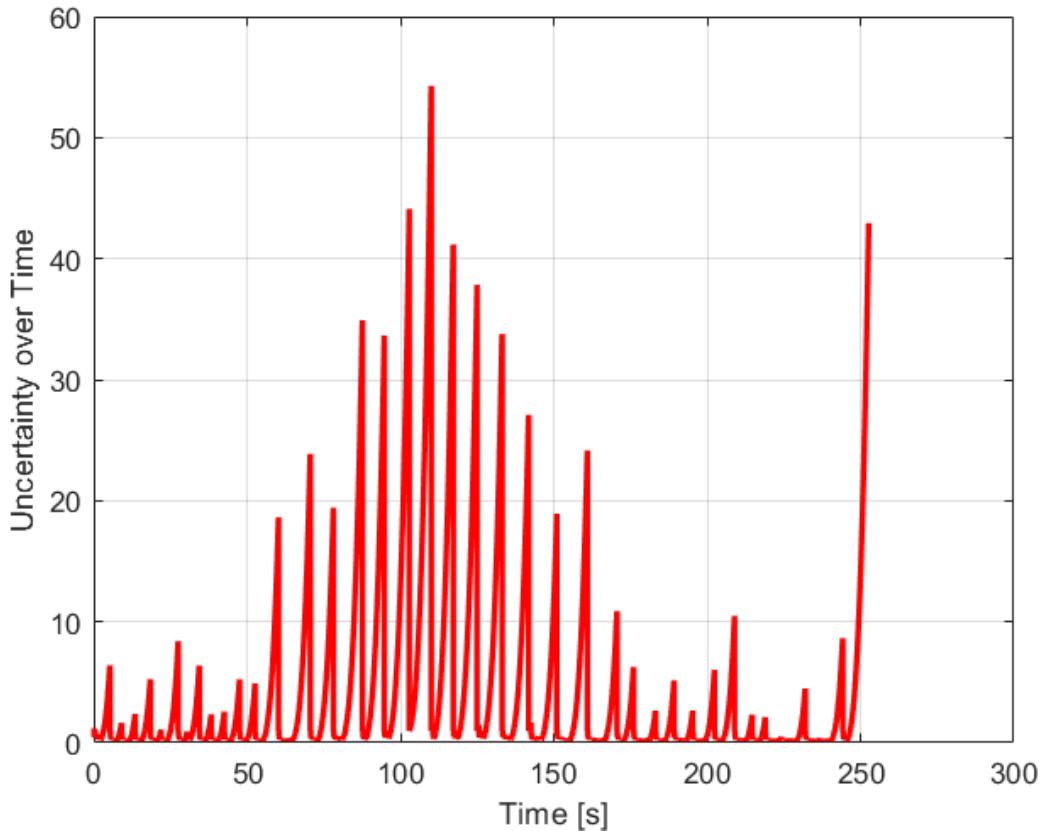


Figure 43: Covariance of the x estimate, *Sweeping* approach

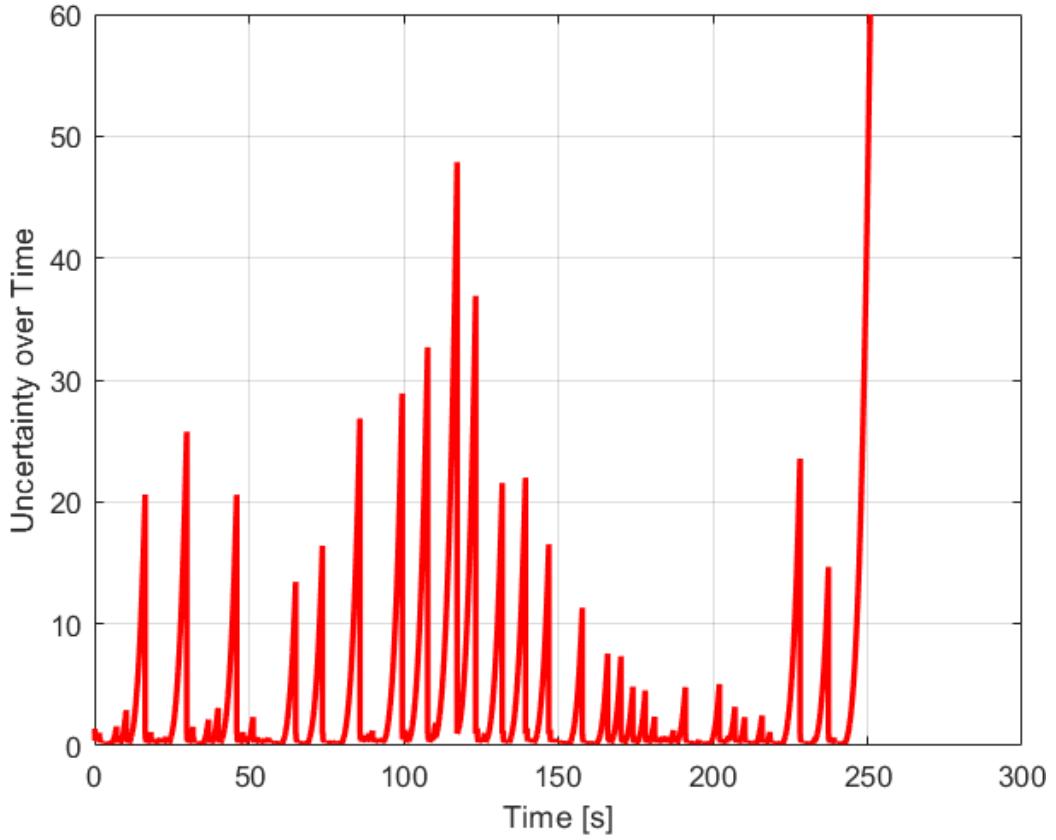


Figure 44: Covariance of the x estimate, *Swinging* approach

It is clearly visible that in the *Swinging* case highlighted in Fig.44, there are fewer and shorter peaks compared to the *Sweeping* case, presented in Fig.43, and above all they are less dense. The frequent number of updates deriving from focusing the motion pattern towards the drone, reduce the uncertainty because the state is corrected more often. Both plots are characterized by a similar final segment, where the curve rapidly increases. This fast rise can be attributed to the drone landing, making it no more visible in the filtered portion of the space and, as consequence, the uncertainty on the estimated position increases that much in the final time instants due to the lack of updates over a significant period.

7.3.4 Density of Detections over Time

An analysis of detection density over time, based on how many of the 16 beams detected the drone, reveals that more beams result in greater accuracy of the recorded and estimated data. Both graphs in Fig.45 and Fig.46 indicate that detections are concentrated at the beginning and end of the recording period, corresponding to the drone's flight path. Initially, the drone was close to the LiDAR, leading to frequent detections with many beams. As the drone moved farther away, fewer beams detected it, resulting in lower detection density in the middle of the recording.

In the final phase of the flight, the drone returned close to the takeoff point for landing, increasing detections again. Comparing the two scenarios, the *Swinging* scenario shows a higher detection density, particularly in the middle of the flight, with more beams detecting the drone. In contrast, the *Sweeping* scenario has fewer detections and beams reaching the target. The increased detection density and beam coverage in the *Swinging* scenario enhance position estimation accuracy, demonstrating its effectiveness in tracking performance.

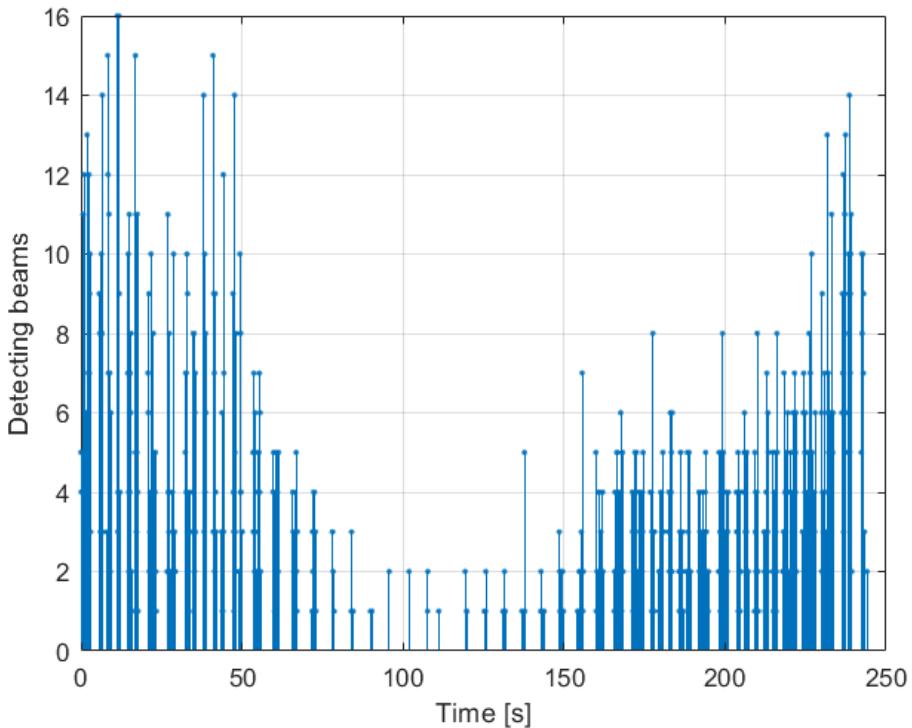


Figure 45: Drone detections with number of beams, *Sweeping*

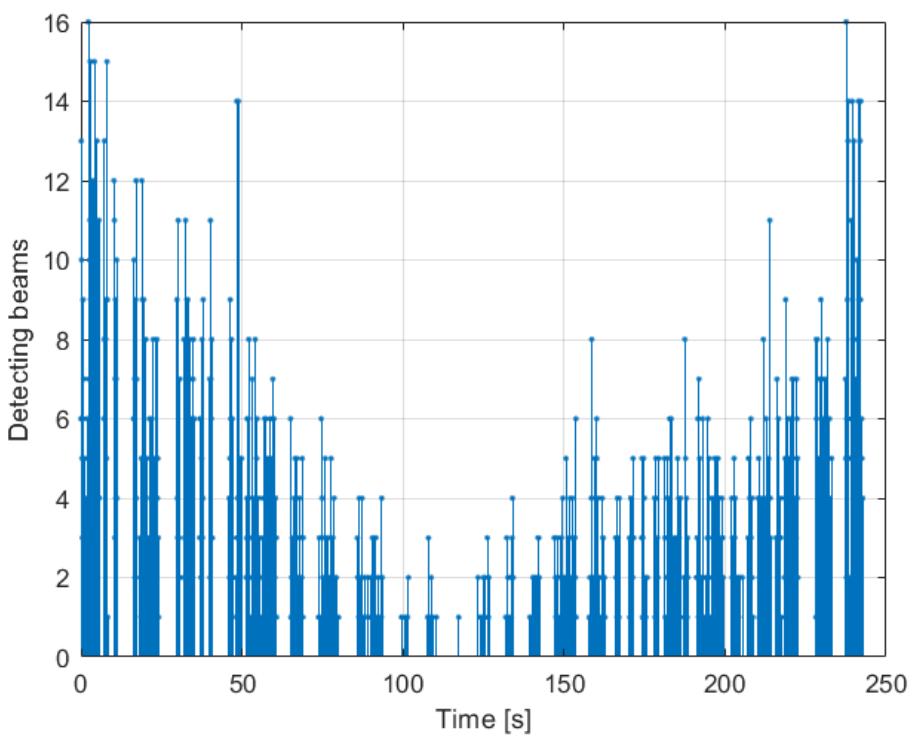


Figure 46: Drone detections with number of beams, *Swinging*

7.4 Analytical Comparison of Motion Patterns

In the previous subsections, the focus has been on comparing just two out of the four motion patterns described in the document. *Sweeping* was chosen as it represents the standard motion type and is used as a benchmark to determine if implementing a different type of pattern has benefits or not. It has been compared to the *Swinging* motion, which came out to be the optimal choice based on graphical analysis in previous sections, for what regards the accuracy of the Extended Kalman Filter estimate, conclusion that is further supported by analytical data gathered in the Tab.1.

Table 1: Comparison of Motion Patterns

	<i>Sweeping</i>	<i>Swinging</i>	<i>SlowingDown</i>	<i>Stopping</i>
Number of detections	2564	3182	3615	4871
Time between detections				
Minimum (s)	0.0717	0.0650	0.0637	0.0665
Maximum (s)	8.21	6.95	7.29	10.6
Mean (s)	0.409	0.303	0.286	0.208
Uncertainty on position	4.0406	3.4503	3.639	4.9169
Uncertainty on speed	0.67555	0.61006	0.62332	0.58151
3D Estimate Error (m)	3.23	1.44	1.56	1.50

To clarify, the reported uncertainty values in Tab.1 pertain to the x -axis and represent averaged values. For what regards the voice '3D Estimate Error', it refers to the Euclidean distance computed using the following formula:

$$d = \sqrt{(X_{\text{EKF}} - X_{\text{DRONE}})^2 + (Y_{\text{EKF}} - Y_{\text{DRONE}})^2 + (Z_{\text{EKF}} - Z_{\text{DRONE}})^2}$$

The 'EKF' subscript refers to the points obtained from the Extended Kalman Filter (EKF), which includes both predictions and updates, while the 'DRONE' subscript relates to the data from the ArduPilot logs, directly taken from the drone's GPS.

Going deeper in the analysis, the table provides a comparison of the four different motion patterns evaluated based on several metrics that are crucial for assessing the quality of tracking, particularly in estimating the drone's path using an Extended Kalman Filter (EKF).

- Uncertainty on Position: *Swinging* has the lowest uncertainty in position (3.4503), indicating that the drone's path is most accurately tracked when it follows a Swinging motion pattern. This low uncertainty is critical for reliable tracking, making Swinging superior in terms of position estimation accuracy.

- 3D Estimate Error: the 3D estimate error for *Swinging* is only 1.44 meters, the lowest among all the motion patterns. This suggests that the EKF is most effective at estimating the drone’s trajectory when it follows a *Swinging* pattern, leading to more precise and accurate tracking.
- Uncertainty on Speed: although the differences in speed uncertainty are relatively minor across the patterns, *Swinging* still shows a slightly better performance (0.61006) compared to other motions like Sweeping and *Slowing – Down*. This further reinforces *Swinging* as the pattern with more consistent and reliable speed estimates.
- Time Between Detections: *Swinging* has a mean time between detections of 0.303 seconds, which is reasonably short, allowing for frequent updates and maintaining a good balance between detection frequency and accuracy. This contributes to maintaining high-quality tracking over time. In contrast, while Stopping and Sweeping have some metrics that might suggest effectiveness in certain aspects (like the number of detections or the mean time between detections), their higher uncertainties in position and larger 3D estimate errors (especially in the case of Stopping) indicate that these motion patterns are less reliable for precise path estimation.

As previously highlighted, *Swinging* comes out to be the best motion pattern for ensuring high-quality tracking and accurate path estimation of the drone by the EKF. Its lower uncertainties in both position and speed, combined with the minimal 3D estimate error, make it the most reliable and precise motion pattern among those compared.

8 Conclusion

The research presented in this document focused on the optimization of drone tracking using LiDAR sensor, specifically exploring the efficacy of different motion patterns: *Sweeping*, *Swinging*, *Slowing – down*, and *Stopping*.

Among the strategies tested, two provided the best results:

- *Swinging*: This motion pattern is most effective when focusing on the accuracy of the estimate produced by the Extended Kalman Filter, as shown in Tab.1.
- *Stopping*: This strategy works best when the priority is to maximize the number of detections.

However, it is important to keep in mind the main goal of this project: improving the tracking system's capabilities while ensuring the protection of people's privacy and safety. The idea is that more accurate tracking allows for better control of drone operations, which reduces the risk of invading privacy or causing harm.

Considering this, the *Swinging* motion pattern is the best choice. This pattern improves detection by focusing the LiDAR's scanning on the area around the drone, which leads to better tracking accuracy and less uncertainty in position estimation.

The use of the PTU-E46 Pan-Tilt Unit enhances this approach by allowing adaptive scanning, which is essential for keeping the drone in view during flight. The ROS-based system manages data flow between nodes (Filter, Controller, EKF), leading to better detection, clustering, and state estimation from LiDAR point clouds.

A key point from this study is how the Extended Kalman Filter improves the accuracy of drone tracking. By constantly refining position estimates based on centroid measurements, the filter reduces uncertainty over time. This has been confirmed through field testing, showing that the system is both precise and reliable in real-world conditions.

8.1 Future Research Directions

The research presented in this document lays a strong foundation for enhancing drone tracking capabilities using LiDAR sensors and specific motion strategies. However, there is significant potential for further advancements that can improve both the effectiveness and efficiency of the tracking system.

8.1.1 Adaptive Motion Strategies Based on Real-Time Environmental Data

One promising direction for future research is the development of adaptive motion strategies that can respond dynamically to real-time environmental conditions. Unlike the static motion patterns explored in this study (Sweeping, Swinging, Slowing-down, Stopping), adaptive strategies would allow the system to adjust its scanning behavior based on factors such as the drone's speed, trajectory, and surrounding obstacles. For instance, if a drone is detected moving rapidly in a cluttered environment, the tracking system could dynamically switch to a Slowing-down or Stopping pattern to ensure more precise measurements, while a Swinging or Sweeping motion might be used in open areas with fewer obstacles. This adaptability could enhance tracking accuracy, minimize resource consumption, and improve the system's overall robustness in diverse operational scenarios [6] [23].

8.1.2 Integration of Machine Learning Techniques for Predictive Tracking

Another area ripe for exploration is the integration of machine learning techniques into the tracking system. By leveraging historical data and patterns observed in drone movements, machine learning algorithms could be employed to predict future drone trajectories, thereby allowing the system to preemptively adjust its motion patterns and scanning focus. For example, a predictive model could identify common evasive maneuvers used by drones and enable the tracking system to anticipate these movements, improving the likelihood of maintaining continuous visibility and accurate tracking. Additionally, machine learning could be used to optimize the Extended Kalman Filter's parameters in real-time, further enhancing the accuracy of position estimates and reducing uncertainty [13] [21].

8.1.3 Exploration of Swarm Intelligence for Coordinated Drone Tracking

Future research could also explore the application of swarm intelligence to coordinate multiple tracking units, each equipped with LiDAR sensors and operating under a distributed control system. By leveraging principles of swarm intelligence, these units could communicate and collaborate in real-time, sharing data and adjusting their positions to maintain optimal coverage of the target drone. This approach could be particularly beneficial in scenarios where a single tracking unit is insufficient to maintain accurate tracking, such as in complex environments or when multiple drones are operating simultaneously. Swarm-based tracking could enhance the system's resilience, allowing for continuous tracking even if one or more units experience failure or signal loss [24] [14].

8.1.4 Integration of Additional Sensors to Improve Robustness

The inclusion of complementary sensors, such as thermal cameras, radar, or acoustic sensors, could be investigated to augment the LiDAR-based tracking system. These additional sensors could provide valuable data in conditions where LiDAR performance is limited, such as in low visibility or adverse weather. By fusing data from multiple sensors, the system could achieve greater robustness and reliability, ensuring consistent performance across a wide range of operational conditions [22].

8.2 Drone Neutralization Methods

As drone technology continues to advance, the development of effective and responsible neutralization methods becomes increasingly important to address potential safety threats posed by unauthorized or malicious drone operations. This section explores various neutralization techniques, as well as their ethical and legal implications.

8.2.1 Signal Jamming

Signal jamming involves the disruption of the communication link between the drone and its operator by emitting radio frequencies that interfere with the control signals. While this method can effectively neutralize a drone by causing it to lose control or return to its takeoff point, it raises significant ethical and legal concerns. Signal jamming can potentially interfere with legitimate communications, including emergency services, and may violate regulations set by telecommunications authorities. Furthermore, jamming may not be effective against drones operating autonomously or those with advanced anti-jamming capabilities. Future research could focus on developing more targeted jamming techniques that minimize collateral interference, as well as exploring legal frameworks that balance the need for security with the protection of communication rights [19] [3].

8.2.2 Spoofing Techniques

Spoofing involves sending false signals to a drone, tricking it into believing it is receiving legitimate commands from its operator. This method can be used to redirect the drone to a safe location or cause it to land. While spoofing can be highly effective, it also presents ethical challenges, particularly regarding the potential misuse of this technology to hijack drones unlawfully. Additionally, as drone manufacturers increasingly incorporate anti-spoofing measures, this method may become less effective over time. Future research could explore advanced spoofing techniques that overcome such countermeasures or focus on the development of regulations to govern the lawful use of spoofing technologies [17] [12].

8.2.3 Laser Interference

Using lasers to disable a drone's electronics or sensors is another potential neutralization method. High-powered lasers can be directed at a drone to damage its cameras, sensors, or communication systems, rendering it inoperable. However, the use of lasers poses significant safety risks, particularly

to people and property if the beam is not accurately targeted. Additionally, there are ethical concerns related to the potential for causing permanent damage to drones that may have been operating innocuously. Legal restrictions on the use of laser technology also vary widely across jurisdictions, complicating the deployment of this method. Research in this area could focus on developing more precise laser targeting systems, as well as studying the legal and ethical implications of their use [20] [10].

8.2.4 Kinetic Projectiles

Physically disabling a drone using kinetic projectiles, such as firearms or net-launching devices, is a direct and potentially effective approach to neutralization. This method, however, carries significant risks, including the potential for collateral damage, harm to bystanders, and unintended consequences if the drone is carrying hazardous materials. Additionally, the use of firearms in civilian areas is heavily regulated, and the deployment of kinetic projectiles may be legally restricted or require specialized authorization. Future research could explore safer kinetic alternatives, such as net-based systems that capture drones without causing destruction, as well as the development of automated targeting systems that minimize the risk of accidental harm [18] [7].

8.2.5 Ethical and Legal Implications

The development and deployment of drone neutralization methods must be carefully considered within the context of ethical and legal frameworks. The potential for these technologies to be misused, either by infringing on privacy rights or by causing unnecessary harm, necessitates the establishment of clear guidelines and regulations. Research could explore the creation of international standards for drone neutralization, ensuring that such actions are taken responsibly and with due consideration for human rights and public safety. Additionally, as drone technology evolves, continuous updates to legal frameworks will be required to address new challenges, such as the emergence of fully autonomous drones or swarms of drones that may require new approaches to neutralization [9] [8].

Bibliography

- [1] Tasnim Azad Abir et al. “Towards Robust Lidar-based 3D Detection and Tracking of UAVs”. In: *Proceedings of the Ninth Workshop on Micro Aerial Vehicle Networks, Systems, and Applications*. DroNet ’23. Helsinki, Finland: Association for Computing Machinery, 2023, pp. 1–7. ISBN: 9798400702105.
- [2] Dirk-Uwe Bartsch, Ilkay K Muftuoglu, and William R Freeman. *Laser pointers revisited*. 2016. URL: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4993644/>.
- [3] Roberto Di Pietro, Gabriele Oligeri, and Pietro Tedeschi. “Jam-me: Exploiting jamming to accomplish drone mission”. In: *2019 IEEE Conference on Communications and Network Security (CNS)*. IEEE. 2019, pp. 1–9.
- [4] Sedat Dogru and Lino Marques. “Drone Detection Using Sparse Lidar Measurements”. In: *IEEE Robotics and Automation Letters* 7.2 (2022), pp. 3062–3069.
- [5] Sedat Dogru and Lino Marques. “Pursuing Drones With Drones Using Millimeter Wave Radar”. In: *IEEE Robotics and Automation Letters* 5.3 (2020), pp. 4156–4163.
- [6] Wenqiang Du and Giovanni Beltrame. *LiDAR-based Real-Time Object Detection and Tracking in Dynamic Environments*. 2024. arXiv: 2407.04115 [cs.RO]. URL: <https://arxiv.org/abs/2407.04115>.
- [7] Kar Ee Ho. “EFFICACY IN LOW-COST KINETIC APPREHENSION COUNTER DRONE SYSTEM”. PhD thesis. Purdue University Graduate School, 2024.
- [8] John C Jarvis and WA Babcock. “The ethical debate of drone journalism: flying into the future of reporting”. In: *Research papers* 475 (2014).

- [9] Wing Commander UC Jha. *Drone Wars: Ethical, Legal and Strategic Implications: Ethical, Legal and Strategic Implications*. KW Publishers Pvt Ltd, 2014.
- [10] Sung-Geon Kim et al. “Review of intentional electromagnetic interference on UAV sensor modules and experimental study”. In: *Sensors* 22.6 (2022), p. 2384.
- [11] Cang Liang et al. “UAV Detection Using Continuous Wave Radar”. In: *2018 IEEE International Conference on Information Communication and Signal Processing (ICICSP)*. 2018, pp. 1–5.
- [12] Georgia Lykou, Dimitrios Moustakas, and Dimitris Gritzalis. “Defending airports from UAS: A survey on cyber-attacks and counter-drone sensing technologies”. In: *Sensors* 20.12 (2020), p. 3537.
- [13] Ihab S Mohamed. “Detection and tracking of pallets using a laser rangefinder and machine learning techniques”. PhD thesis. European Master on Advanced Robotics+(EMARO+), University of Genova, Italy, 2017.
- [14] Sinan Oğuz et al. “An Open-Source UAV Platform for Swarm Robotics Research: Using Cooperative Sensor Fusion for Inter-Robot Tracking”. In: *IEEE access* 12 (2024), pp. 43378–43395.
- [15] Mattes Ohlenbusch et al. “Robust Drone Detection for Acoustic Monitoring Applications”. In: *2020 28th European Signal Processing Conference (EUSIPCO)*. 2021, pp. 6–10.
- [16] Alain Quentel. “A scanning LiDAR for long range detection and tracking of UAVs”. Theses. Normandie Université, Jan. 2021.
- [17] PD Rathika, BG Jaya Suriya, et al. “Review on Anti-Drone Techniques”. In: *NVEO-NATURAL VOLATILES & ESSENTIAL OILS Journal— NVEO* (2021), pp. 3498–3508.
- [18] Cyril Robbe, Alexandre Papy, and Nestor Nsiampa. “Using Kinetic Energy Non-Lethal Weapons to Neutralize Low Small Slow Unmanned Aerial Vehicles”. In: *Human Factors and Mechanical Engineering for Defense and Safety* 2 (2018), pp. 1–15.
- [19] Ondřej Šimon and Tomáš Göththans. “A survey on the use of deep learning techniques for uav jamming and deception”. In: *Electronics* 11.19 (2022), p. 3025.
- [20] Jacob Tewes. “Lasers, jammers, nets, and eagles: Drone defense is still illegal”. In: *Available at SSRN 3304914* (2017).

- [21] AN Wilson et al. “Embedded sensors, communication technologies, computing platforms and machine learning for UAVs: A review”. In: *IEEE Sensors Journal* 22.3 (2021), pp. 1807–1826.
- [22] Yifan Zhang, Junhui Hou, and Yixuan Yuan. “A comprehensive study of the robustness for lidar-based 3d object detectors against adversarial attacks”. In: *International Journal of Computer Vision* 132.5 (2024), pp. 1592–1624.
- [23] Zixu Zhao et al. “Efficient and adaptive lidar–visual–inertial odometry for agricultural unmanned ground vehicle”. In: *International Journal of Advanced Robotic Systems* 19.2 (2022), p. 17298806221094925.
- [24] Fangcheng Zhu et al. “Decentralized lidar-inertial swarm odometry”. In: *arXiv preprint arXiv:2209.06628* (2022).

Acknowledgements

Desidero esprimere la mia profonda gratitudine a coloro che sono stati di fondamentale importanza per me durante il percorso che mi ha portato al raggiungimento di questo prestigioso traguardo.

In primo luogo, desidero ringraziare i miei genitori, Loredana e Massimo, per avermi dato piena libertà, a partire dalla scelta di questo nuovo ambito in cui mi sono specializzato negli ultimi due anni, fino all'intero percorso di studi. Mi hanno sostenuto quando ne avevo bisogno e mi hanno elogiato nei momenti opportuni, ed è anche grazie a loro se ho potuto affrontare questi anni con serenità, raggiungendo risultati gratificanti.

Un ringraziamento speciale va a mio fratello Luca, al quale, anche se non lo esprimo spesso, voglio un bene immenso. Sono felice dell'evoluzione del nostro rapporto negli ultimi anni, sempre più caratterizzato da quel legame fraterno che, ne sono certo, sarà fondamentale nelle nostre vite. (Detto questo, anche se sei più alto, comando ancora io!)

Grazie poi ai Boys: Bani, Kriol, Fec, Fil e Pit. Siete stati compagni di serate spensierate, fatte di risate e, a volte, di discorsi più profondi. Ci vorrebbe un libro con il doppio delle pagine di questa tesi per racchiudere tutti i ricordi dei momenti che abbiamo condiviso in questi anni. Grazie anche alle girlssss, che, fortunatamente, nei momenti di necessità riescono sempre a riportarci un po' di calma.

Un ringraziamento speciale ad Ale :), che, nonostante sia entrata nella mia vita da poco tempo, lo ha fatto come un uragano, sconvolgendola in meglio. Il legame che ci unisce è davvero forte, e spero con tutto il cuore che il nostro rapporto non cambi mai, che i prossimi anni siano belli come questi primi mesi. Te se ama.

La mia gratitudine si estende anche alla mia grande famiglia, agli zii, e ai cugini, con i quali ho condiviso momenti indimenticabili dai miei primi anni di vita. In particolare grazie a mio cugino Lorenzo, che è sempre stato al mio fianco come se fosse un secondo fratello e con cui ho condiviso un'infinità di momenti felici nel corso di questi ormai 23 anni che siamo insieme.

Non posso non ringraziare i miei nonni, che porto sempre nel cuore. Penso

spesso a loro e sono certo che, da lassù, mi abbiano aiutato nei momenti in cui ne avevo più bisogno.

Grazie di cuore a Lazzino e Lori. Nonostante il passare degli anni, sono contento di aver mantenuto un legame speciale, e spero davvero che continui così anche in futuro.

Grazie poi al mio squadrone dei colleghi di meccatronica, quante fatiche abbiamo superato insieme, lo testimoniano direttamente i vocali del sior Fabio sul nostro gruppo: sono convinto avremo modo di coltivare il nostro rapporto ulteriormente negli anni a venire.

Un ringraziamento sincero al mio tutor in Portogallo presso l'ISR, Sedat, che mi ha guidato con pazienza (c'è da ammettere veramente molta) e competenza in ogni fase della realizzazione della mia tesi, dall'ideazione del lavoro alla messa in atto finale: lavoro di cui sono davvero fiero che mi ha portato alla stesura del mio primo paper scientifico.

Infine, desidero ringraziare i miei amici dell'Erasmus in Portogallo, che rimarranno per sempre nel mio cuore come un ricordo bellissimo, grazie a Cosimo, Alice, Fra, Vale e Lallush.