Methodology for Precision Landing of Unmanned Aerial Vehicles on a Mobile Base

Luciano Bonzatto Junior

Dissertation presented to the School of Technology and Management of Bragança to obtain the master's degree in Informatics within the scope of the double degree program with the Federal University of Technology – Paraná

Work oriented by:

Prof. Dr. José Luís Sousa de Magalhães Lima

Prof. Dr. Marco Aurélio Wehrmeister

Bragança

March 2024

Methodology for Precision Landing of Unmanned Aerial Vehicles on a Mobile Base

Luciano Bonzatto Junior

Dissertation presented to the School of Technology and Management of Bragança to obtain the master's degree in Informatics within the scope of the double degree program with the Federal University of Technology – Paraná

Work oriented by:

Prof. Dr. José Luís Sousa de Magalhães Lima

Prof. Dr. Marco Aurélio Wehrmeister

Bragança

March 2024

Dedication

I would like to express my sincere gratitude to my friends, my parents and my partner for their unconditional support. Without the help and encouragement of each one of you, this journey would not have been possible. It is with great affection and gratitude that I dedicate this dissertation to all of you, as a symbol of my eternal gratitude.

Abstract

The integration of heterogeneous robotic systems is a constant topic today as a promising strategy to overcome the inherent limitations of each system. With this in view, this study explores the development of a precision landing system for Unmanned Aerial Vehicles (UAVs), designed to land autonomously on static and moving targets. To achieve this, a detailed analysis of aspects of the system is first carried out, such as the definition of the fiducial marker, the control architecture, and the definition of gains, followed by the development of the code, which includes the architecture and the interface with an operator. After development, tests begin which are divided into two stages: the first verifies the UAV's ability to identify and follow moving targets, and the second consists of precision landing experiments in different scenarios. The results of the investigation indicate that the combination of a complete PID controller with Aruco markers is more effective, which is why they were selected for the development of the system. Tracking tests have proven the driver's ability to guide the UAV to autonomously follow a UGV, although it presents difficulties with high angular speeds. On the other hand, autonomous landing tests showed high efficiency in constant speed scenarios but revealed some failures in situations with sudden changes and requests to the rotation driver.

Keywords: Cooperative Robotics; Unmanned Aerial Vehicles; Unmanned Ground Vehicles; Precision Landing

Resumo

A integração de sistemas robóticos heterogêneos é um tópico constante atualmente como uma estratégia promissora para ultrapassar as limitações inerentes a cada sistema individualmente. Com isso, este estudo explora o desenvolvimento de um sistema de pouso de precisão para Veículos Aéreos Não Tripulados (UAVs), destinado a aterragens em alvos estáticos e em movimento autonomamente. Para isso, primeiro é feita uma análise detalhada de aspectos do sistema, como a definição do marcador fiducial, da arquitetura de controle e definição de ganhos, seguido do desenvolvimento do código, que inclui a arquitetura e a interface com o operador. Após o desenvolvimento, inicia-se os testes que se dividem em duas etapas: a primeira verifica a capacidade do UAV de identificar e seguir alvos em movimento, e a segunda consiste em experimentos de pouso de precisão em diversos cenários. Os resultados da investigação indicam que a combinação de um controlador PID completo com marcadores Aruco é mais eficaz, razão pela qual foram selecionados para o desenvolvimento do sistema. Os testes de rastreamento comprovaram a habilidade do controlador em orientar o UAV para seguir autonomamente um UGV, embora apresente dificuldades com velocidades angulares elevadas. Por outro lado, os testes de pouso autônomo mostraram alta eficiência em cenários de velocidade constante, mas revelaram algumas falhas em situações com mudanças bruscas e desafiadoras para o controlador de rotação.

Palavras-chave: Robótica Cooperativa; Veículos Aéreos Não Tripulados; Veículos Terrestres Não Tripulados; Pouso De Precisão

Contents

List of Tables

List of Figures

Acronyms

ESTiG Escola Superior de Tecnologia e Gestão.

IPB Instituto Politécnico de Bragança.

CeDRI Research Centre in Digitalization and Intelligent Robotics.

AGV Automated Guided Vehicle.

AMR Autonomous Mobile Robots.

GUI Graphical User Interface.

LiDAR Light Detection and Ranging.

UAV Unmanned Aerial Vehicle.

UGV Unmanned Ground Vehicle.

ROS Robot Operating System.

Chapter 1

Introduction

The intelligent production environment has been an emerging theme about the ascension of the Industry 4.0 concept in multiple application domains, whether in the logistical, agricultural, or industrial fields. The automation of tasks by mobile robots makes it possible to minimize human intervention, optimize processes, and increase the productivity of the sector. In this sense, the use of aerial robots and ground robots is constantly advancing in a diversified range of applications. However, using a single type of mobile robot often restricts the range of applications due to the limitations of each of the architectures [1].

The use of heterogeneous robotic systems emerges as a promising solution since this strategy aims to exploit the specific advantages of each system while compensating for their deficiencies. While an aerial robot can offer extensive reconnaissance and mobility capabilities, a ground robot can perform tasks that require greater stability and physical strength, such as transporting large amounts of weight. The key to the success of these systems resides in effective integration and coordination, ensuring that the advantages of each architecture complement the limitations of the others [2].

For cooperation between Unmanned Ground Vehicles and Unmanned Aerial Vehicles to be efficient, several key steps need to be carried out with high reliability. Communication between UGVs and UAVs is essential to synchronize operations and exchange critical information. And the control of movement and trajectory, including the ability to avoid obstacles, ensuring safe and efficient navigation in different environments.

Additionally, more specific steps depending on the application are also required, for example, for precision agriculture, cooperation may be involved with the refueling of UAV materials, in this context, it is crucial that the UAV can dock and land accurately on the UGV, ensuring that refueling takes place smoothly. This integrated approach is not only limited to precision agriculture, but also proves useful in many other applications, such as inspections, or in the military.

Solutions revolving around mobile docking stations have been presented in the literature, allowing the aircraft to land safely, and charge or replace batteries or loads to extend the system's autonomy. Precision landing becomes even more challenging if the UAV must track the moving ground target while simultaneously performing landing operations, making it a major challenge to develop solutions for precision landing operations on mobile docking stations [3].

The motivation for developing systems that allow UAVs to land autonomously on moving UGVs is based on practical applications and potential benefits, opening up new horizons in different sectors. From rescue and humanitarian operations, where speed and efficiency are crucial to saving lives in disaster areas, to continuous surveillance missions in military contexts. In logistics and delivery, the ability to land on moving vehicles can mean optimizing routes and reducing waiting times, especially in dense urban environments or remote locations. These advances not only promise to increase the autonomy and efficiency of operations involving UAVs and UGVs, but also provide the basis for innovations in areas including environmental research, space exploration and much more, representing a significant milestone in the progress of robotics and automation.

Especially as it is a recent field of research that has great potential for solving problems related to the autonomy and flight range of UAVs in different application domains, research into the autonomous landing of multi-rotor UAVs has gained prominence, given the complexity and importance of the safety of the aircraft and other parties during the procedure.

1.1 Objectives

The main focus of this project is developing an advanced precision landing methodology, capable of acting on both static and moving targets and tracking moving targets. The aim of using fiducial markers and PID controllers is to establish a robust and efficient system. For the initial testing phase, a commercial UAV (Unmanned Aerial Vehicle) will be used to ensure the safety and stability of the system. However, the long-term goal is to adapt this control system for non-commercial drones operating with MAVROS (MAVLink to ROS), which requires a code architecture prepared for this transition.

The specific objectives of the project are as described below:

1. Detailed definition of system aspects: fiducial marker and control architecture.
2. Development of a technique for tracking a moving target.
3. Development of a technique for precise landing on moving and static targets.
4. Construction of a control system using ROS, which is easily adaptable for use with MAVROS.
5. Verification of the reliability of the landing and target tracking system.

1.2 Document Structure

This work has been divided into five chapters as follows:

Introduction: chapter 1 provides the basis for the research, highlighting the objectives and the importance of the study in the context of autonomous robots and automated guided vehicles. The structure of the document is presented, preparing the reader for the technical and theoretical details that will follow.

State of the Art and Study of Tools: chapter 2 is a review of the literature and existing tools in the field of autonomous robotics used during development, this chapter

addresses specific research areas, including unmanned aerial and ground vehicles, cooperative robotics, the Robot Operating System and virtual environments such as the Gazebo simulator. Technical aspects such as PID controllers, state machines, camera analysis, and marker detection are also discussed.

Development: chapter 3 details the path taken to develop the control system, as well as describes the complete control system. It shows how the comparative analysis of fiducial markers and controllers was carried out and, finally, the final architecture of the code is presented, including the control system, user interface, and definition of the controller gains.

Results: chapter 4 is dedicated to the results obtained in the research. The focus is on moving target tracking results, including targets in constant linear and angular motion, and precision landing results on stationary targets and in different motions. Detailed analysis of these results is crucial to understanding the effectiveness of the proposed techniques and approaches.

Conclusion and Future Work: The final chapter, chapter 5 summarizes the main conclusions of the research and discusses their implications. In addition, directions for future work are suggested, indicating how the research can be expanded or improved. This chapter is essential for understanding the contribution of the research to the field of autonomous robotics and for guiding future investigations.

Chapter 2

State of art and Study of tools

2.1 Autonomous Mobile Robots and Automated Guided Vehicles

As defined in [4], Autonomous Mobile Robots (AMRs) perceive their surroundings and make intelligent decisions, contributing significantly to sectors like healthcare, industry, and civil engineering. These robots have evolved over the past 30 years, leading to significant advancements in locomotion, perception, localization, mapping, ego-motion tracking, and dynamic navigation [4].

Autonomous Mobile Robots represent a significant advancement in robotic technology, evolving from their predecessors, Automated Guided Vehicles (AGVs). While AGVs, as described in [5], have been instrumental in logistics and manufacturing due to their ability to follow predefined paths and require external guidance, AMRs mark a more advanced stage in robotic autonomy. Equipped with sophisticated sensors and algorithms, AMRs allow dynamic navigation and intelligent decision-making capabilities [5]. This shift marks a significant development in how robotic systems are utilized across various industries, ranging from civil engineering to high-risk work.

2.1.1 Research Areas

AMRs consist of several integral components that enable them to perform complex tasks autonomously. These components include advanced sensor arrays for environmental perception, autonomous navigation capabilities, and sophisticated algorithms for decision-making.

Perception

Perception in AMR involves interpreting and understanding the environment using various sensors. This process includes collecting and processing sensory data to identify objects, obstacles, and relevant environmental features. According to [4], perception is a fundamental aspect of AMRs, enabling safe and effective interaction in diverse settings.

As defined in [4], AMRs perceive their surroundings and make intelligent decisions, contributing significantly to sectors like healthcare, industry, and civil engineering. These robots have evolved over the past 30 years, leading to significant advancements in locomotion, perception, localization, mapping, ego-motion tracking, and dynamic navigation [4].

Navigation

Navigating AMRs entails determining and following a path to a target location while avoiding obstacles. This process combines sensory data with algorithms for real-time decision-making. Navigation, including dynamic path planning and decision-making in response to environmental changes, is a complex task highlighted in [4].

The challenge of path planning, especially in dynamic environments, is addressed in recent research. For instance, a focus on multi-agent policy learning and improved algorithms for efficient path planning is described in [6].

An example of advanced navigation is seen in the development of a mobile robot equipped with gas sensors and a LIDAR device for scanning areas and detecting gas leaks while navigating and avoiding obstacles [7]. Additionally, optimizing motion planning for

speed and safety, particularly in semi-structured scenarios, is another research focus, as detailed in [8].

Significant research in AMRs includes developing machine learning systems for obstacle detection and avoidance. This area involves using artificial neural networks and FPGA distributive processing algorithms to enhance prediction speed and accuracy, as highlighted in [9]. The study [10] uses object detection to identify and classify objects as discrepant from a location point of view, thus improving location accuracy in dynamic environments.

Localization

Localization, the process by which an AMR determines its position within an environment, is crucial for accurate navigation and task execution. Authors in [4] emphasize the importance of localization for the precision and reliability of AMRs.

Innovative approaches in localization in Industry 4.0 factories are being developed. For instance, an indoor localization system based on the Extended Kalman Filter (EKF) and ArUco markers is described in [11]. This method involved testing different innovation methods for observations within the EKF and was validated using a real robot in a factory mockup.

Additionally, a machine learning approach to the localization problem is proposed in [12], where the relative pose of an onboard camera concerning fiducial markers is estimated using algorithms such as the Random Forest Regressor. This method presents high accuracy and does not require explicit knowledge of the exact positions of fiducial markers, showcasing the potential of AI in enhancing traditional localization techniques.

Cognition

Cognition in AMRs refers to the ability to process information, make decisions, and learn from experiences. This includes using advanced algorithms and artificial intelligence to interpret data, predict outcomes, and adapt to new situations. Cognition allows AMRs

to operate autonomously and make intelligent decisions based on their perception of the environment, as suggested in [4].

In conclusion, the evolution of AMRs from AGVs represents a leap in robotic capabilities, driven by advancements in sensor technology, autonomous navigation algorithms, and machine learning. While challenges remain, particularly in dynamic path planning and real-time responsiveness, the future of AMRs is promising, with expanding applications across numerous sectors. In this area, the solutions for these challenges are fostered by robotics competitions worldwide, incentivizing researchers to develop technology to address these problems in the industry and society. Some of these challenges can be seen in [13]–[15]. They can also foster education, as seen in [16]–[19].

AMRs are a subcategory of Unmanned Ground Vehicles (UGVs). There is also another category of autonomous vehicles which is Unmanned Aerial Vehicles (UAVs). Recent research works are described below under these categories.

2.1.2 Unmanned aerial vehicle

Unmanned Aerial Vehicles (UAVs) serve as essential robotic platforms extensively employed in diverse research domains, meeting demands in areas such as agriculture [20], infrastructure inspection and monitoring [21], goods delivery [22], cooperative robotics [23], among others. The versatility of UAVs, particularly multi rotors, is attributed to their ability to hover precisely, their exceptional maneuverability, and their reasonable payload capacity for transportation purposes [24].

There are also studies designed to test alternative ways of maintaining the vehicle's stability in adverse situations. In [25], a Fast Control Allocation (FCA) is used on a Tilt-Rotor quadrotor, in which the motor angles are variable, to improve wind safety.

In summary, UAVs play a vital role in various research fields due to their unique capabilities. Nevertheless, the restricted flight duration of multirotor UAVs remains a significant impediment to achieving prolonged and autonomous operations.

2.1.3 Unmanned Ground Vehicle

Recent research in UGVs has made significant strides across various domains, from civil infrastructure and autonomous exploration to sensor technology, intelligent traffic systems, and adaptive control strategies.

Ali et al. (2023), in "Enhanced Application of UGVs in Civil Infrastructure", highlight the increasing use of UGVs in civil infrastructure, emphasizing the need for comprehensive reviews of applications and methodologies, including data sensing techniques and analytics [26]. Concurrently, Zuo et al. (2023) in "Real-Time Action Planning for UGVs" address the challenge of autonomous UGV exploration, introducing novel approaches for real-time action planning in 3D spaces, significantly improving coverage efficiency and performance [27].

Lu et al. (2024), in "Adaptive Control of UGVs", propose an adaptive heading tracking control strategy for UGVs with variable wheelbases, focusing on optimizing driving performance under different conditions and incorporating robust control methods for enhanced maneuverability and stability [28]. Lastly, Szpaczynska et al. (2024) in "Running Gear Mobility in UGVs" present a method for assessing the mobility of light UGVs with rubber-tracked running gears, providing insights into the design of more efficient running gears for overcoming terrain obstacles [29].

Collectively, these studies underscore the rapid evolution of UGV technology, marked by innovative approaches in computational methods, sensor technologies, and control strategies. The integration of these advancements promises to expand the capabilities and efficiency of UGVs, facilitating their broader adoption in diverse applications.

2.1.4 Cooperative robotics

Remote operations offer increased autonomy and decision-making capabilities, particularly suited for tasks allowing the simultaneous operation of multiple robots over extended periods. In this sense, there are several limitations to using only one robotic architecture, which promotes the use of multiple architectures cooperating to resolve the same task.

Consequently, the main advantage of cooperative heterogeneous systems, as in the use of UAVs and UGVs, consists in combining the operational characteristics of each architecture to maximize the resolution of tasks in different application domains, be it in the field of precision agriculture [30], logistics inspections [31], industrial [32], search and rescue [33] or military [34].

Multi-rotor UAVs have operational advantages around their ability to hover with precision and move with a high degree of maneuverability [35]. But a notable challenge for this architecture is their limited flight time, which averages only 20 to 30 minutes [36], making them unviable for operations of long duration and consequently long range [35]. In this sense, several works have been explored to prolong the flight time of multirotor UAVs. For this, autonomous precision landing systems have been presented as one of the main research topics related to recharging or replacing UAV batteries [37].

UGVs, in contrast, are capable of transporting a large amount of weight, have high operating autonomy and can be used as mobile landing platforms [38]. This allows the multi-rotor UAVs to land and take off safely on top of UGVs, thus allowing it to increase the UAV's operating endurance and minimize the need for human intervention [37].

An interesting review is presented by Ding et al. [39], which describes various applications involving cooperation between Unmanned Aerial Vehicles and Unmanned Ground Vehicles. As part of the [39] research, the complexity of the precision landing procedure is elucidated, which involves a preliminary step where the UAV must follow the UGV before proceeding with the landing procedures. In another article [40], there is an overview of robotic cooperation work applied to precision agriculture, including the use of various robotic architectures, including cooperation between UAVs and UGVs.

2.2 Land approach techniques

The autonomous landing of multi-rotor UAVs on a moving target is a challenge within the context of cooperative robotics. In this sense, the robustness of the UAV system is characterized in the order of greatest importance, considering that the environment of

cooperative operation between the UAV and the mobile platform can present different types of disturbances related to its environment of employment, such as wind gusts, low visibility or even the dynamics of movement of the landing platform (as in maritime platforms) [3].

This operation involves different stages that are crucial to making a successful landing. The first stage consists of the approach between the UAV and the mobile platform. At this stage, GNSS systems are usually used to provide the UAV with global references that indicate the location of the landing platform when operated outdoors. On the other hand, when the cooperative operation is conducted indoors, and in this scenario there is no GPS signal, the main strategies aim to adopt computer vision techniques for tracking and locating static or mobile platforms, and may even make use of auxiliary sensors that are less affected outdoors, such as Light Detection and Ranging (LiDAR), infrared (IR) or even ultrasonic [41] [42].

This is followed by precision alignment of the aircraft with the mobile platform. This phase requires precise estimation of the landing platform's location, which is why UAVs usually use computer vision for this, such as the use of fiducial markers [43], IR sensors [44] or even thermal markers [45], but there are also techniques in the literature based on estimating the location of mobile platforms using Ultra Wide Band (UWB) waves [46].

Then proceed with the effective precision landing, which involves the aircraft maintaining alignment with the UGV while slowly decreasing its altitude. This phase requires robust control over the positioning of the aircraft by the landing platform location system in order to chase the landing platform while performing the simultaneous landing maneuver [41]. Finally, the UAV executes the land command in which it descends vertically to the ground and switches off the engines.

The paper [47] presents a comprehensive study on methods for identifying safe landing zones for UAVs. It explores techniques for both indoor and outdoor environments, distinguishing between static and dynamic landing zones. The study includes camera-based and LiDAR methods, as well as hybrid approaches. Challenges and future research directions are also discussed, highlighting the importance of this field in autonomous aviation.

Another work, [48], presents a strategy for a multi-rotor UAV to land vertically on a maritime platform in a simulated environment. In this work, PID control techniques were applied as well as the use of computer vision resources to detect fiducial markers and control the aircraft's altitude during the landing procedure on the constantly moving platform.

A vast study involving topics related to movement control, target detection, and communication between systems is presented and discussed in [49], where the use of a set of fiducial markers of different sizes is exposed to improve the dynamic tracking process between a UAV and a UGV. A PID control algorithm was designed which, in combination with the DJI M100 aircraft control system, demonstrated practical feasibility in UAV tracking and landing operations over the UGV in an outdoor environment.

2.3 Robot Operating System

The Robot Operating System, or ROS, is a middleware with a set of libraries and tools that simplify the process of creating and operating advanced robotic systems.

ROS is widely adopted in academic and industrial research and therefore has a large collection of packages already tested for various applications. This allows developers to create complex robotic systems by reusing existing components and integrating new ones [50].

In addition to creating robotic systems, ROS includes tools for testing these systems, including visualization and simulation. Such as Rviz, a powerful 3D visualization tool, allows developers to see sensor data and the state of the robot in a virtual environment [51]. Gazebo, on the other hand, offers an advanced robotics simulator, ideal for testing algorithms, robot design, and simulating complex scenarios in a safe and controlled environment [52]. Another essential tool is rosbag, which allows ROS messages to be recorded and played back, making it easier to debug and test systems. These tools are fundamental in the process of developing, testing, and implementing innovative and efficient robotic solutions [53].

ROS uses a system of specialized codes that exchange messages to execute a more complex system. The main components are:

- Packages: This is an organizational unit that contains the elements needed to provide functionality to a robotic system, such as configurations, initializations, libraries, and nodes.

- Nodes: Nodes are individual processes that perform specific tasks in a robotic system, for example, controlling actuators, reading sensors, processing data, and managing the robotic system. Nodes communicate with each other via topics and services.

- Topics: Topics are asynchronous communication channels that allow nodes to exchange messages. For example, a node that captures sensor data can publish this data in a topic, and other nodes can subscribe to this topic to receive this data.

- Services: Services are used for synchronous communication between nodes. A node can offer a service that provides a specific functionality and other nodes can request that functionality via the service.

However, despite the popularity and extensive adoption of the Robot Operating System, it presents some challenges and limitations that are important to consider. One of the biggest challenges of ROS is the absence of native support for real-time operations. In many robotic applications, especially those that require fast and accurate responses, such as industrial robotics or autonomous systems, the ability to operate in real-time is crucial. ROS was also not originally designed with strong security measures, which can make it vulnerable to cyber attacks or security breaches. Despite these challenges and limitations, ROS remains an extremely valuable tool in robotics, and many of these problems are being actively addressed by the community, especially in the transition to ROS 2, which aims to address many of the real-time, security, and scalability issues [54] [55].

2.4 Virtual environments

A virtual environment in robotics is a useful tool that offers a computerized simulation for developing robot algorithms. These simulations make it possible to test algorithms, behaviors, and navigation strategies in a safe and controlled manner, without the risk of damaging real equipment. They are also an economical way of conducting experiments since they reduce the costs associated with building and maintaining physical robots and facilitate remote collaboration between teams from different locations.

In addition to testing algorithms, virtual environments are widely used to teach robotics, as in [56], which provides a literature review of different robotic simulators with a graphical interface. Development of algorithms in complex or dangerous environments that would be difficult to replicate in the real world, such as the inspection of electrical towers or oil extraction platforms. They are frequently used to train machine learning and artificial intelligence algorithms, as they offer a consistent and repeatable platform for evaluating performance in a variety of scenarios[57] [58].

A widely used robotic simulation tool for testing in virtual environments is Gazebo Simulator [59]. It is a component of Gazebo, a collection of libraries designed to simplify the development of robotic applications, with a special focus on mobile robotics, although its versatility allows it to be used in many other applications.

This tool is commonly used to test and validate different models of robotic algorithms, and robotic kinematics, as well as enable the development of virtual environments dedicated to different types of robotic operations, the intention of which is to evaluate and analyze performance in simulation before its use in the real world. In addition, Gazebo natively integrates with ROS, simplifying the simulation testing process, as well as enhancing different types of simulation implementation for later transfer to a real physical system [60].

Several studies have been published in the literature based on the results obtained using the Gazebo simulation software. In [61], the development of a trajectory planning

model for multi-rotor UAVs is presented, taking into account an environment with aggressive environmental adversity caused by wind gusts. The test scenario was developed in the Gazebo virtual environment using the PX4-SITL and ROS interfaces. The performance shown by the work in question demonstrated superiority in dealing with wind disturbances and was highly suitable for deployment in real-world applications.

A quadcopter model, erle-copter, interfaced with Matlab and Simulink software is presented in [62]. In this work, Gazebo was used to analyze the performance of the kinematic model of a quadcopter multirotor UAV, to analyze the operational capacity of the aircraft in simulation. In [63], the cooperative use of a multi-rotor UAV and an Unmanned Ground Vehicle (UGV) in search and rescue operations was explored, using the Gazebo simulator as a tool to validate the positioning, communication, and coordination tasks between the multiple robotic platforms.

2.5 PID controller

In the realm of control theory, the Proportional-Integral-Derivative (PID) controller is a testament to both time-tested reliability and widespread applicability. Despite the vast and well-developed landscape of control theory, encompassing many control approaches and sophisticated tuning methods, PID control remains a cornerstone in both industrial and academic settings.

PID control, characterized by its simplicity and effectiveness, has established itself as a fundamental tool in the control systems arsenal. The PID controller operates on a straightforward principle: it calculates an error value as the difference between a desired setpoint and a measured process variable. This error is then used to compute a corrective action by applying proportional, integral, and derivative terms, which can be tuned to optimize the system's performance.

The literature is replete with various methodologies to tune these controllers, and software tools like MATLAB have further streamlined this process, allowing for more efficient and accurate tuning. The advancements in control strategies have led to more

complex systems, such as Fractional-Order PID (FOPID) controllers and Nonlinear PID (NPID) controllers. As discussed in various scientific papers, these advanced controllers offer more flexibility and can be more effective in specific specialized applications.

2.5.1 Application cases

For instance, in the realm of electrical energy transmission, researchers have explored the use of FOPID controllers to maintain system stability amidst low-frequency oscillations (LFOs), a critical requirement for interconnected power systems. The increased number of design parameters in FOPID controllers allows for finer control, albeit at the cost of more complex tuning processes. This is highlighted in the work on adaptive power system control methodologies, combining FOPID with fuzzy logic systems for dynamic tuning, significantly improving control performance in multi-machine interconnected power systems [64]. Similarly, FOPID controllers have found applications in the medical field, specifically in automatic blood pressure regulation. The optimization of FOPID parameters using genetic algorithms, as demonstrated in the management of blood pressure for post-operative patients, showcases the controller's effectiveness in critical healthcare applications [65].

The development of NPID controllers for soft dielectric elastomer actuators in soft robots is another example of the evolution of PID control. These controllers address the nonlinear characteristics of such systems, fine-tuning their parameters through iterative optimization algorithms and practical experiments [66].

Moreover, PID controllers have been successfully applied to optimize DC-DC buck converters, with improvements like the Improved Sine Cosine Algorithm (ISCA) enhancing their performance and robustness against disturbances and uncertainties [67].

Despite these advancements and the emergence of more complex control strategies, PID controllers continue to be widely used in industry and academia. Their enduring popularity stems from their simplicity, robustness, and proven track record in various applications. While the field of control theory continues to evolve, introducing more

sophisticated and specialized controllers, the PID controller remains a fundamental and highly relevant tool in theoretical and practical applications. One example is displayed in paper [68]. The authors described methods like the Internal Model Control (IMC) and the second-order Bessel prototype employed to tune speed and position controllers. These methodologies have been tested across PID controller architectures, including PID, Cascade, and feedforward combinations, demonstrating their versatility and effectiveness.

2.5.2 Anti-windup

An anti-windup filter is an important feature in control systems that use Proportional-Integral-Derivative controllers. The term windup refers to the problem that occurs when the integrative part of the PID controller accumulates a huge error during periods when the controller's output is saturated.

Saturation of a controller occurs when the output of the controller exceeds the capacity of the actuator, so the error of the integrative term increases indefinitely but the actuator can not match it. For example, when controlling a motor, if the motor is already rotating at its maximum speed, any additional command to increase the speed is useless but the integrative term continues to increase.

The windup problem occurs when the system stops demanding a high performance. The integrative term needs several iterations to eliminate the accumulated error, so the controller continues to send high values although it no longer requires them. This can cause oscillations and instability in the system.

To ensure the efficiency and stability of systems, an anti-windup filter is applied which adjusts the integrative term of the controller to reduce the error accumulation when the output is saturated. There are various techniques for implementing the anti-windup filter, such as the integrator freeze, where the calculation of the integrative term is paused during saturation, or the feedforward method. This helps to avoid excessive responses and oscillations, contributing to smoother and more precise control.

In [69] an analysis is made, both analytical and experimental, of three different filtering

methods, back-calculation, Incremental algorithm, and Error recalculation. Considering the stability, integrating and unstable processes with dead time subjected to saturation of the actuator.

2.6 State machine

A state machine is a conceptual model used in computer science and engineering to describe a system that can be in one of several states and how the system transits between them. Each state is a specific condition or situation of the system and the machine can change from one state to another in response to some external impulses, for example, user commands, sensor measurements, the passing of time, etc. These state transitions are governed by predefined rules.

State machines are widely used in various areas, including software development for user interfaces, games, industrial process control, and programming languages for text analysis and pattern recognition. Their relevance and application continue to be very pertinent today because they provide a clear and efficient way of managing complex systems that have multiple operational states and interactions.

2.7 Camera analysis

The standard RGB pinhole cameras used today have image distortions that must be taken into account when determining the relative position of the fiducial marker. To do this, it is first necessary to measure the scale of the distortion of the camera that will be used in the experiments [70].

Because of this, some works focus on defining methods for measuring and correcting camera distortions, such as [71], which proposes a method for correcting images focusing on radial and tangential distortions.

2.7.1 Radial distortion

Radial distortion causes straight lines to have a slight curvature, as seen in Figure 2.1, where the edges of the paper and the edges of the squares have a slight angle in relation to the straight lines in red [72].

This type of distortion has to do with the increase in the apparent size of the objects (Magnification) in relation to the optical center of the camera, so this distortion is more evident in regions further away from the center of the camera.

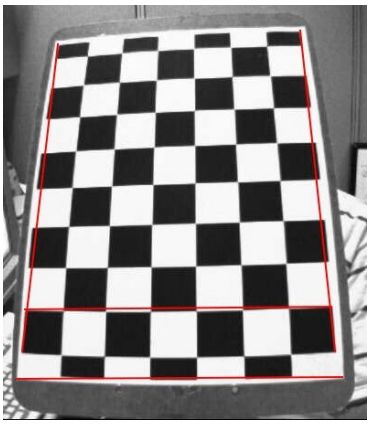

Figure 2.1: Radial distortion example (OpenCV)

This type of distortion can be classified in three different ways (Figure 2.2): Barrel distortion, in which the Magnification of the image decreases with distance from the camera's optical axis, Pincushion distortion, in which the Magnification of the image increases with distance from the optical axis or Complex distortion, which is a mixture of the other types [73].

2.7.2 Tangential distortion

Tangential distortion, on the other hand, occurs when the camera lens is not perfectly aligned and parallel to the sensor plane, resulting in the lengthening or shortening of straight lines that are near or tangent to the center of the image. This distortion can make objects that are close to the center of the image appear to be closer or further away

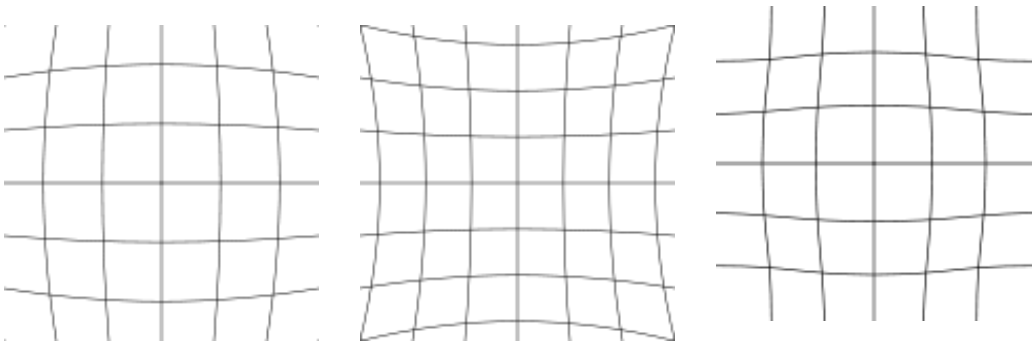

Figure 2.2: Classification of radial distortion. Barrel on the left, Pincushion in the center, and Complex on the right

from the camera than they are, impacting the accuracy of the representation of reality in the images captured [74].

2.7.3 Calibration

In this work, the camera was calibrated using the OpenCV tools [75], which use five coefficients to parameterize the images. Three parameters are used, k_1, k_2 and k_3, shown in the equations Equation 2.1 and Equation 2.2, where x and y are the coordinates of a point in the undistorted image, $x_{\text{radial distorted}}$ and $y_{\text{radial distorted}}$ are the coordinates of the same point in the radially distorted image, and r is the radial distance of the point (x, y) from the center of distortion.

For tangential distortion, two other parameters are used, p_1 and p_2, shown in the equations Equation 2.3 and Equation 2.4, where $x_{\text{tangential distorted}}$ and $y_{\text{tangential distorted}}$ are the coordinates of the point in the tangentially distorted image. These coefficients are organized in a matrix shown in Equation 2.6.

$$x_{\text{radial distorted}} = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \tag{2.1}$$

$$y_{\text{radial distorted}} = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \tag{2.2}$$

$$x_{\text{tangential distorted}} = x + (2p_1 xy + p_2(r^2 + 2x^2)) \tag{2.3}$$

$$y_{\text{tangential distorted}} = y + (p_1(r^2 + 2y^2) + 2p_2 xy) \tag{2.4}$$

In addition, the camera's intrinsic parameters, focal length (fx,fy), and optical centers (cx,cy), are measured and organized in a matrix shown in Equation 2.5.

$$\text{cameramatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{2.5}$$

$$\text{distortioncoeffs} = \begin{bmatrix} k_1 & k_2 & p_1 & p_2 & k_3 \end{bmatrix} \tag{2.6}$$

2.8 Marker detection

Computer vision-based localization tasks enable vehicles to determine their positions without relying on GNSS. This is particularly crucial for precision missions or operations in GNSS-denied environments. Therefore, computer vision is an alternative sensing method that aids robot localization. As stated in Coelho et al. [76], algorithms using computer vision through a camera can detect artificial features, such as fiducial markers, extract their characteristics, and estimate the position and relative orientation of the robot.

Fiducial markers are reference points easily distinguishable for object tracking, motion capture, augmented reality, and visual localization [77]. A wide range of fiducial marker packages is available, including open-source options. Some open-source packages are also

available to the robotics community as ROS packages, such as ARTag, ArUco markers, Artrack, etc. [50]. A good general overview of fiducial markers used for robot pose estimation is provided by Kalaitzakis et al. [77].

Several studies have compared the detection of markers at different angles and distances. For instance, Jurado-Rodriguez et al. [78] developed a system to design, detect and track personalized fiducial markers. The study compared the accuracy, response time, and occlusions of six different types of markers, capturing photos of the markers at three different distances using a camera with a resolution of 1280 x 720 pixels. Their experiment concluded that the custom markers while experiencing a slight loss in accuracy and detection speed compared to usual markers, were a valid option for commercial and research applications.

Similarly, in [79], a UAV delivery system was developed using an optical flow module for hover and cruise while using visual processing for target detection. This strengthens the use case of fiducial markers in UAV applications where accurate and reliable target detection is crucial.

In Kalaitzakis et al. [77], the authors presented a review and experimental comparison of four types of fiducial markers (ARTag, AprilTag, ArUco, and STag) for their localization performance in autonomous systems and robotics applications. The study investigated location accuracy, robustness under varying lighting conditions and motion blur, and computational performance. The experiments used different tag configurations and employed the Logitech C270 Webcam and Raspberry Pi camera for data acquisition. The fiducial markers were rotated during the experiment, and photographs were captured at various distances. This approach enabled the evaluation of measurement errors and the determination of detection rates for each type of fiducial marker. The results indicated that AprilTag, ArUco, and STag demonstrated high detection rates, AprilTag had the best orientation measures, and STag had the best position measures.

However, not all environments may be suitable for vision-based localization. As indicated in [80], methods like the precise cable-guided landing of tethered drones can prove helpful in degraded environments where GNSS and vision are not feasible. This draws

attention to the need for alternative strategies in addition to the fiducial marker-based approaches to achieve precision in landing.

2.9 Tello UAV

Tello is a quadcopter developed in collaboration between Ryze Tech and DJI, Figure 2.3. It stands out for its lightweight and compact dimensions, with only 87 grams and 98mm x 92.5mm x 41mm. Designed to be accessible and secure, it is an option for initial testing in both indoor and outdoor applications in controlled conditions. To communicate with the drone, a Wi-Fi network is used to send commands and read status, operating via UDP (User Datagram Protocol) packets [81].

One of the important aspects of the model is the incorporation of a flight stabilization system. This system ensures a stable and controlled flight operation, which is essential when performing activities in confined spaces or taking images and videos. Additionally, it comes equipped with advanced speed measurement and odometry capabilities, making it possible to monitor its position and movement [82].

The UAV's documentation specifies that it has an 1100 mAh battery, offering up to 13 minutes of flight time, and two varieties of 3044P propellers. The forward-facing camera has a field of view of 82.6° and includes electronic image stabilization. It can capture 5-megapixel images and record 720p videos, providing satisfactory image quality for a variety of applications. In terms of flight performance, it can reach speeds of up to 7.5 m/s and is capable of flying at altitudes ranging from 30 cm to 30 meters, presenting improved performance at altitudes of up to 6 meters. Moreover, the drone has a transmission range of up to 100 meters, expanding its possibilities of use [83].

In addition to the forward-facing camera, the Tello is equipped with a downward-facing visual positioning system composed of an additional camera and a 3D infrared module. This system is used exclusively by the drone's internal system for stability control. However, the effectiveness of this visual positioning system can be limited in extreme lighting conditions, as in excessively dark (less than 10 lux) or excessively bright (more than 100,000 lux) environments, as well as on surfaces with repetitive patterns. These conditions can affect Tello's ability to maintain stability and precision in flight.

Figure 2.3: Tello Ryze [84]

2.10 Magni UGV

The Magni Silver Robot Base mobile robot, Figure 2.4, developed by Ubiquity Robotics [85], has a robust metal structure and the capacity to transport a payload of up to 100 kg. In addition, the Magni robot has a modular hardware architecture, making it possible to attach different types of sensors and actuators to its structure, enhancing its application in different application domains.

The fact that it is completely modular makes it possible to adapt different types of embedded systems to control it. Currently, this robot is equipped with a Raspberry Pi 4B and operates with the Ubuntu operating system, as well as having Noetic ROS integrated natively, making it ideal for optimizing and customizing the robot's system at a low level.

Magni can reach an average speed of 1 m/s (or more by software update), making it ideal for simulating and validating experiments conducted indoors. In addition, this robot can establish communication via Wi-Fi through different access points.

Figure 2.4: Magni Silver Robot Base [85]

2.11 Iris UAV

The Iris aircraft model provided in the Gazebo robotic simulation library is a multirotor developed by 3D Robotics [86] and operating with the PX4 firmware [87]. The PX4 firmware is an open-source flight control software system and is prepared to run Software in the Loop (STIL) simulations, making it ideal for preliminarily evaluating new hardware and software implementations in simulation.

Communication between the Gazebo simulator and the autopilot, i.e. the PX4 firmware, is carried out using the MAVROS MAVLink protocols. The ROS and the Gazebo software interact with the Iris autopilot, making it possible to receive data from the aircraft linked to its sensory devices, such as the IMU, barometer, GPS, camera and others.

In this sense, different solutions can be explored in a virtual environment by the Iris multirotor, for example, validating PID control models as well as modeling fault-tolerant systems in the face of different engagement scenarios, mitigating potential failures observed in simulation before implementation in an autopilot (especially in controllers of the Pixhawk line) for experiments conducted in a real environment [88].

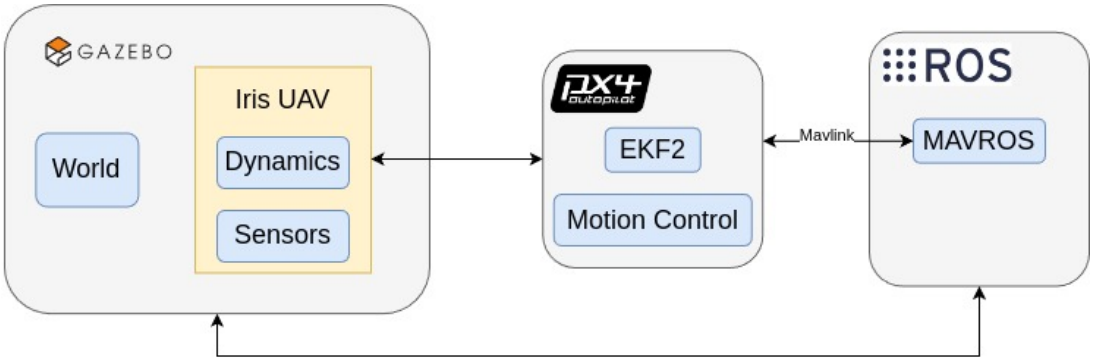

Figure 2.5: Interconnection of the Gazebo, the PX4 firmware, and the ROS for development with the simulated Iris UAV.

Chapter 3

Development

3.1 Comparative Analysis of Fiducial Markers

For the continuation of the work, it is first necessary to define the fiducial marker model that will be used. In this study, three different models were analyzed: ARTag, ArUco, and ArUco board, a collection of ArUco markers. While these markers share some common characteristics, they rely on precise detection and interpretation packages. For the ARTag markers, the Ar Track Alvar package is used, which is based on ALVAR, a set of SDKs that offer a range of tools for developing augmented reality applications using marker-based 2D imaging, multi-marker, or 3D point cloud-based tracking. As for the ArUco markers, the ArUco library from OpenCV is used, which provides a set of functions for marker detection, tracking, and pose estimation.

The fiducial markers type used in this experiment was selected because they are widely referenced in the literature in several applications involving UAVs, are open source, and are already integrated with the ROS platform.

This research adopts a comprehensive methodological approach to assess the performance of various fiducial markers in precision landing operations of multirotor UAVs, specifically in indoor environments. The core objective is to investigate the UAV Tello's

ability to accurately detect these markers, which are strategically positioned at predetermined distances and angles. This study not only evaluates the accuracy of the information provided by these fiducial markers to the UAV but also explores the potential limits of their detectability. By doing so, it aims to enhance the UAV's precision landing capabilities, considering the practical applications and constraints faced during indoor operations.

3.1.1 Proposed approach

As the objective of this work is to evaluate the ability of the UAV Tello to detect the different models of fiducial markers when applied to precision landing operations, and considering that the UAV Tello has only one camera pointing forward, it was adapted in the frontal region of the aircraft a 3×3 cm mirror positioned at $70°$ perpendicularly to the Tello camera, whose main function is to reflect the image from the front camera to the lower region of the UAV. With this, the objective was to extract results as close as possible to the UAV Tello application scenario, i.e., when used in precision landing operations. Figure 3.1 illustrates the mirror's adaptation under the UAV Tello's frontal region.

Figure 3.1: Mirror's adaptation under the UAV Tello's.

The marker needs to be large enough to be detected on the flight ceiling, and also small enough to prevent occlusion as the UAV approaches the landing. With this, it was defined that the dimensions of the ARTag and ArUco are 8×8 cm, with a white margin

of approximately 1 cm. In comparison, the ArUco marker measures 8×8 cm with an internal space of approximately 9 mm. As the Tello camera has a field of view of 82.6°, these markers will stop being detected at approximately 5 cm. These markers are shown in Figure 3.2.

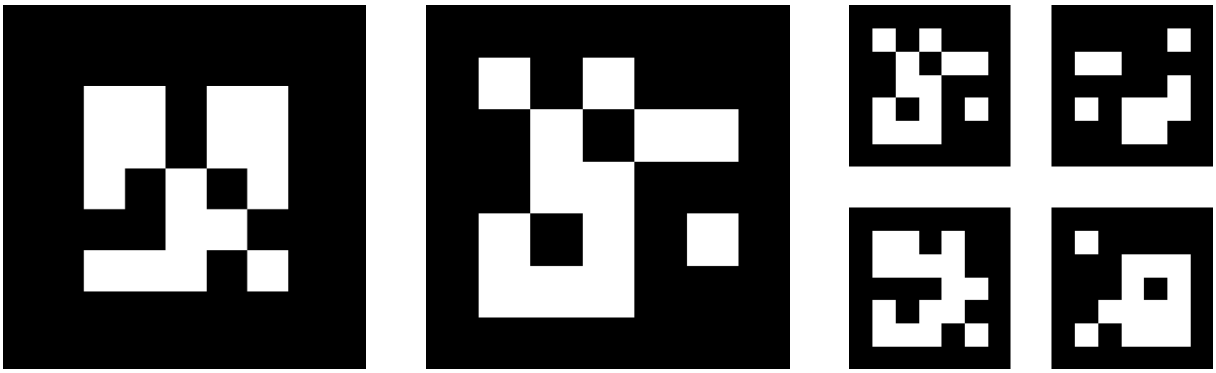

Figure 3.2: Fiducial Markers used (ARTag being arranged on the left, ArUco in the center, and the ArUco board on the right)

To evaluate the detection accuracy of the markers in different situations several photographs were captured for each fiducial marker, acquiring images in predefined positions and angles. From these images, it was possible to calculate the error associated with each measurement obtained, thus allowing us to determine which fiducial marker has the lowest mean error in all situations. To ensure precision in moving the markers, a robot from Universal Robotics - UR3 was used. To acquire the images, a set of procedures was performed, as shown in Figure 3.3.

- In a controlled environment, i.e., without interference from external factors, the UAV Tello was positioned with the optical center of its camera pointed toward the front of the UR3 robot. Under the end of the UR3 is positioned the fiducial marker used during the experiments.
- A LASER gauge accurate to approximately 2 mm was used to guide the distance between the Tello UAV camera and the fiducial markers. In addition, the LASER meter was used to orient the fiducial markers to the center of the mirror, keeping them centralized under the image captured by the UAV Tello.

- A tripod was used to guide the UAV Tello in front of the UR3 robot.
- Four ranges of distances were determined to perform the acquisition of UAV Tello images in relation to the fiducial markers. These bands comprise 75, 100, 125, and 150 cm distances.

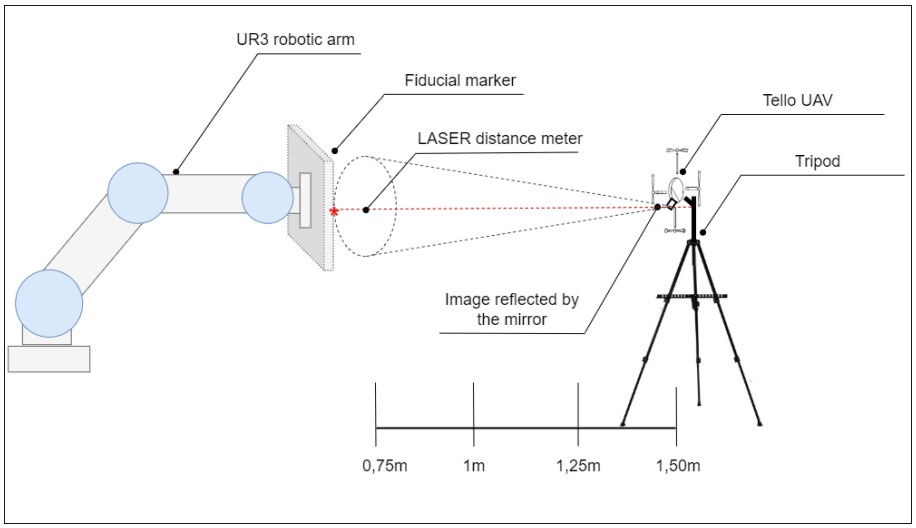

Figure 3.3: Procedures adopted for data acquisition.

For image acquisition, a script was developed in which the Tello UAV captures a set of 5 images at the moment each position of the UR3 robot is assumed. In total, nine positions were programmed for the UR3, as shown in Figure 3.4. The actions adopted for image acquisition are:

- The Tello UAV remains centered 75 cm away from the fiducial marker.
- Initially, the UR3 robot goes to position one and remains with the fiducial marker at point $(0, 0)$, that is, in the center of the Tello UAV camera.
- The UR3 goes to the fist position (P1).
- The fiducial marker is rotated to $30°$, $20°$, $10°$, $0°$, $-20°$, $-30°$, $-10°$ in pitch and yaw, and finally, at $0°$, $-15°$, $-30°$, $-45°$, $-60°$, $-75°$ in roll. As 0 in each of them is the same angle, this gives a total of 18 angles in each position. As five photos

were taken at each angle, in total, a set of 90 images were acquired by the UAV in position one. One example of these images is in Figure 3.5.

- Then, the UR3 goes to position two, and the procedures adopted in position one are repeated until the nine pre-established positions are completed. Upon completing the nine positions (see Table 3.1), the UAV Tello is moved to the following distance on the z-axis, 1m away from the fiducial marker.

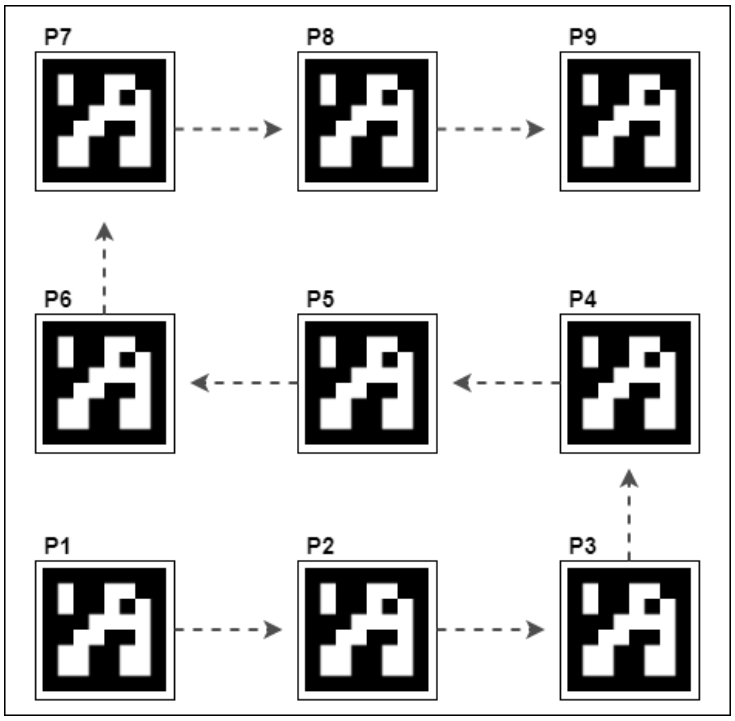

Figure 3.4: Positions assumed by the UR3 robot.

Table 3.1: Set of predetermined positions for UR3 robot.

	P1	P2	P3	P4	P5	P6	P7	P8	P9
X	0.360	0.360	0.360	0.360	0.360	0.360	0.360	0.360	0.360
Y	-0.150	0.070	0.290	0.290	0.070	-0.150	-0.150	0.070	0.290
Z	0.080	0.080	0.080	0.080	0.080	0.080	0.080	0.080	0.080

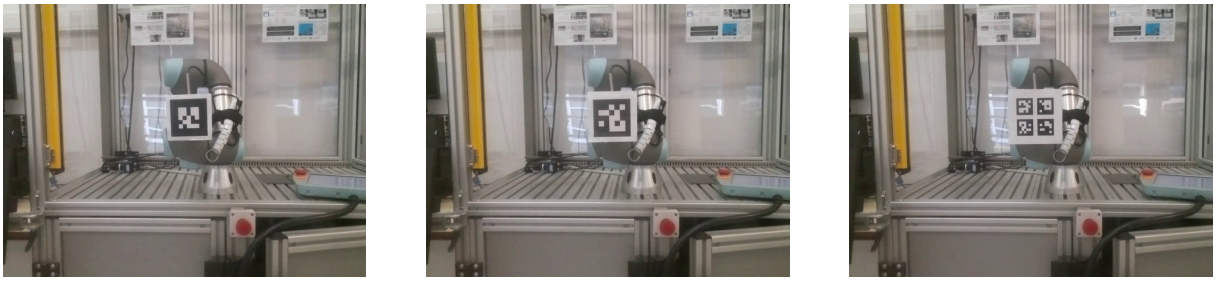

Figure 3.5: Images example (ARTag being arranged on the left, ArUco in the center, and the ArUco board on the right)

3.1.2 Results and Discussion

This section discusses the results obtained through the methodology adopted to classify the measurement accuracy of the fiducial markers selected for this case study, identifying their effectiveness for different distances and angles.

Performance towards position

To determine the accuracy of the positions obtained in detecting the fiducial markers, it is necessary to establish the expected values for each position to calculate the measurement errors.

The center point of the fiducial markers placed on the UR3 is assumed to be precisely aligned with the camera, so at this point, is expected to get the values of $(0, 0)$ m for the X and Y axes. Based on the defined values for the UR3 motion (see Table 3.1), the expected positions for the other points can be determined, as shown in Table 3.2.

Table 3.2: Expected markers positions.

	A	B	C	D	E	F	G	H	I
X	-0.220	-0.220	-0.220	0.000	0.000	0.000	0.220	0.220	0.220
Y	-0.160	0.000	0.160	-0.160	0.000	0.160	-0.160	0.000	0.160

Figure 3.6 shows the values obtained by detecting the fiducial markers (blue dots) and the expected position (i.e., red dots). It can be seen that the measurements follow an error pattern. The measures have a downward and leftward shift in all positions, showing an intrinsic error in the system.

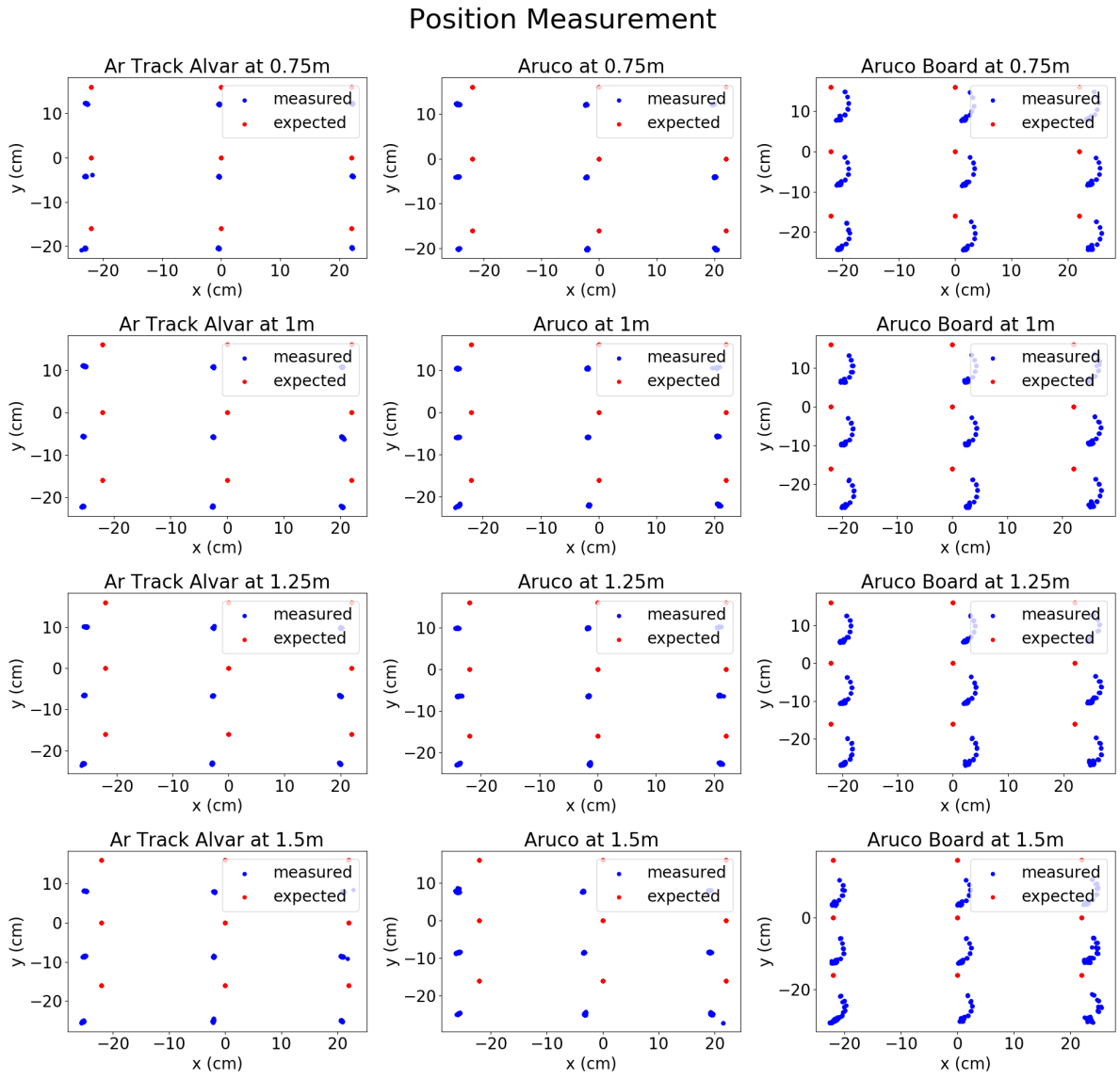

Figure 3.6: Positions measured with the fiducial markers (ARTag being arranged on the left, ArUco measurements in the center, and the ArUco board on the right) at different distances (0.75, 1, 1.25, and 1.5 meters).

As mentioned, the mirror has an angle of 70° concerning the y-axis of the camera. With this, the image suffers a distortion, causing an intrinsic error in the measurements of this coordinate that grows along the distance between the marker and the camera, as a consequence also of the decrease of accuracy in detecting the fiducial markers at long

distances.

It is possible to see this error more clearly in Figure 3.7, Figure 3.8 and Figure 3.9, which show that the error in the measurements at 0.75m is closer to zero than the of 1.5m. This error is less prominent on the X-axis, as the mirror angle on this axis is close to 0°. This offset can be calculated based on these angles and must be considered during the UAV precision landing.

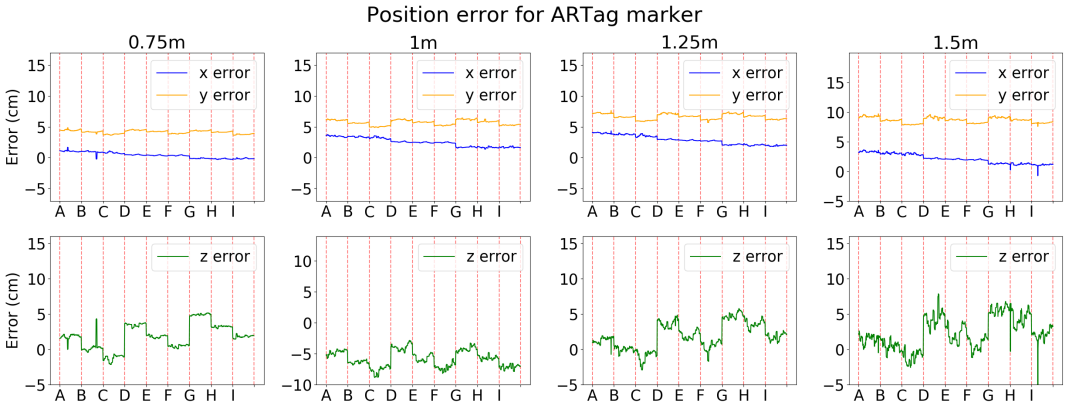

Figure 3.7: ARTag position measurement error per marker position (Table 3.2).

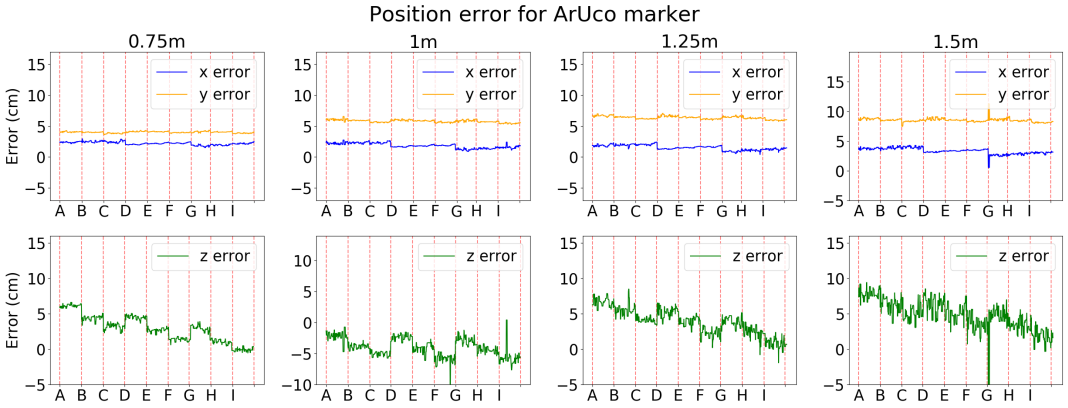

Figure 3.8: ArUco position measurement error per marker position (Table 3.2).

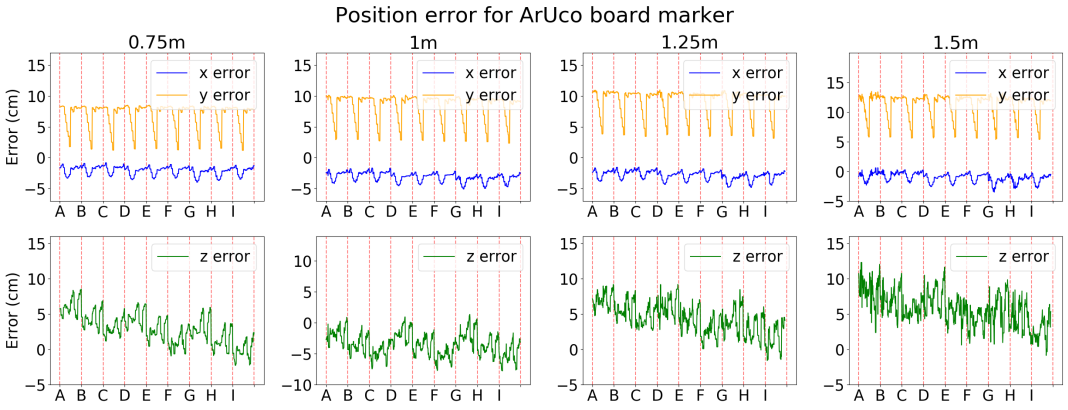

Figure 3.9: ArUco board position measurement error per marker position (Table 3.2).

Note that the ArUco and the ARTag suffer little interferences in the X and Y axis measurements about the position and angle of the marker. In contrast, with the ArUco board, a significant variation in the error with the yaw rotations is presented.

On the z-axis, it is possible to verify sudden changes in the error in the position changes in all graphs, showing a significant interference in the measurement of the three markers. Besides, both the ArUco and the ArUco board suffer more interference from the angle of the marker.

Performance towards orientation

In the case of orientation measurements, the expected values align with the predefined ones for the UR3 rotation, Table 3.3. Subsequently, the error is calculated for each measurement, presented in Figure 3.10, Figure 3.11, and Figure 3.12. Notably, the results obtained from the three markers exhibit striking similarities, with most measurements yielding errors between -10 and 10 degrees.

However, when considering yaw and pitch measurements taken from a distance of 1m, it becomes apparent markers exhibit errors surpassing 10 degrees on multiple occasions. Hence, to determine more accurate outcomes, it is crucial to compare the standard deviation and average error (Table 3.4, Table 3.5, and Table 3.6).

Table 3.3: Expected Marker Angles.

Pitch	Yaw	Roll
-30°	0°	0°
-20°	0°	0°
-10°	0°	0°
10°	0°	0°
20°	0°	0°
30°	0°	0°

Pitch	Yaw	Roll
0°	0°	0°
0°	0°	-15°
0°	0°	-30°
0°	0°	-45°
0°	0°	-60°
0°	0°	-75°

Pitch	Yaw	Roll
0°	-30°	0°
0°	-20°	0°
0°	-10°	0°
0°	10°	0°
0°	20°	0°
0°	30°	0°

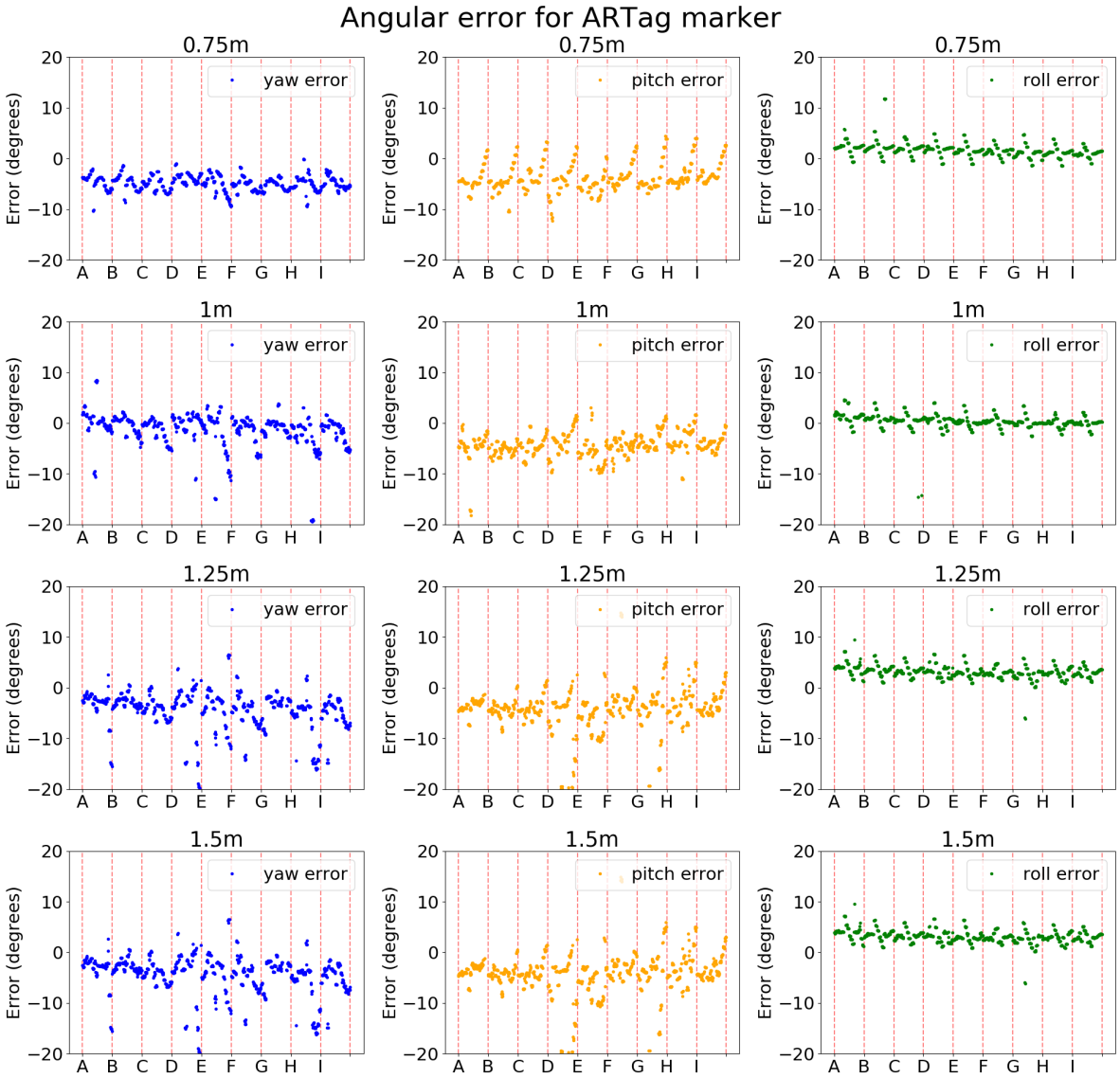

Figure 3.10: ARTag angular measurement error.

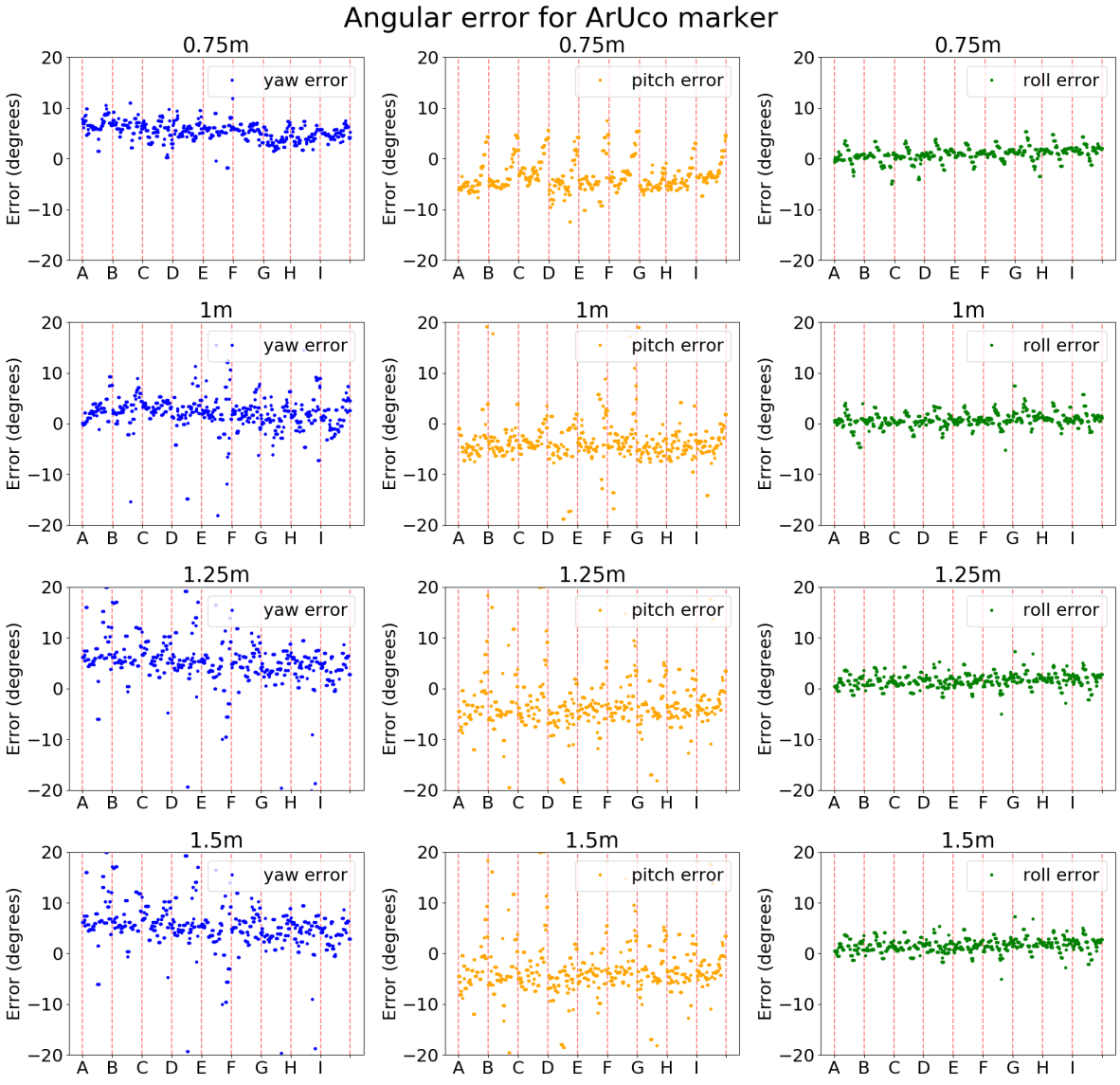

Figure 3.11: ArUco angular measurement error.

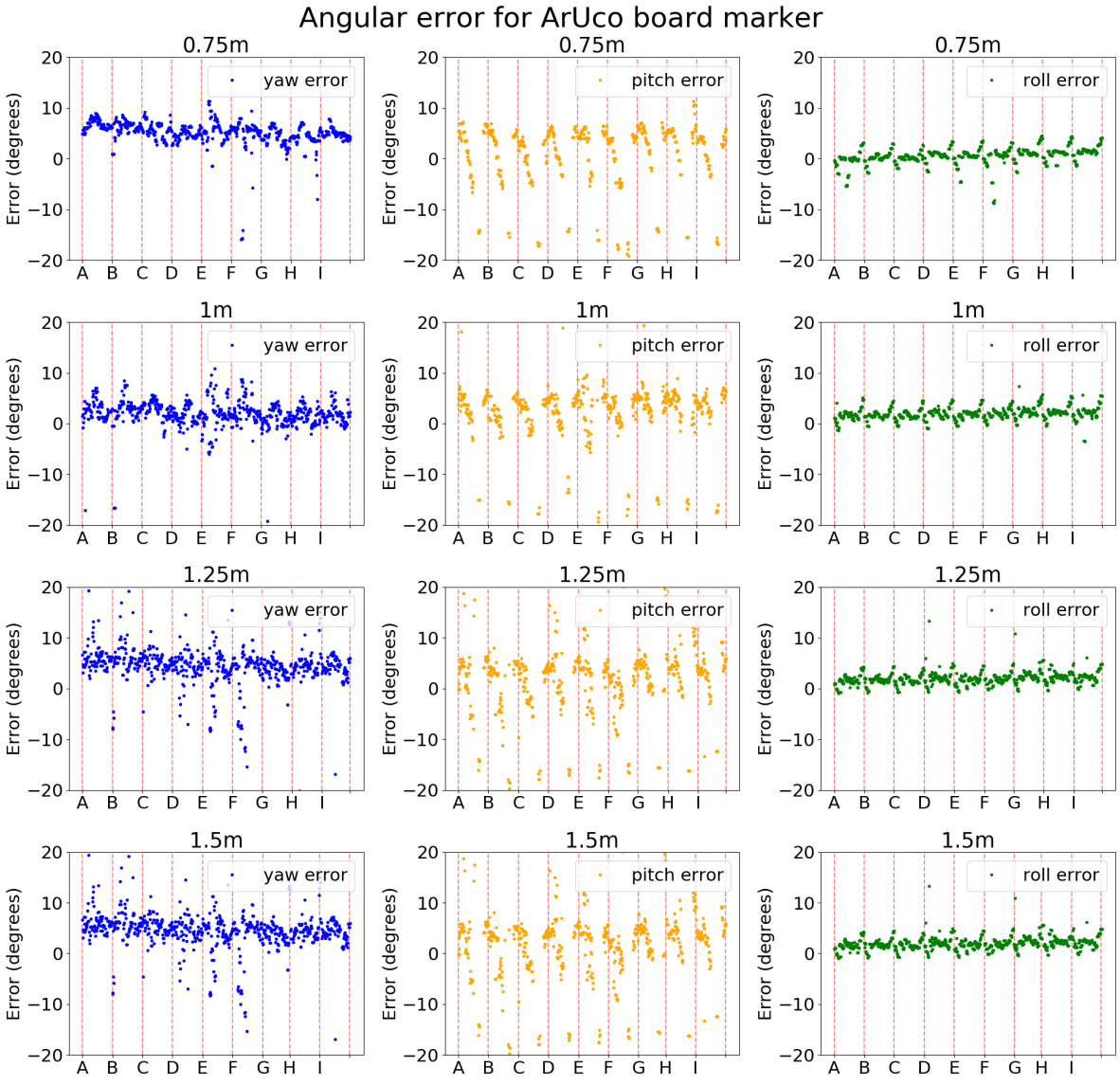

Figure 3.12: ArUco Board angular measurement error.

Table 3.4: Std and mean error of the ARTag measurements.

	ARTag							
	0.75m		1m		1.25m		1.5m	
	std	mean	std	mean	std	mean	std	mean
x_error	0.460	0.376	0.692	2.497	0.719	2.858	0.794	2.089
y_error	0.263	4.140	0.416	5.684	0.446	6.638	0.469	8.599
z_error	1.799	1.832	1.345	-5.848	1.863	1.868	2.281	2.274
yaw_error	1.506	-4.852	3.158	-1.207	5.269	-4.857	8.797	-5.704
pitch_error	3.074	-3.782	3.586	-4.751	5.334	-4.645	6.761	-5.535
roll_error	1.385	1.646	1.402	0.424	1.262	3.168	1.449	1.444

Table 3.5: Std and mean error of the ArUco measurements.

	ArUco							
	0.75m		1m		1.25m		1.5m	
	std	mean	std	mean	std	mean	std	mean
x_error	0.247	2.201	0.493	1.821	0.408	1.519	0.403	3.309
y_error	0.127	3.997	0.317	5.778	0.221	6.349	0.252	8.474
z_error	1.867	2.899	2.266	-4.052	1.488	3.869	1.851	4.843
yaw_error	1.798	5.360	7.720	1.825	4.567	4.858	6.401	4.506
pitch_error	3.871	-3.139	9.701	-3.711	4.399	-3.193	5.903	-2.205
roll_error	1.459	0.949	1.689	0.745	1.450	1.605	1.375	1.502

Table 3.6: Std and mean error of the ArUco board measurements.

	ArUco board							
	0.75m		1m		1.25m		1.5m	
	std	mean	std	mean	std	mean	std	mean
x_error	0.687	-2.091	0.755	-3.101	0.713	-2.845	0.715	-1.056
y_error	2.020	6.964	2.037	8.383	2.076	9.177	2.072	11.234
z_error	2.239	2.874	2.459	-3.553	1.892	4.353	2.234	5.647
yaw_error	2.447	5.154	9.840	1.901	2.836	4.003	5.862	3.597
pitch_error	25.102	3.235	23.782	3.849	25.055	3.322	25.375	5.594
roll_error	1.683	0.546	1.588	1.988	1.228	2.005	1.280	1.181

None of the markers showed a significantly high average error, with the largest recorded error being 5.704 degrees. However, it is important to mention that the standard deviation revealed some problematic values.

It can be seen that in the tilt measurements of the ArUco plate at the four distances, the standard deviation is greater than 20 degrees, showing that the measurements were very dispersed in relation to the average, highlighting a problem in the detection of the marker. For ArUco and ARTag, a less dispersed pattern was observed, with a maximum of 9.701 and 8.797, respectively.

Overall performance of the fiducial markers

Upon examining the error tables (Table 3.4, Table 3.5, Table 3.6), the best and worst values for each marker type are indicated in green and red, respectively. Based on these results, it can be concluded that the ArUco board marker consistently shows the weakest results among the three markers. It also shows that the ArTag and ArUco markers have similar results, as they both have many of the best values.

A closer look at the values reveals that the ArTag and ArUco markers have similar standard deviation and mean error values. However, ArUco has a slightly lower mean error than ArTag and also has a more consistent result by having lower standard deviations.

Comparing the ArUco board with the ArUco, where the detection method is the same, shows that using more than one marker made the results worse, especially on rotation. This worsening is associated with the size of the markers. In the ArUco board, the internal markers are about 3.5 cm, half that used in the regular ArUco, which makes the measurements much worse the further away the camera is from the marker. This fact is also demonstrated in [77].

3.1.3 Final Considerations

Upon thorough analysis of the collected data, it has been observed that the ArUco board marker exhibited comparatively weaker performance in both positional and rotational

measurements. However, when comparing ArUco and ARtag markers, it was found that both demonstrated similar levels of accuracy, with the ArUco marker showing marginally better accuracy and consistency. Within the tested range of 0.75m to 1.5m, the markers achieved, in the worst-case scenario, a positional accuracy of approximately 3cm on the x-axis, 5cm on the z-axis, and 8cm on the y-axis. It is noteworthy that these results could be further enhanced by compensating for the errors associated with the system.

Given the constraints of dimensions and the indoor flight ceiling limitations, this study demonstrated the feasibility and effectiveness of utilizing these techniques, highlighting their potential to improve the landing process of these heterogeneous vehicles.

After evaluating the outcomes, it can be concluded that the Aruco system displayed a marginally superior performance compared to the tested alternatives. This effectiveness substantiates its selection for implementation in subsequent experiments. Moreover, it was observed that the integration of a mirror in the measurement mechanism introduced significant distortions, particularly on the y-axis. This discovery underscores the critical need for implementing error correction to mitigate such inaccuracies.

Nevertheless, to circumvent the complexity introduced by the necessity for error correction, a significant alteration was made to the drone's architecture: the removal of the mirror followed by the rotation of the main camera. This modification, although considerable, is deemed beneficial as it not only simplifies the system but also enhances the code's compatibility with a diverse range of handmade drones, which typically feature cameras oriented vertically toward the ground.

3.2 Control architecture

After establishing the fiducial marker model, it is possible to proceed with developing the UAV system. The next step is to define the control architecture. In this context, the central focus of the present study is the implementation and comparative analysis of different Proportional-Integral-Derivative control architectures on a multi-rotor UAV, whose main objective is its application in the precise tracking of a moving object using the tracking of an ArUco fiducial marker positioned on top of a UGV as localization system. This research aims to identify the control architecture that best achieves the final objectives, emphasizing the need for a reliable method to adjust the gain values of these architectures, as this directly impacts the efficiency of the controllers.

To ensure an effective analysis, it was decided to conduct this analysis in a simulated environment. This choice allows for a more precise evaluation, offering control over the environment and accurate position and velocity data. In addition, it minimizes the risks associated with real tests, a crucial aspect in the early stages of development, where controllers may not yet be fully optimized, avoiding accidents and material losses. Simulation provides a solid basis for UAV system development, ensuring that decisions on control architecture are made with confidence, safety, and efficiency.

For this simulated environment, was utilized the Gazebo integrated with ROS (Robot Operating System) to conduct the tests. Regarding the robotic architectures, was selected the Iris UAV, known for its efficiency and versatility, and the Magni UGV by Ubiquity Robots, noted for their robustness and reliability. This combination of simulation tools and robotic platforms provides a detailed and realistic simulation. Figure 3.13 illustrates this robotics units.

The intrinsic camera parameters defined in the iris model were used to detect and measure the fiducial markers. A landing platform with an ArUco marker measuring 12x12 cm was modeled and added to the top of the Magni UGV, making it approximately 53 cm high. The fiducial marker aims to estimate the aircraft's position and pitch relative to the UGV. In this way, the aircraft has a visual reference to assist in pursuing the UGV

along the routes proposed in this work.

Two routes were predefined to evaluate the control architectures used, where the UAV must follow the UGV and stay at 2 meters in height. The first route executed by the robotic platforms consists of a straight line of four meters with a constant speed, performing forward and backward movements, while in the second route, the robotic platforms cover a section whose shape is square of two meters side, Figure 3.14, adding more complexity to the UAV movement during the tracking operation. The different control architectures are applied and compared on each route defined by the experiments.

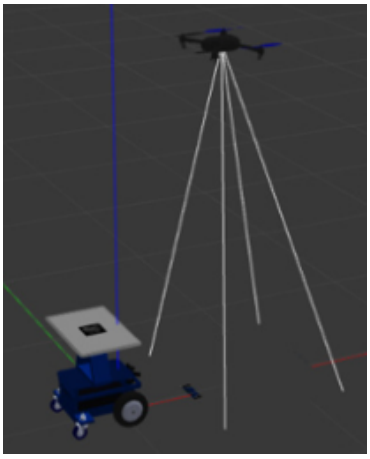

Figure 3.13: UAV Iris and UGV Magni are used in the Gazebo virtual environment.

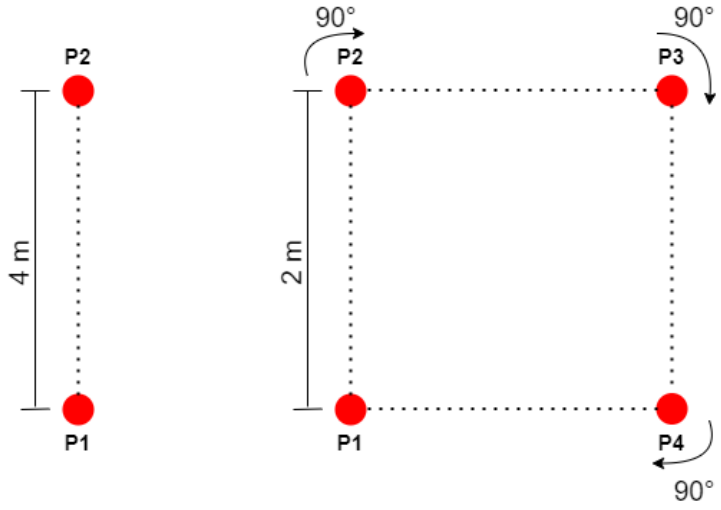

Figure 3.14: Predetermined routes are chosen to evaluate the different controllers.

3.2.1 Proposed Controllers

The proposed control system regulates the UAV's attitude (h), and inertial position, ensuring stable and precise flight control during the process. The overall control structure is presented in Figure 3.15.

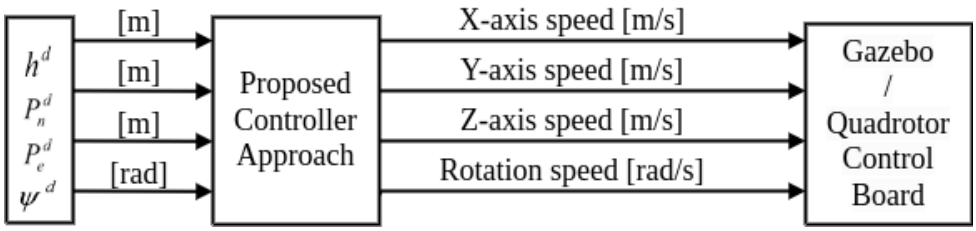

Figure 3.15: Overall schematic of the proposed controllers approach.

The system is designed to align the drone with a fixed marker, using the UAV's relative position to the marker. To achieve this alignment, the reference inputs are fixed at a height of 1.5 meters (z=1.5 m), without rotation (yaw=0 rad) and with coordinates x=0 m and y=0 m. This setup is typical in control systems where the goal is to keep the unmanned aerial vehicle (UAV) stabilized concerning a reference point, which is vital for surveillance, mapping, and other applications that require a fixed position or precise movement to a specific object or location.

As previously mentioned, it has been tested three different architectures, which are: PD Controller (Figure 3.16), PD-PI Cascade Controller (Figure 3.17), and PD-PI Parallel Controller (Figure 3.18), inspired from the research work presented in [68]. These figures (3.16, 3.17, 3.18) show the diagram of the blocks tested in the specific placement shown in Figure 3.15.

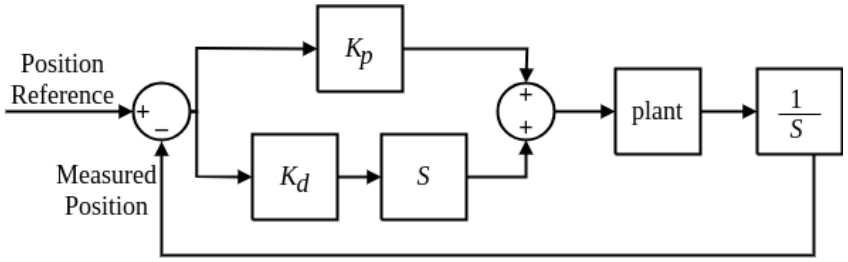

Figure 3.16: Proportional and Derivative controller.

From the Figure 3.16, it is possible to see that a position variable is a setpoint for the PD controller, where the respective velocity is the output driven to the Gazebo environment.

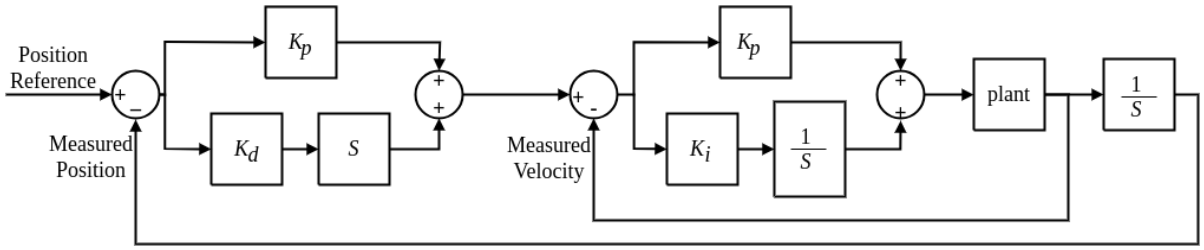

Figure 3.17: Proportional and Derivative - Proportional Integral cascade controller

Figure 3.17 illustrates a PD-PI cascade controller where the outer loop receives a position the outer loop receives a position variable and drives a velocity variable as a setpoint for the inner one. The output of the whole control structure is also a velocity variable. The last proposed control approach is a PD-PI parallel structure, as seen in Figure 3.18.

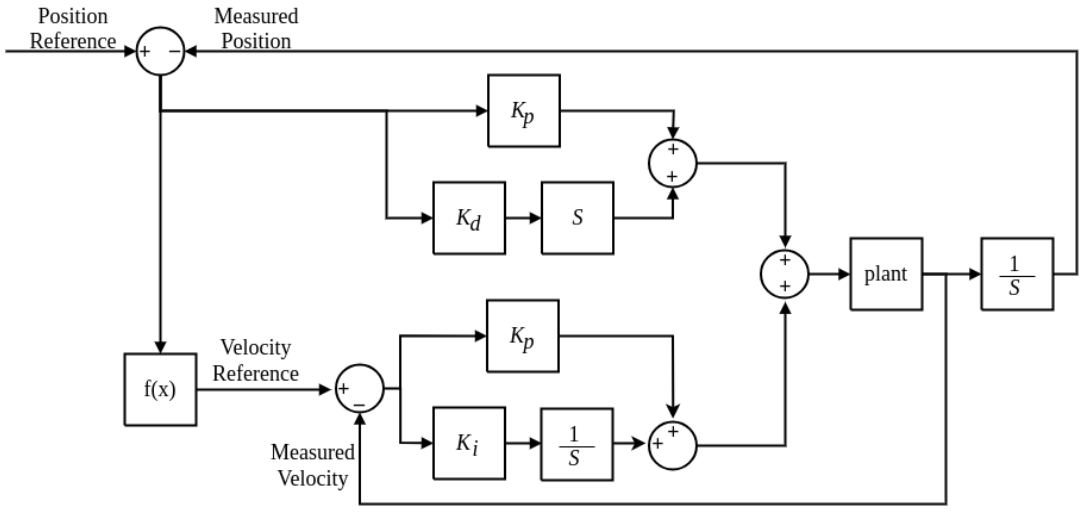

Figure 3.18: Proportional and Derivative - Proportional Integral parallel controller

Then, the position and speed must be entered as set points, one for the PD and one for the PI controller. Finally, the respective velocities are added and directed towards the Gazebo environment. For a speed reference, the represented by f(x) in Figure 3.18 was

used, based on the distance between the UAV and the UGV acquired with the fiducial marker measurement, thus establishing a relationship between the speed and position reference. The curve used for the four axes is in Figure 3.19, only for the yaw command, the speed axis is in radians per second.

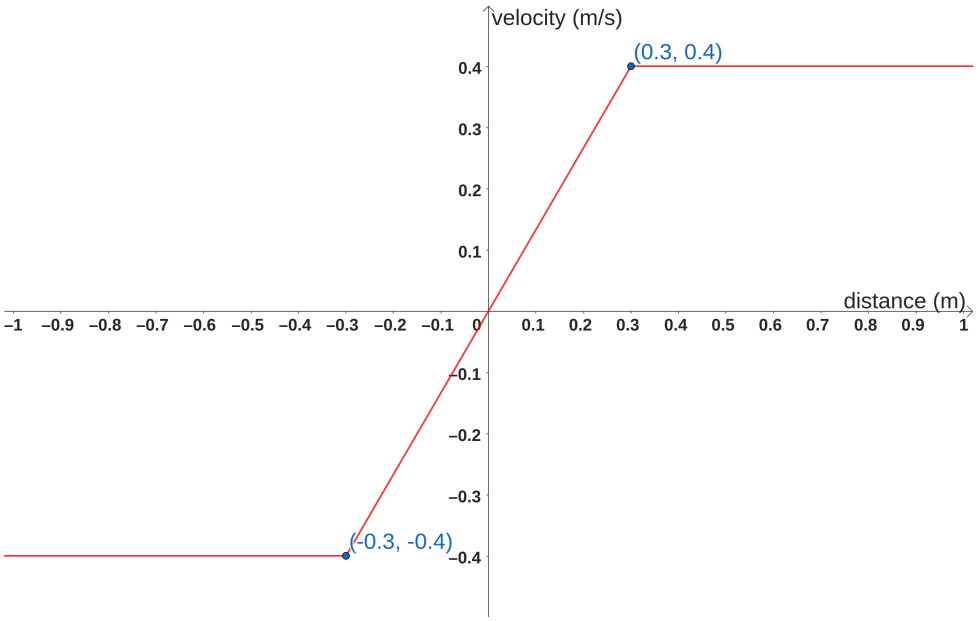

Figure 3.19: PD-PI parallel controller speed reference.

Finally, all three PID architectures employed for drone control utilized the conditional integration anti-windup approach. This method effectively manages integral wind-up during boundary constraints.

3.2.2 Definition of controller's gains

To optimize the performance of UAV controllers it is essential a precise calibration of their gains. This calibration was conducted separately for each UAV motion axis: x, y, z, and yaw, following the same criteria. The first step in this process involved approximating the UAV's transfer function for each axis. This was achieved by recording the UAV's step response, where a constant velocity was applied to each axis until a stationary state was observed.

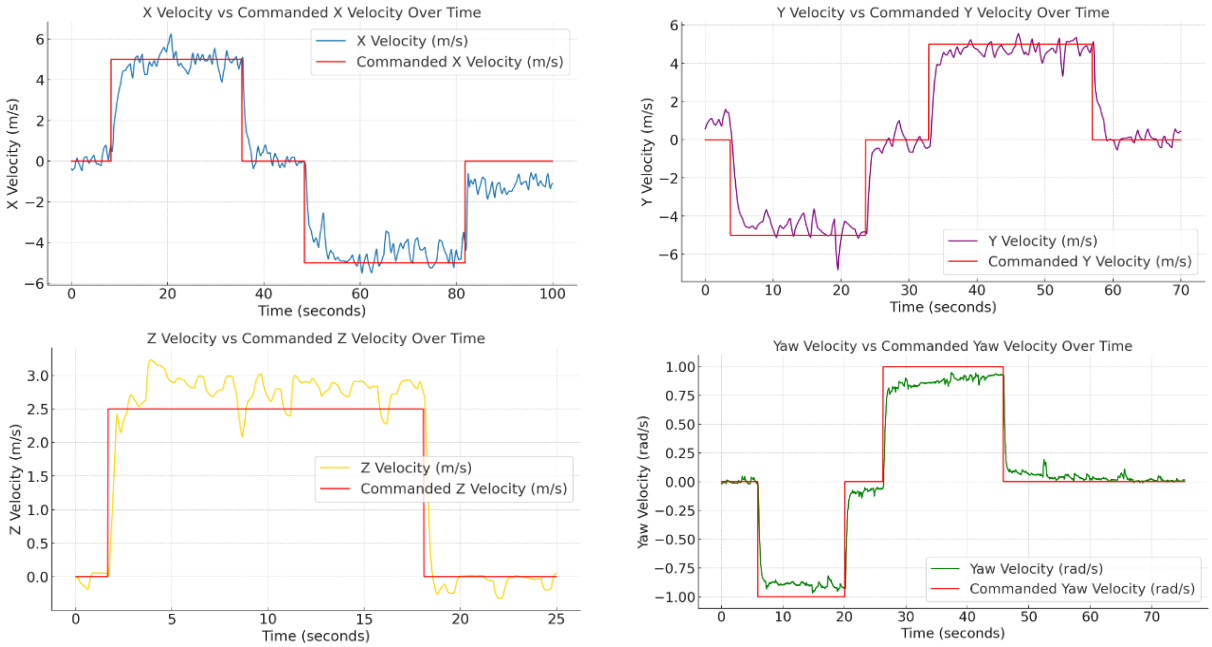

Figure 3.20: Step Response of the UAV System

The step responses, illustrated in Figure 3.20, were crucial to understanding the dynamics of the Iris UAV. These responses were analyzed using Matlab's system identification tool. For each axis (X, Y, Z, and R rotation), a model consisting of two poles and a zero was used to approximate the transfer function, indicated in the equations Equation 3.1, Equation 3.2, Equation 3.3, Equation 3.4.

This modeling method was selected because of its accuracy in representing system dynamics. System identification is a powerful technique for modeling dynamic systems based on input and output data. In this case, the step responses provided data that, when analyzed, allowed the creation of mathematical models representing the dynamic behavior of the Iris UAV.

Using the transfer function modeled for each axis, it was possible to understand how the UAV responds to different control commands. For example, the $Iris_T F_x$ transfer function, referred to in the equation Equation 3.1, describes how the UAV responds to changes along the X axis. This analysis is key to developing more efficient control algorithms and improving the UAV's stability and maneuverability in different flight conditions.

The Figure 3.21 compares the measured step response with the transfer function derived on each axis. It is worth noting that the functions obtained closely reflect the UAV's responses, assuming that abrupt variations are excluded. These abrupt variations, usually due to external disturbances or sensor noise, are not considered in this analysis to focus on the inherent dynamic behavior of the UAV.

$$Iris\_TF_x(s) = \frac{0.541s + 3.400}{s^2 + 5.258s + 3.394} \tag{3.1}$$

$$Iris\_TF_y(s) = \frac{-1.419s + 13.656}{s^2 + 11.416s + 14.577} \tag{3.2}$$

$$Iris\_TF_z(s) = \frac{2.373s + 0.768}{s^2 + 2.252s + 0.697} \tag{3.3}$$

$$Iris\_TF_{yaw}(s) = \frac{2.503s + 0.017}{s^2 + 2.826s + 3.758 \times 10^{-7}} \tag{3.4}$$

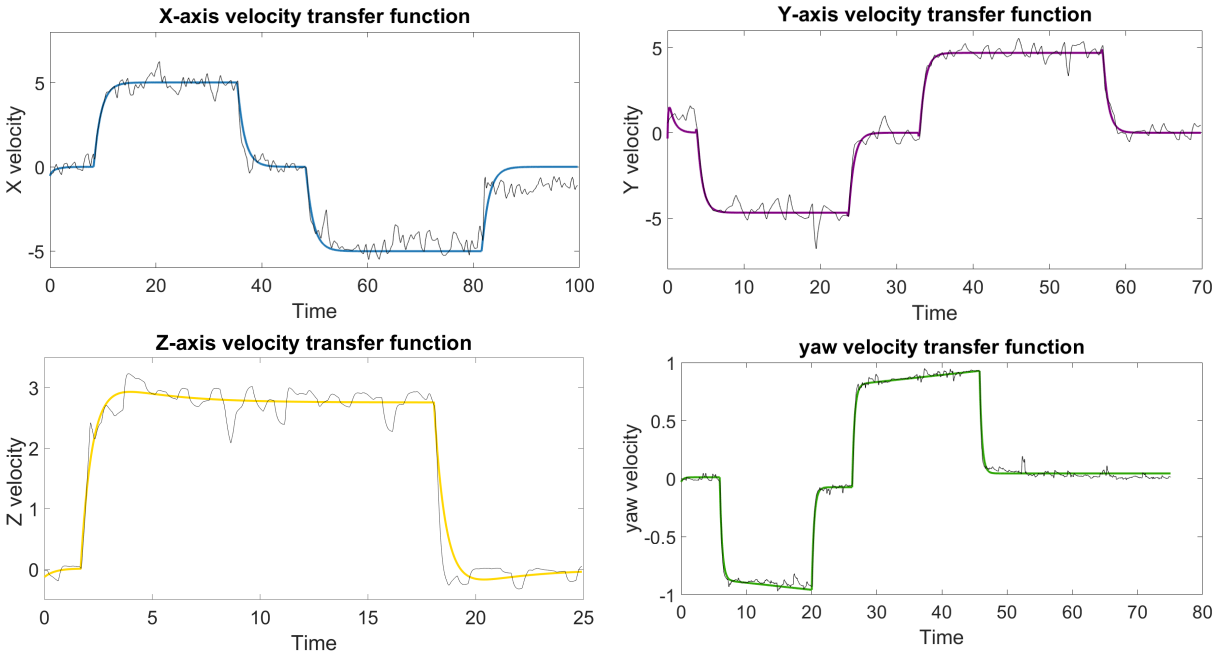

Figure 3.21: transfer functions of the iris UAV system

To determine the controller gains, Matlab's Autotuning system was utilized. This system calculates gain values based on specified transient behavior and response time. In our approach, the closed loops for the speed PI controller and the position PD controller were configured separately.

Both controllers were based on the transfer functions identified previously. However, the PD controller required the integration of its respective transfer function to generate a position response. For each function, was meticulously adjusted parameters related to transient behavior and response time, aiming to maintain system stability. These adjustments were crucial for minimizing overshoot and settling time in the step response, key factors in ensuring precise control.

As a result of this process, the optimal gains for each controller architecture were determined. These gains, which are integral to the controllers' effective operation, are detailed in Table Table 3.7.

	PD				**PI**			
	RT	**TB**	**KP**	**KD**	**RT**	**TB**	**KP**	**KI**
X	2.909	0.495	0.93	0.11	1.348	0.550	1.28	2.78
Y	2.48	0.586	0.97	0.15	2.58	0.9	0.74	0.74
Z	1.819	0.667	1.02	0.087	0.9507	0.9	0.906	1.67
YAW	1.759	0.6	1.40	0.014	0.832	0.9	1.017	2.651

Table 3.7: PD and PI Controller Gains

3.2.3 Results and Discussion

As mentioned, two trajectories were analyzed, one straight and one square. Figure 3.22 shows the path the two robotic platforms take in these two trajectories for each control architecture.

Regarding the position of the Z axis, the Iris UAV must maintain a height of 1.5 meters above the top of the Magni. Since the UGV measures approximately 53 cm, the UAV must hover approximately 2 meters above the ground. Figure 3.22 shows that the

cascade controller and the parallel controller have a low variance, while the PD controller has a high variance. This observation is further confirmed by the error analysis presented in Table 3.8, which shows that the standard deviation of the PD controller is higher than that of the others. Comparing the average error of the parallel controller and the cascade controller, the cascade controller has a lower error on both trajectories, concluding that the cascade PD-PI controller performed better on this axis.

Table 3.8: Standard Deviation and Mean error of the Z-axis position in each control architecture.

		Z		X		Y	
		mean	std	mean	std	mean	std
line trajectory	pd	0.1	0.11	0.39	0.38	0.02	0.01
	cascade	0.03	0.03	0.38	0.38	0.02	0.02
	parallel	0.14	0.03	0.29	0.28	0.02	0.01
square trajectory	pd	0.05	0.06	0.23	0.32	0.23	0.32
	cascade	0.05	0.05	0.25	0.34	0.25	0.35
	parallel	0.13	0.02	0.2	0.27	0.2	0.27

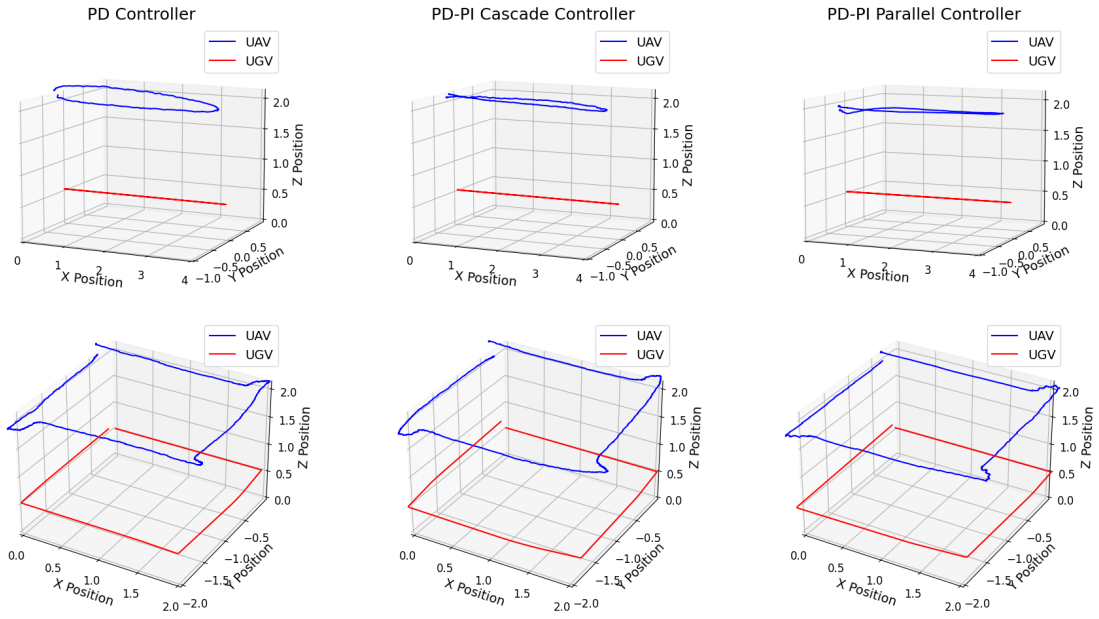

Figure 3.22: UGV and UAV trajectories. The PD controller is shown on the left, the cascade PD-PI controller in the center and the PD controller on the right.

In figure Figure 3.22, the trajectory of the UGV is represented in red, while the

trajectory of the UAV is represented in blue. Focusing on the X and Y axes, that the UAV was able to follow the entire trajectory of the UGV with only a slight deviation while the Magni made a rotation, which is expected because the IRIS camera is on at the front of the UAV, not at the center. This effect is also seen in the Figure 3.23, which shows that the iris can correctly imitate the rotations made by the Magni, but during this movement, the X and Y axes are affected.

In terms of the average error in the linear trajectory, see Table 3.8. As the UGV's movement is only in the direction of the X axis, alignment in Y becomes easier, meaning that the average error in the Y axis is lower. In contrast, the error is much higher for the X-axis, so it can be concluded that the UAV could not stay aligned with the UGV while it was moving in this direction, always staying slightly behind the Magni. In this case, the parallel controller seems to have performed better, with an error close to 30 centimeters, and the other controllers had an error of 40 centimeters. In addition, looking at the standard deviation, it appears that the pattern repeats itself, with a good value on the Y axis, but a worse value on the X axis.

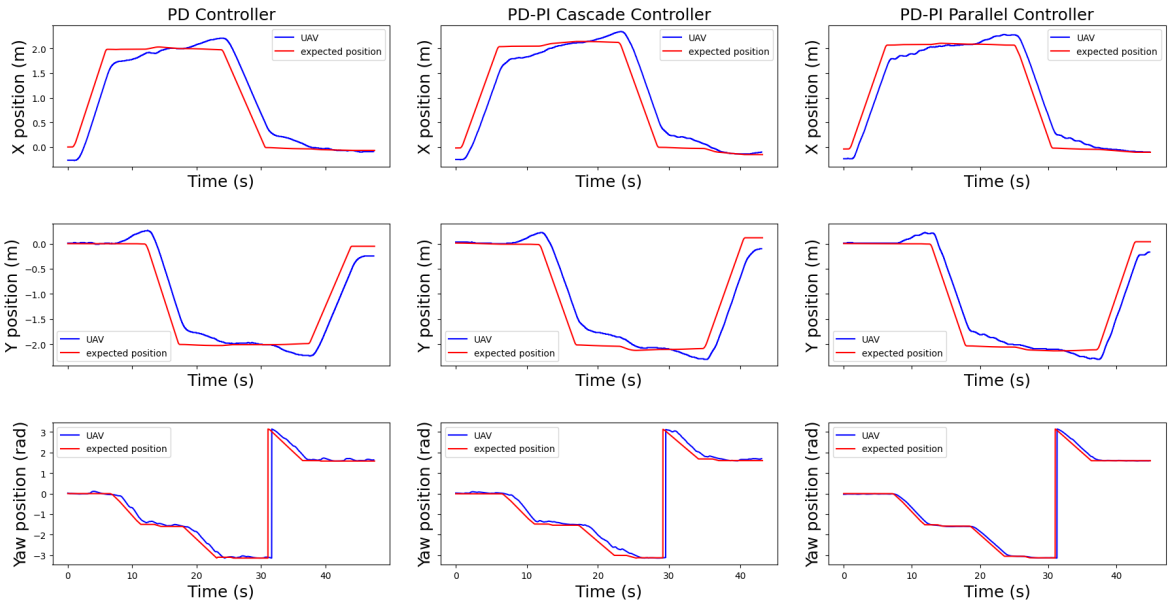

Figure 3.23: Comparison of UAV and UGV position with the three controllers, PD, PD-PI Cascade, and PD-PI Parallel, on the quadrangle trajectory.

In the square trajectory, the two axes, X and Y, have a high average error and a high standard deviation because the UGV moves on both axes in this trajectory. This is confirmed by the observation of Figure 3.23 in which the position of Iris is always behind the position of Magni during the movements. It is also clear that this delay is slightly shorter in the case of the parallel PD-PI controller.

3.2.4 Final Considerations

The SITL tests performed on the Gazebo software, implemented using ROS, showed that the proposed approach is a suitable solution, presenting effectiveness and robustness. It also shows that PID control architectures based on fiducial marker detection successfully performed the tracking task, keeping the UAV coordinated with the ground robot's movement in a simulated environment.

The results showed that all the architectures tested performed satisfactorily, allowing them to move synchronously and that the parallel PD-PI performed slightly better than the other two among the three architectures investigated.

Based on the comparative analysis of the three controllers, PD , PD-PI cascade, and PD-PI parallel, and the observation of their similar performance outcomes, it has been concluded that utilizing a complete PID controller on the marker position is preferable. This conclusion is primarily driven by the simplicity and straightforward implementation that a full PID controller offers. Despite the nuanced differences and specific advantages that the PD, PD-PI cascade and PD-PI parallel controllers may present, the overarching effectiveness of a complete PID controller in managing the UAV's positional control makes it a more practical and efficient choice. This approach not only streamlines the control architecture but also potentially reduces the complexity involved in the system's tuning and maintenance, thereby enhancing the overall robustness and reliability of the UAV control system in various operational scenarios.

In addition to the simplicity, the choice of a full PID controller was also based on its

characteristic of being a third-order system. This particularity is important to guarantee good performance in ramp response, a particularly important aspect when tracking a moving object, and it is a technically suitable alternative to meet the system's requirements.

3.3 Real experiments

To perform the experiments in the real world, after defining the marker and controller, it is necessary to complete the software architecture and develop an effective approach methodology. In the practical tests, two types of robots were used: the Tello, for aerial operations, and the Magni, for ground operations. The Tello is controlled using the tello_driver package, which is based on DJI's SDK 2.0. This package allows code to be run on a computer, which then sends commands and receives status information from the robot via Wi-Fi. For Magni, the official ROS package provided by Ubiquity Robotics was used. This integrated approach aims to ensure efficient and precise communication between the robots and the control system.

To make the experiments practicable, two essential modifications were carried out on the robots. The first one was the addition of a landing platform, shown in Figure 3.24, measuring 34.6 cm by 53.8 cm, on top of the UGV. This platform serves as a landing point for the aerial robot. On top of this base, an ArUco-type fiducial marker measuring 12x12 centimeters was installed, which enables the aerial robot to be guided and land.

The second modification, as described in Section section 3.1, was the change to the frontal camera of the Tello, which was rotated 90º, Figure 3.25, to capture images of the underside of the Unmanned Aerial Vehicle. This adjustment is crucial to enable the detection of the fiducial marker positioned on the UGV's landing pad. These modifications are essential to ensure precision and safety during the landing and take-off operations of the aerial robot on the ground robot.

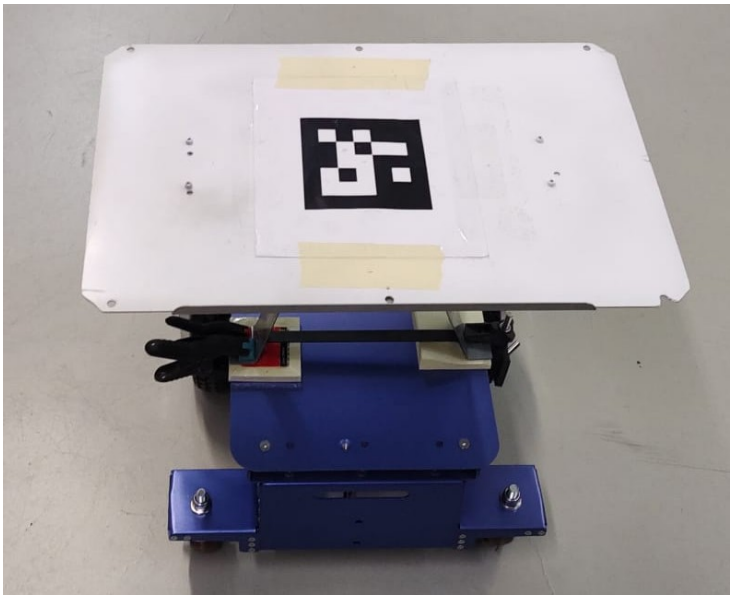

Figure 3.24: Unmanned ground vehicle.

Figure 3.25: Unmanned aerial vehicle.

3.3.1 System's architecture

The system's architecture, as illustrated in the Figure 3.26, is modeled on the Robot Operating System (ROS) framework and comprises a modularized nodes network that interacts by using defined topics to exchange data efficiently. There are two main nodes: */tello_driver* which acts as a communication bridge between the Tello and the ROS,

allowing operational control of the drone, including take-off or landing commands and speed manipulation; and */uav_manager* which is responsible for high-level management and autonomous control logic, providing management of the aerial robot, as well as the controllers of follow up and precision landing.

The remaining nodes are responsible for more specialized actions. The node */aruco_node* is responsible for detecting ArUco markers and estimating the location of the UAV, which is used for navigation and for localization in space. The user interface is provided by two nodes: */joy_node*, which interprets signals from a joystick and transmits control commands through the */joy_control* topic, and */controller_gui*, which enables some important system settings and facilitates data acquisition.

This infrastructure constitutes a robust platform for research and innovation in aerial robotics. It is also designed for flexibility and can be easily adapted to integrate with Mavros, a widely recognized ROS package for communicating with a variety of robotic controllers.

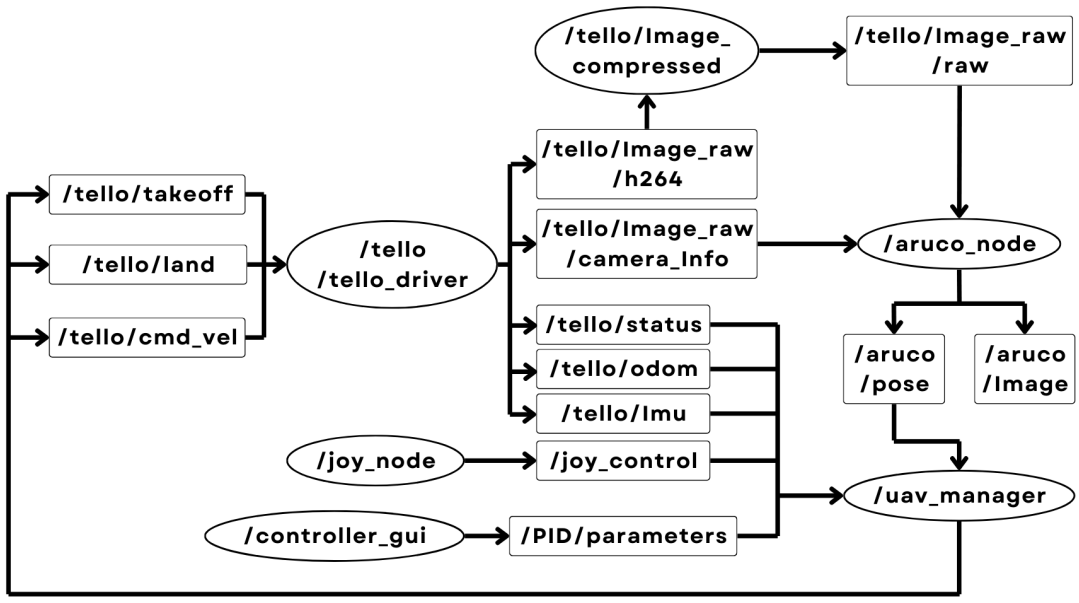

Figure 3.26: ROS nodes and topics.

User interface

To improve the experience of the operator and the efficiency of managing the UAV's control system, a graphical interface was developed using the Tkinter library in Python. This interface allows to define parameters of the UAV manager, as illustrated in Figure 3.27.

During the development process, four different controllers were implemented, as mentioned in section 3.2. The graphical interface provides a selection option between them, allowing the user to independently determine and adjust the gains for each one. The final architecture used only the PID controller, whose gains were defined as documented in subsection 3.3.2, but all four controllers are usable.

In addition to the controllers' selection, the interface incorporated functionality for setting the default altitude that the UAV Tello model should maintain during follow-up operations, as well as for setting the linear and angular velocities during manual control. To optimize data collection, an automated system for recording the relevant parameters was also implemented, which organizes the data collected from each experiment in ways that simplify subsequent processing.

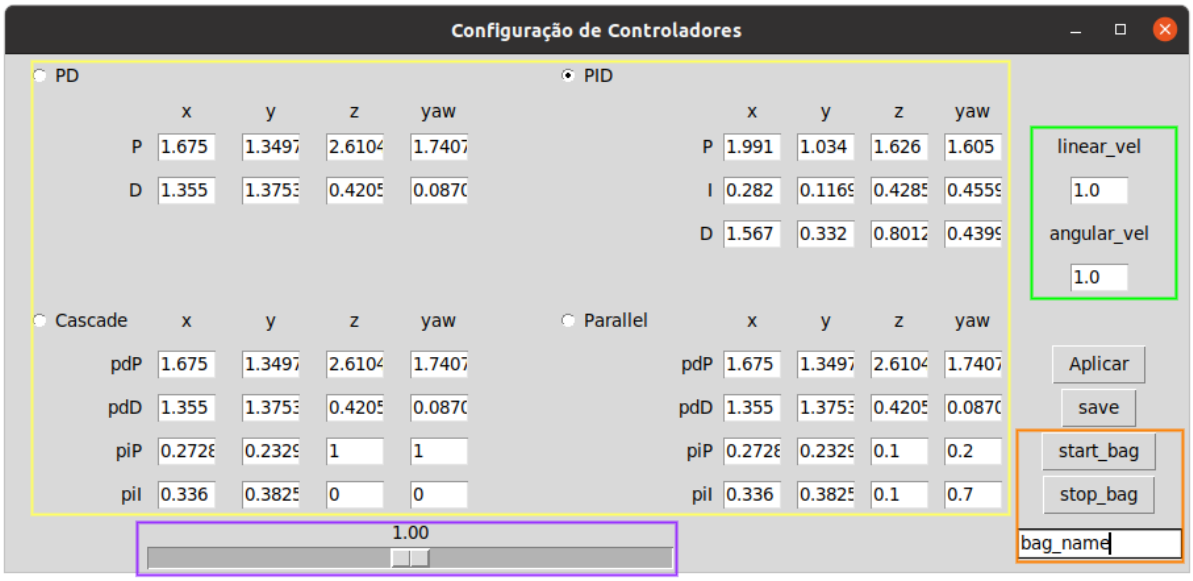

Figure 3.27: Uav control system graphic user interface. in yellow: controller settings, in green: speed settings, in purple: follow-up altitude settings, in orange: data saving

State machine

The efficient operation of an Unmanned Aerial Vehicle (UAV) depends on a sophisticated operating system capable of responding dynamically to a variety of commands and situations. The state machine is a critical component of this system, offering a structure that allows the UAV operator to switch between different operating modes in a predetermined and intuitive way. In addition, the state machine facilitates the inclusion of essential safety mechanisms, such as the UAV's immediate landing capability, which are key to minimizing risks during aerial operations. The states of the utilized state machine and the transitions between them are shown in Figure 3.28.

The UAV's state machine is structured in six states:

1. STOPPED: This is the UAV's resting state, where all systems are deactivated. The UAV is on the ground, not performing flight operations. This state ensures a safe point for initiating and terminating operations.

2. TAKE_OFF: Activated by a take-off command. The engines will be started and the UAV will ascend to a predefined altitude.

3. LAND: State destined for the UAV's landing.

4. JOY_CONTROL: Allows manual control. The operator uses a joystick to control the movement and adjust the altitude of the UAV. This status is essential if direct human intervention is required.

5. LAND_CONTROL: This is the autonomous landing execution state, where the UAV uses a fiducial marker to identify the exact landing point.

6. FOLLOW_CONTROL: Autonomously follows a fiducial marker, ideal for monitoring or surveillance tasks.

Transitions between states triggered by joystick commands:

1. A: Initiates operations, triggering a take-off command if the uav is stationary, and activates manual control.

2. B: Terminates operations, immediately inducing the transition to ”LAND” from any state, acting as a safety mechanism.

3. X: Activates follow-up mode, assuming that take-off has already taken place.

4. Y: Activates precision landing mode, assuming that take-off has already taken place.

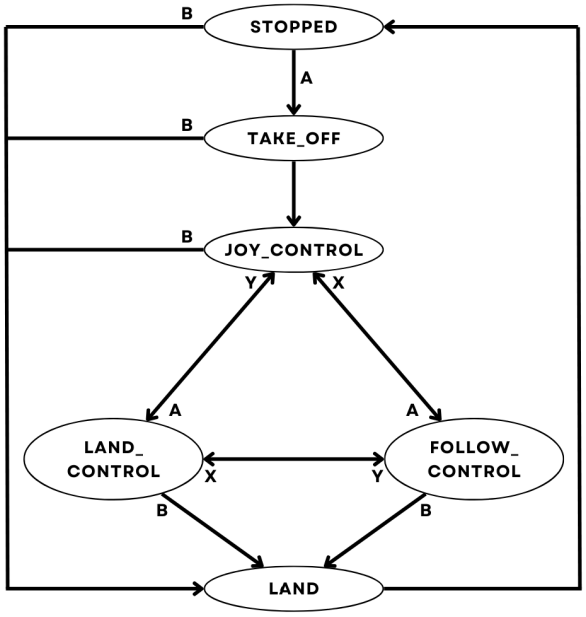

Figure 3.28: State machine

The implementation of this state machine exemplifies the practical application of automation principles in improving the functionality and safety of advanced autonomous systems. With this implementation, operators can perform a variety of complex tasks while maintaining consistent control over the system at any moment.

3.3.2 Definition of controller's gains

As discussed in section 3.2, it is crucial to set the controller parameters precisely to guarantee optimum performance. For this purpose, the approach used in simulations will also be applied in a real-world environment. The calibration for each axis of movement of the Unmanned Aerial Vehicle will be conducted independently. The data collected can be found at Figure 3.29.

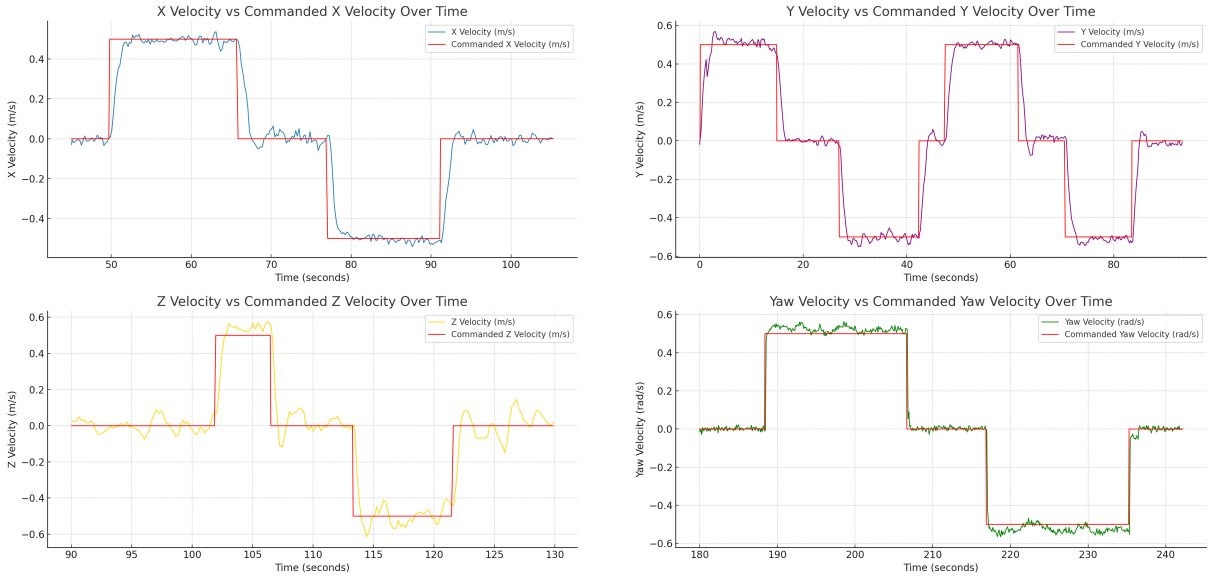

Figure 3.29: Step Response of the Tello UAV System

Based on the step response, an approximation of the transfer functions for each axis of the UAV was defined. Using Matlab's system identification tool again, the UAV's dynamic characteristics were modeled. In the real environment, it was decided to characterize the system using only one pole, without the inclusion of zeros. This process resulted in the transfer functions named $Tello\_TF\_x$ for the X axis, $Tello\_TF\_y$ for the Y axis, $Tello\_TF\_z$ for the Z axis, and $Tello\_TF\_r$ for the rotation, respectively represented in the equations: Equation 3.5, Equation 3.6, Equation 3.7, Equation 3.8.

$$Tello\_TF_x(s) = \frac{1.061}{s + 1.049} \tag{3.5}$$

$$Tello\_TF_y(s) = \frac{1.119}{s + 1.089} \tag{3.6}$$

$$Tello\_TF_z(s) = \frac{2.603}{s + 2.509} \tag{3.7}$$

$$Tello\_TF_{yaw}(s) = \frac{6.028}{s + 5.732} \tag{3.8}$$

A comparison of these transfer functions with the real movement is shown in Figure 3.30. This comparison shows that the transfer functions are accurate approximations of the values obtained experimentally, confirming the suitability of the single-pole model to represent the UAV's dynamics.

Following the methodology described in section 3.2, the Matlab Autotuning system was used to determine the control gains. These gains are detailed in Table 3.9. Through Autotuning, it was possible to automatically calibrate the control parameters, optimizing the UAV's response to commands.

With the adjusted gains, the UAV is expected to operate with greater stability and precision, improving the flight experience and allowing for more precise and controlled maneuvers.

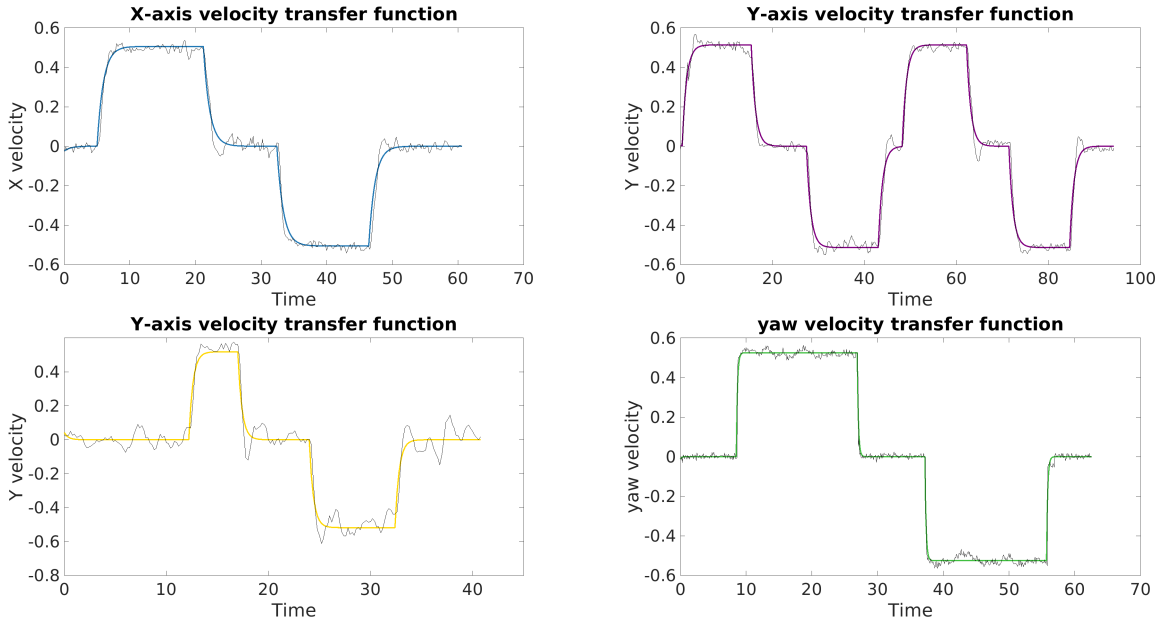

Figure 3.30: transfer functions of the tello UAV system

	RT	TB	KP	KI	KD
X	1.174	0.828	1.991	0.282	1.567
Y	2.154	0.432	1.034	0.1169	0.332
Z	1.177	0.9	1.626	0.4285	0.8012
YAW	1.18	0.855	1.605	0.4559	0.4399

Table 3.9: PID Controller Gains

3.4 Land Approach

The methodology implemented for landing the Unmanned Aerial Vehicle was developed with a focus on precision and safety. This part of the flight is particularly challenging, requiring strict control and a well-defined strategy. As described in the chapter 2 section, the landing process is made up of several critical stages, each playing a vital role in ensuring a successful landing operation.

1. Approach between the UAV and the mobile platform: in this phase, the UAV navigates toward the ground robot, starting the landing process.

2. Precision alignment: at this stage, the aim is to align the UAV as precisely as possible with the mobile platform, ensuring optimal positioning for landing.

3. Effective precision landing: the UAV begins its controlled descent, maintaining the alignment established in the previous stage, ensuring a precise landing.

4. Landing command: at this final point, the UAV performs a vertical descent, regardless of the previous alignment, and deactivates the engines when it touches the ground. Ideally, this phase is carried out with the drone already in contact with the ground to maximize landing accuracy, although this is not always possible.

This work considers that the first stage has already been completed and therefore begins with aligning the aircraft with the mobile base. To perform this alignment, the

Proportional-Integral-Derivative (PID) controller is used, which has been defined in detail in the previous sections of the document.

The UAV reads the position of the fiducial marker placed on the mobile base and adjusts its position to align itself above it. Initially, the alignment is done at a distance of one meter from the marker and, during the landing, this altitude is decreased by one centimeter as soon as the alignment is done correctly.

When the UAV reaches 20 centimeters from the mobile base, the landing command is activated. The UAV descends vertically and deactivates its engines after reaching the landing base. This is the critical moment when the effectiveness of the landing strategy is tested, and the precision of the PID control is essential to ensure that the UAV lands exactly where it is supposed to.

At this last stage, the landing process should ideally be carried out with the drone already on top of the landing pad, but in commercial drones it is common for there to be a minimum flight altitude, as is the case with the Tello where the minimum operating altitude is 30 cm. This minimum altitude limit presents an additional challenge in the landing process. Executing the landing command at an altitude of 20 cm requires even more precision from the PID controller to ensure a successful landing.

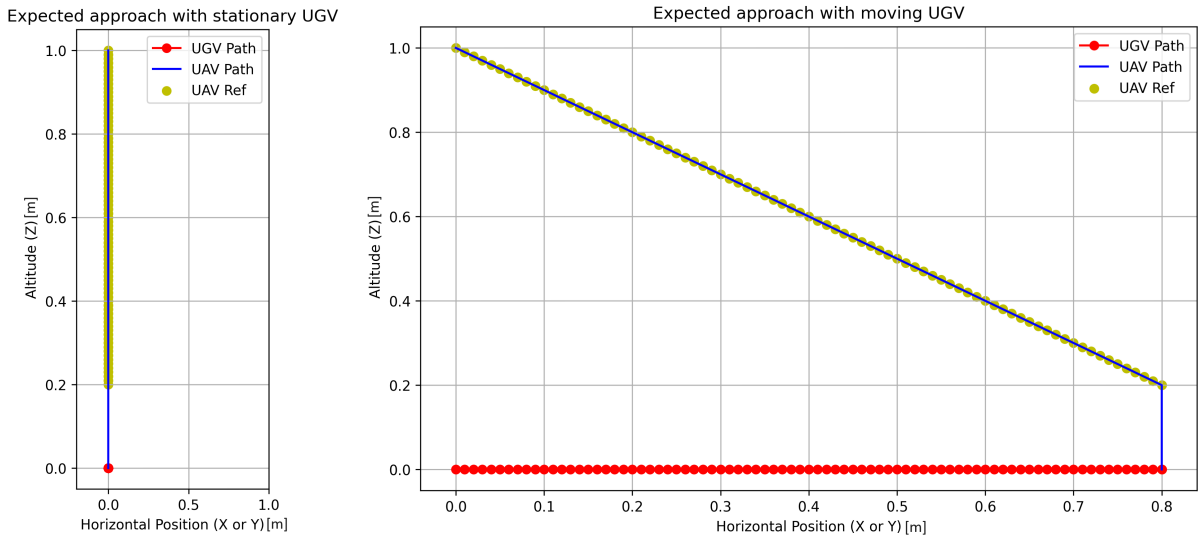

Figure 3.31: Trajectory graph of follow-up mission with linear velocity

Chapter 4

Results

This chapter presents the results of the tests carried out to evaluate the performance of the control system developed. Two types of tests were conducted: Follow-up and Precision Landing. These tests are crucial to ensuring the reliability and effectiveness of drones in critical operations. They make it possible to identify and solve problems in flight control and navigation, ensuring that the drone can operate safely and efficiently in real environments.

4.1 Follow-up Results

The follow-up tests aimed to verify the drone's ability to follow a ground robot at different trajectories and speeds. These tests assess the drone's ability to maintain precise alignment with a moving target on the ground, which is crucial for applications such as security monitoring and tracking moving targets.

4.1.1 Target at constant linear velocity

In the initial testing phase, the Unmanned Ground Vehicle moved forward on a straight trajectory of five meters, before returning to the starting point, maintaining a constant linear velocity in both directions. The focus was on the Unmanned Aerial Vehicle's ability

to maintain a constant distance from the UGV. Four different speeds were used to evaluate the UAV's response. The trajectories executed are shown in Figure 4.1.

It was observed that the UAV followed the UGV effectively at all four speeds tested. Interestingly, in the highest speed test, the UGV made a turn, introducing an angular component to the movement. Despite this, the UAV proved capable of successfully replicating the UGV's curvilinear trajectory.

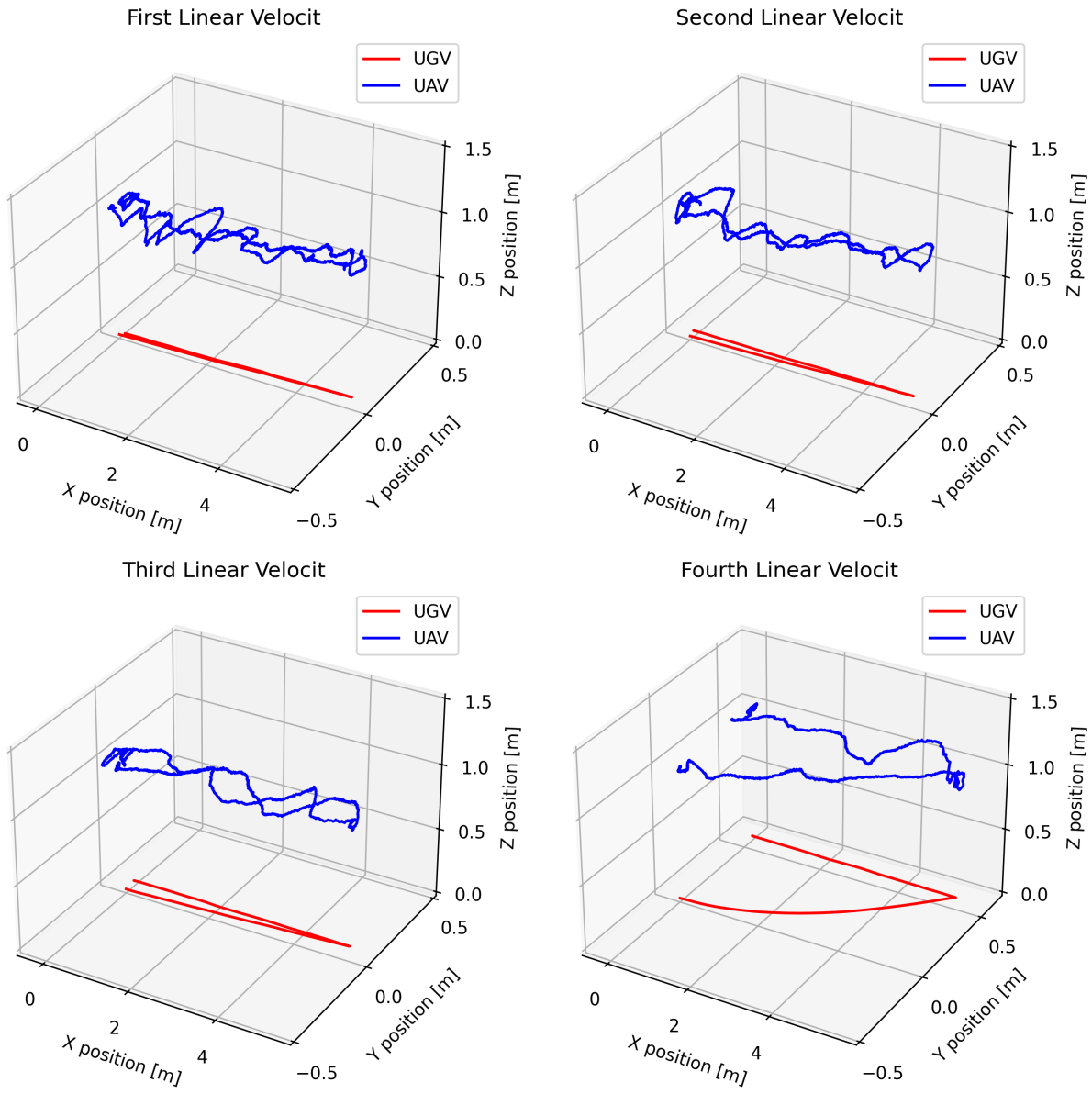

Figure 4.1: Trajectory graph of follow-up mission with linear velocity

For a detailed examination, the graphs representing each axis as a function of time were analyzed, as illustrated in Figure 4.2, which shows the UAV's position at the third speed tested. In the X and Z axes, the UAV showed good alignment with the predicted trajectory, indicating effective longitudinal and vertical tracking. However, on the Y axis, a maximum deviation of around 20 centimeters was recorded, a contrast with the results obtained in simulations, where the misalignment was more noticeable in the direction of movement. Concerning rotation, the UAV remained consistently within an acceptable error of ± 0.1 radians, denoting stable orientation control.

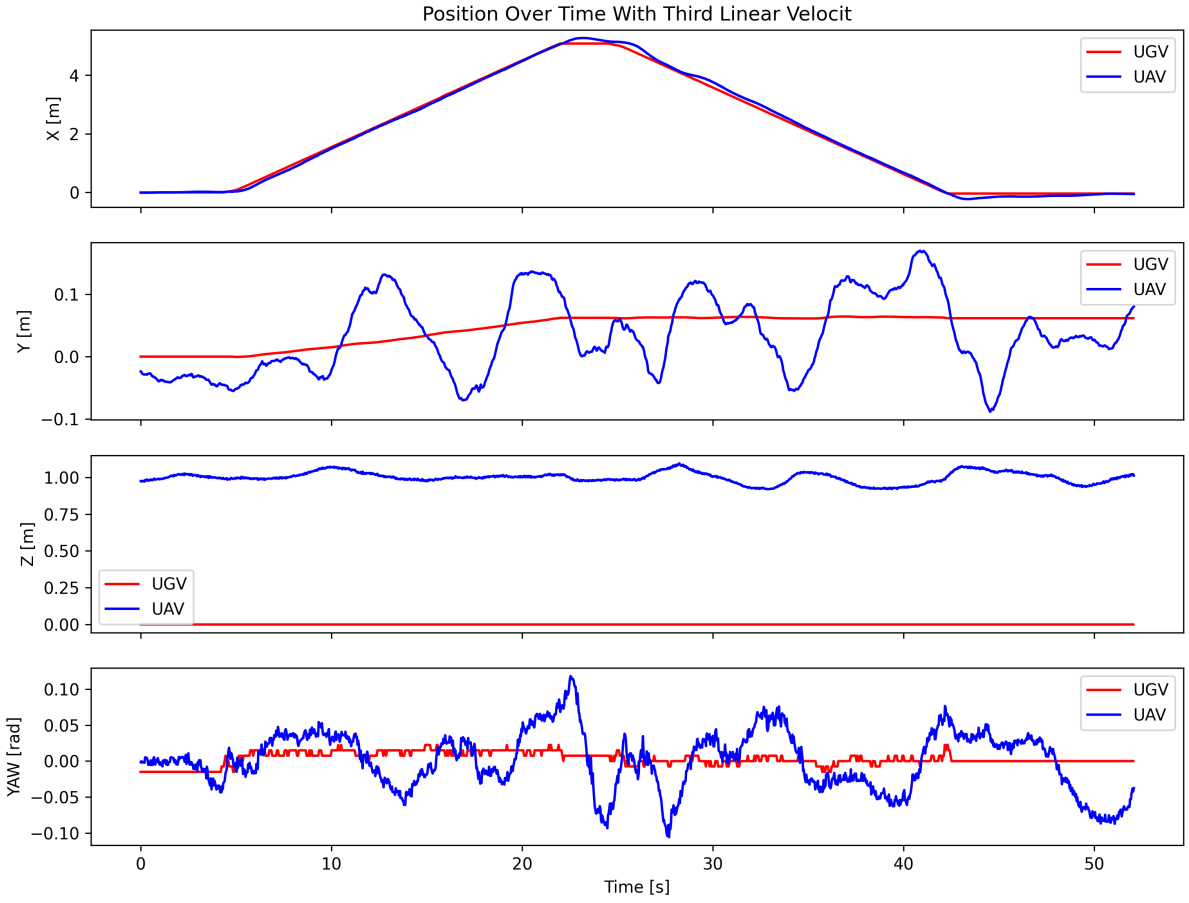

Figure 4.2: Position graph over time in follow-up mission at linear speed

The tests demonstrated the UAV's proficiency in accurately tracking a UGV at constant linear displacement, even with the introduction of smaller angular movements and at varying speeds. The UAV showed a remarkable ability to maintain alignment in the

longitudinal and vertical directions. The need to improve lateral control to minimize the error observed in the Y axis is clear. Maintaining consistent angular orientation with minimal error also underscores the efficiency of the UAV's yaw control system.

Figure 4.3: Real images of the linear trajectory.

4.1.2 Target at constant angular velocity

The test procedures were expanded beyond linear tracking to assess the UAV's competence in following a UGV performing rotational movements at a constant angular velocity. This phase of testing is essential to determine the UAV's ability to maintain vigilance and relative positioning in the face of changes in a target's orientation, a common challenge in dynamic environments.

To complement the tracking tests at linear speeds, the UAV's ability to track a UGV performing constant rotational movements was also tested. This part of the test is crucial to understanding the effectiveness of the UAV's flight system when faced with changes in target orientation, a common situation in dynamic operational environments. In this scenario, the UGV performed a rotation around its axis at a constant angular velocity. The evaluation focused on the UAV's ability to compensate for changes in the UGV's orientation and maintain stable tracking, despite the absence of significant linear displacement.

The UGV performed two complete rotations on its axis during the tests at three

different angular velocities. The UAV's task was to remain aligned with the UGV during these rotations. The captured trajectories are illustrated in Figure 4.4.

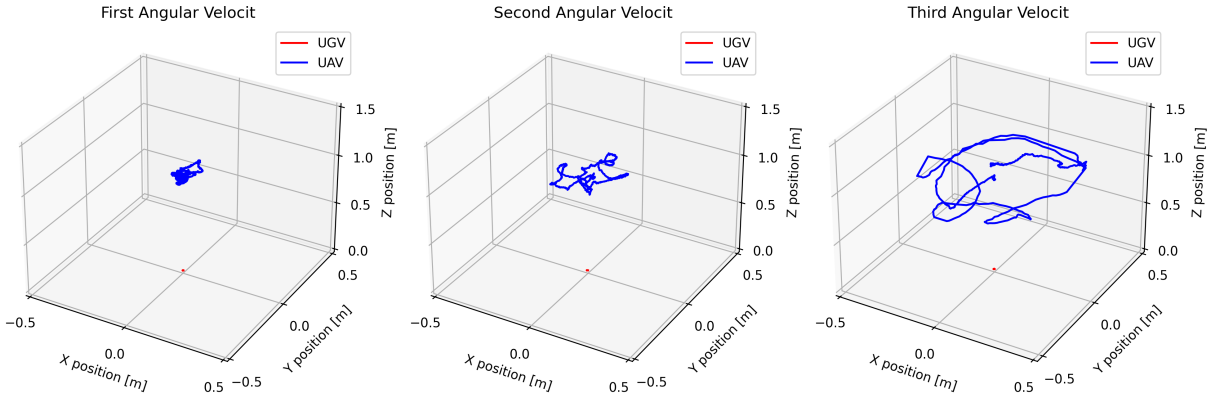

Figure 4.4: Trajectory graph of follow-up mission with angular velocity

The graphical representations in Figure 4.4 show that although the Tello UAV managed to maintain proximity to the Magni during rotations, its effectiveness was compromised as the angular velocity increased. At higher angular velocities, the UAV struggled to maintain perfect alignment with the UGV. However, these graphs are not sufficient to discern whether the UAV maintained the same orientation as the UGV over time; for this, it is necessary to refer to the temporal graphs, presented in Figure 4.5, Figure 4.6 and Figure 4.7.

At the first two speed settings (Figure 4.5 and Figure 4.6), the UAV maintained satisfactory alignment with the UGV, despite a slight delay in response. However, at the highest speed (Figure 4.7), the UAV was unable to synchronize its movements with those of the UGV, resulting in misalignments that impacted the accuracy of the other axes as well.

It is concluded that the Tello UAV is capable of following the Magni UGV at low to moderate angular velocities, but not at high speeds. This limit in the UAV's performance suggests the need for optimizations in the control system, especially to deal with tracking challenges involving high angular velocities.

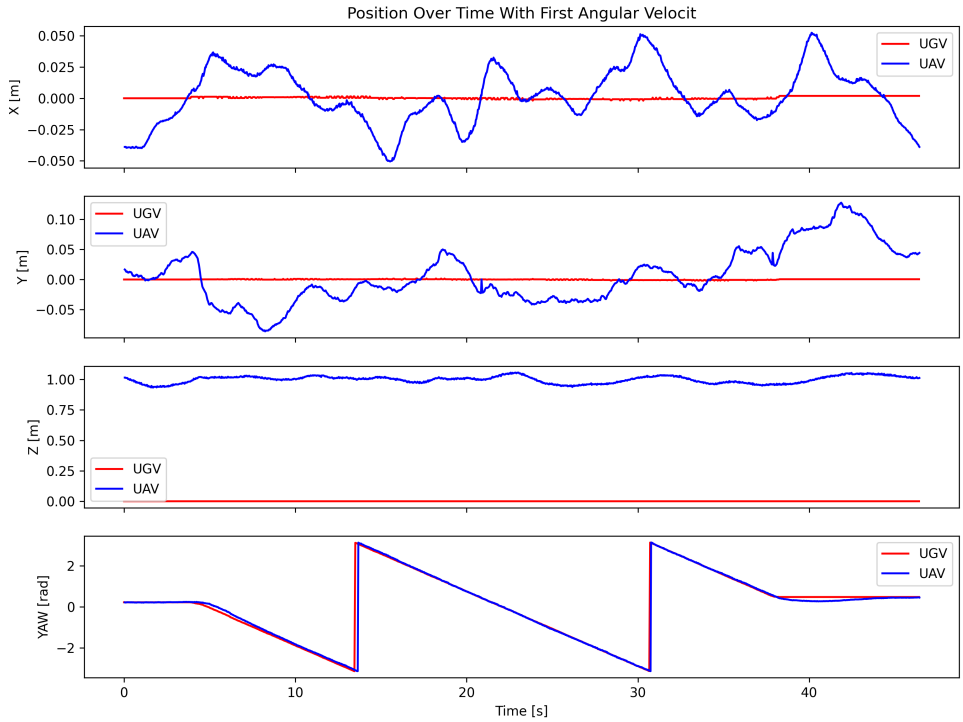

Figure 4.5: Position graph over time in follow-up mission at first angular speed

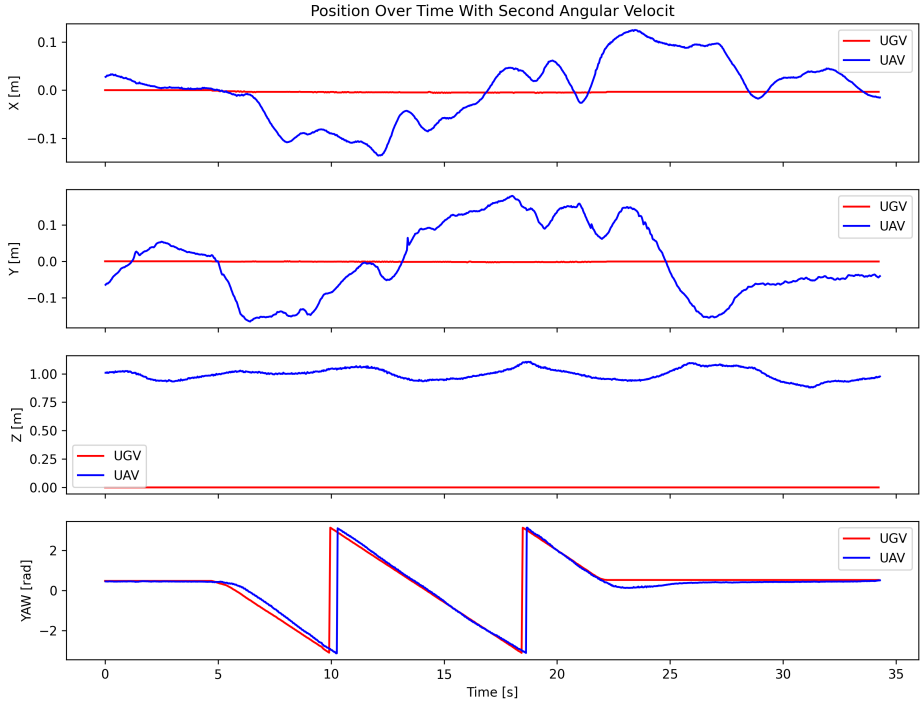

Figure 4.6: Position graph over time in follow-up mission at second angular speed

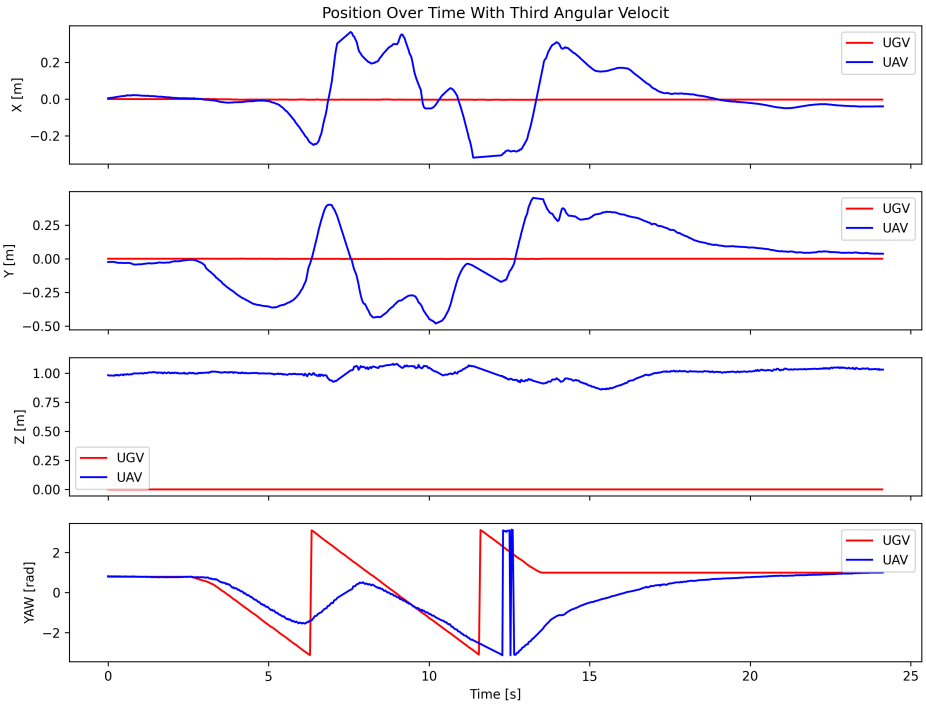

Figure 4.7: Position graph over time in follow-up mission at third angular speed

4.2 Precision landing results

This section evaluates the UAV's ability to perform precise landings in different conditions: with the landing pad stationary, in constant linear movement, with interrupted movement, and with combined linear and angular movements.

The objectives of these tests are: first, to measure landing accuracy, determined by the proximity of the UAV's landing point to the center of the fiducial marker; second, to evaluate the repeatability and reliability of the drone's control system in meeting various landing challenges; and third, to analyze the path taken by the UAV during the landing procedure.

To ensure a comprehensive evaluation, the landing procedure was repeated ten times in each scenario. These tests are vital to understanding the effectiveness of the UAV's control and navigation algorithms. In addition, they are key to identifying areas that need future improvement, with a view to applications in real, dynamic environments where landing accuracy and reliability are essential.

4.2.1 Stationary target

In this test, the focus is on assessing the UAV's ability to make an accurate landing on the stationary landing pad. To begin the test, the UAV was positioned at a height of one meter directly over the magni, already aligned with the marker. From this initial position, the landing procedure began. The aim was for the UAV to land exactly on the marker, minimizing deviation from the marker's center.

This test configuration was chosen to simulate a controlled landing scenario, where the UAV has a direct descent path without the need for additional alignment maneuvers. This scenario represents the most basic case of precise landing, serving as a baseline for evaluating the UAV's performance in more complex conditions.

Repeatability and precision

In this initial test, all ten landings were successful, with the UAV landing correctly on the ground robot. The positions of the Tello's camera after landing, to the center of the marker, are detailed in Table 4.1.

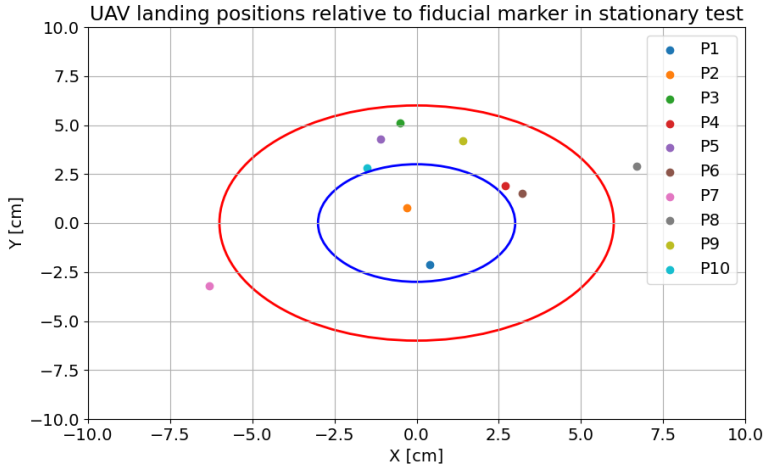

Figure 4.8: UAV Landing Positions Relative to Fiducial Marker in Stationary Target Test

In Figure 4.8, the graphical analysis reveals that most of the final positions are within the red circle, which has a radius of 6 centimeters, while two are located within the

blue circle, with a radius of 3 centimeters. This pattern suggests that the UAV showed good landing accuracy, always landing very close to the marker, which is considered a satisfactory result.

	P1	**P2**	**P3**	**P4**	**P5**	**P6**	**P7**	**P8**	**P9**	**P10**
X (cm)	0.4	-0.3	-0.5	2.7	-1.1	3.2	-6.3	6.7	1.4	-1.5
Y (cm)	-2.1	0.8	5.1	1.9	4.3	1.5	-3.2	2.9	4.2	2.8

Table 4.1: UAV landing position with stationary landing base

UAV path

The trajectory covered during the P6 landing is illustrated in Figure 4.9. This figure shows that the UAV followed a consistent and almost linear route, with a slight disturbance at the beginning, attributed to the initial alignment. On the other hand, Figure 4.9 shows the UAV's position as a function of time. Here, there is a slight deviation in Tello's rotation, although the other axes behave adequately. It is also worth noting that the total landing time was approximately 6 seconds, which is efficient.

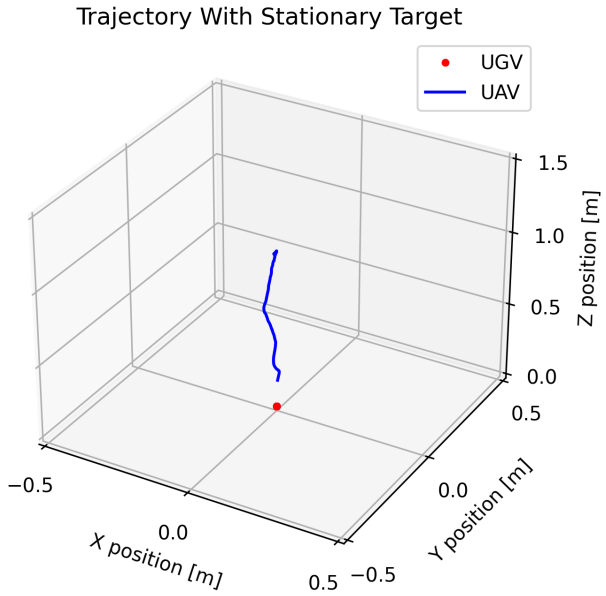

Figure 4.9: Visualization of UAV's Flight Path in 3D During Precision Landing on the stationary test

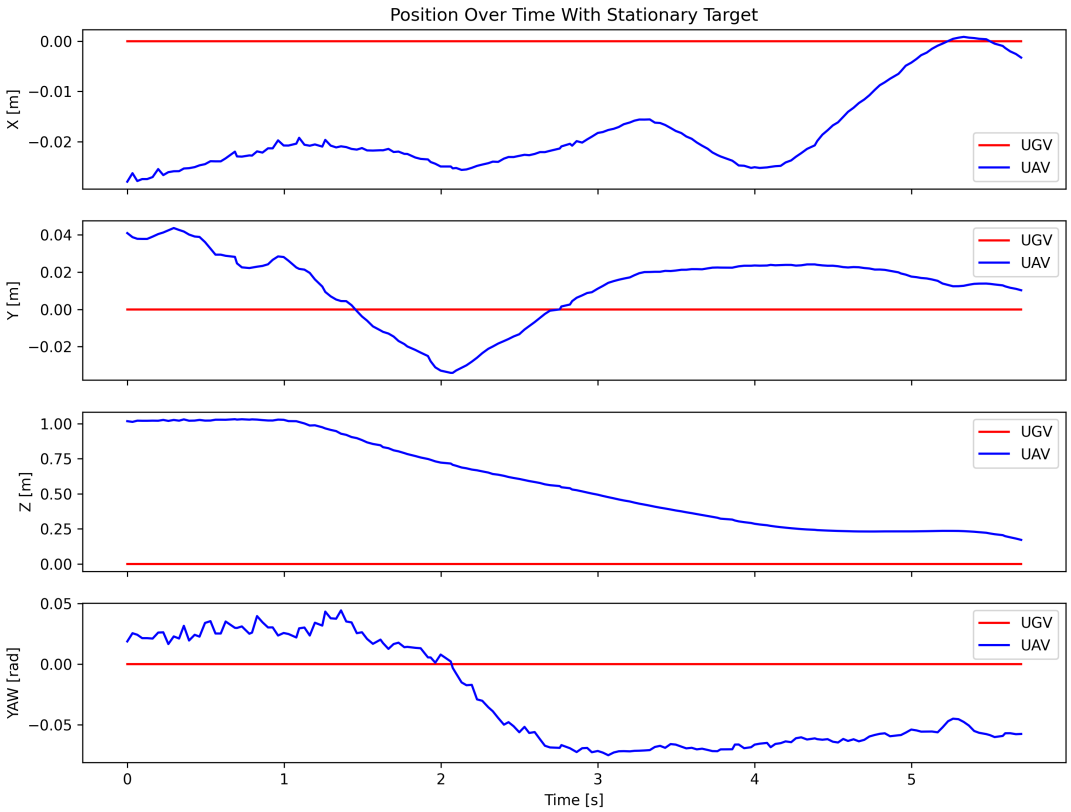

Figure 4.10: Visualization of UAV's Position Over Time During Precision Landing on the stationary test

4.2.2 Target at constant linear velocity

In the second test, the aim was to test Tello's ability to land accurately on a landing base that moves linearly at a constant speed of 0.2 m/s, without making any rotational movements. The procedure began with aligning the Tello with the fiducial marker. Once this alignment was achieved, the Magni moved linearly and then the Tello initiated the landing procedure. Throughout the landing operation, the Magni maintained its constant speed.

This test put a significant demand on the UAV's control system. To ensure a successful landing, it is crucial that the UAV not only aligns itself with the fiducial marker, but also synchronizes its movement with that of the moving landing pad, implying a complex coordination between the two architectures.

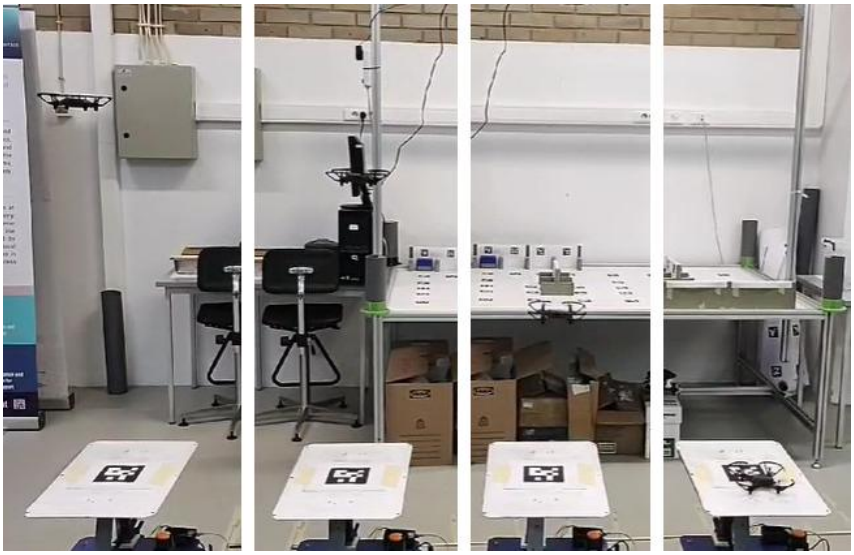

Figure 4.11: UAV Landing Perform in Linear Motion

Repeatability and precision

Again, in this test, the UAV landed correctly on the ground robot in all ten experiments. Tello's camera positions after landing are detailed in Table 4.2.

	P1	**P2**	**P3**	**P4**	**P5**	**P6**	**P7**	**P8**	**P9**	**P10**
X (cm)	1.9	-3	-3	6.3	-3	5.3	8.4	1.1	-0.1	-0.1
Y (cm)	0.4	-2.5	-2.8	1.2	0.1	-0.8	-1.5	-2.2	-5.5	-4.3

Table 4.2: UAV landing position with linear motion landing base

Figure 4.12 shows that most of the lands were inside the red circle and three were inside the blue one. Again, this suggests that the UAV demonstrated good landing accuracy. However, unlike the first test, this time most of the landings were made at the bottom of the marker, which can be explained by the problem mentioned earlier that, because it is a limited aerial robot, it is necessary to start the landing 20 centimeters from the marker, so if the UAV is aligned with the ground robot, after landing it will be a few centimeters behind.

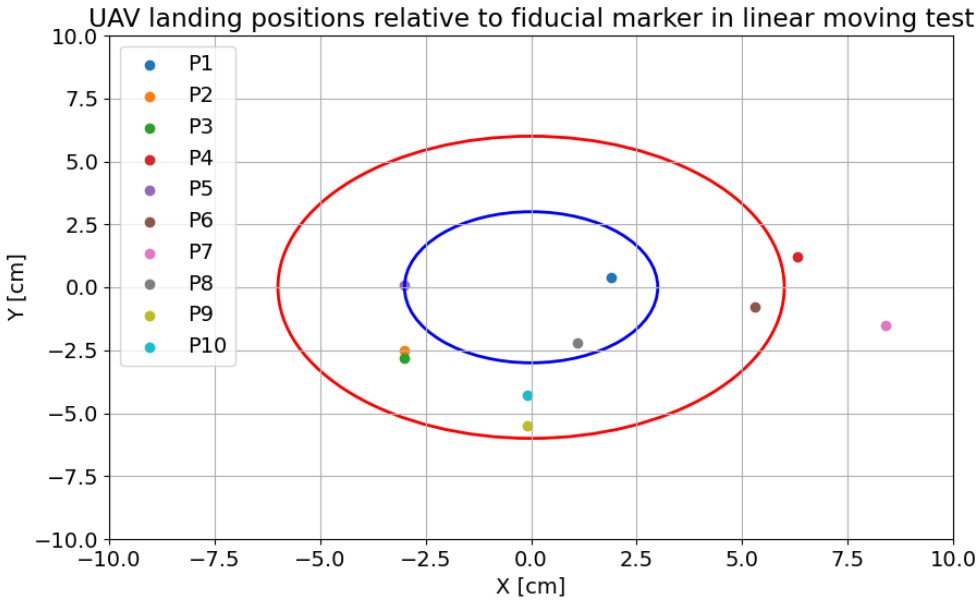

Figure 4.12: UAV Landing Positions Relative to Fiducial Marker in Linear Motion Target Test

UAV path

The trajectory of the P5 landing is illustrated in Figure 4.13. Again, the UAV followed a route consistent with the movements of the UGV, clearly showing the decrease in altitude during the movement of the landing base. In Figure 4.14 there is a slight deviation in the Y axis, which was also observed in the follow-up tests. The landing time was approximately 9 seconds, as expected, slightly longer than the static test, and the ground robot covered more than 1.5 meters during the procedure.

Looking at all the landing attempts in this experiment, it was observed that the Magni moved between 120 cm and 250 cm, with this maximum occurring in experiments with some disturbance in the Tello's camera image. Considering this, it is concluded that the control system is adequately capable of landing in this scenario.

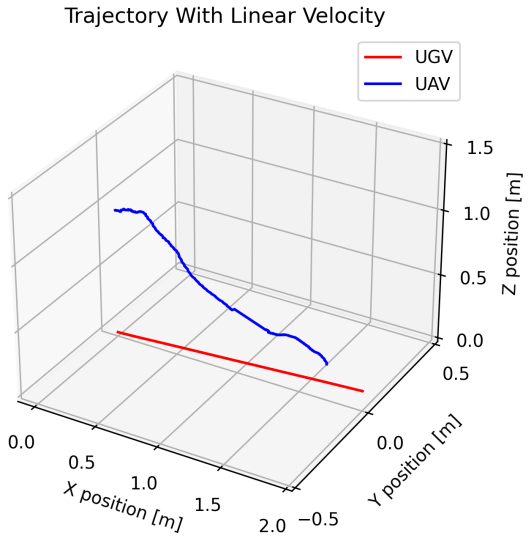

Figure 4.13: Visualization of UAV's Flight Path in 3D During Precision Landing on the linear moving test

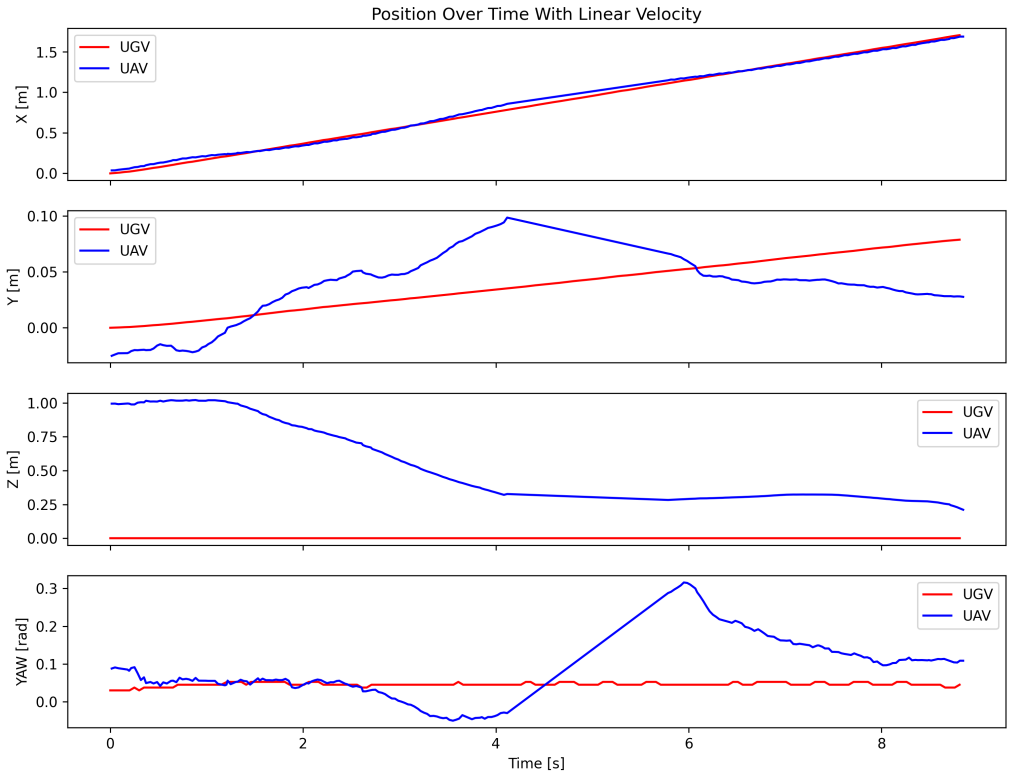

Figure 4.14: Visualization of UAV's Position Over Time During Precision Landing on the linear moving test

4.2.3 Target with abruptly stopped

This phase of the tests was designed to assess how the UAV (Tello) reacts to sudden changes in the target's movement. This scenario simulates real situations in which the target may stop unexpectedly, requiring the UAV to quickly adapt to complete the landing successfully. First, the Tello is aligned with the fiducial marker. Next, the Magni starts moving forward at a constant speed of 0.2 meters per second. During this movement, the Tello begins its precision landing procedure. In this test, there is a variation: after covering a distance of 120 centimeters, the Magni abruptly stops its movement.

This specific distance of 120 centimeters was selected based on the observations of the second experiment. In this experiment, it was noted that at this distance, the UAV had already reduced its altitude to the point where it was almost reaching the landing base. Therefore, the Magni's sudden stop at this point is crucial to test Tello's ability to adapt quickly to unexpected changes, while maintaining landing accuracy despite the target's unpredictable behavior.

Repeatability and precision

As Table 4.3 shows, this test had some failures on landing, three out of ten attempts failed. This shows that the system has certain limitations in adapting to abrupt changes in movement. These errors are attributed to the fact that the UAV loses sight of the marker due to the UGV stopping abruptly while the Tello is close to the marker. At the moment of stopping, Tello suffers an overshoot due to the PID controller and maintains movement, so as it is close to the marker, it loses its position reference and is unable to complete the landing correctly.

	P1	**P2**	**P3**	**P4**	**P5**	**P6**	**P7**	**P8**	**P9**	**P10**
X (cm)	-2.5	fail	7.3	-1.3	-2.3	fail	fail	-1.2	-2.1	4.3
Y (cm)	0	fail	-0.5	0.6	-1.9	fail	fail	-1.6	-3.7	-1.6

Table 4.3: UAV Landing Position with Abruptly Stopped Landing Base

In Figure 4.15, there are several landing points (P1, P4, P5 and P8) inside or very close to the blue circle, suggesting that these landings were very successful. Other points (P9 and P10) are located outside the blue circle, but inside the red circle, indicating less precision, but still within an acceptable zone. There is one point (P3) that is noticeably further away from the others, suggesting a landing attempt with greater deviation.

In this experiment, the successful landings did not show the same pattern as in the previous test because at the time of the land command, the UGV was not moving, indicating that the justification given was correct.

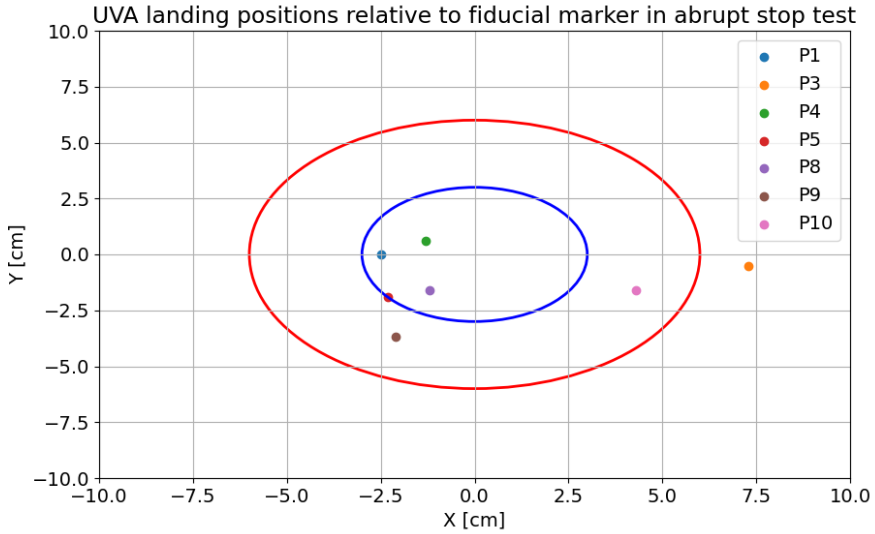

Figure 4.15: UAV Landing Positions Relative to Fiducial Marker in Abruptly Stopped Target Test

UAV path

The trajectory of the P5 landing is illustrated in the Figure 4.16, it can be seen that during the constant movement, the UAV follows the UGV correctly, however, at the moment of stopping, there is a significant oscillation, although it managed to resume the landing.

The Figure 4.17 shows some interesting points, firstly, concerning the X axis, it shows disturbances in the UAV's section. It is also noticeable that the Tello's altitude decreases during the movement until, at the moment of stopping, it increases slightly until it aligns

and decreases again. The landing time was approximately 18 seconds, much longer than the static test.

The results of this test reveal the importance of a robust navigation and control system for UAVs, capable of dealing with dynamic changes in the environment. Improving the UAV's ability to respond quickly to unforeseen changes in the target's movement can significantly increase reliability and safety in real operations. In addition, it shows the importance of including an error-checking system, which signals when an error occurs in the landing and treats this error to try to resume the landing.

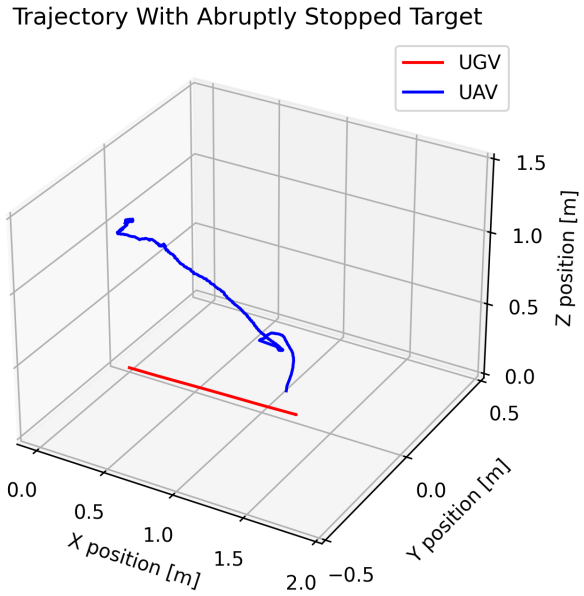

Figure 4.16: Visualization of UAV's Flight Path in 3D During Precision Landing on the Abruptly Stopped test

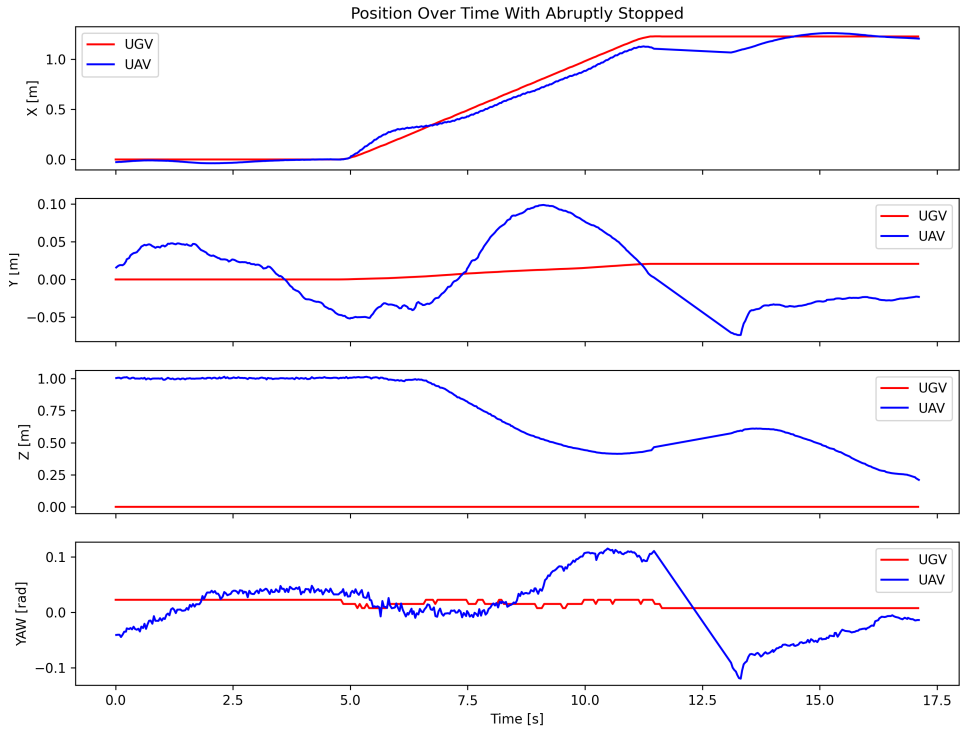

Figure 4.17: Visualization of UAV's Position Over Time During Precision Landing on the Abruptly Stopped test

4.2.4 Target with linear and angular movement

The objective of these experiments was to evaluate the UAV's ability to perform an accurate landing on a target that not only moves linearly but also incorporates angular movement. This test aims to stress the UAV's control system, challenging it to compensate for angular variations in addition to the linear position.

The test began with the UGV (Magni) moving linearly at a constant speed of 0.2 meters per second. After traveling 80 centimeters, the UGV inserts an angular movement at a speed of 0.1 radians per second until it completes a quarter circle and then stops. During this time, the UAV (Tello) was tasked with making the landing, which required the controller to make corrections not only to the position but also to the orientation in response to the UGV's angular movement.

Figure 4.18: UAV Landing Perform in Angular Motion

Repeatability and precision

The complexity of the test, which combined linear and angular movement of the target, resulted in a higher error rate, where 4 out of 10 landings resulted in failures, Table 4.4. A crucial observation was that the failed landings were associated with the UAV's previously mentioned minimum flight altitude limitation. In these attempts, the UAV started the

landing procedure, but the UGV left its position below Tello before the landing could be completed, causing a misalignment between the UAV and the target.

	P1	**P2**	**P3**	**P4**	**P5**	**P6**	**P7**	**P8**	**P9**	**P10**
X (cm)	16.0	fail	-15.9	3.5	fail	fail	5.3	5.0	fail	14.0
Y (cm)	1.5	fail	4.0	6.4	fail	fail	2.4	11.0	fail	-1.5

Table 4.4: UAV landing position with angular moving landing base

In 4.19 it shows that most of the landings were outside the red circle, reinforcing that this test did not have a good result in terms of either repeatability or accuracy. Most of the landings were made in the direct upper part of the landing base, which can be explained by Tello's inability to update its position at the end of the process.

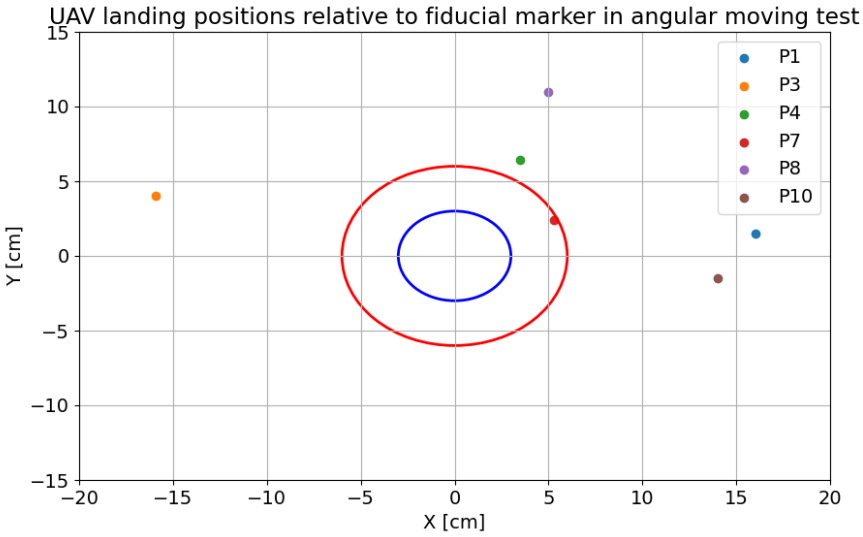

Figure 4.19: UAV Landing Positions Relative to Fiducial Marker in angular moving target test

UAV path

The P8 landing, Figure 4.13, shows a route consistent with the UGV's movements and shows the decrease in altitude during the circular movement and landing at the moment when the ground robot has stopped. The graphs to time, Figure 4.14, show that the UAV

was able to mimic the movements of the UGV very well, with only a small deviation on the Y axis.

In this experiment, the UAV took approximately 20 seconds to land and was only able to complete the procedure once the movements had been completed.

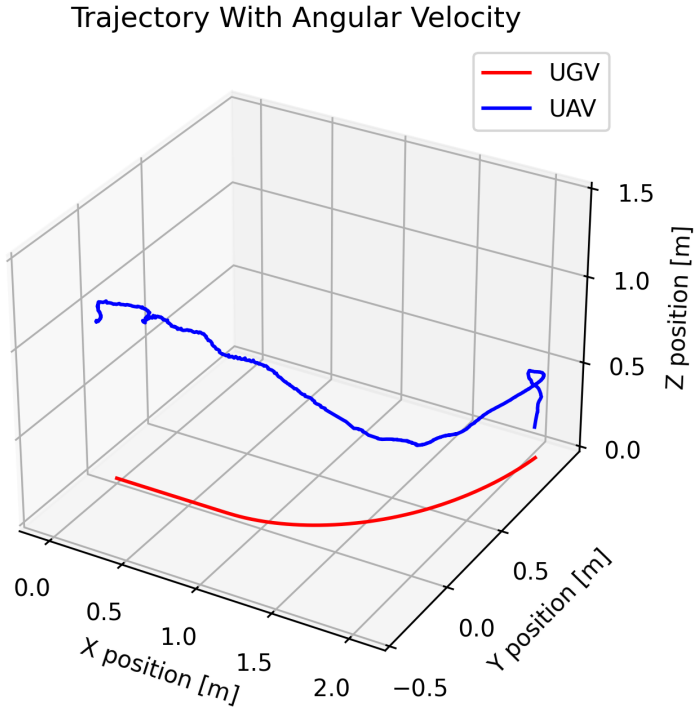

Figure 4.20: Visualization of UAV's Flight Path in 3D During Precision Landing on the angular moving test

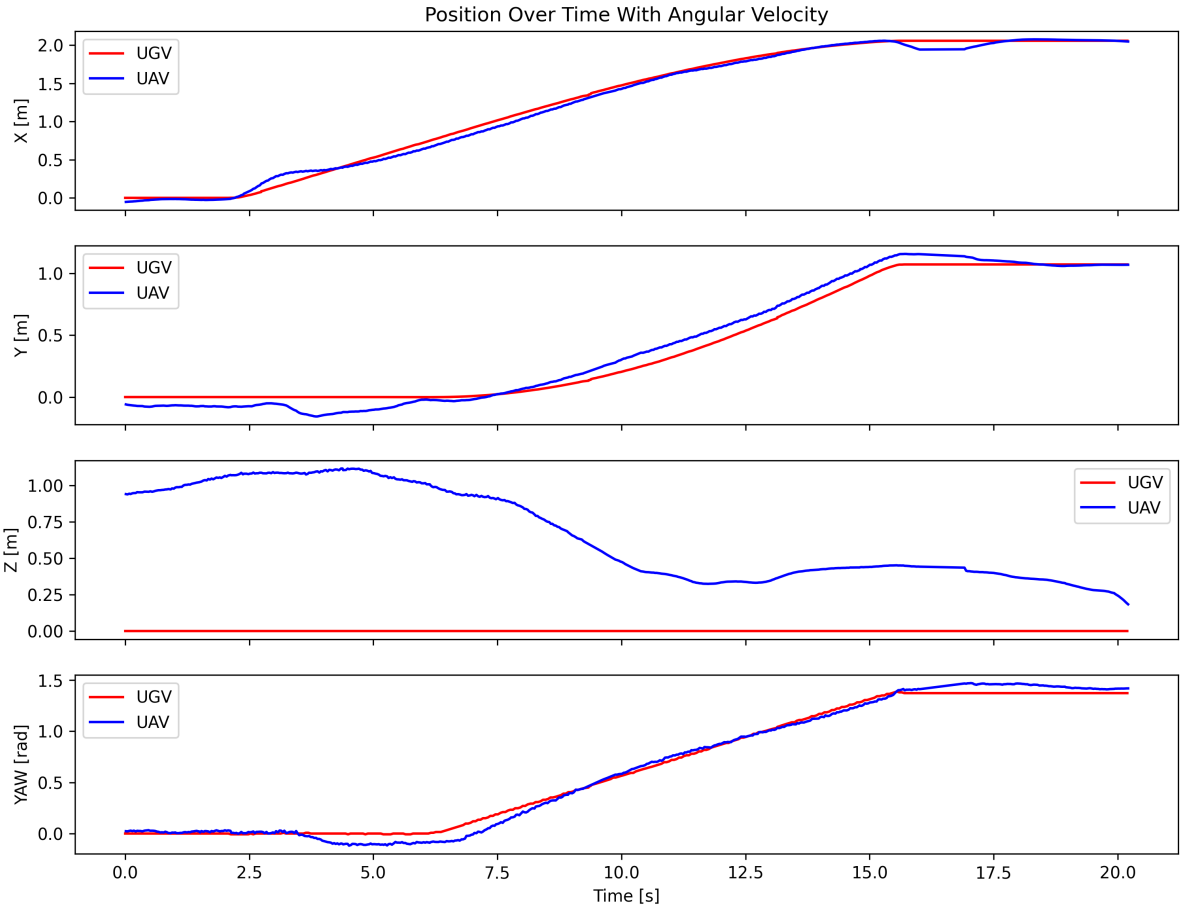

Figure 4.21: Visualization of UAV's Position Over Time During Precision Landing on the angular moving test

Chapter 5

Conclusion and Future Work

This dissertation addressed the development of a precision landing methodology for unmanned aerial vehicles (UAVs), focusing on static and moving targets, using PID controllers and fiducial markers. This methodology involved defining the most suitable fiducial marker for the application, the best performance control architecture, and a technique for defining the controllers' gains. In addition, the development of a control system applicable to different UAVs can be expanded for different activities.

Different configurations of controllers and markers were analyzed and it was concluded that the use of a complete PID together with Aruco markers would have better performance. Then the testing stage began, in which the efficiency of the controller in making the aerial robot autonomously follow the UGV was verified. In this respect, it was concluded that the UAV was able to keep in line with the UGV with good precision at contained speeds, while the rotation control performed worse.

This was followed by landing tests in some different scenarios, which showed good performance in landings with smooth, steady movements, but revealed the importance of a navigation and control system with better capabilities for dealing with dynamic changes in the environment. The limitations of commercial drones were also clear: they are good for initial tests, but they create barriers to development and reduce the efficiency of completing important tasks during cooperation.

5.1 Future works

For future work, tests using customizable aircraft are of the essence, allowing the control system to perform better due to different configurations and the insertion of different sensors. In addition, tests in outdoor environments are important due to their wide applicability, requiring the integration of solutions for different variables such as lighting and wind.

In the context of the system's application, it is important to integrate the developed controller into different mission planning modules, including navigation, path planning, obstacle detection, coordination between multiple robots of different architectures, and so on. In addition, it focuses on the ground robot, and the development of equipment to enable autonomous recharging of aerial robots.

5.2 Articles

During the research period for the development of the system discussed in this dissertation, the diffusion of information was an objective, resulting in the publication of many papers at conferences. These articles not only reinforce the relevance of the work carried out but also contribute to the scientific community.

Among these publications, two papers are directly related to this dissertation:

- Luciano Bonzatto junior, Guido Szekir Berger, et al. "A comparison of PID controller architectures applied in autonomous UAV follow up of UGV". Paper presented in Sixth Iberian Robotics Conference ROBOT 2023, Coimbra, 2023.

- Luciano Bonzatto junior, Guido Szekir Berger, et al. "A Comparison of Fiducial Markers Pose Estimation for UAVs Indoor Precision Landing". Paper presented in International Conference on Optimization, Learning Algorithms and Applications (OL2A), Açores, 2023.

In addition, two articles focused on the application of UAVs in precision agriculture, an emerging field of great practical relevance.

- Guido Szekir Berger; Luciano Bonzatto junior; Milena F. Pinto; Alexandre de Oliveira Júnior; João Mendes; Ana Pereira; Yago M. R. da Silva; Antônio Valente; José Lima. "UAV-Assisted Navigation for Insect Traps in Olive Groves". Paper presented in Sixth Iberian Robotics Conference ROBOT 2023, Coimbra, 2023.

- Guido Szekir Berger; João Mendes; Chellal, Arezki Abderrahim; Luciano Bonzatto junior; Yago M. R. da Silva; Zorawski, Matheus; Ana Pereira; et al. "A YOLO-based insect detection: Potential use of small multirotor Unmanned Aerial Vehicles (UAVs) monitoring". Paper presented in International Conference on Optimization, Learning Algorithms and Applications (OL2A), Açores, 2023.

And, two articles were dedicated to the construction and planning of ground robots, both presented at the CLAWAR 2023 conference:

- Braun, João. "Energy Efficiency Analysis of Differential and Omnidirectional Robotic Platforms: A Comparative Study". Paper presented in CLAWAR 2023 - 26th issue of the International Conference Series on Climbing and Walking Robots and the Support Technologies for Mobile Machines., 2023

- Braun, João. "Design and Development of an Omnidirectional Mecanum Platform for the RobotAtFactory 4.0 Competition". Paper presented in CLAWAR 2023 - 26th issue of the International Conference Series on Climbing and Walking Robots and the Support Technologies for Mobile Machines., 2023.

In conclusion, the articles published during this research period not only reflect the depth and breadth of the work carried out but also establish a course for future research and practical applications in the areas of mobile robotics.

Bibliography

[1] Y. Khosiawan, A. Khalfay, and I. Nielsen, "Scheduling unmanned aerial vehicle and automated guided vehicle operations in an indoor manufacturing environment using differential evolution-fused particle swarm optimization," *International journal of advanced robotic systems*, vol. 15, no. 1, p. 1 729 881 417 754 145, 2018.

[2] C. C. Ribeiro, L. H. dos Santos, and D. G. Macharet, "Collaborative ugv/uav path planning for inventory management in warehouses," in *2022 Latin American Robotics Symposium (LARS), 2022 Brazilian Symposium on Robotics (SBR), and 2022 Workshop on Robotics in Education (WRE)*, IEEE, 2022, pp. 121–126.

[3] L. Xin, Z. Tang, W. Gai, and H. Liu, "Vision-based autonomous landing for the uav: A review," *Aerospace*, vol. 9, no. 11, 2022, ISSN: 2226-4310. DOI: 10.3390/aerospace9110634. [Online]. Available: https://www.mdpi.com/2226-4310/9/11/634.

[4] M. A. Niloy, A. Shama, R. K. Chakrabortty, *et al.*, "Critical design and control issues of indoor autonomous mobile robots: A review," *IEEE Access*, vol. 9, pp. 35 338–35 370, 2021.

[5] J. Zhang, X. Yang, W. Wang, J. Guan, L. Ding, and V. C. Lee, "Automated guided vehicles and autonomous mobile robots for recognition and tracking in civil engineering," *Automation in Construction*, vol. 146, p. 104 699, 2023.

[6] L. Zhang, Z. Cai, Y. Yan, C. Yang, and Y. Hu, "Multi-agent policy learning-based path planning for autonomous mobile robots," *Engineering Applications of Artificial Intelligence*, vol. 129, p. 107 631, 2024.

[7] J. Braun, L. Piardi, T. Brito, *et al.*, "Indoor environment monitoring in search of gas leakage by mobile robot," in *Robot 2019: Fourth Iberian Robotics Conference: Advances in Robotics, Volume 2*, Springer, 2020, pp. 339–350.

[8] H. Liu, W. Dong, Z. Zhang, C. Wang, R. Li, and Y. Gao, "Optimization-based local planner for a nonholonomic autonomous mobile robot in semi-structured environments," *Robotics and Autonomous Systems*, vol. 171, p. 104 565, 2024.

[9] M. Preetha, A. Archana, A. Suresh, T. Kalaichelvi, and M. Venkatesan, "A preliminary analysis by using fcga for developing low power neural network controller autonomous mobile robot navigation," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 12, no. 9s, pp. 39–42, 2024.

[10] J. Braun, J. Mendes, A. I. Pereira, J. Lima, and P. Costa, "Object detection for indoor localization system," in *International Conference on Optimization, Learning Algorithms and Applications*, Springer, 2022, pp. 788–803.

[11] J. Braun, A. O. Júnior, G. Berger, *et al.*, "A robot localization proposal for the robotatfactory 4.0: A novel robotics competition within the industry 4.0 concept," *Frontiers in Robotics and AI*, vol. 9, p. 1 023 590, 2022.

[12] L. C. Klein, J. Braun, J. Mendes, *et al.*, "A machine learning approach to robot localization using fiducial markers in robotatfactory 4.0 competition," *Sensors*, vol. 23, no. 6, p. 3128, 2023.

[13] J. Saunders, S. Saeedi, and W. Li, "Autonomous aerial robotics for package delivery: A technical review," *Journal of Field Robotics*, vol. 41, no. 1, pp. 3–49, 2024.

[14] F. Ferreira and G. Ferri, "Marine robotics competitions: A survey," *Current Robotics Reports*, vol. 1, pp. 169–178, 2020.

[15] L. Brancalião, J. Gonçalves, M. Á. Conde, and P. Costa, "Systematic mapping literature review of mobile robotics competitions," *Sensors*, vol. 22, no. 6, p. 2160, 2022.

[16] G. Roggi, S. Meraglia, and M. Lovera, "Leonardo drone contest autonomous drone competition: Overview, results, and lessons learned from politecnico di milano team," *Journal of Intelligent & Robotic Systems*, vol. 108, no. 2, p. 23, 2023.

[17] A. Bermúdez, R. Casado, G. Fernández, M. Guijarro, and P. Olivas, "Drone challenge: A platform for promoting programming and robotics skills in k-12 education," *International Journal of Advanced Robotic Systems*, vol. 16, no. 1, p. 1 729 881 418 820 425, 2019.

[18] J. Braun, A. O. Júnior, G. S. Berger, J. Lima, A. I. Pereira, and P. Costa, "Robotat-factory 4.0: A ros framework for the simtwo simulator," in *2022 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, IEEE, 2022, pp. 205–210.

[19] L. E. Luiz, L. Pilarski, K. Baidi, *et al.*, "Robot at factory lite-a step-by-step educational approach to the robot assembly," in *Iberian Robotics Conference*, Springer, 2022, pp. 550–561.

[20] G. G. d. Castro, G. S. Berger, A. Cantieri, *et al.*, "Adaptive path planning for fusing rapidly exploring random trees and deep reinforcement learning in an agriculture dynamic environment uavs," *Agriculture*, vol. 13, no. 2, p. 354, 2023.

[21] R. Sharma and R. Arya, "Uav based long range environment monitoring system with industry 5.0 perspectives for smart city infrastructure," *Computers & Industrial Engineering*, vol. 168, p. 108 066, 2022, ISSN: 0360-8352. DOI: https://doi.org/10.1016/j.cie.2022.108066.

[22] T. Benarbia and K. Kyamakya, "A literature review of drone-based package delivery logistics systems and their implementation feasibility," *Sustainability*, vol. 14, no. 1, 2022, ISSN: 2071-1050. DOI: 10.3390/su14010360. [Online]. Available: https://www.mdpi.com/2071-1050/14/1/360.

[23] G. S. Berger, J. Braun, A. O. Júnior, *et al.*, "Sensor architecture model for unmanned aerial vehicles dedicated to electrical tower inspections," in *Optimization, Learning*

Algorithms and Applications: Second International Conference, OL2A 2022, Póvoa de Varzim, Portugal, October 24-25, 2022, Proceedings, Springer, 2023, pp. 35–50.

[24] S. A. H. Mohsan, M. A. Khan, F. Noor, I. Ullah, and M. H. Alsharif, "Towards the unmanned aerial vehicles (uavs): A comprehensive review," *Drones*, vol. 6, no. 6, 2022, ISSN: 2504-446X. DOI: 10.3390/drones6060147. [Online]. Available: https://www.mdpi.com/2504-446X/6/6/147.

[25] M. F. Santos, L. M. Honório, A. P. G. M. Moreira, M. F. Silva, and V. F. Vidal, "Fast real-time control allocation applied to over-actuated quadrotor tilt-rotor," *Journal of Intelligent & Robotic Systems*, vol. 102, no. 3, p. 65, 2021.

[26] A. M. Ali, M. A. Ngadi, R. Sham, and I. I. Al_Barazanchi, "Enhanced qos routing protocol for an unmanned ground vehicle, based on the aco approach," *sensors*, vol. 23, no. 3, p. 1431, 2023.

[27] X. Zuo, J. Zhou, F. Yang, F. Su, H. Zhu, and L. Li, "Real-time global action planning for unmanned ground vehicle exploration in three-dimensional spaces," *Expert Systems with Applications*, vol. 215, p. 119 264, 2023.

[28] S. Lu, Y. Jiang, X. Xu, H. Qian, and W. Zhang, "Adaptive heading control strategy for unmanned ground vehicle with variable wheelbase based on robust-active disturbance rejection control," *Control Engineering Practice*, vol. 142, p. 105 786, 2024.

[29] D. Szpaczyńska, M. Łopatka, and P. Krogul, "The running gear construction impact on overcoming obstacles by light high-mobility tracked ugv," *Journal of Terramechanics*, vol. 112, pp. 1–17, 2024.

[30] G. S. Berger, M. Teixeira, A. Cantieri, *et al.*, "Cooperative heterogeneous robots for autonomous insects trap monitoring system in a precision agriculture scenario," *Agriculture*, vol. 13, no. 2, 2023, ISSN: 2077-0472. DOI: 10.3390/agriculture13020239. [Online]. Available: https://www.mdpi.com/2077-0472/13/2/239.

[31] C. O. Barcelos, L. A. Fagundes-Júnior, D. K. D. Villa, *et al.*, "Robot formation performing a collaborative load transport and delivery task by using lifting electromagnets," *Applied Sciences*, vol. 13, no. 2, 2023, ISSN: 2076-3417. [Online]. Available: https://www.mdpi.com/2076-3417/13/2/822.

[32] C. Liao and Z. Liu, "Cooperative path planning of ground-air robots for distributed photovoltaic inspection," in *Journal of Physics: Conference Series*, IOP Publishing, vol. 2658, 2023, p. 012 015.

[33] J. Bravo-Arrabal, M. Toscano-Moreno, J. Fernandez-Lozano, A. Mandow, J. A. Gomez-Ruiz, and A. Garcıa-Cerezo, "The internet of cooperative agents architecture (x-ioca) for robots, hybrid sensor networks, and mec centers in complex environments: A search and rescue case study," *Sensors*, vol. 21, no. 23, p. 7843, 2021.

[34] J. Zhang, X. Yue, H. Zhang, and T. Xiao, "Optimal unmanned ground vehicle—unmanned aerial vehicle formation-maintenance control for air-ground cooperation," *Applied Sciences*, vol. 12, no. 7, p. 3598, 2022.

[35] R. Chaurasia and V. Mohindru, "Unmanned aerial vehicle (uav): A comprehensive survey," *Unmanned Aerial Vehicles for Internet of Things (IoT) Concepts, Techniques, and Applications*, pp. 1–27, 2021.

[36] Q. Liu, M. Li, J. Yang, *et al.*, "Joint power and time allocation in energy harvesting of uav operating system," *Computer Communications*, vol. 150, pp. 811–817, 2020, ISSN: 0140-3664. DOI: https://doi.org/10.1016/j.comcom.2019.12.009. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0140366419315300.

[37] C. G. Grlj, N. Krznar, and M. Pranjić, "A decade of uav docking stations: A brief overview of mobile and fixed landing platforms," *Drones*, vol. 6, no. 1, 2022, ISSN: 2504-446X. DOI: 10.3390/drones6010017. [Online]. Available: https://www.mdpi.com/2504-446X/6/1/17.

[38] X. Liang, S. Zhao, G. Chen, G. Meng, and Y. Wang, "Design and development of ground station for uav/ugv heterogeneous collaborative system," *Ain Shams Engineering Journal*, vol. 12, no. 4, pp. 3879–3889, 2021.

[39] Y. Ding, B. Xin, and J. Chen, "A review of recent advances in coordination between unmanned aerial and ground vehicles," *Unmanned Systems*, vol. 9, no. 02, pp. 97–117, 2021.

[40] C. Lytridis, V. G. Kaburlasos, T. Pachidis, *et al.*, "An overview of cooperative robotics in agriculture," *Agronomy*, vol. 11, no. 9, 2021, ISSN: 2073-4395. DOI: 10.3390/agronomy11091818. [Online]. Available: https://www.mdpi.com/2073-4395/11/9/1818.

[41] A. Gautam, M. Singh, P. B. Sujit, and S. Saripalli, "Autonomous quadcopter landing on a moving target," *Sensors*, vol. 22, no. 3, 2022, ISSN: 1424-8220. DOI: 10.3390/s22031116. [Online]. Available: https://www.mdpi.com/1424-8220/22/3/1116.

[42] Y. Chang, Y. Cheng, U. Manzoor, and J. Murray, "A review of uav autonomous navigation in gps-denied environments," *Robotics and Autonomous Systems*, vol. 170, p. 104 533, 2023, ISSN: 0921-8890. DOI: https://doi.org/10.1016/j.robot.2023.104533. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0921889023001720.

[43] A. Khazetdinov, A. Zakiev, T. Tsoy, M. Svinin, and E. Magid, "Embedded aruco: A novel approach for high precision uav landing," in *2021 International Siberian Conference on Control and Communications (SIBCON)*, IEEE, 2021, pp. 1–6.

[44] J. Lim, T. Lee, S. Pyo, J. Lee, J. Kim, and J. Lee, "Hemispherical infrared (ir) marker for reliable detection for autonomous landing on a moving ground vehicle from various altitude angles," *IEEE/ASME Transactions on Mechatronics*, vol. 27, no. 1, pp. 485–492, 2022. DOI: 10.1109/TMECH.2021.3066643.

[45] R. M. Claro, D. B. Silva, and A. M. Pinto, "Artuga: A novel multimodal fiducial marker for aerial robotics," *Robotics and Autonomous Systems*, vol. 163, p. 104 398, 2023.

[46] Q. Zeng, Y. Jin, H. Yu, and X. You, "A uav localization system based on double uwb tags and imu for landing platform," *IEEE Sensors Journal*, vol. 23, no. 9, pp. 10 100–10 108, 2023. DOI: 10.1109/JSEN.2023.3260311.

[47] M. Shah Alam and J. Oluoch, "A survey of safe landing zone detection techniques for autonomous unmanned aerial vehicles (uavs)," *Expert Systems with Applications*, vol. 179, p. 115 091, 2021, ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2021.115091. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417421005327.

[48] W. Li, Y. Ge, Z. Guan, and G. Ye, "Synchronized motion-based uav–usv cooperative autonomous landing," *Journal of Marine Science and Engineering*, vol. 10, no. 9, p. 1214, 2022.

[49] Z. Li, Y. Chen, H. Lu, H. Wu, and L. Cheng, "Uav autonomous landing technology based on apriltags vision positioning algorithm," in *2019 Chinese Control Conference (CCC)*, 2019, pp. 8148–8153. DOI: 10.23919/ChiCC.2019.8865757.

[50] M. Quigley, K. Conley, B. Gerkey, *et al.*, "Ros: An open-source robot operating system," in *ICRA workshop on open source software*, Kobe, Japan, vol. 3, 2009, p. 5.

[51] S. Moon, J. J. Bird, S. Borenstein, and E. W. Frew, "A gazebo/ros-based communication-realistic simulator for networked suas," in *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*, IEEE, 2020, pp. 1819–1827.

[52] J. Galvez-Serna, F. Vanegas, S. Brar, J. Sandino, D. Flannery, and F. Gonzalez, "Uav4pe: An open-source framework to plan uav autonomous missions for planetary exploration," *Drones*, vol. 6, no. 12, 2022, ISSN: 2504-446X. DOI: 10.3390/

drones6120391. [Online]. Available: https://www.mdpi.com/2504-446X/6/12/391.

[53] K. Xiao, S. Tan, G. Wang, X. An, X. Wang, and X. Wang, "Xtdrone: A customizable multi-rotor uavs simulation platform," in *2020 4th International Conference on Robotics and Automation Sciences (ICRAS)*, 2020, pp. 55–61. DOI: 10.1109/ICRAS49812.2020.9134922.

[54] Y. Yang and T. Azumi, "Exploring real-time executor on ros 2," in *2020 IEEE International Conference on Embedded Software and Systems (ICESS)*, 2020, pp. 1–8. DOI: 10.1109/ICESS49830.2020.9301530.

[55] Z. Li, A. Hasegawa, and T. Azumi, "Autoware$_perf$: A tracing and performance analysis framework for ros 2 applications," *Journal of Systems Architecture*, vol. 123, p. 102 341, 2022, ISSN: 1383-7621. DOI: https://doi.org/10.1016/j.sysarc.2021.102341. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1383762121002344.

[56] S. Tselegkaridis and T. Sapounidis, "Simulators in educational robotics: A review," *Education Sciences*, vol. 11, no. 1, 2021, ISSN: 2227-7102. DOI: 10.3390/educsci11010011. [Online]. Available: https://www.mdpi.com/2227-7102/11/1/11.

[57] N. Soares, J. Teixeira, and V. Teichrieb, "Robot training in virtual environments using reinforcement learning techniques," in *Anais Estendidos do XXII Simpósio de Realidade Virtual e Aumentada*, Evento Online: SBC, 2020, pp. 25–29. DOI: 10.5753/svr_estendido.2020.12950. [Online]. Available: https://sol.sbc.org.br/index.php/svr_estendido/article/view/12950.

[58] J. Kim, G. Lim, Y. Kim, B. Kim, and C. Bae, "Deep learning algorithm using virtual environment data for self-driving car," in *2019 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, 2019, pp. 444–448. DOI: 10.1109/ICAIIC.2019.8669037.

[59] Gazebo Simulator Team, *Gazebo simulator documentation*, https://gazebosim.org/docs, 2023.

[60] R. Mengacci, G. Zambella, G. Grioli, D. Caporale, M. G. Catalano, and A. Bicchi, "An open-source ros-gazebo toolbox for simulating robots with compliant actuators," *Frontiers in Robotics and AI*, vol. 8, 2021, ISSN: 2296-9144. DOI: 10.3389/frobt.2021.713083. [Online]. Available: https://www.frontiersin.org/articles/10.3389/frobt.2021.713083.

[61] H. M. P. C. Jayaweera and S. Hanoun, "Path planning of unmanned aerial vehicles (uavs) in windy environments," *Drones*, vol. 6, no. 5, 2022, ISSN: 2504-446X. DOI: 10.3390/drones6050101. [Online]. Available: https://www.mdpi.com/2504-446X/6/5/101.

[62] K. Kumar, S. I. Azid, A. Fagiolini, and M. Cirrincione, "Erle-copter simulation using ros and gazebo," in *2020 IEEE 20th Mediterranean Electrotechnical Conference (MELECON)*, 2020, pp. 259–263. DOI: 10.1109/MELECON48756.2020.9140476.

[63] S. M. Nogar, "Autonomous landing of a uav on a moving ground vehicle in a gps denied environment," in *2020 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, 2020, pp. 77–83. DOI: 10.1109/SSRR50563.2020.9292607.

[64] M. Shafiee, M. Sajadinia, A.-A. Zamani, and M. Jafari, "Enhancing the transient stability of interconnected power systems by designing an adaptive fuzzy-based fractional order pid controller," *Energy Reports*, vol. 11, pp. 394–411, 2024.

[65] P. S. Krishna and P. G. K. Rao, "Fractional-order pid controller for blood pressure regulation using genetic algorithm," *Biomedical Signal Processing and Control*, vol. 88, p. 105 564, 2024.

[66] P. Huang, J. Wu, C.-Y. Su, and Y. Wang, "Tracking control of soft dielectric elastomer actuator based on nonlinear pid controller," *International Journal of Control*, vol. 97, no. 1, pp. 130–140, 2024.

[67] N. F. Nanyan, M. A. Ahmad, and B. Hekimoğlu, "Optimal pid controller for the dc-dc buck converter using the improved sine cosine algorithm," *Results in Control and Optimization*, vol. 14, p. 100 352, 2024.

[68] A. P. Moreira, J. Lima, and P. Costa, "Improving a position controller for a robotic joint," in *2021 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, IEEE, 2021, pp. 97–103. DOI: 10.1109/ICARSC52212.2021.9429809.

[69] L. R. da Silva, R. C. Flesch, and J. E. Normey-Rico, "Analysis of anti-windup techniques in pid control of processes with measurement noise∗∗this work was supported by the brazilian national council for scientific and technological development (cnpq) under grants 311024/2015-7 and 305785/2015-0.," *IFAC-PapersOnLine*, vol. 51, no. 4, pp. 948–953, 2018, 3rd IFAC Conference on Advances in Proportional-Integral-Derivative Control PID 2018, ISSN: 2405-8963. DOI: https://doi.org/10.1016/j.ifacol.2018.06.100. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405896318303975.

[70] T. Tsoy, R. Safin, R. Sultanov, S. K. Saha, and E. Magid, "Recommended criteria for qualitative comparison of fiducial markers performance," in *2022 International Siberian Conference on Control and Communications (SIBCON)*, 2022, pp. 1–5. DOI: 10.1109/SIBCON56144.2022.10003018.

[71] Y. Liu, D. Cheng, Q. Wang, *et al.*, "Optical distortion correction considering radial and tangential distortion rates defined by optical design," *Results in Optics*, vol. 3, p. 100 072, 2021, ISSN: 2666-9501. DOI: https://doi.org/10.1016/j.rio.2021.100072. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2666950121000201.

[72] Z. Shi, Y. Shang, X. Zhang, and G. Wang, "Dlt-lines based camera calibration with lens radial and tangential distortion," *Experimental Mechanics*, vol. 61, no. 8, pp. 1237–1247, 2021.

[73] S. Peña-Haro, R. Ljubičić, and D. Strelnikova, "Geometric correction and stabilization of images collected by uass in river monitoring," in *Unmanned Aerial Systems for Monitoring Soil, Vegetation, and Riverine Environments*, Elsevier, 2023, pp. 203–230.

[74] B. Lin, L. Wu, Y. Niu, H. Zhou, and Z. Ma, "A multi-target detection framework for multirotor uav," in *2020 Chinese Automation Congress (CAC)*, 2020, pp. 1063–1068. DOI: 10.1109/CAC51589.2020.9327450.

[75] *Camera calibration with OpenCV - python tutorial*, https://docs.opencv.org/4.x/dc/dbb/tutorial_py_calibration.html.

[76] F. O. Coelho, J. P. Carvalho, M. F. Pinto, and A. L. Marcato, "Ekf and computer vision for mobile robot localization," in *2018 13th APCA International Conference on Automatic Control and Soft Computing (CONTROLO)*, IEEE, 2018, pp. 148–153.

[77] M. Kalaitzakis, B. Cain, S. Carroll, A. Ambrosi, C. Whitehead, and N. Vitzilaios, "Fiducial markers for pose estimation: Overview, applications and experimental comparison of the artag, apriltag, aruco and stag markers," *Journal of Intelligent & Robotic Systems*, vol. 101, pp. 1–26, 2021.

[78] D. Jurado-Rodrıguez, R. Muñoz-Salinas, S. Garrido-Jurado, and R. Medina-Carnicer, "Design, detection, and tracking of customized fiducial markers," *IEEE Access*, vol. 9, pp. 140 066–140 078, 2021.

[79] Y.-L. Shi and S.-Z. Zou, "A design of an indoor delivery uav," in *Journal of Physics: Conference Series*, IOP Publishing, vol. 2457, 2023, p. 012 036.

[80] R. R. Lima, B. M. Rocamora, and G. A. Pereira, "Continuous vector fields for precise cable-guided landing of tethered uavs," *IEEE Robotics and Automation Letters*, 2023.

[81] W. Giernacki, J. Rao, S. Sladic, A. Bondyra, M. Retinger, and T. Espinoza-Fraire, "Dji tello quadrotor as a platform for research and education in mobile robotics and control engineering," in *2022 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2022, pp. 735–744. DOI: 10.1109/ICUAS54217.2022.9836168.

[82] P. Yue, J. Xin, Z. Mao, Y. Lu, and Y. Zhang, "Vision-based autonomous landing of uav on dynamic apron," in *2023 IEEE International Conference on Unmanned Systems (ICUS)*, 2023, pp. 1613–1619. DOI: 10.1109/ICUS58632.2023.10318423.

[83] V. 1. RYZE DJI, "Tello sdk 2.0 user guide," p. 22, 2018. [Online]. Available: https://dl-cdn.ryzerobotics.com/downloads/Tello/Tello%20User%20Manual%20v1.4.pdf. (visited on 01/09/2024).

[84] D. RYZE, "Ryze tello," [Online]. Available: https://www.ryzerobotics.com/tello..

[85] U. Robotics, "Robot base magni silver," 2024. [Online]. Available: https://www.ubiquityrobotics.com/product/magni-silver/. (visited on 01/09/2024).

[86] "3dr inc," 2024. [Online]. Available: https://www.3dr.com/ (visited on 01/11/2024).

[87] P. Autopilot, "Open-source flight control," 2024. [Online]. Available: https://px4.io/ (visited on 01/11/2024).

[88] T. Chen, X. Zhuang, Z. Hou, and H. Chen, "A pixhawk-ros based development solution for the research of autonomous quadrotor flight with a rotor failure," in *2022 IEEE International Conference on Unmanned Systems (ICUS)*, IEEE, 2022, pp. 590–595.