

Independent University

Bangladesh (IUB)

IUB Academic Repository

Computer Science and Engineering

Undergraduate Thesis

2024-12

Optimizing UAV Performance using ArUco Marker-Enabled Vision-based Precision Landing and Autonomous Charging

Islam, Tabriji

IUB

<http://ar.iub.edu.bd/handle/11348/999>

Downloaded from IUB Academic Repository



Optimizing UAV Performance using ArUco Marker-Enabled Vision-based Precision Landing and Autonomous Charging

December 2024

Prepared by:

Tabriji Islam

ID: 2030660

Khandaker Umme Salma

ID: 2030664

Ishtiaq Hossen

ID: 1921532

Department of Computer Science and Engineering
Independent University, Bangladesh

Supervised by

M Ashraful Amin, PhD

Professor

Department of Computer Science and Engineering
Independent University, Bangladesh

Attestation

Plagiarism is strictly prohibited and conflicts with our university's rules and regulations. We guarantee the authenticity of our work. We have used some copyrighted material and models of others in our project which have been properly cited following international standards and guidelines.

Tabriji Islam

ID: 2030660

Undergraduate, CSE

Signature:

.....

Khandaker Umme Salma

ID: 2030664

Undergraduate, CSE

Signature:

.....

Ishtiaq Hossen

ID: 1921532

Undergraduate, CSE

Signature:

.....

Evaluation Committee

Supervisor

Name: M Ashraful Amin, PhD
Professor, CSE

Signature: _____

Internal Examiner 1

Name: Amin Ahsan Ali, PhD
Professor, CSE

Signature: _____

Internal Examiner 2

Name: AKM Mahbubur Rahman,
PhD
Associate Professor, CSE

Signature: _____

Internal Examiner 3

Name: Ashraful Islam, PhD
Assistant Professor, CSE

Signature: _____

External Examiner

Name: Md Wahidul Hasan
Lecturer, CSE

Signature: _____

Acknowledgment

We are deeply grateful to all those who contributed to the successful completion of this project. First and foremost, we extend our heartfelt thanks to our professor, M Ashraful Amin, of Independent University, Bangladesh, for his invaluable guidance and support. His insightful suggestions, boundless patience, and generous investment of time were crucial to the depth and quality of our work. Without his expert mentorship, this project would not have been possible, and we are grateful for his tireless commitment to our academic growth. Furthermore, we thank Md Wahidul Hasan from the South Dakota School of Mines and Technology, who provided invaluable remote guidance throughout the project. Despite the geographical distance, his timely advice, constructive feedback, and expert insight helped shape the direction of our project and ensured its successful completion.

We would also like to thank the CCDS (Centre for Computational & Data Sciences) for their continuous assistance throughout this journey. The collaborative environment and the valuable resources provided by the CCDS played a key role in the completion of this project. The support of the research assistants has greatly shaped our academic and research experiences, allowing us to explore new ideas and improve our understanding in meaningful ways.

This project is supported by a grant from the Independent University, Bangladesh (IUB), under the Vice-Chancellor (VC) Research Fund.

Lastly, we wish to acknowledge the contributions of all those who offered their support, directly and indirectly, throughout this endeavor. Their encouragement and assistance were highly appreciated and we are thankful for their impact on our project.

Letter of Transmittal

December 19, 2024

M Ashraful Amin, PhD
Senior Project Supervisor
Department of Computer Science and Engineering,
Independent University, Bangladesh.

Md Wahidul Hasan
Senior Project Co-Supervisor
Department of Mechanical Engineering,
South Dakota School of Mines and Technology.

Subject: Optimizing UAV Performance using ArUco Marker-Enabled Vision-Based Precision Landing and Autonomous Charging.

Dear Sir,

We are delighted to present our senior project report titled “*Optimizing UAV Performance using ArUco Marker-Enabled Vision-based Precision Landing and Autonomous Charging*”. This report showcases the significant research and technical efforts we have undertaken during the project under your expert guidance.

Through this project, we investigated critical aspects of UAV operations, including autonomous navigation, precision landing systems, and wireless power transfer technology. Our efforts involved designing, building, and testing a custom UAV system that addresses operational challenges such as GPS inaccuracies and battery inefficiency. These experiences have greatly enriched our understanding and technical proficiency in UAV technologies and their applications.

We sincerely appreciate your consistent guidance, support, and insightful mentorship, which have been crucial in completing this project. We are confident that this report meets the academic and professional standards expected, and we hope to receive your approval.

We hope that this report meets your expectations and is deemed satisfactory. Thank you for your continuous cooperation and invaluable assistance.

Yours sincerely,

Tabriji Islam (ID: 2030660)
Khandaker Umme Salma (ID: 2030664)
Ishtiaq Hossen (ID: 1921532)

Abstract

Unmanned Aerial Vehicles (UAVs) have become indispensable in applications such as precision agriculture and environmental monitoring. Despite their potential, their widespread deployment is hindered by three critical challenges: limited flight time due to battery constraints, declining navigation accuracy at higher altitudes, and reliance on Global Positioning System (GPS) signals, which degrade with altitude, environmental factors, and battery depletion. To address these issues, this study proposes a system integrating vision-based ArUco marker detection with a proportional integral derivative (PID) controller for precision landing, paving the way for autonomous recharge and extending operational capabilities. Our approach leverages a combination of GPS-based navigation and vision-based marker detection to ensure accurate landings at target waypoints. At lower altitudes (10 meters), GPS accuracy declines with deviations averaging 0.15 meters, increasing to 0.41 meters at 25 meters. In addition, battery depletion worsens positional inaccuracies, with up to a 152.3% increase in GPS-based landing deviations as the voltage drops from 12.6 to 9.1 V. Vision-based guidance, enabled by ArUco markers, compensates for these inaccuracies, achieving a detection rate of 83.33% at 4.5 meters but decreasing to 33.33% at 6.0 meters. The Vision-guided landing system with PID controller further refines the descent by dynamically adjusting the position of the UAV, reducing the deviation by 56.5% in simulation tests, demonstrating its effectiveness in enhancing the landing accuracy over the GPS-only system. Experimental results validate the effectiveness of the system, showcasing its potential to allow UAVs to autonomously locate charging stations, extend mission durations, and maintain consistent operational performance under varying environmental and battery conditions.

Contents

1	Introduction	11
1.1	Background of the Project	11
1.2	Problem Statement	12
1.2.1	Research Questions	13
1.3	The Outcome of the Project	14
2	Literature Review	16
2.1	Existing Technology and Applied Solutions	16
2.2	Issues with the Existing Solutions	17
3	Understanding UAVs: Mechanics, Types, and Flight	18
3.1	Drone: The Most Versatile UAV	18
3.2	Core Components of a Multi-Rotor Drone	18
3.2.1	Drone Assembly	21
3.3	Autopilot and On-board Computer Configuration	22
3.4	Autonomous Mission Planning using MAVLink	25
3.5	Power and Performance: Batteries, Charging, and Efficiency	26
3.5.1	Battery Selection Criteria for UAVs	26
3.5.2	Challenges in Battery Selection	26
3.5.3	Charging Mechanisms and Efficiency Considerations	27
3.6	Battery and Power Consumption Calculations	27
3.6.1	Summary of Battery Performance and Power Consumption	28
3.6.2	Verification and Relative Error	29
4	OpenCV ArUco Markers for Detection	30
4.1	Configuring OpenCV	30
4.1.1	Installing Essential Libraries and Packages	30
4.1.2	Downloading and Building OpenCV and OpenCV-Contrib	30
4.1.3	Building and Installing OpenCV	31
4.1.4	Verifying the Installation	31
4.2	Overview of ArUco Markers	31
4.2.1	Structure of ArUco Markers	32
4.2.2	ArUco Dictionaries	32
4.2.3	Marker Identifiers (IDs)	33
4.3	Detection Mechanism and Pose Estimation	33
4.3.1	Implementation Details	34
4.4	Precision Landing Algorithm	35
4.4.1	Precision Landing Logic	35

5	Role of PID Controller in Precision Landing	39
5.1	Integration with ArUco Markers	39
5.1.1	Understanding PID Controllers	39
5.1.2	PID Controller Algorithm and System Design	41
5.1.3	Importance in Drone Flight and Precision Landing	41
6	Wireless Power Transfer System	44
6.1	Introduction to Wireless Power Transfer (WPT)	44
6.2	Principle and Operation of WPT	44
6.3	Implementation of WPT for Drone Charging	45
6.3.1	System Design	45
6.3.2	Experimental Results	45
6.3.3	Flight Time Enhancement	45
7	Experimental Design and Testing	46
7.1	Methodology	46
7.1.1	Requirements Analysis	46
7.1.2	Autonomous Flight Mission Configuration	46
7.1.3	Data Collection and Retrieval Strategy	48
8	Results and Analysis	50
8.1	Overview of Results	50
8.1.1	GPS-Guided Landing Accuracy Analysis	50
8.1.2	Analysis of GPS Accuracy and Altitude Impact	53
8.1.3	Battery Depletion and Navigation Reliability	54
8.1.4	Performance of Vision-Guided Landing Using ArUco Markers and PID Controller	60
9	Conclusion	66
9.1	Limitations and Challenges	66
9.2	Ethical Considerations	67
9.3	Sustainability Of The Work	67
9.4	Social and Environmental Effects And Analysis	67
9.5	Future Plan	68
A	Appendix	70
A.1	Turnitin Originality Report	70
	Bibliography	70

List of Figures

1	Drone Operations: (a) Preparation on the ground and (b) Flight in action.	11
2	Example of Detected ArUco Markers with IDs.	12
3	Different types of Drones.	18
4	System Block Schematic of a Quadcopter Drone.	20
5	Snapshots of Drone Assembly Steps.	23
6	Connection Pinout between Pixhawk 2.4.8 and Raspberry Pi 4	24
7	Communication between Drone and Ground Station using MAVLink Protocol.	25
8	Sample ArUco Marker with Pose Axis and Basic Detection Process Flowchart.	32
9	ArUco Markers with Different Matrix Sizes (4x4, 5x5, 6x6, 7x7 matrices). .	32
10	ArUco markers with different IDs	33
11	Detected Marker IDs and Pose Estimation Output	36
12	Output of Marker Detection Code.	36
13	Vision-based Precision Landing Process Flowchart.	38
14	Basic PID Controller Block Diagram.	40
15	Generic Block Diagram of a Wireless Power System.	44
16	Proposed Flowchart for Precision Landing Procedure for Autonomous Drones.	47
17	Figure (a) Shows the Test Flight Location and Figure(b) Illustrates the Process of Autonomous Mission Programming via Mission Planner.	47
18	Snapshot of Mission Planner Interface for Retrieving Flight Data Logs using MAVLink.	48
19	Comparison of Smartphone GPS with Drone's built-in GPS Module Exper- iment Setup.	50
20	The Drone's built-in GPS Data Compared with the GPS Data Collected from the Smartphones at Waypoints A and B.	51
21	The Drone's Actual Flight Path Deviating from the Ideal Straight-line Path Between Waypoint A and Waypoint B.	52
22	Scatter Plot showing the Correlation between Different Altitude and Preci- sion Landing Deviation.	53
23	Regression Line visualizing the Deviation in Precision Landing at Higher Altitudes.	54
24	Battery Voltage Trend during a 10-Minute Autonomous Flight Mission. . .	55
25	Relationship between decreasing Battery Voltage and GPS Deviation. . . .	59
26	Snapshots from the Camera Module showing the ArUco Markers Detected at different Altitudes.	61

27	Challenges in ArUco marker detection due to environmental conditions. The first image shows the impact of glare, while the second illustrates the distortion caused by uneven terrain.	62
28	3D model of a Proposed Charging Station with ArUco Marker.	62
29	Simulation of a drone adjusting its position toward a target using PID control over 100 steps. At each step, control adjustments are applied to update the drone's X, Y, and Z positions, and the loop stops early if the drone is close enough to the target.	63
30	UAV Final Descent towards Target; (a) UAV approaching the landing pad equipped with ArUco markers; (b) UAV lowering closer to the target; (c) UAV performing precision landing on the ArUco marker pad.	64
31	Comparison of Landing Deviations for GPS-Only and Vision-guided + PID Systems	64
32	Turnitin Originality Report Indexes	70

List of Tables

3.1	Comparison of Different Drone Types	19
3.2	Components List	22
3.3	Comparison of Battery Specifications for UAV	26
3.4	Comparison of Charging Methods for UAV Batteries	27
3.5	Battery Performance and UAV Endurance Summary	29
6.1	Specifications of Transmitter and Receiver Coils	45
8.1	GPS Data from Pixhawk M8N Module and Smartphones	51
8.2	Target Altitude, Actual Altitude, and Precision Landing Deviation	53
8.3	Battery Voltage Over Time during Flight	56
8.4	Battery Voltage and GPS Deviation Data Across 15 Missions	57
8.5	GPS Deviation Calculations	58
8.6	Detection Rates of ArUco Markers at different Altitudes	61
8.7	Simulation Data of Landing Performance with GPS-Only and Vision-guided + PID Control	63

1 Introduction

1.1 Background of the Project

The advancement of Unmanned Aerial Vehicles (UAVs), commonly known as drones, has revolutionized several industries, including agriculture, environmental monitoring, and disaster response [1, 2, 3]. UAVs offer unparalleled advantages, such as real-time data collection, remote access, and reduced operational costs. However, despite these benefits, several challenges persist in their deployment, mainly concerning limited flight duration, dependence on GPS for navigation, and inconsistent landing accuracy [4, 5]. These challenges become particularly pronounced in environments where GPS signals are not reliable.

One critical issue UAVs face is achieving precise landings, especially when GPS reliability decreases at higher altitudes or during battery depletion. Existing navigation systems often rely on standalone GPS technology, which, while essential, is prone to inaccuracies caused by factors such as altitude variations, environmental interference, and signal degradation. For example, a study by Sriram and Gade [6] demonstrated that GPS-based navigation systems show significant deviations, with positional errors increasing to 0.5 meters at higher altitudes, compromising the reliability of autonomous operations. This limitation becomes even more critical when UAVs are required to dock at autonomous charging stations, where precision is vital.



(a) Drone on the ground.



(b) Drone in flight.

Figure 1: Drone Operations: (a) Preparation on the ground and (b) Flight in action.

Alternative solutions, such as vision-based systems and infrared beacons, have been explored to address this [7, 8]. Vision-based systems have emerged as a particularly effective solution due to their ability to provide accurate real-time feedback without the need for additional external infrastructure. For example, Ge et al. [8] developed a vision-based landing strategy that used visual markers to improve positional accuracy under adverse conditions, such as high wind speeds and uneven terrain. Their results showed a 40% reduction in landing deviations compared to GPS-only systems, emphasizing the potential of visual guidance to improve UAV precision.

ArUco markers¹ have gained popularity in UAV navigation due to their simplicity, robustness, and high detection accuracy [9]. These markers are unique visual identifiers that cameras onboard can easily detect, making them ideal for precision landing tasks, as shown in Figure 2. This study [10] highlights the effectiveness of ArUco markers when combined with proportional integral derivative (PID) controllers for real-time positional adjustments. These systems demonstrated notable improvements in landing precision, with average deviations reduced to less than 0.2 meters, even under challenging conditions such as poor lighting and wind interference.

Despite these advances, challenges remain. Marker detection reliability decreases with altitude, and the computational load of real-time image processing can strain the systems on board. Addressing these issues requires a comprehensive integration of visual guidance with control algorithms and robust hardware to support precise operations. This project builds on these findings, taking advantage of the strengths of ArUco markers and PID controllers to design a system that achieves high-precision autonomous landings while addressing the limitations of GPS-only navigation. In doing so, it aims to enable UAVs to autonomously dock at charging stations, extending their operational range and utility in diverse applications.

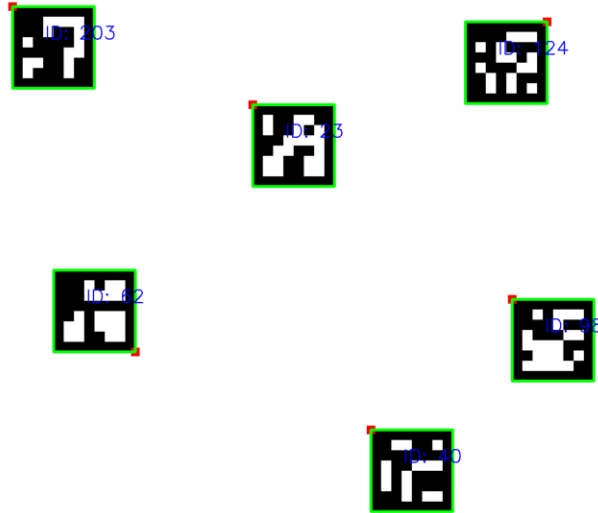


Figure 2: Example of Detected ArUco Markers with IDs.

1.2 Problem Statement

In the expanding domain of UAV technologies, where autonomous flight operations are becoming increasingly essential, it is evident that UAVs still face significant limitations in achieving consistent precision during autonomous landings. These challenges are primarily due to the inherent inaccuracies of GPS-based navigation, which become especially pronounced under battery depletion, altitude changes, and external environmental factors. These issues are compounded by the limited flight duration of UAVs, which further restricts their operational capabilities, particularly in applications requiring continuous monitoring, such as precision agriculture and environmental assessments. Precision landing, a critical aspect of UAV autonomy, becomes increasingly difficult due to several operational factors,

¹https://docs.opencv.org/3.4/d5/dae/tutorial_aruco_detection.html

including altitude-related GPS inaccuracy, battery performance variations, and susceptibility to external interferences such as wind and obstacles. In this context, developing effective landing solutions must address these operational inefficiencies and support autonomous battery charging for extended deployment. Although conventional methods, such as Real-Time Kinematics (RTK) GPS and infrared beacons [11], have been employed to achieve better landing precision, they have limitations, ranging from high costs to susceptibility to environmental interference [12]. Alternatively, vision-based systems that employ ArUco markers provide a scalable and adaptable solution to address landing precision without extensive infrastructure requirements.

This study aims to optimize the operational capabilities of autonomous UAVs by integrating vision-based ArUco markers with a PID control system to enable high-accuracy precision landings. Using a vision-guided closed-loop control system, this research intends to overcome the limitations of GPS-based navigation, ensuring reliable precision landing and paving the way for sustainable autonomous charging stations [2]. Ultimately, this research seeks to improve UAV navigation capabilities, reduce positional deviations during landing, and extend operational durations, thus broadening the applicability of UAVs for continuous and precise field operations.

1.2.1 Research Questions

The following research questions are crucial for this study, which guide the analysis to enhance the precision and reliability of autonomous UAV landings. Each question addresses a critical aspect of the UAV landing system, ensuring a comprehensive analysis of the integrated technologies and their performance under various conditions.

- i. **RQ1:** *How does integrating ArUco markers and a PID controller improve the precision of autonomous UAV landings compared to GPS-only landings?*

Rationale: This question investigates whether combining ArUco markers with a PID controller improves UAV landing precision compared to GPS-only approaches. By examining how visual guidance and error correction improve accuracy, particularly when GPS reliability declines, this study aims to demonstrate the benefits of this integrated landing strategy.

- ii. **RQ2:** *What are the effects of battery depletion on the accuracy of GPS-based autonomous UAV landing, and how can autonomous charging mitigate these effects?*

Rationale: This question explores how declining battery levels impact the accuracy of GPS-based landings and whether autonomous charging can help maintain consistent landing precision. Understanding these effects is key to improving UAV reliability by ensuring optimal power levels for precise navigation and landing.

- iii. **RQ3:** *To what extent do environmental factors such as wind, obstacles, and varying altitudes impact the reliability of ArUco marker detection during UAV landings?*

Rationale: This question examines how environmental factors like wind, obstacles, and altitude affect the reliability of ArUco marker detection during UAV landings. Investigating these influences helps determine the operational limitations and required adaptations to ensure consistent marker detection and precision landing under different field conditions.

1.3 The Outcome of the Project

Existing UAV systems designed for autonomous navigation and landing often rely heavily on GPS or infrared beacons for precision tasks. However, while reliable in certain situations, these conventional methods face significant limitations in challenging environments, such as areas with poor satellite visibility, high altitudes, or complex environmental conditions. Addressing these gaps, we introduce an innovative system for UAV precision landing that integrates ArUco marker detection with a PID controller. This approach enhances the autonomy of UAVs and provides practical solutions for real-world applications requiring continuous monitoring, such as precision agriculture and environmental assessments.

This research aims to advance the field of autonomous UAV navigation by introducing an innovative and more efficient approach to precision landing and navigation, significantly improving UAV reliability and operational capacity in diverse environments. Our contributions are as follows.

- **Development of Vision-Guided Precision Landing System:** We introduce an ArUco marker-based vision-guided precision landing system for UAVs, supported by a PID controller. This system significantly addresses the shortcomings of GPS-only navigation by providing reliable visual guidance even when GPS signals are unreliable due to altitude or battery depletion. This integration aims to improve landing precision and enable consistent UAV performance across various environmental conditions.
- **Comprehensive Performance Evaluation:** The system was rigorously tested under varying conditions, including different altitudes and battery levels, to assess its robustness. The introduction of ArUco markers combined with PID control significantly reduced positional deviations, achieving high-accuracy autonomous landings even when GPS reliability was compromised. Marker detection success was notably higher at lower altitudes, reaching 93% at 7 meters, demonstrating the effectiveness of the vision-based system in challenging conditions.
- **Integration for Autonomous Charging Stations:** By facilitating precise landings, the developed system paves the way for seamless integration with autonomous charging stations. This capability is crucial for extending UAV operational lifespans, particularly for use cases in agriculture and environmental monitoring where continuous field coverage is essential. Integrating the precision landing system with a closed-loop autonomous charging set-up allows UAVs to perform extended operations with minimal manual intervention, contributing to sustainable and efficient mission management.

Comparison with Existing Solutions: To evaluate the impact of our approach, we compared the precision landing performance using ArUco markers and PID control with GPS-only approaches. The vision-based system proved to be a more cost-effective and adaptable solution for environments with limited infrastructure. The results underscored its reliability in achieving precision landing, particularly in scenarios where GPS accuracy was compromised by altitude or battery constraints, highlighting the potential for broader deployment in various critical real-world applications.

These developments mark a significant step forward in making autonomous charging stations for UAVs allowing them to be more effective tools for environmental monitoring, precision agriculture, and other mission-critical tasks requiring prolonged flight time with

minimal downtime for charging. Our system addresses critical limitations of existing technologies, providing practical solutions that enhance UAV autonomy and reliability. By incorporating vision-based landing, precision control, and autonomous charging, we create a foundation for UAV operations that are more efficient and capable of adapting to the demands of real-world environments.

2 Literature Review

Numerous initiatives have been undertaken to enhance the UAVs' precision landing and operational capabilities, addressing the challenges associated with limited GPS reliability and power efficiency.

2.1 Existing Technology and Applied Solutions

- Um, Inseop, et al. developed a redundancy-enhanced architecture for Real-Time Kinematic GPS (RTK-GPS) systems to improve availability in multi-drone operations [13]. They introduced cold-standby, hot-standby, and parallel systems for seamless Radio Technical Commission for Maritime Services (RTCM) message switching, ensuring drones maintain accurate positioning even during network instability. Empirical tests demonstrated successful implementation, with parallel systems achieving nearly 100% availability when multiple RTCM sources were used. Their design reduced GPS errors by up to 30% and maintained system reliability during source switching.
- Joshua et al. proposed a precision drone landing system leveraging visual and infrared (IR) fiducial markers and a gimbal-mounted, multi-payload camera. Their method reduces data dependencies, requiring only the direction to the landing pad and its pixel size, avoiding altitude or distance calculations. The system uses dynamic switching between zoom, wide-angle, and IR cameras for long-range and nighttime operations. They introduced active and passive IR markers, achieving an average landing error of 0.19 meters and enabling reliable recovery when the landing pad is temporarily obscured [6].
- The authors in this study [14] explored autonomous drone navigation and landing systems, emphasizing solutions that mitigate the limitations of GPS reliability. Their approach incorporated visual markers and sensor fusion techniques to enhance accuracy when GPS signals are weak or obstructed, such as in urban environments or dense forests. By leveraging alternative localization methods, the study addressed the prevalent challenges of GPS reliance in critical drone operations.
- Patoliya et al. presented a hybrid autonomous navigation and mapping (ANM) system [15] integrating GPS and LiDAR data for precise localization and mapping in unstructured environments. The system employs a Rao-Blackwellized particle filter-based Gmapping algorithm within the ROS framework to generate 2D maps while compensating for the cumulative error common in relative localization. LiDAR ensures obstacle detection, while GPS aids in defining navigation trajectories. The proposed approach achieves an impressive localization accuracy with an RMSE of 0.883 cm and a mean absolute error of 0.094 cm.

2.2 Issues with the Existing Solutions

- The system developed by Um, Inseop, et al. [13] faces limitations such as reliance on local base stations and Networked Transport of RTCM via Internet Protocol (NTRIP) servers, which restrict usability in remote areas. Position shifts during RTCM source switching and complexity in managing multiple RTCM sources highlight potential operational challenges. Additionally, the system’s scalability with large drone fleets and its ability to address broader GPS signal issues remain untested, suggesting future work is needed to enhance its robustness and practical applicability.
- Joshua’s proposed system [6] was validated across various conditions, including distances up to 168 meters and nighttime operations. While passive IR pads require no power, they introduce challenges such as occlusion due to the drone’s heat reflection. The authors acknowledged limitations, including weather dependence and reliance on specific camera capabilities. They suggested future work to test the system in diverse environments and explore adaptive control policies for moving landing pads.
- Sriram highlighted the inherent unreliability of GPS for precision tasks, particularly in areas with significant signal interference or multi-path effects, such as urban canyons or natural obstacles [14]. While their proposed system improves navigation and landing precision, it remains partially dependent on environmental conditions and the availability of high-quality visual markers. Future research should focus on enhancing the robustness of alternative navigation systems to reduce reliance on GPS entirely and adapt to more complex operational scenarios.
- The authors [15] addressed the challenges of GPS reliability, noting that limited satellite signals in specific environments, such as caves or tunnels, resulting in inaccurate or unavailable positional data. While LiDAR effectively complements GPS for localization, the system struggles with large open areas where mapping precision declines. Additionally, the design is limited to planar surfaces, making it unsuitable for complex terrains like staircases or uneven ground. The proposed model emphasizes integrating diverse sensor systems to mitigate GPS weaknesses and improve performance under dynamic conditions. Future research should focus on refining algorithms to enable mapping over larger areas and adapting to non-planar environments.

Despite notable advancements in autonomous UAV operations, prior systems have exhibited limitations such as dependency on local base stations, restricted scalability, and environmental constraints that impact GPS reliability and visual marker accuracy. In response, our approach integrates ArUco markers with a PID controller to mitigate the challenges of fluctuating GPS accuracy and environmental interference while ensuring adaptability for varying operational conditions. This strategy specifically aims to address scalability, reduce reliance on fixed infrastructures, and enhance the precision of autonomous landings in real-world environments.

3 Understanding UAVs: Mechanics, Types, and Flight

3.1 Drone: The Most Versatile UAV

An unmanned aerial vehicle (UAV), often called a drone or unmanned aircraft system (UAS), is operated without a human pilot, crew, or passengers. Initially developed in the 20th century for military tasks deemed too monotonous, hazardous, or unclear for human involvement, drones became vital tools for military operations by the 21st century [16]. As advancements in control systems emerged and production costs, they are now widely utilized in areas [17] such as aerial photography, large-area mapping, precision agriculture [18, 19], wildfire monitoring, environmental assessment, infrastructure inspections [20], product delivery, entertainment, and competitive drone racing.

Drones vary widely in size, capabilities, and design, ranging from small recreational models to advanced surveillance and control systems, as illustrated in Figure 3. We have organized drones into four categories based on wing type, their applications, benefits, and drawbacks, as shown in Table 3.1.

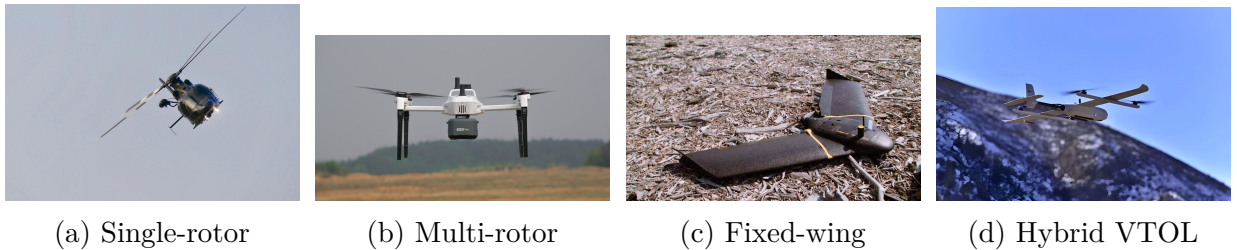


Figure 3: Different types of Drones.

A quadcopter drone was chosen for this project due to its excellent maneuverability, Vertical Take-Off and Landing (VTOL) capability, and ease of control, which are essential for achieving precise landings using ArUco markers and a PID controller. Its ability to hover and make minor adjustments makes it ideal for accurate alignment during descent, enhancing the overall reliability of the autonomous landing and charging system. Moreover, quadcopters' relatively low cost and versatility make them an effective solution for testing and implementing precision landing strategies.

3.2 Core Components of a Multi-Rotor Drone

A multi-rotor drone comprises several essential components, each contributing to its functionality and performance. The core components are as follows:

Table 3.1: Comparison of Different Drone Types

Drone Type	Advantages	Disadvantages	Use Cases
Multi-rotor drones	(i) Easy to control and maneuver. (ii) VTOL and hover flight (iii) Often lower price (iv) Portability	(i) Limited flying time (ii) Small payload capabilities (iii) Less stability in the wind (iv) Lower flight speeds	(i) Aerial photography (ii) Aerial inspection (iii) Landing surveying (iv) Agriculture
Fixed-wing drones	(i) Longer flight time (ii) Can carry a heavier payload (iii) Greater stability in the wind (iv) Higher flight speeds	(i) More training is needed (ii) No VTOL/hover (iii) Expensive (iv) Difficult to land, more space needed	(i) Aerial mapping (ii) Utility inspection (iii) Surveillance (iv) Agriculture
Single-rotor helicopter drones	(i) VTOL and hover flight (ii) Long endurance (with gas power) (iii) Heavier payload capability	(i) More dangerous (ii) Harder to fly, more training is needed (iii) Expensive	(i) Aerial LiDAR laser scanning
Fixed-wing hybrid VTOL drones	(i) VTOL and hover (ii) Long-endurance flight (iii) Fast speed (iv) Heavier payload capability	(i) More training is needed (ii) Expensive	(i) Aerial mapping (ii) Utility inspection (iii) Surveillance (iv) Agriculture (v) Search and rescue

- **Frame:** The structural body of the drone is typically constructed from lightweight materials like carbon fiber or aluminum. It serves as the base to which all other components are mounted.
- **Motors:** Multi-rotor drones use multiple brushless electric motors (BLDC) connected to a propeller. These motors provide the thrust required for lift and are crucial to maintaining stability during flight.
- **Electronic Speed Controllers (ESCs):** ESCs regulate each motor's speed, allowing the drone to adjust in real time for balance and maneuverability.
- **Propellers:** Attached to each motor, propellers convert rotational energy into thrust. The number of propellers is determined by the type of multi-rotor (e.g., quadcopter, hexacopter), which affects stability and lifting capacity.
- **Flight Controller:** The flight controller is the “brain” of the drone. It processes sensor inputs and commands from the pilot or pre-programmed instructions, using them to adjust the ESCs and maintain stable flight. Advanced controllers also enable autonomous flight capabilities.

- **Battery:** A Lithium Polymer (LiPo) battery typically provides the power required for all the drone's systems, including motors and electronics. Battery capacity directly impacts the flight time of the drone.
- **GPS Module:** The GPS module is used for navigation, enabling the drone to determine its position in space, follow waypoints, and achieve autonomous flight. In the case of precision landing, GPS helps guide the drone to the general vicinity of the target.
- **Sensors:** A multi-rotor may include various sensors such as accelerometers, gyroscopes, barometers, and cameras. These sensors provide data to the flight controller, allowing the drone to maintain altitude and orientation and detect obstacles or landing markers.
- **Telemetry System:** The telemetry system provides a communication link between the drone and the ground control station, enabling real-time monitoring of parameters such as altitude, speed, and battery level.
- **Camera Module (Optional):** Drones can have a camera module for visual feedback applications. This is especially useful for precision landing using visual markers in autonomous flight scenarios.

These components work together to ensure stable and effective flight operations. A system block diagram, as shown in Figure 4, illustrates the basic design schematic of a quadcopter drone. Each component's specific configuration and quality determine the drone's performance characteristics, such as stability, flight time, and payload capacity.

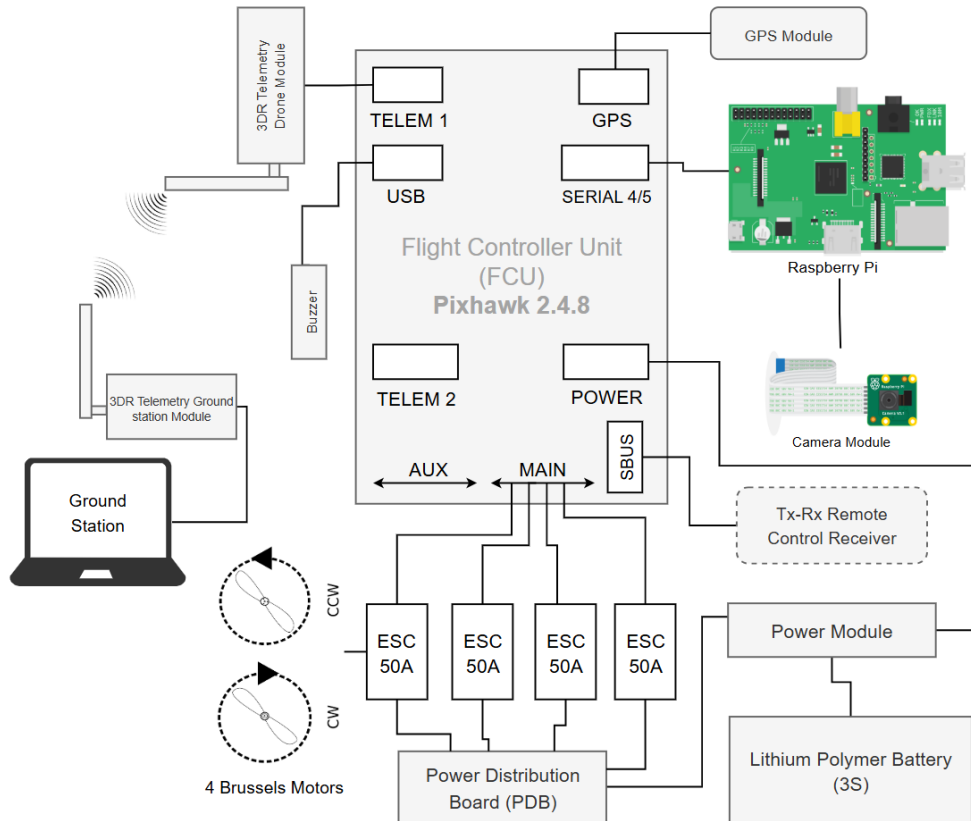


Figure 4: System Block Schematic of a Quadcopter Drone.

3.2.1 Drone Assembly

To assemble the drone, we first conducted a feasibility study of available components in the domestic market, ensuring their compatibility with the project’s objectives. Some of the core components have been listed in Table 3.2. After procuring the components, we followed a systematic approach provided by an online drone assembly guide¹.

The assembly procedure involved the following steps:

1. **Frame and Motor Installation:** The first step was to attach the motors to the F450 quadcopter frame. The frame served as the base for securing all components. Each of the four Emax XA2212-980kv brushless motors was mounted at the ends of the frame’s arms as shown in Figure 5d, ensuring balanced positioning to provide stability in flight.
2. **Power Distribution Board (PDB) and ESC Wiring:** The PDB was installed at the center of the frame to manage power distribution from the LiPo battery depicted in Figure 5b. The Electronic Speed Controllers (ESCs) were connected to each motor and then wired to the PDB as in Figure 5c to regulate the power supplied to each motor.
3. **Flight Controller Mounting and Configuration:** The Pixhawk 2.4.8 flight controller was secured in the center of the frame, above the PDB. The flight controller was then connected to the ESCs and additional sensors, such as the GPS module, to enable autonomous flight. Shock absorbers are placed to ensure the flight controller is mounted with minimal vibration.
4. **Camera Module and Raspberry Pi Integration:** The Raspberry Pi 5 and the Raspberry Pi Camera Module 3 were attached to the frame. The camera was positioned to provide a downward view to assist with ArUco marker detection during precision landing illustrated in Figure 5f. The Raspberry Pi was interfaced with the camera and linked to the Pixhawk flight controller for communication, as shown in Figure 5e.
5. **Telemetry System Setup:** The 3DR telemetry module was installed to facilitate real-time communication between the drone and the ground control station shown in Figure 5g. This allowed for remote monitoring and control during testing.
6. **Battery and Landing Gear Installation:** The Gens Ace LiPo battery was mounted securely using Velcro to provide adequate power for all components. Figure 5a depicts how the landing gear was attached to the frame to provide stability during take-off and landing.
7. **Calibration and Testing:** After completing the physical assembly, the drone was calibrated using Mission Planner² software as shown in Figure 5h. Calibration included setting up the accelerometer, compass, radio, and ESCs to optimize flight performance. This calibration process was crucial to ensure stable and precise flight control, particularly for executing autonomous landings guided by ArUco markers.

¹<https://www.instructables.com/Building-Your-First-ArduCopter/>

²<https://ardupilot.org/planner/>

Table 3.2: Components List

Sl. No.	Components Name	Quantity
1	Pixhawk 2.4.8 32 bit Flight Controller board + Safety switch + 4 gigabyte (GB) SD card	1 set
2	Emax XA2212-980kv Brushless Motor	4
3	Emax BLHeli 50A ESC with 5V3A BEC	4
4	3DR Telemetry Module 433/915MHz	1 set
5	Gens Ace 11.1V 60C 3S1P 5000mAh LiPo Battery	1
6	Raspberry Pi 5 8GB Starter Kit	1 set
7	Raspberry Pi Camera Module 3 Standard - 12MP Autofocus	1
8	NEO-M8N GPS with Compass for Pixhawk	1
9	PDB + Shock Absorber + Power Module + 5&12V	1
10	F450 Quadcopter Frame with Landing Gear	1 set

This structured approach ensured all components were properly integrated, calibrated, and tested for optimal functionality. Following the systematic guide reduced the chances of assembly errors and provided a strong foundation for implementing the precision landing features developed in this study. Figure 5i showcases the fully assembled drone.

3.3 Autopilot and On-board Computer Configuration

To establish communication between the companion computer and the autopilot, it is first necessary to adjust the read and write permissions for the serial/USB ports. This can be achieved using the following commands:

```
$ sudo adduser $USER $(stat --format="%G" /dev/ttyACM0 )
$ sudo adduser $USER $(stat --format="%G" /dev/ttyAMA0 )
```

Hardware connections can be configured with all the necessary packages and libraries installed. For the quadcopter, the telemetry port is used for serial communication between the Pixhawk 2.4.8 and Raspberry Pi 5. The connection follows the pinout configuration shown in Figure 6. Once the hardware connection is complete, the companion computer can communicate with the autopilot using Python and the DroneKit library. A script, such as

`01_connection_parameters_reading.py`, allows for establishing the connection and retrieving basic information, including firmware type, position, and, ground speed.

Setting Up the Serial Communication

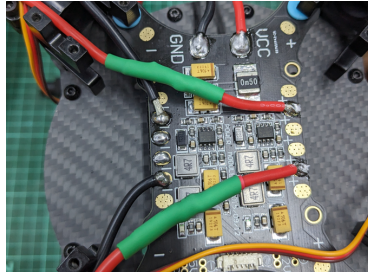
The port used for serial communication depends on the hardware setup:

- `/dev/ttyACM0`: USB connection
- `/dev/ttyAMA0`: Serial/telemetry port

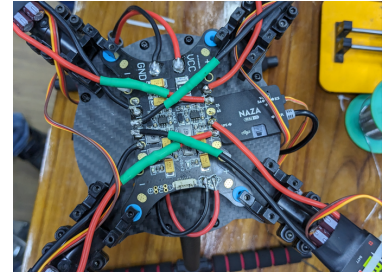
The connection can be established as follows:



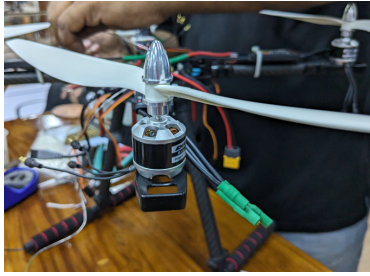
(a) Frame and Landing Gear



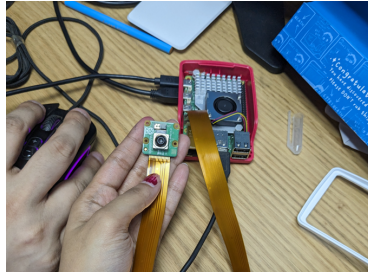
(b) PDB Wiring



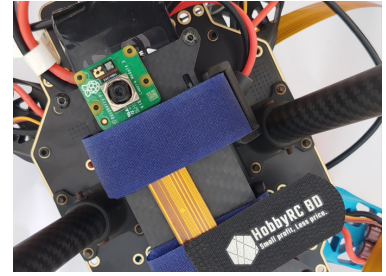
(c) ESC Wiring



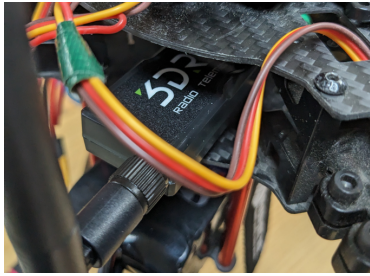
(d) Mounting Motors



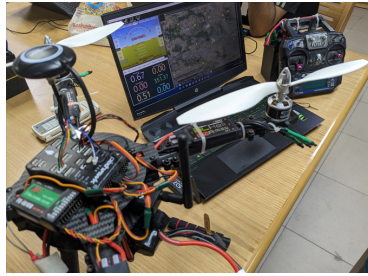
(e) Raspberry Pi Setup



(f) Pi Camera Mounting



(g) Telemetry Module



(h) Mission Planner Setup



(i) Assembled Drone

Figure 5: Snapshots of Drone Assembly Steps.

```
connection_string = "/dev/ttyACM0"
baud_rate = 921600
print(">>>> Connecting with the UAV <<<<")
vehicle = connect(connection_string, baud=baud_rate, wait_ready=True)
# The wait_ready flag ensures all parameters are loaded before proceeding.
```

Retrieving Information from the Autopilot

After the connection is established, the following data can be retrieved:

```
# Firmware version
print('Autopilot version: %s' % vehicle.version)
# Position
print('Position: %s' % vehicle.location.global_relative_frame)
# Attitude (Roll, Pitch, Yaw)
print('Attitude: %s' % vehicle.attitude)
# Velocity (North, East, Down)
print('Velocity: %s' % vehicle.velocity)
# Last Heartbeat
print('Last Heartbeat: %s' % vehicle.last_heartbeat)
```

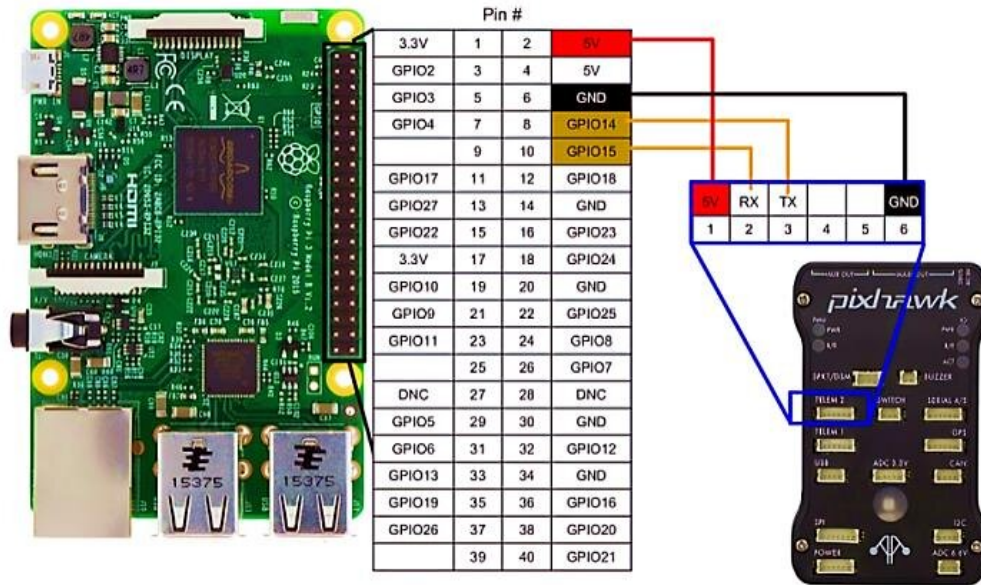


Figure 6: Connection Pinout between Pixhawk 2.4.8 and Raspberry Pi 4

```
# Armability status
print('Is the vehicle armable: %s' % vehicle.is_armable)
# Ground speed
print('Groundspeed: %s' % vehicle.groundspeed)
# Current flight mode
print('Mode: %s' % vehicle.mode.name)
# Armed status
print('Armed: %s' % vehicle.armed)
# EKF status
print('EKF Ok: %s' % vehicle.ekf_ok)
```

Adding a Listener for Real-Time Data

To monitor specific attributes over some time, a listener function can be added. For example, the attitude information can be printed every 5 seconds as follows:

```
# Adding an attitude listener
def attitude_callback(self, attr_name, value):
    print(vehicle.attitude)
print("Adding an attitude listener")
vehicle.add_attribute_listener('attitude', attitude_callback)
time.sleep(5)
# Removing the listener
vehicle.remove_attribute_listener('attitude', attitude_callback)
# Closing the connection
vehicle.close()
print("done")
```

This concludes the setup and initial interaction with the autopilot. The communication established here forms the basis for further control and data acquisition in the project.

3.4 Autonomous Mission Planning using MAVLink

MAVLink (Micro Air Vehicle Link) is a lightweight communication protocol designed for small unmanned vehicles [21]. It facilitates seamless communication between a Ground Control Station (GCS) and UAVs and manages vehicle subsystem interactions [22]. The protocol supports the transmission of critical flight data, such as orientation, GPS coordinates, and velocity, which makes it essential for monitoring and controlling UAVs.

In addition to basic communication, MAVLink is integral in configuring drones' autonomous missions. Mission Planner and MAVLink can be utilized to set up waypoints, altitude control, and other mission parameters. Mission Planner offers a powerful interface where users can plan complex flight paths, upload them to the drone, and monitor real-time telemetry during operations, as shown in Figure 7.

The Python code snippet below further illustrates how MAVLink can be used for mission planning. It demonstrates the use of the `pymavlink` library to establish a connection and define a waypoint mission:

```
from pymavlink import mavutil
master = mavutil.mavlink_connection('udpin:0.0.0.0:14550')
master.wait_heartbeat()
master.mav.mission_item_send(master.target_system, master.target_component,
                             0, mavutil.mavlink.MAV_FRAME_GLOBAL_RELATIVE_ALT,
                             mavutil.mavlink.MAV_CMD_NAV_WAYPOINT, 0, 1, 0, 0, 0, 0,
                             47.397742, 8.545594, 10)
```

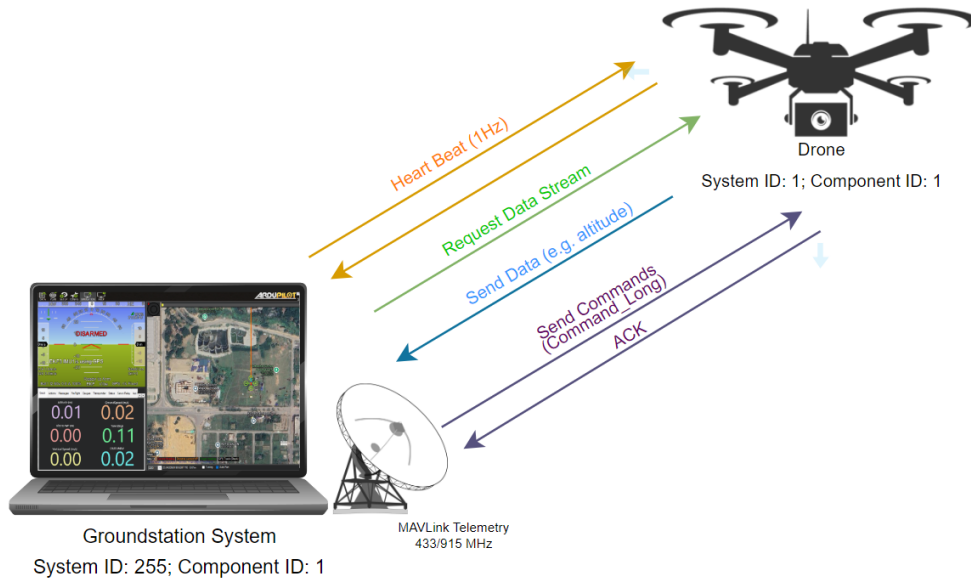


Figure 7: Communication between Drone and Ground Station using MAVLink Protocol.

3.5 Power and Performance: Batteries, Charging, and Efficiency

3.5.1 Battery Selection Criteria for UAVs

Selecting the appropriate battery for UAV applications is a critical factor that directly impacts flight performance, endurance, and mission capabilities. A recent study by Shakhatreh et al. [3] emphasized the trade-off between payload capacity and flight duration, with drones typically achieving average flight times ranging from 15 to 25 minutes under optimal conditions. Factors such as battery chemistry, charge-discharge cycles, and environmental conditions (e.g., wind resistance and temperature) worsen the problem, leading to reduced operational lifespan [4]. The complexities in choosing the optimal power source involve evaluating numerous technical requirements, such as capacity (measured in milliampere-hours, mAh), cell configuration (S-rating), discharge rate (C-rating), weight, and form factor. In our UAV project, we aimed to identify a battery that provides adequate power for an autonomous mission involving multiple sensors while ensuring that the UAV remains lightweight for increased maneuverability.

A major challenge during this selection process was finding a suitable battery with a configuration higher than 3S, a substantial capacity, and a high discharge rate to handle sudden spikes in power demands from motors and sensors. After assessing multiple candidates, the Gens Ace 11.1V 60C 3S1P 5000mAh LiPo battery was chosen to balance power capacity, weight, and high current supply capability. Table 3.3 summarizes the key specifications of this battery in comparison with other potential candidates evaluated during the selection process.

Table 3.3: Comparison of Battery Specifications for UAV

Battery Model	Configuration (S)	Capacity (mAh)	Discharge Rate (C)	Weight (g)	Max Output (A)
Gens Ace 11.1V 60C	3S1P	5000	60	410	300
Tattu 14.8V 45C	4S1P	4500	45	480	202.5
Zippy 22.2V 35C	6S1P	4000	35	625	140
Turnigy 11.1V 40C	3S1P	5200	40	430	208

As seen in Table 3.3, the Gens Ace battery offered the highest discharge capacity, which was crucial for handling the UAV's peak current requirements without causing excessive heating or voltage drops. Furthermore, its moderate weight helped maintain an acceptable power-to-weight ratio, vital for UAV flight efficiency.

3.5.2 Challenges in Battery Selection

Multiple parameters influence the selection of a power source for UAV applications:

1. **Capacity vs. Weight Trade-Off:** Higher capacity batteries increase flight time but add weight, negatively impacting the drone's thrust-to-weight ratio and maneuverability.
2. **Discharge Rate (C-Rating):** The discharge rate represents the battery's ability to deliver power immediately. For our application, the drone required a C-rating of

at least 50C to support high-thrust flight maneuvers without risking power failure or damage to the battery.

3. **Configuration (S-Rating):** A higher S-rating (e.g., 4S or 6S) increases power but often leads to heavier and bulkier battery packs. Balancing power output with system compatibility was an ongoing challenge, as many components, such as the speed controllers (ESCs), were tuned for 3S operation.

3.5.3 Charging Mechanisms and Efficiency Considerations

To maximize UAV uptime, efficient charging solutions were considered. The research investigated both conventional charging and autonomous charging approaches. The latter involved a combination of vision-guided docking using ArUco markers and a Proportional–Integral–Derivative (PID) control mechanism to ensure precision landings for reliable charging.

Table 3.4 compares various charging solutions investigated for this study. The key attributes considered included charging time, efficiency, and the feasibility of integration with the UAV system.

Table 3.4: Comparison of Charging Methods for UAV Batteries

Charging Method	Charging Time	Efficiency	Integration Complexity
Manual Plug-in	1-1.5 hours	High	Low
Wireless Power Transfer (WPT)	2-3 hours	Moderate	High
Solar-Assisted Autonomous Charging	Variable	Low-Medium	Very High

Wireless Power Transfer (WPT) emerged as a promising solution to eliminate manual intervention. However, WPT’s efficiency remained lower than traditional plug-in methods, and precision docking was needed to align the UAV with the charging coils. The proposed solar-assisted charging solution aims to enhance long-term sustainability, albeit at the cost of increased system complexity.

3.6 Battery and Power Consumption Calculations

The following section outlines the calculations performed to determine our drone’s battery performance and flight endurance. The battery utilized for this analysis was a 5000 mAh, 11.1V (3S) LiPo battery. The calculations include the total current draw from all components, power consumption, and estimated flight time.

Total Current Draw Calculation

The total current draw (I_{total}) of the drone can be calculated by summing the current draw of each motor and additional components:

$$I_{\text{total}} = I_{\text{motor}} \times 4 + I_{\text{controller}} + I_{\text{GPS}} \quad (3.1)$$

where:

- $I_{\text{motor}} = 9.5 \text{ A}$ (per motor)

- Number of motors = 4
- $I_{\text{controller}} = 0.3 \text{ A}$
- $I_{\text{GPS}} = 0.06 \text{ A}$

Thus:

$$I_{\text{total}} = (9.5 \times 4) + 0.3 + 0.06 = 38.36 \text{ A} \quad (3.2)$$

Power Consumption Calculation

The total power consumption (P_{total}) is determined by multiplying the total current draw by the battery voltage:

$$P_{\text{total}} = V_{\text{battery}} \times I_{\text{total}} \quad (3.3)$$

where $V_{\text{battery}} = 11.1 \text{ V}$. Therefore:

$$P_{\text{total}} = 38.36 \times 11.1 = 425.796 \text{ W} \quad (3.4)$$

Battery Capacity in Watt-Hours

The battery capacity needs to be converted from mAh to Ah and then into watt-hours:

$$C_{\text{battery (Ah)}} = \frac{C_{\text{battery (mAh)}}}{1000} = \frac{5000}{1000} = 5 \text{ Ah} \quad (3.5)$$

$$C_{\text{battery (Wh)}} = C_{\text{battery (Ah)}} \times V_{\text{battery}} \times N_{\text{cell}} \quad (3.6)$$

$$C_{\text{battery (Wh)}} = 5 \times 11.1 \times 3 = 166.5 \text{ Wh} \quad (3.7)$$

Estimated Flight Time (Endurance) Calculation

The endurance of the drone can be calculated by dividing the battery capacity in watt-hours by the total power consumption:

$$\text{Endurance (Hours)} = \frac{C_{\text{battery (Wh)}}}{P_{\text{total}}} \quad (3.8)$$

$$\text{Endurance} = \frac{166.5}{425.796} \approx 0.391 \text{ hours} = 23.46 \text{ minutes} \quad (3.9)$$

3.6.1 Summary of Battery Performance and Power Consumption

Table 3.5 summarizes the calculated values for battery capacity, power consumption, and expected flight endurance.

Table 3.5: Battery Performance and UAV Endurance Summary

Parameter	Value
Battery Capacity (mAh)	5000
Battery Voltage (V)	11.1
Total Current Draw (A)	38.36
Total Power Consumption (W)	425.796
Battery Capacity (Wh)	55.5
Estimated Flight Time (minutes)	$23.46 \pm 10\%$

3.6.2 Verification and Relative Error

The results were compared to online calculators and estimations from ArduPilot resources. Using an online tool³, the estimated flight time was found to be similar to the calculated value with a relative error of approximately 16.31%, which is within acceptable limits for our testing environment.

$$\text{Relative Error (\%)} = \left| \frac{\text{Exact Value} - \text{Approximated Value}}{\text{Exact Value}} \right| \times 100 \quad (3.10)$$

$$\text{Relative Error} = \left| \frac{20.17 - 23.46}{20.17} \right| \times 100 \approx 16.31\% \quad (3.11)$$

³<https://www.ecalc.ch/xcoptercalc.php?ecalc>

4 OpenCV ArUco Markers for Detection

4.1 Configuring OpenCV

As part of the project development, two advanced tools in the field of computer vision were utilized to enhance functionality and performance.

The first tool is OpenCV¹ (Open Source Computer Vision Library), a versatile, cross-platform, open-source software library designed specifically for real-time computer vision applications. OpenCV provides a comprehensive suite of functions for acquiring, analyzing, and manipulating visual data captured by cameras, video files, or other devices. While it includes basic functions for drawing shapes and lines, its true strength lies in its advanced algorithms for tasks such as face detection, motion tracking, and shape analysis [23, 24].

4.1.1 Installing Essential Libraries and Packages

Begin by installing the fundamental build tools and necessary development libraries using the following commands:

```
1 sudo apt-get update
2 sudo apt-get install build-essential cmake pkg-config
3 sudo add-apt-repository 'deb http://ports.ubuntu.com/ubuntu-ports xenial-
  security main'
4 sudo apt-get update
5 sudo apt-get install libjpeg-dev libtiff5-dev libjasper-dev libpng-dev
6 sudo apt-get install libavcodec-dev libavformat-dev
7 sudo apt-get install libswscale-dev libv4l-dev
8 sudo apt-get install libxvidcore-dev libx264-dev
9 sudo apt-get install libgtk2.0-dev
10 sudo apt-get install libatlas-base-dev gfortran
11 sudo pip install numpy
12 sudo pip3 install numpy
```

Listing 4.1: Installing Build Essentials and Development Libraries

4.1.2 Downloading and Building OpenCV and OpenCV-Contrib

After installing the necessary libraries, download and build the latest versions of OpenCV and OpenCV-Contrib. Execute the following commands:

```
1 wget -O opencv.zip https://github.com/Itseez/opencv/archive/4.6.0.zip
2 unzip opencv.zip
```

¹<https://opencv.org/>

```
3 wget -O opencv_contrib.zip https://github.com/Itseez/opencv_contrib/archive
  /4.6.0.zip
4 unzip opencv_contrib.zip
```

Listing 4.2: Downloading OpenCV and OpenCV-Contrib

4.1.3 Building and Installing OpenCV

Once you have downloaded both OpenCV and OpenCV-Contrib, create a build directory, configure the build with CMake, and compile the source code. Depending on the system's performance, the compilation process may take over two hours.

```
1 cd opencv-4.6.0/
2 mkdir build
3 cd build/
4 cmake -D CMAKE_BUILD_TYPE=RELEASE \
5       -D CMAKE_INSTALL_PREFIX=/usr/local \
6       -D INSTALL_PYTHON_EXAMPLES=ON \
7       -D OPENCV_EXTRA_MODULES_PATH=~/opencv_contrib-4.6.0/modules/ \
8       -D BUILD_EXAMPLES=ON ..
9 make -j4
10 sudo make install
11 cd
12 sudo ldconfig
```

Listing 4.3: Building and Installing OpenCV

4.1.4 Verifying the Installation

To confirm that OpenCV has been successfully installed, open a Python interpreter and attempt to import the `cv2` module. If no error messages appear, the configuration is correct.

```
1 python
2 >>> import cv2
```

Listing 4.4: Verifying OpenCV Installation

4.2 Overview of ArUco Markers

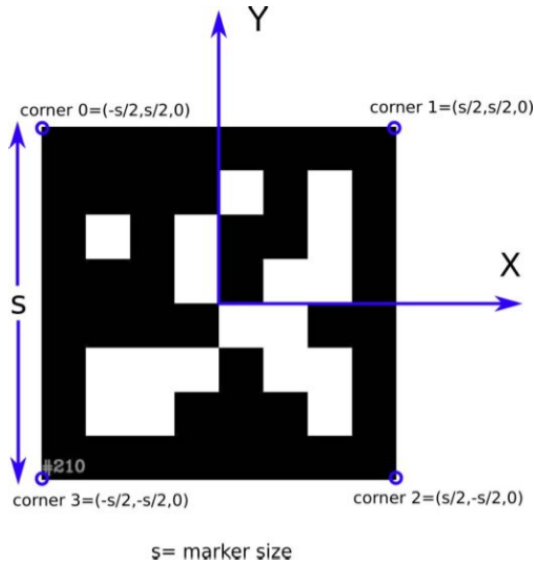
The second tool utilized in this project is the fiducial marker system, which plays a crucial role in the field of computer vision. Various types of fiducial markers exist, such as ArUco and AprilTags, each offering unique functionalities. One of the key features leveraged in this project is Pose Estimation. This process involves identifying correspondences between points in the physical environment and their 2D projections in an image. A widely adopted approach for this purpose is the use of binary square fiducial markers.

The primary advantage of these markers lies in their ability to provide sufficient correspondences for camera pose estimation using just a single marker. Specifically, the four corners of the marker are used for this calculation. Additionally, their internal binary encoding enhances robustness by enabling error detection and correction mechanisms. For this application, ArUco markers were selected due to their reliability and compatibility

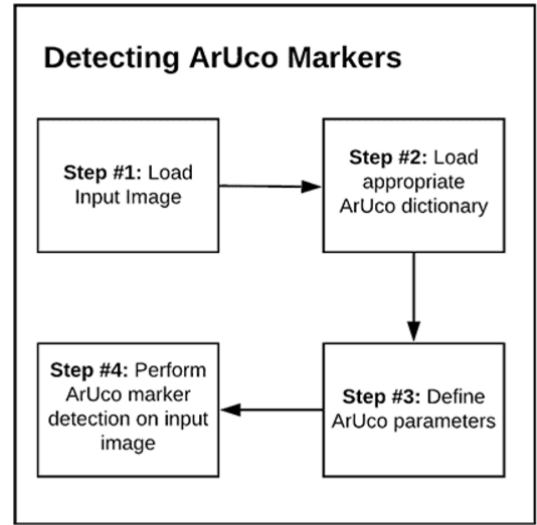
with the project requirements. These markers are detected and processed using the popular ArUco library [25], a widely used tool for square fiducial marker detection. An Example of ArUco marker is presented in Figure 8.

4.2.1 Structure of ArUco Markers

The size of the marker directly determines the size of its internal matrix. For example, a marker with a size of 4×4 contains 16 bits in its binary matrix. It is important to note that markers may appear rotated within an environment. However, the detection process is robust enough to identify the original orientation of the marker and assign correct labels to its corners. This orientation detection is achieved by interpreting the binary encoding. A sample marker and the basic detection process are illustrated in Figure 8.



(a) Basic Structure of an ArUco Marker



(b) ArUco Detection Process

Figure 8: Sample ArUco Marker with Pose Axis and Basic Detection Process Flowchart.

4.2.2 ArUco Dictionaries

An ArUco dictionary is a collection of markers of various sizes used within a specific application, as shown in Figure 9. Each dictionary is defined by the binary codes of its markers. The two main attributes of a dictionary are:

- **Dictionary Size:** The number of markers included in the dictionary.
- **Marker Size:** The size of each marker's binary matrix, measured in bits or modules.

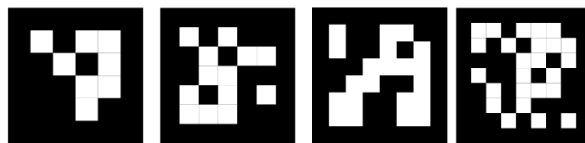


Figure 9: ArUco Markers with Different Matrix Sizes (4×4 , 5×5 , 6×6 , 7×7 matrices).

The `aruco` module provides several predefined dictionaries, accommodating a variety of dictionary sizes and marker sizes.

Contrary to initial assumptions, a marker’s *id* is not derived by converting its binary code into a decimal number. This approach is impractical for larger markers due to the many bits involved. Instead, the *id* corresponds to the index of the marker within its dictionary. For instance, in a dictionary, the first five markers are assigned the IDs: 0, 1, 2, 3, and 4 depicted in Figure 10.



The ArUco marker detection process involves several key steps. The detailed steps for detection are given below:

- 33

- The marker will be generated using the University of Maryland’s online generator [26].

The above process has been summarized in Algorithm 1

Algorithm 1 ArUco Marker Detection Process

Require: Input image I , ArUco dictionary D , Camera calibration parameters (K , $dist$)

Ensure: Detected markers with pose estimations (IDs, corners, R , T)

```

1: Step 1: Preprocess the Image
2: Convert  $I$  to grayscale:  $I_{gray} \leftarrow \text{grayscale}(I)$ 
3: Apply noise reduction if necessary
4: Step 2: Detect ArUco Markers
5: Load ArUco dictionary  $D$ 
6: Use aruco.detectMarkers to find markers in  $I_{gray}$ 
7: Retrieve marker corners  $C$  and IDs  $ID$ 
8: Step 3: Validate Detected Markers
9: if no markers detected then
10:   Return None
11: end if
12: Step 4: Estimate Marker Poses
13: for each detected marker  $i$  in  $C$  do
14:   Estimate rotation ( $R_i$ ) and translation ( $T_i$ ) vectors using
15:   aruco.estimatePoseSingleMarkers
16:   Store  $ID_i$ ,  $C_i$ ,  $R_i$ ,  $T_i$ 
17: end for
18: Step 5: Return Results
19: return Marker IDs  $ID$ , corners  $C$ , and pose estimations ( $R$ ,  $T$ )

```

4.3.1 Implementation Details

The integration of ArUco markers for precision landing in drones involves two primary components: marker detection and drone navigation. ArUco markers serve as fiducial markers that are uniquely identifiable and easily detected under varying lighting conditions. The implementation pipeline is as follows:

1. **Marker Detection:** Using the OpenCV library, the onboard camera identifies the ArUco marker. The marker’s ID and pose (position and orientation) are computed using the `cv2.aruco.detectMarkers()` function.
2. **Pose Estimation:** The 3D pose of the marker is estimated relative to the drone’s camera using the `cv2.aruco.estimatePoseSingleMarkers()` function.
3. **Navigation Adjustment:** The drone dynamically adjusts its position based on the computed relative pose to align with the marker for a smooth descent.

The code used for implementation can be found in our project’s GitHub repository².

²https://github.com/KhandakerUmmeSalma/PrecisionLanding_DroneProject

Below is a sample pseudocode implementation of the detection and pose estimation process:

```
1 1. Import required libraries (cv2, numpy, matplotlib).
2
3 2. Load the input image:
4   - Read the image file (e.g., 'marker.png').
5   - Check if the image was loaded successfully.
6
7 3. Preprocess the image:
8   - Convert the image to grayscale.
9   - Optionally enhance the contrast for better marker detection.
10
11 4. Configure the ArUco marker detection:
12   - Define the ArUco dictionary (e.g., DICT_6X6_250).
13   - Set up detection parameters.
14
15 5. Detect ArUco markers in the image:
16   - Use the ArUco detector to find marker corners and IDs.
17
18 6. If markers are detected:
19   - Print detected marker IDs and their corner coordinates.
20
21 7. For each detected marker:
22   - Estimate the pose (rotation and translation vectors) based on known
23     marker size and camera parameters.
24   - If pose estimation is successful:
25     - Draw marker axes (X, Y, Z) on the image.
26     - Draw a bounding box around the marker.
27     - Display marker ID and position information.
28
29 8. If no markers are detected:
30   - Print a message indicating detection failure.
31
32 9. Display the image with visualizations:
33   - Show the processed image with detected markers, bounding boxes, axes,
34     and pose data.
```

Listing 4.5: Pseudocode for ArUco Marker Detection and Pose Estimation

4.4 Precision Landing Algorithm

Following the successful recognition of the marker in the previous implementation, the functionality will be encapsulated in a Python library. This library will then be integrated into a precision landing script. The script will establish a connection with the UAV, utilize the marker recognition library, navigate the craft above the marker, and execute the landing procedure.

4.4.1 Precision Landing Logic

The precision landing algorithm is designed to execute the landing in a series of seven systematic steps, as outlined below:

- The marker is detected, and its position is determined within the camera frame.
- This position is transformed from the camera to the body frame.

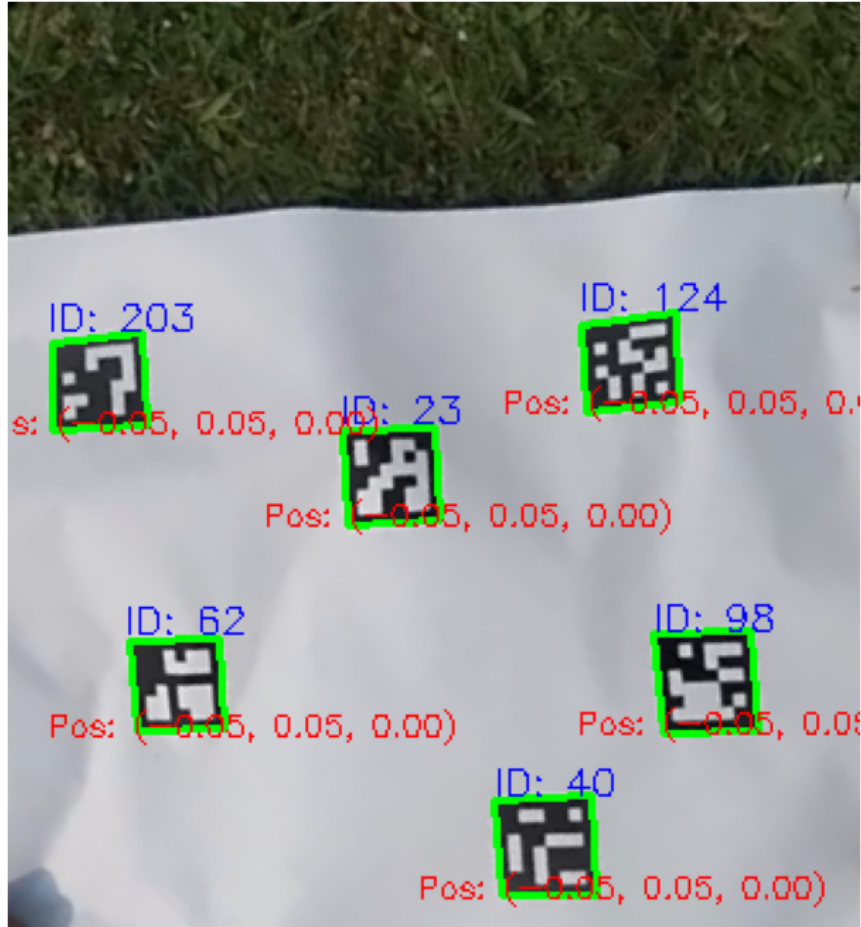


Figure 11: Detected Marker IDs and Pose Estimation Output

```

Detected marker IDs: [ 98 40 23 124 203 62]
Corners of detected markers: (array([[251., 246.],
[289., 245.],
[292., 284.],
[255., 285.]]], dtype=float32), array([[189., 311.],
[227., 309.],
[229., 348.],
[192., 348.]]], dtype=float32), array([[130., 167.],
[166., 166.],
[168., 202.],
[132., 204.]]], dtype=float32), array([[260., 121.],
[261., 156.],
[225., 160.],
[223., 126.]]], dtype=float32), array([[17., 133.],
[52., 129.],
[54., 165.],
[17., 167.]]], dtype=float32), array([[84., 283.],
[50., 284.],
[47., 249.],
[82., 248.]]], dtype=float32))
Marker ID 98:
Rotation Vector (rvec): [[[-0.94612733 -0.24390233 2.56906737]]]
Translation Vector (tvec): [[[-0.05 0.05 0. ]]

[[ 0.05 0.05 0. ]]

[[ 0.05 -0.05 0. ]]

[[ -0.05 -0.05 0. ]]
Marker ID 40:
Rotation Vector (rvec): [[[-1.13067004 -0.08128352 2.62482121]]]
Translation Vector (tvec): [[[-0.05 0.05 0. ]]

[[ 0.05 0.05 0. ]]

[[ 0.05 -0.05 0. ]]

[[ -0.05 -0.05 0. ]]
Marker ID 23:
Rotation Vector (rvec): [[[-1.35080343 -0.48215514 2.75256874]]]
Translation Vector (tvec): [[[-0.05 0.05 0. ]]

[[ 0.05 0.05 0. ]]

[[ 0.05 -0.05 0. ]]

[[ -0.05 -0.05 0. ]]
Marker ID 124:
Rotation Vector (rvec): [[[-1.07740842 -0.59404415 2.70871203]]]
Translation Vector (tvec): [[[-0.05 0.05 0. ]]

[[ 0.05 0.05 0. ]]

[[ 0.05 -0.05 0. ]]

[[ -0.05 -0.05 0. ]]
Marker ID 62:
Rotation Vector (rvec): [[[-1.67871381 -0.58694064 2.77473523]]]
Translation Vector (tvec): [[[-0.05 0.05 0. ]]

[[ 0.05 0.05 0. ]]

[[ 0.05 -0.05 0. ]]

[[ -0.05 -0.05 0. ]]
Marker ID 203:
Rotation Vector (rvec): [[[-1.64712779 -0.26955193 2.86829392]]]
Translation Vector (tvec): [[[-0.05 0.05 0. ]]

[[ 0.05 0.05 0. ]]

[[ 0.05 -0.05 0. ]]

```

Figure 12: Output of Marker Detection Code.

- The body frame coordinates are converted into a North-East frame, with the UAV's center of gravity as the reference point.
- The latitude and longitude of the marker are calculated by adding the North-East offset to the UAV's current location.
- The UAV is directed to the computed marker location.
- Upon reaching the marker, the UAV descends toward the marker if the positional error is within an acceptable threshold.
- The UAV initiates the landing sequence Once the altitude is sufficiently low.

The block diagram in Figure 13 illustrates the logical flow of the precision landing algorithm.

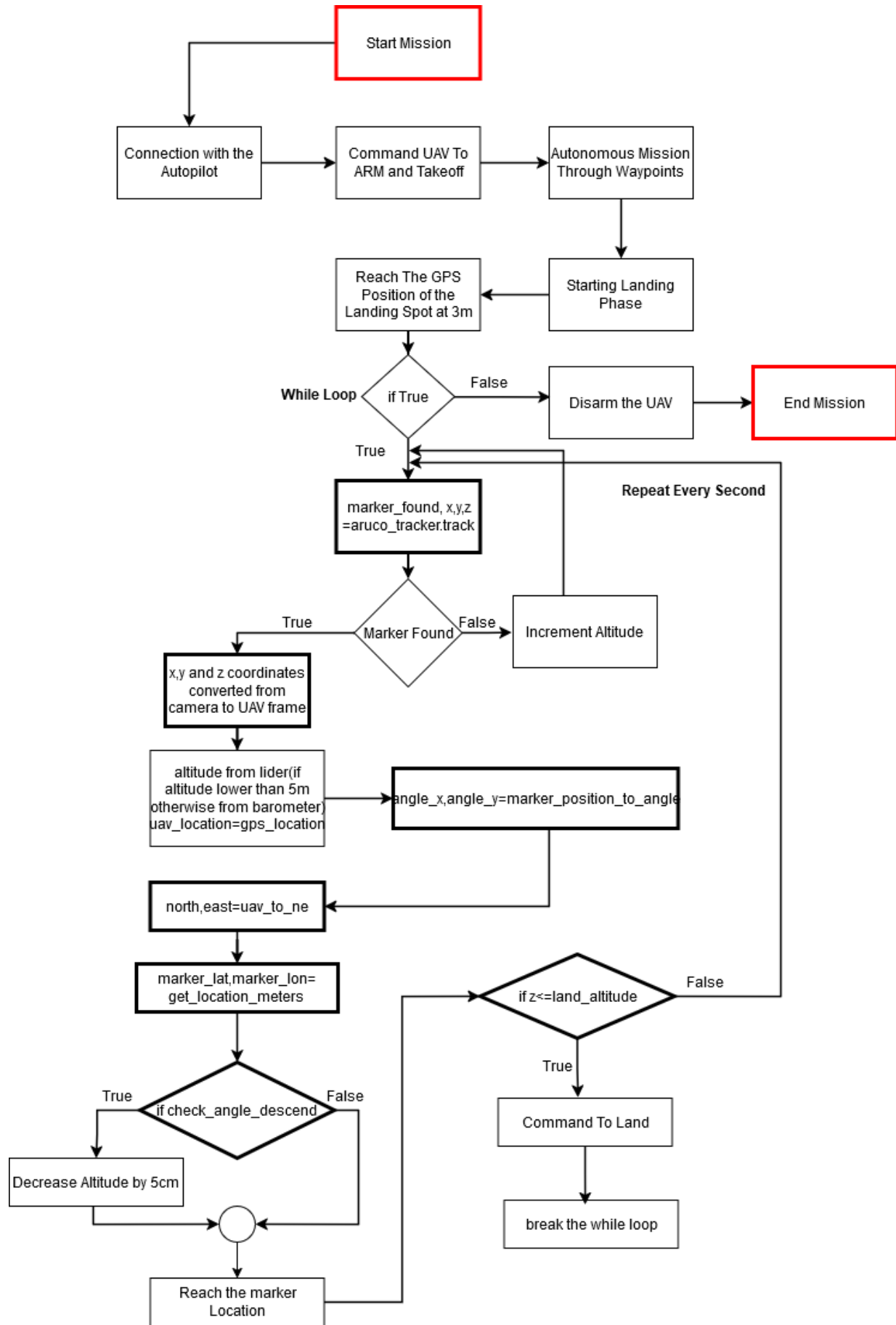


Figure 13: Vision-based Precision Landing Process Flowchart.

5 Role of PID Controller in Precision Landing

5.1 Integration with ArUco Markers

The PID (Proportional-Integral-Derivative) controller [27] plays a crucial role in achieving precision landing in UAVs, especially when integrated with ArUco markers. These markers provide a visual reference for the drone to identify the landing pad, while the PID controller ensures that the drone can land accurately and stably, even in the presence of external disturbances. The combination of these technologies allows for highly precise and autonomous landing operations, which are essential for applications such as autonomous charging stations and mission-critical operations.

5.1.1 Understanding PID Controllers

To maintain a stable and precise hover or landing position, a UAV relies on a PID control system [28]. This system combines three components: proportional, integral, and derivative controllers. Each of these components addresses a specific aspect of flight stability and precision, working together to manage the drone's position and velocity effectively. Figure 14 depicts how each controller functions together.

Proportional Controller

The proportional controller (P-controller) responds to the immediate difference between the desired position (setpoint) and the current position (process variable). Mathematically, the proportional control action can be expressed as:

$$P = K_p \cdot e(t)$$

where:

- P is the proportional output,
- K_p is the proportional gain,
- $e(t)$ is the error (difference between the set-point and process variable).

Larger deviations result in stronger corrective actions, allowing the drone to respond quickly to positional errors. However, while effective for immediate corrections, the P-controller alone cannot eliminate persistent steady-state errors.

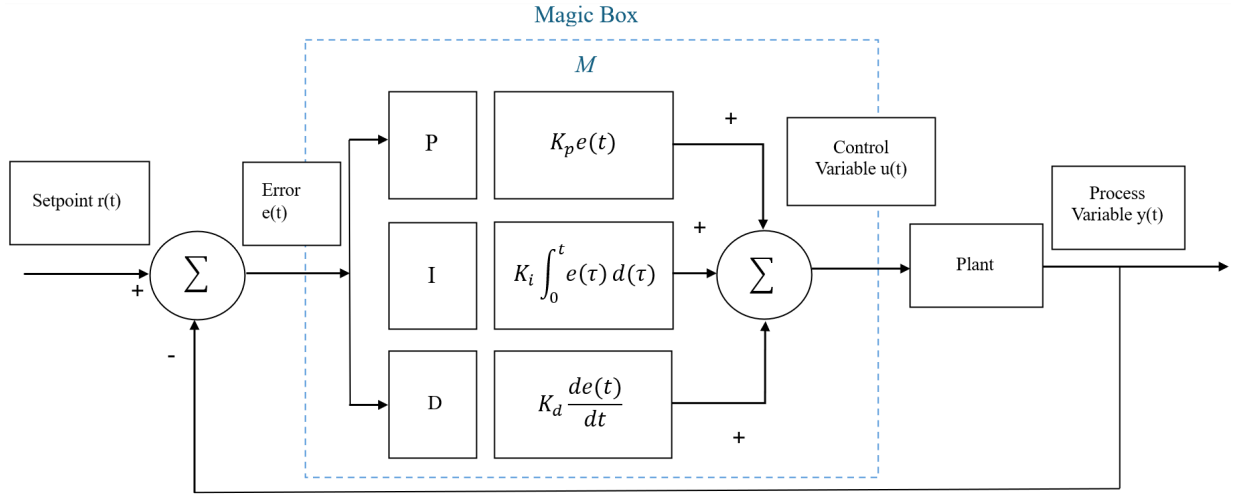


Figure 14: Basic PID Controller Block Diagram.

Derivative Controller

The derivative controller (D-controller) anticipates future errors by analyzing the rate of change of the positional error. It ensures smooth adjustments by moderating rapid changes, such as those caused by gusts of wind or abrupt movements. The derivative control action is given by:

$$D = K_d \cdot \frac{de(t)}{dt}$$

where:

- D is the derivative output,
- K_d is the derivative gain,
- $\frac{de(t)}{dt}$ is the rate of change of the error.

This enhances the system's stability and prevents oscillations, making it particularly valuable in maintaining a steady descent during precision landing.

Integral Controller

The integral controller (I-controller) addresses the limitations of the P- and D-controllers by considering the accumulation of past errors. It gradually corrects persistent deviations from the target position, eliminating steady-state errors. The integral control action is expressed as:

$$I = K_i \cdot \int_0^t e(\tau) d\tau$$

where:

- I is the integral output,
- K_i is the integral gain,
- $\int_0^t e(\tau) d\tau$ represents the accumulated error over time.

This ensures that the drone can maintain precise altitude over extended periods, even in dynamic environments.

5.1.2 PID Controller Algorithm and System Design

A PID controller is implemented to correct deviations in the drone's position based on ArUco marker detection to ensure precise autonomous landing of the UAV on target. The PID controller reduces the error between the drone's current position and the target center of the landing pad in three-dimensional space. The detailed algorithm of the PID-assisted landing is described in Algorithm 2.

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt} \quad (5.1)$$

where:

- $u(t)$ is the control signal applied to the UAV.
- $e(t)$ is the error: $e(t) = \text{setpoint} - \text{current_value}$.
- K_p , K_i , and K_d are the proportional, integral, and derivative gains, respectively.

The PID controller activates when the ArUco marker is detected within the camera's field of view. The detection reliability improves at lower altitudes (below 10 meters), triggering the PID controller to reduce positional deviations precisely. The PID controller adjusts the UAV's position using Equation 5.1. Using the onboard camera, the center of the target landing pad is identified, and the relative offset $(x_{\text{off}}, y_{\text{off}}, z_{\text{off}})$ is measured from the UAV. The PID controller computes the control signals for each axis (x, y, z) using the respective errors (e_x, e_y, e_z) :

$$u_i = K_p e_i + K_i \int e_i dt + K_d \frac{de_i}{dt}; \quad \text{for } i \in \{x, y, z\} \quad (5.2)$$

The control signals u_x , u_y , and u_z adjust the drone's motors, gradually aligning the UAV with the center of the landing pad, as described in Equation 5.2. The gains K_p , K_i , and K_d are empirically tuned to balance responsiveness and stability. The controller terminates when positional error falls below $|e_x|, |e_y|, |e_z| < 0.01$ meters.

5.1.3 Importance in Drone Flight and Precision Landing

The integration of the PID control system with ArUco markers is critical for precision landing. ArUco markers provide the drone with a visual guide to locate the landing pad accurately, while the PID controller ensures that the drone's movements are stable and responsive.

The overall PID control action combines these three components:

$$u(t) = K_p \cdot e(t) + K_i \cdot \int_0^t e(\tau) d\tau + K_d \cdot \frac{de(t)}{dt}$$

As the drone descends, the proportional controller corrects immediate positional errors, the derivative controller moderates rapid altitude changes, and the integral controller compensates for any accumulated deviations. Together, these controllers enable the UAV to make precise adjustments, ensuring a smooth and accurate landing even under challenging conditions such as uneven terrain or external disturbances.

Algorithm 2 PID for Drone Precision Landing Assist

```
1: Input: PID parameters  $K_p, K_i, K_d$ ; initial UAV position; ArUco marker position;
   camera feedback of the marker's offset from the UAV center
2: Output: Estimated UAV position relative to the landing pad center
3: Initialize PID parameters  $K_p, K_i, K_d$  and setpoint values for each axis  $(x, y, z)$ 
4: while drone has not reached the target within an acceptable threshold do
5:   Detect marker using camera and calculate offset from the UAV's center:  $(x_{\text{off}}, y_{\text{off}}, z_{\text{off}})$ 

6:   for each axis  $i \in \{x, y, z\}$  do
7:     Compute current error:  $e_i = \text{setpoint}_i - \text{current\_value}_i$ 
8:     Compute proportional term:  $P_i = K_p \times e_i$ 
9:     Update integral term:  $\text{integral}_i += e_i \times dt$ 
10:    Compute integral control:  $I_i = K_i \times \text{integral}_i$ 
11:    Calculate derivative term:  $D_i = K_d \times \frac{e_i - e_{i,\text{prev}}}{dt}$ 
12:    Compute the control signal:  $u_i = P_i + I_i + D_i$ 
13:    Update UAV's position along axis  $i$ :  $\text{current\_value}_i += u_i$ 
14:    Update previous error:  $e_{i,\text{prev}} = e_i$ 
15:   end for
16: end while
17: Return UAV's final position
```

Additional Considerations

Tuning PID Parameters Proper tuning of the PID gains (K_p, K_i, K_d) is essential for optimal performance. Common tuning methods include:

- **Ziegler-Nichols Method:** A heuristic tuning method that involves setting K_i and K_d to zero and increasing K_p until the system oscillates at a constant amplitude. The ultimate gain and oscillation period are then used to set K_p, K_i , and K_d .
- **Trial and Error:** Manually adjusting the gains while observing system performance.
- **Software-Based Optimization:** Using algorithms to optimize PID parameters based on predefined criteria.

Handling Nonlinearities and External Disturbances In real-world scenarios, UAVs encounter various nonlinearities and disturbances such as wind gusts, sensor noise, and payload variations. Advanced PID control strategies or complementary control mechanisms (e.g., feedforward control, adaptive PID) can be employed to enhance robustness.

Sensor Fusion and State Estimation Integrating data from multiple sensors (e.g., IMU, GPS, camera) through sensor fusion techniques like the Kalman filter can improve state estimation accuracy, leading to more precise control actions.

Real-Time Implementation Ensuring that the PID controller operates in real-time with minimal latency is crucial for maintaining stability and responsiveness during landing. Efficient algorithms and optimized code execution on the UAV's onboard computer are necessary to achieve this.

Redundancy and Safety Mechanisms Implementing redundancy in sensors and control pathways, along with safety mechanisms such as emergency landing protocols, can enhance the reliability and safety of the precision landing system.

Extended Mathematical Model

Consider the UAV's dynamics in three-dimensional space for a more comprehensive understanding. The PID controller can be integrated into the UAV's state-space representation:

$$\begin{cases} \dot{x} = v_x \\ \dot{v}_x = \frac{u_x}{m} - g \sin(\theta) \\ \dot{y} = v_y \\ \dot{v}_y = \frac{u_y}{m} - g \sin(\phi) \\ \dot{z} = v_z \\ \dot{v}_z = \frac{u_z}{m} - g + w(t) \end{cases}$$

where:

- x, y, z are the UAV's positions in three-dimensional space.
- v_x, v_y, v_z are the velocities along each axis.
- u_x, u_y, u_z are the control inputs from the PID controller.
- m is the mass of the UAV.
- g is the acceleration due to gravity.
- θ and ϕ are the pitch and roll angles, respectively.
- $w(t)$ represents external disturbances.

In this model, the PID controller adjusts u_x, u_y , and u_z to control the UAV's position and velocity, compensating for disturbances and maintaining stability during the landing process.

When integrated with ArUco markers, the PID controller forms a robust framework for precision landing in UAVs. This synergy ensures that drones can autonomously and accurately land on designated pads, which is vital for various applications ranging from automated delivery systems to automated wireless charging capabilities. By meticulously tuning the PID parameters and incorporating advanced control strategies, UAVs can achieve reliable and stable landings despite environmental challenges.

6 Wireless Power Transfer System

6.1 Introduction to Wireless Power Transfer (WPT)

Wireless Power Transfer (WPT) is a technology that uses electromagnetic fields to transmit electrical power without physical connections [29]. This technique is particularly advantageous for drones, which are constrained by limited battery capacity and require frequent charging to sustain prolonged operation in autonomous applications such as precision agriculture, monitoring, and surveillance.

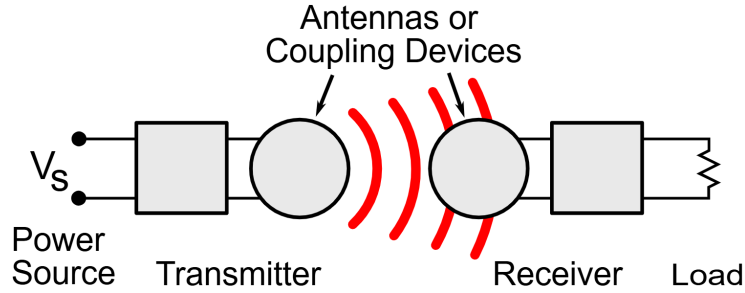


Figure 15: Generic Block Diagram of a Wireless Power System.

6.2 Principle and Operation of WPT

WPT operates based on electromagnetic coupling between transmitter and receiver coils. The efficiency of power transfer depends on the alignment, distance, and resonant frequency of these coils. The key parameters for evaluating WPT systems include:

- **Coupling Factor (K):** Indicates the efficiency of magnetic field linkage between coils. It is defined as:

$$K = \frac{M}{\sqrt{L_r L_s}} \quad (6.1)$$

where M is mutual inductance, and L_r and L_s are the self-inductances of the receiver and transmitter coils, respectively.

- **Resonant Frequency (f_o):** Determines the operating frequency of the system, calculated as:

$$f_o = \frac{1}{2\pi\sqrt{LC}} \quad (6.2)$$

where L is inductance, and C is capacitance.

- **Transfer Efficiency (η_T):** Defined as:

$$\eta_T = \frac{P_{\text{DC, output}}}{P_{\text{DC, input}}} \times 100 \quad (6.3)$$

6.3 Implementation of WPT for Drone Charging

In the following system by Jawad, Aqeel Mahmood, et al. [2], WPT is implemented using:

- A flat spiral coil (FSC) transmitter with 150 turns of 18 AWG copper wire.
- A multiturn coil (MTC) receiver with 60 turns of 24 AWG copper wire mounted on the drone.

6.3.1 System Design

The transmitter coil is fixed at the charging station, while the receiver coil is integrated into the drone. Key system parameters are shown in Table 6.1.

Table 6.1: Specifications of Transmitter and Receiver Coils

Parameter	Transmitter (FSC)	Receiver (MTC)
Number of turns	150	60
Wire gauge	18 AWG	24 AWG
Diameter (cm)	40	40
Inductance (μH)	293.17	152.34

6.3.2 Experimental Results

The alignment and misalignment performance of the WPT system was evaluated. Under ideal alignment, a transfer efficiency of 85.25% was achieved with a power output of 20.46 W at a 0 cm air gap. For a 1 cm misalignment, the efficiency dropped to 71.25%.

6.3.3 Flight Time Enhancement

The drone, equipped with a 5.5 Ah Li-Po battery, achieved a flight time of 25.84 minutes under conventional operation. By integrating the WPT system and a sleep/active strategy, the battery life was extended to 851 minutes, enabling approximately 170 missions over 28 days without manual recharging.

The proposed system by [2] demonstrated high efficiency and reliability for drone charging in smart agriculture applications. The integration of WPT with drones mitigates battery limitations and enhances operational sustainability.

7 Experimental Design and Testing

7.1 Methodology

The methodology is divided into the following phases: requirements analysis, UAV setup and sensor interfacing, autonomous flight mission configuration, and the design and implementation of the PID controller. The ultimate goal of this experiment is to test the efficacy of wireless charging stations for drones to extend their operational efficiency. Figure 16 illustrates the potential real-world implementation schematic using a flow chart.

7.1.1 Requirements Analysis

The study began with an extensive requirements analysis to identify the operational limitations of UAVs, particularly in the context of autonomous navigation and landing. The key challenges addressed included

- Limited flight duration due to battery constraints.
- Declining GPS accuracy with reduced battery charge and increased altitude.
- The need for precise autonomous landing in varying environmental conditions.

The system requirements were defined as follows:

- i. Autonomous navigation capabilities using GPS for waypoint guidance with the aid of Mission Planner.
- ii. Integration of a vision-based detection system to enhance landing accuracy.
- iii. Implementation of a PID controller to assist ArUco marker-guided precision landing during descent.

7.1.2 Autonomous Flight Mission Configuration

Field experiments were conducted in an open field at Rupganj, Dhaka (23.8394°N, 90.5369°E), under varying wind conditions. Autonomous missions were programmed using Mission Planner version 1.3.82, ArduPilot's ground control software, designed for Pixhawk flight controllers, as shown in Figure 17. Regulatory approvals were obtained for UAV operations. The test included a predefined landing target equipped with ArUco markers for visual detection. The mission setup followed these steps:

- i. **Waypoint Navigation:** Predefined waypoints were configured in the Mission Planner software for autonomous navigation.

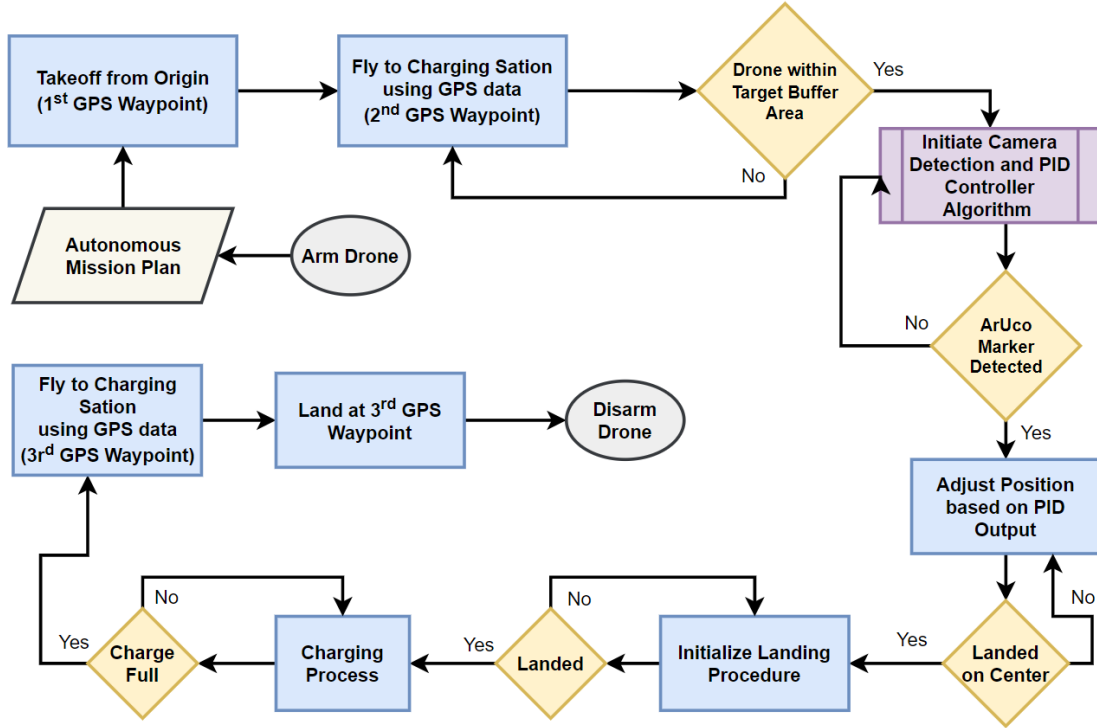


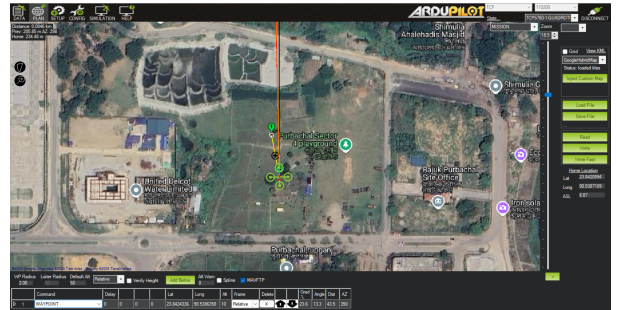
Figure 16: Proposed Flowchart for Precision Landing Procedure for Autonomous Drones.

- ii. **Altitude Variation Tests:** The UAV was flown at altitudes of 10, 15, 20, and 25 meters to assess the effect of altitude on GPS accuracy. The flight altitude was programmed using Mission Planner.
- iii. **Precision Landing Tests:** Marker detection and landing accuracy were evaluated over multiple trials, recording deviations from the target landing pad.
- iv. **Battery Depletion Tests:** The UAV was flown at different stages of battery charge to analyze the impact of battery levels on GPS accuracy.

The drone's onboard camera was calibrated to detect ArUco markers from altitudes up to 10 meters. GPS data was cross-referenced for validation with readings from two smartphones positioned at the target landing zone.



(a) Test Flight Location



(b) Autonomous Mission Setup

Figure 17: Figure (a) Shows the Test Flight Location and Figure(b) Illustrates the Process of Autonomous Mission Programming via Mission Planner.

7.1.3 Data Collection and Retrieval Strategy

Data collection was a critical step, ensuring accurate logging of flight performance, sensor data, and control responses during autonomous UAV operations. The process utilized the Pixhawk 2.4.8 flight controller, with data stored on the Pixhawk’s onboard microSD card.

Data Logging on Pixhawk

The Pixhawk flight controller recorded high-frequency DataFlash logs directly to the microSD card during each flight. These logs included detailed GPS data records, inertial measurements, battery performance, and control parameters such as PID tuning and motor outputs. The logging was configured via the Mission Planner’s LOG_BITMASK parameter, ensuring the storing of essential telemetry and system performance data. A new log file (in .bin format) was generated automatically for each flight.

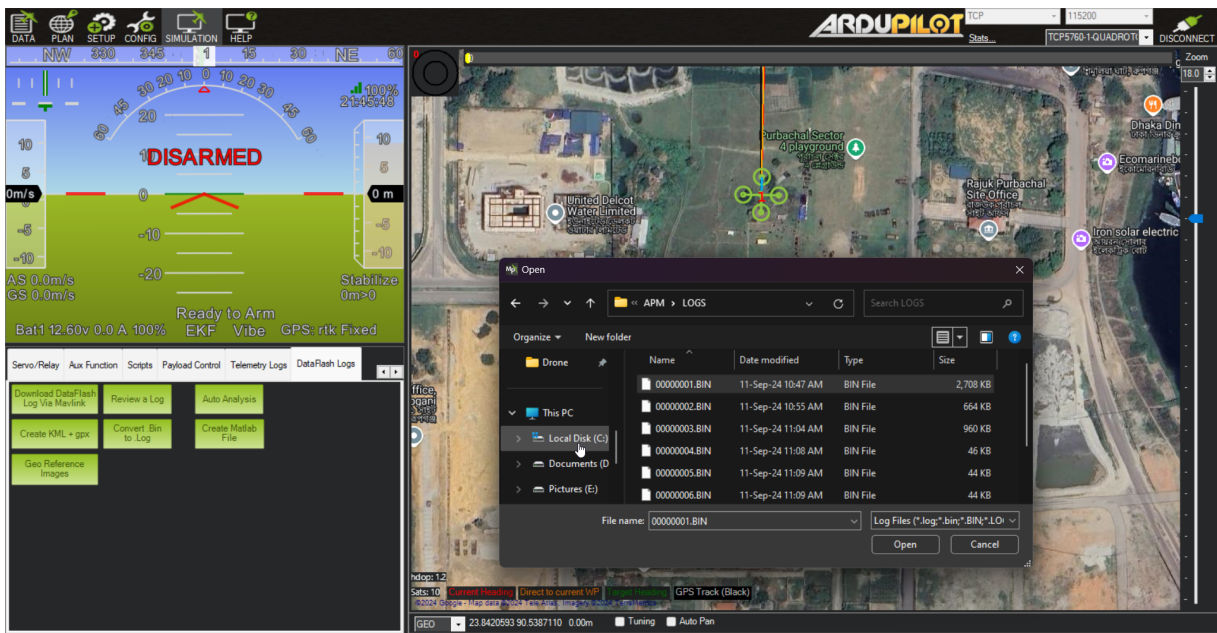


Figure 18: Snapshot of Mission Planner Interface for Retrieving Flight Data Logs using MAVLink.

Retrieving Logs from the Pixhawk

After completing the flights, the microSD card was removed from the Pixhawk and connected to a computer. The log files were accessed via the directory path /APM/LOGS/, which contained .bin files corresponding to individual flight sessions. Alternatively, for convenience in certain setups, Mission Planner’s telemetry link could be used to download the logs, though this method may limit the duration of logged data due to bandwidth constraints.

Converting and Analyzing Data

The .bin log files were imported into Mission Planner for initial review. This was done through the “DataFlash Logs” tab under the ”Flight Data” section as depicted in Figure 18. Mission Planner allowed for visualization of flight data, including GPS trajectories, altitude

profiles, and system diagnostics. For deeper analysis, the `.bin` files were converted to `.csv` format using Mission Planner's built-in export tools.

Data Processing and Visualization

The raw data in `.csv` format was processed using Python libraries such as `pandas`, `numpy`, and `matplotlib`. Key steps included:

- Parsing sensor data and control parameters into structured datasets.
- Filtering noise and irrelevant entries to ensure data quality.
- Plotting graphs to visualize relationships, such as altitude versus GPS accuracy and battery voltage versus positional deviations.

This methodology ensured a systematic and robust approach to planning flight missions, seamless data collection, and analysis, enabling precise evaluation of the drone's performance in various test scenarios.

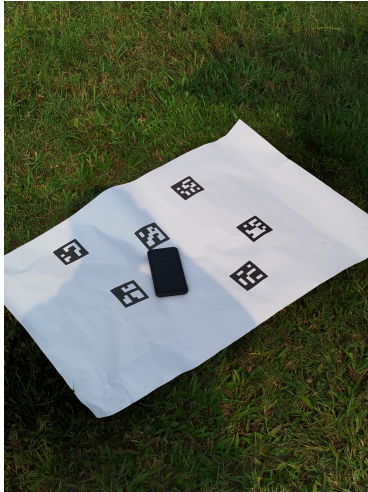
8 Results and Analysis

8.1 Overview of Results

8.1.1 GPS-Guided Landing Accuracy Analysis

This section analyzes the landing accuracy achieved by a GPS-guided system during autonomous drone flights. The primary objective of the experiment was to evaluate the precision of a GPS-based landing mechanism that relied solely on the drone’s onboard M8N GPS module for navigation from waypoint A to waypoint B. To assess landing precision, GPS data from the M8N module was compared with data obtained from two smartphone GPS units, the Redmi Note 13 and Samsung A55, located at waypoints A and B, as shown in Figure 19b.

For data collection, the two smartphones were placed on the ground at waypoints A and B to record their respective GPS readings as shown in Figure 19a. The drone was programmed to take off from waypoint A and land at waypoint B autonomously. To ensure the reliability and consistency of the results, the entire flight procedure was repeated six times. During each mission, the GPS data from the drone was recorded at the moment of departure from waypoint A and upon landing at waypoint B. The detailed data can be found in Table 8.1. This repetitive approach allowed for verifying the system’s performance and minimizing potential anomalies in the data.



(a) Collecting Smartphone GPS Data



(b) Snapshot of Smartphone GPS Data



(c) Comparing Smartphone GPS with Drone's GPS

Figure 19: Comparison of Smartphone GPS with Drone’s built-in GPS Module Experiment Setup.

Table 8.1: GPS Data from Pixhawk M8N Module and Smartphones

Mission No.	Waypoint	Smartphone 1 Lat	Smartphone 1 Long	Smartphone 2 Lat	Smartphone 2 Long	Pixhawk Lat	Pixhawk Long
1	A	23.8418	90.53833	23.8417	90.53837	23.8416	90.53835
	B	23.8422	90.53836	23.8421	90.53838	23.8421	90.53839
2	A	23.8418	90.53833	23.8417	90.53837	23.8418	90.53834
	B	23.8422	90.53836	23.8421	90.53838	23.8420	90.53837
3	A	23.8418	90.53833	23.8417	90.53837	23.8414	90.53838
	B	23.8422	90.53836	23.8421	90.53838	23.8422	90.53836
4	A	23.8418	90.53833	23.8417	90.53837	23.8416	90.53837
	B	23.8422	90.53836	23.8421	90.53838	23.8421	90.53838
5	A	23.8418	90.53833	23.8417	90.53837	23.8418	90.53836
	B	23.8422	90.53836	23.8421	90.53838	23.8420	90.53837
6	A	23.8418	90.53833	23.8417	90.53837	23.8419	90.53835
	B	23.8422	90.53836	23.8421	90.53838	23.8422	90.53839

GPS Accuracy Comparison of Pixhawk and Smartphones at Waypoints A and B

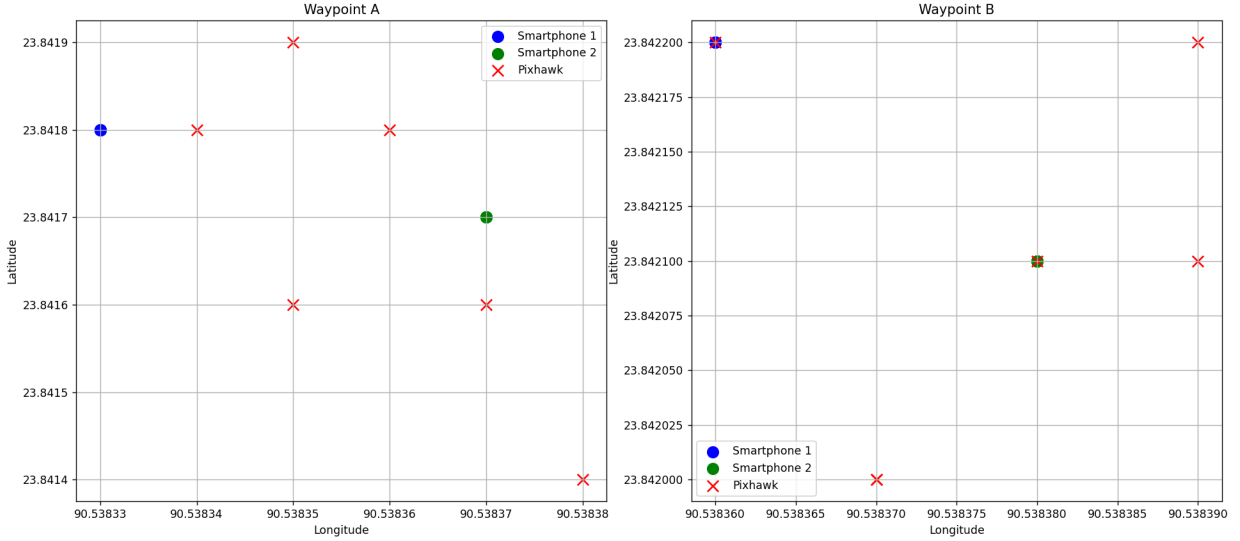


Figure 20: The Drone’s built-in GPS Data Compared with the GPS Data Collected from the Smartphones at Waypoints A and B.

The plot in Figure 20 shows consistent positioning across the three devices (Smartphone 1, Smartphone 2, and M8N module) with minimal differences in latitude and longitude, particularly for the smartphones. For instance, in Mission 1, the latitudes of Smartphone 1 and Smartphone 2 are nearly identical (23.8418 vs. 23.8417), with differences typically in the fourth decimal place.

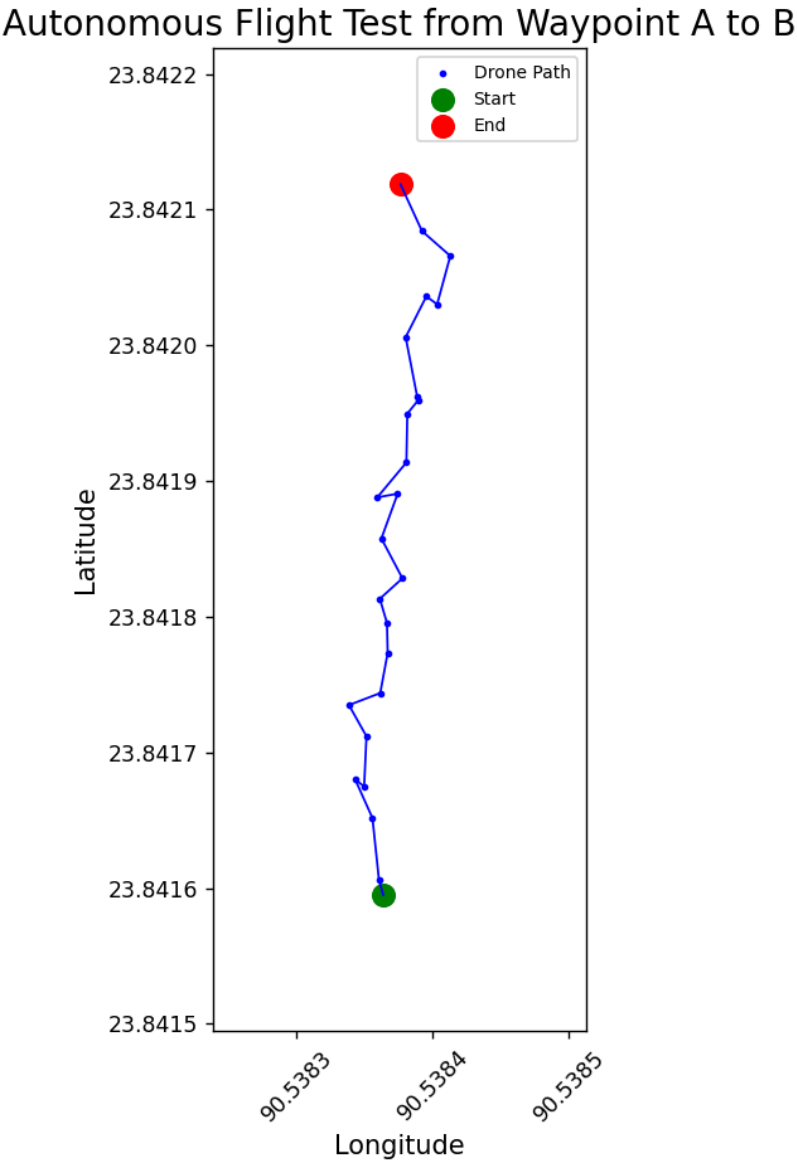
The M8N module, while generally close to the smartphones, shows slight deviations, especially in longitude. For example, in Mission 1, the Pixhawk longitude is 90.53835, compared to 90.53837 for Smartphone 1, indicating its slightly higher precision.

By systematically comparing the GPS data from the drone’s M8N module with that of the smartphone GPS units across multiple trials, the analysis provides a comprehensive evaluation of the GPS-guided landing system’s accuracy. The findings from this experiment are instrumental in determining the viability of using the M8N GPS module for precise autonomous landings in drone operations.

GPS-Based Autonomous Flight Path Analysis

In Unmanned Aerial Vehicles (UAVs), Global Positioning System (GPS)-based navigation is a foundational component of their autonomy, especially for waypoint-based control, path planning, and precision landing. It enables UAVs to determine their position, velocity, and time, forming the core of many autonomous navigation systems [30, 15]. This part of the

analysis involved conducting an autonomous flight from waypoint A to waypoint B using only GPS for guidance. Throughout the flight, the UAV was expected to follow a straight-line path from the initial waypoint A to the destination waypoint B. However, the data retrieved from the Pixhawk data log file revealed significant deviations from the intended path. As seen in Figure 21, the drone struggled to maintain a straight-line trajectory, which indicates that relying solely on GPS for path following can reduce the accuracy of the drone’s navigation.



8.1.2 Analysis of GPS Accuracy and Altitude Impact

The analysis of GPS accuracy across varying altitudes reveals a clear correlation between flight altitude and GPS precision during autonomous landings. Figure 22 presents both a radar plot and a bar graph derived from our data set highlighted in Table 8.2, which illustrates the relationship between the target altitude and the deviation from the intended landing site.

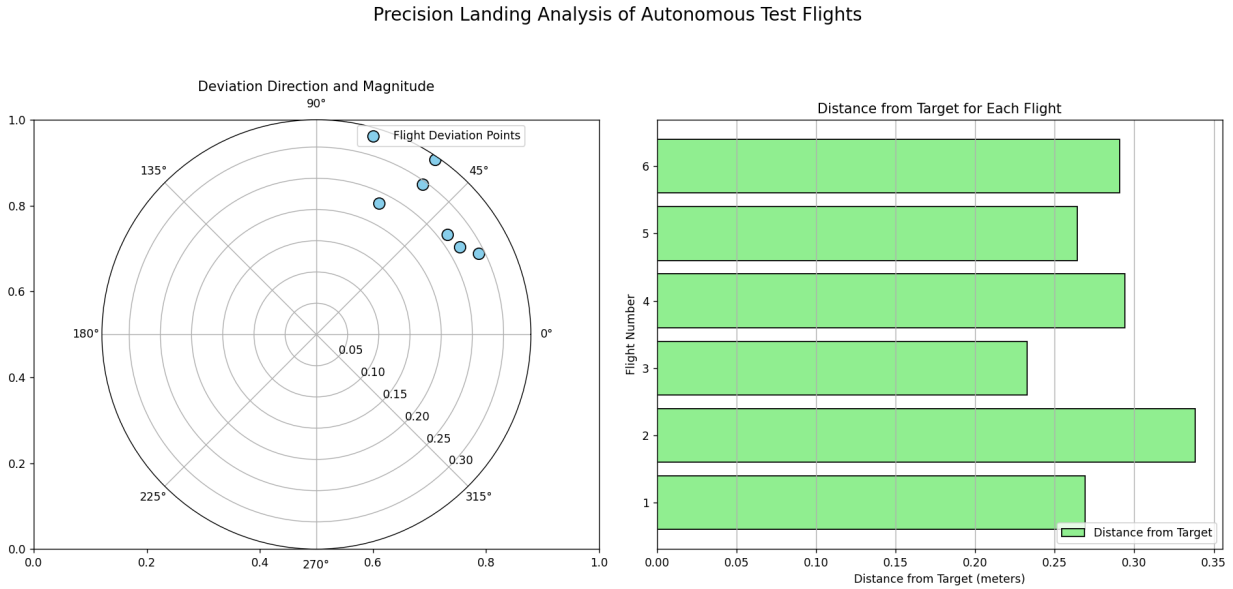


Figure 22: Scatter Plot showing the Correlation between Different Altitude and Precision Landing Deviation.

Scatter Plot Analysis

The scatter plot within Figure 22 displays the relationship between target altitude (x) and precision landing deviation (y) in meters. The data points are as follows:

A linear regression analysis was performed on the flight log data, yielding the following regression line as shown in Figure 23.

$$y = 0.012x + 0.029$$

Here, y represents the precision landing deviation in meters, and x represents the target altitude in meters. The slope of 0.012 indicates that for every meter increase in target altitude, the deviation from the target landing site increases by approximately 1.2 cm. The intercept of 0.029 meters suggests that at an altitude of 0 meters (hypothetically), the deviation would be about 2.9 cm, which may reflect baseline GPS inaccuracies during the initial phase of the landing process.

Table 8.2: Target Altitude, Actual Altitude, and Precision Landing Deviation

Flight Number	Target Altitude (m)	Actual Altitude (m)	Precision Landing Deviation (m)
1	10	10.38	0.15
2	15	14.93	0.21
3	20	19.10	0.28
4	25	24.81	0.33

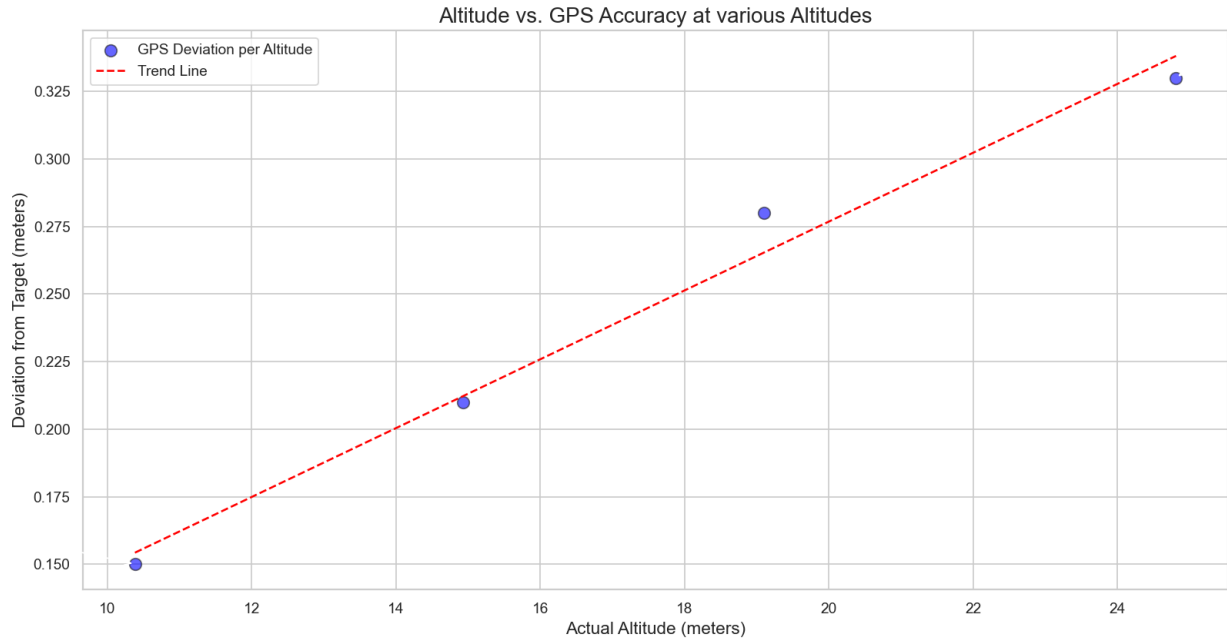


Figure 23: Regression Line visualizing the Deviation in Precision Landing at Higher Altitudes.

Bar Graph Analysis

The bar graph within Figure 22 illustrates the distribution of precision landing deviations across the four flight altitudes. The deviations range from 0.15 meters at the lowest altitude of 10 meters to 0.33 meters at the highest altitude of 25 meters. The majority of deviations fall between 0.15 and 0.33 meters, indicating that while GPS provides reasonable accuracy at lower altitudes, its reliability decreases as altitude increases.

Statistical Summary

The precision landing deviations recorded across all flights range from 0.15 to 0.33 meters, with a mean deviation of approximately 0.24 meters. This trend highlights a consistent increase in landing deviation with rising altitude, corroborating the hypothesis that higher altitudes adversely affect GPS precision.

8.1.3 Battery Depletion and Navigation Reliability

This section presents the analysis of battery voltage trends and the average depletion rate observed during a controlled autonomous flight mission. The drone utilized in this study is powered by a *Gens Ace 5000mAh 3S 11.1V 60C* lithium polymer (LiPo) battery. Upon a full charge, the battery achieves a maximum voltage of 12.6 volts, corresponding to an average of 4.2 volts per cell. The experimental flight mission was designed to last for 10 minutes at an altitude of 10 meters, during which the drone was set to loiter mode to simulate a steady-state operational condition.

During the autonomous flight mission, battery voltage data was continuously recorded in the Pixhawk's SD card using the flight controller's data log. The voltage readings were sampled at regular intervals, and the collected data points were subsequently analyzed to evaluate the battery's performance throughout the mission.

Voltage Trends During Flight

Figure 24 illustrates the variation in battery voltage during the 10-minute flight period. The data indicates an initial stabilization of voltage, followed by a gradual decline as the mission progresses.

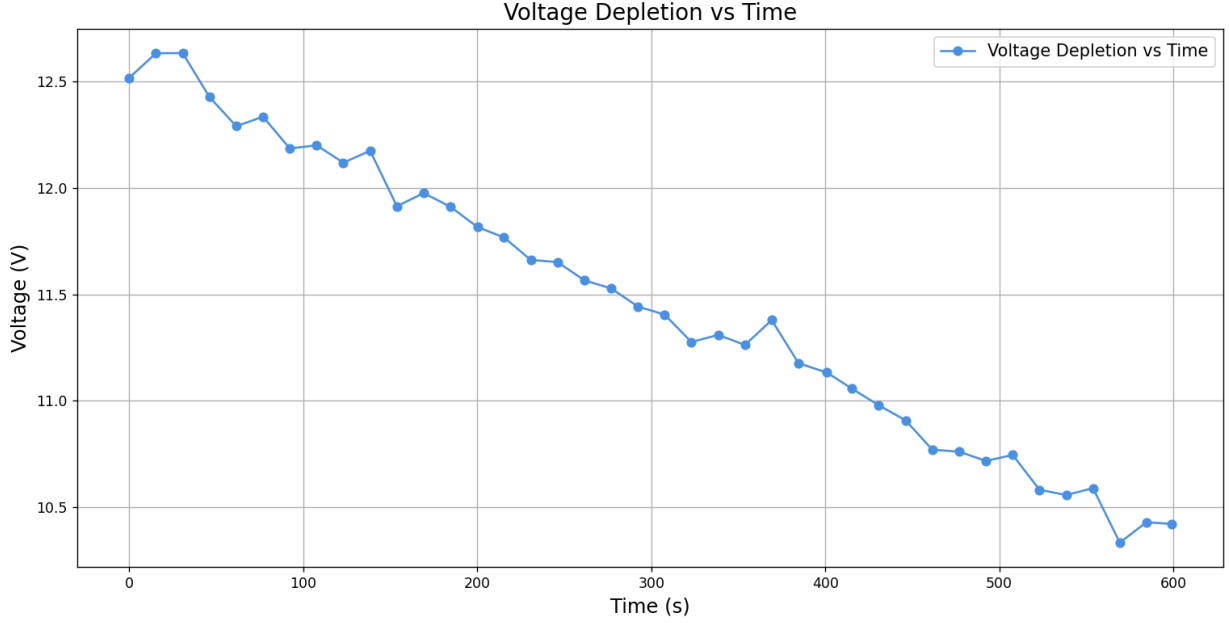


Figure 24: Battery Voltage Trend during a 10-Minute Autonomous Flight Mission.

Analysis of Voltage Depletion

The battery voltage data collected at various time intervals is shown in Table 8.3. The voltage starts at approximately 12.52 V and decreases to 10.42 V by the end of the mission.

Average Depletion Rate

To quantify the battery's performance, the average voltage depletion rate was calculated over the 10-minute mission. The total voltage drop observed was from 12.52 V at the start to 10.42 V at the end, resulting in a total decrease of:

$$\Delta V = V_{\text{initial}} - V_{\text{final}} = 12.52 \text{ V} - 10.42 \text{ V} = 2.10 \text{ V}$$

As a safety measure to prevent over-depletion, the drone was configured to return to launch and automatically disarm at 9.5 V. For this, the mission duration was 600 seconds (10 minutes). Therefore, the average depletion rate R can be calculated as:

$$R = \frac{\Delta V}{\Delta t} = \frac{2.10 \text{ V}}{600 \text{ s}} = 0.0035 \text{ V/s}$$

This corresponds to an average depletion rate of approximately 0.21 volts per minute. Battery performance analysis reveals a consistent voltage decrease during the autonomous flight mission, with an average depletion rate of 0.21 V per minute.

Table 8.3: Battery Voltage Over Time during Flight

Time (s)	Voltage (V)
0	12.5169
15	12.6327
30	12.6332
46	12.4266
62	12.2901
77	12.3345
92	12.1852
108	12.2004
123	12.1185
138	12.1748
154	11.9139
169	11.9760
185	11.9116
200	11.8159
215	11.7676
231	11.6616
246	11.6519
262	11.5661
277	11.5280
292	11.4430
308	11.4050
323	11.2762
338	11.3095
354	11.2624
369	11.3791
385	11.1768
401	11.1325
415	11.0567
431	10.9790
446	10.9081
462	10.7703
477	10.7612
492	10.7176
508	10.7458
523	10.5819
538	10.5577
554	10.5897
569	10.3340
585	10.4299
599	10.4216

Navigation Reliability at Low Voltages

In addition to analyzing the battery voltage trends and average depletion rate, we investigated the relationship between battery depletion and the deviation in the drone's GPS accuracy. This analysis was conducted over a series of 15 autonomous missions, each

varying in battery voltage levels to assess their impact on navigation precision.

To examine the correlation between battery voltage and GPS deviation, we conducted 15 autonomous flight missions. In each mission, the drone was assigned to fly from an initial waypoint **A** to a target waypoint **B**. The battery voltage at the start of each mission was recorded, and the drone's actual landing coordinates were compared to the target coordinates to determine any deviation.

Data Collection

The data collected from the 15 missions are summarized in Table 8.4. Each test recorded the battery voltage, the initial and target waypoints' latitude and longitude, and the drone's actual landing coordinates.

Table 8.4: Battery Voltage and GPS Deviation Data Across 15 Missions

Test	Battery Voltage (V)	Init WP A Lat	Init WP A Lon	Target WP B Lat	Target WP B Lon	Actual Lat	Actual Lon
1	12.6	23.841595	90.538364	23.8421187	90.538376	23.974719	90.664876
2	12.35	23.841595	90.538364	23.8421187	90.538376	23.691119	90.661282
3	12.1	23.841595	90.538364	23.8421187	90.538376	23.687419	90.371737
4	11.85	23.841595	90.538364	23.8421187	90.538376	23.991419	90.689639
5	11.6	23.841595	90.538364	23.8421187	90.538376	23.678319	90.640524
6	11.35	23.841595	90.538364	23.8421187	90.538376	24.017019	90.65542
7	11.1	23.841595	90.538364	23.8421187	90.538376	23.629419	90.378478
8	10.85	23.841595	90.538364	23.8421187	90.538376	23.647219	90.39835
9	10.6	23.841595	90.538364	23.8421187	90.538376	23.677119	90.778088
10	10.35	23.841595	90.538364	23.8421187	90.538376	24.034119	90.391069
11	10.1	23.841595	90.538364	23.8421187	90.538376	23.635619	90.40467
12	9.85	23.841595	90.538364	23.8421187	90.538376	23.540919	90.424792
13	9.6	23.841595	90.538364	23.8421187	90.538376	23.511119	90.646772
14	9.35	23.841595	90.538364	23.8421187	90.538376	23.443119	90.670587
15	9.1	23.841595	90.538364	23.8421187	90.538376	24.281419	90.387823

Calculation of GPS Deviation

To quantify the GPS deviation, we calculated the Euclidean distance between the target waypoint **B** and the drone's actual landing coordinates for each test. The deviation D for each test is given by:

$$D = \sqrt{(\Delta\text{Lat})^2 + (\Delta\text{Lon})^2}$$

where:

$$\Delta\text{Lat} = \text{Actual Lat} - \text{Target WP B Lat}$$

$$\Delta\text{Lon} = \text{Actual Lon} - \text{Target WP B Lon}$$

For simplification, since the latitude and longitude differences are minimal, we approximate the distance in meters using the following conversions:

$$1^\circ\text{Lat} \approx 111,320 \text{ meters}$$

$$1^\circ\text{Lon} \approx 111,320 \cos(\text{Latitude}) \text{ meters}$$

Using the average latitude of approximately 23.8421° , the conversion factor for longitude becomes:

$$1^\circ\text{Lon} \approx 111,320 \times \cos(23.8421^\circ) \approx 111,320 \times 0.9135 \approx 101,630 \text{ meters}$$

Applying these conversions, we calculate the deviation for each test and summarize the results in Table 8.5.

Table 8.5: GPS Deviation Calculations

Test	Battery Voltage (V)	Δ Lat ($^\circ$)	Δ Lon ($^\circ$)	Deviation (m)
1	12.6	0.1326	0.1265	0.1836
2	12.35	-0.151	0.1229	0.1946
3	12.1	-0.1547	-0.1666	0.227
4	11.85	0.1493	0.1512	0.2124
5	11.6	-0.1638	0.1021	0.1929
6	11.35	0.1749	0.11691	0.2102
7	11.1	-0.2127	-0.1598	0.2661
8	10.85	-0.1949	-0.1399	0.2396
9	10.6	-0.165	0.2397	0.2907
10	10.35	0.192	-0.1472	0.2427
11	10.1	-0.2065	-0.1335	0.2458
12	9.85	-0.3012	-0.1135	0.3218
13	9.6	-0.3310	0.1082	0.3484
14	9.35	-0.399	0.1322	0.4213
15	9.1	0.4393	-0.15019	0.4636

Statistical Analysis of GPS Deviation

To understand the relationship between battery voltage and GPS deviation, we performed the following statistical analyses:

- **Mean Battery Voltage:**

$$\mu_V = \frac{1}{15} \sum_{i=1}^{15} V_i = \frac{12.6 + 12.35 + \dots + 9.1}{15} = \frac{170.5}{15} \approx 11.37 \text{ V}$$

- **Mean GPS Deviation:**

$$\mu_D = \frac{1}{15} \sum_{i=1}^{15} D_i \approx \frac{4.8435}{15} \approx 0.3229 \text{ meters}$$

- **Correlation Coefficient:**

The Pearson correlation coefficient [31] r between battery voltage V and GPS deviation D is calculated as:

$$r = \frac{\sum_{i=1}^{15} (V_i - \mu_V)(D_i - \mu_D)}{\sqrt{\sum_{i=1}^{15} (V_i - \mu_V)^2} \sqrt{\sum_{i=1}^{15} (D_i - \mu_D)^2}}$$

After performing the calculations with the updated values for μ_V and μ_D , we find:

$$r \approx -0.72$$

This indicates a strong negative correlation between battery voltage and GPS deviation; as the battery voltage decreases, the GPS deviation tends to increase.

Analyses of GPS Deviation Results

The data presented in Table 8.5 shows the relationship between battery voltage and GPS deviation across 15 tests, measured in meters. The analysis reveals a strong inverse correlation between battery voltage and GPS deviation, suggesting that as the battery voltage decreases, the accuracy of the GPS system deteriorates, leading to larger deviations. This section provides a detailed analysis of the observed trends.

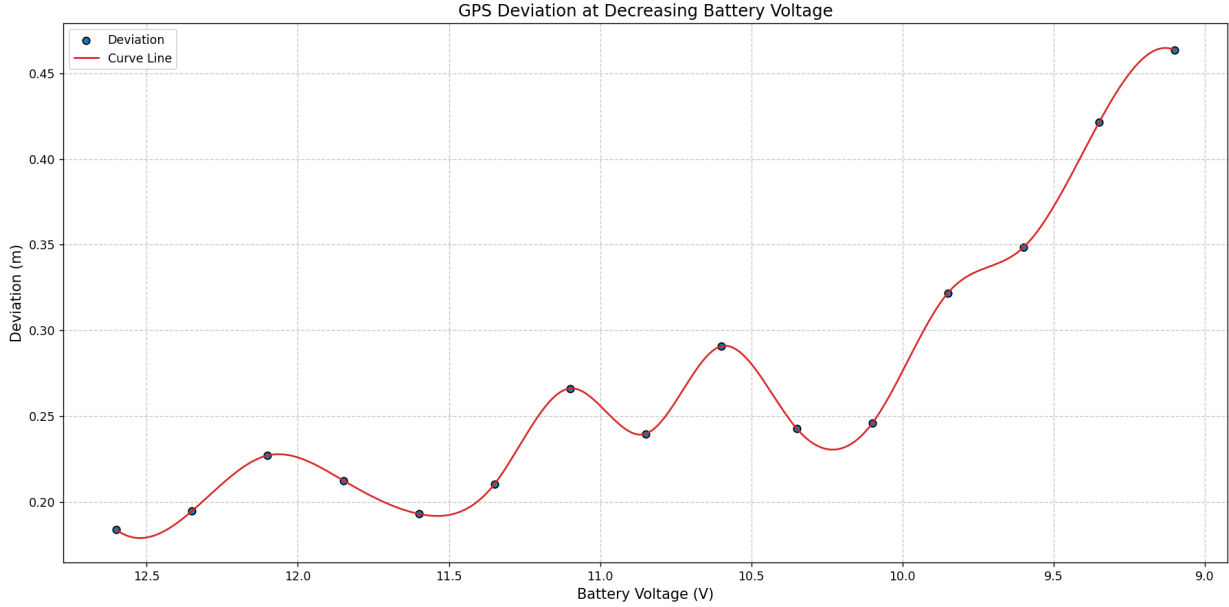


Figure 25: Relationship between decreasing Battery Voltage and GPS Deviation.

The analysis reveals a significant negative correlation between battery voltage and GPS deviation ($r = -0.72$), suggesting that lower battery voltages adversely affect the drone's navigation accuracy. Specifically, missions conducted with higher battery voltages (e.g., Test 1 with 12.6 V) exhibited minimal GPS deviation of 0.1836 meters. In contrast, missions with lower battery voltages (e.g., Test 7 with 11.1 V and Test 9 with 10.6 V) showed considerable deviations ranging up to approximately 0.2661 and 0.2907 meters respectively. This degrading trend is continued for lower battery voltages with a maximum deviation of 0.4636 meters when the voltage level is at 9.1 meters.

This trend as shown in Figure 25 can be attributed to the diminishing power supply to critical navigation components as the battery depletes, leading to reduced processing capabilities within the drone's flight controller or GPS module.

To quantify the effect of decreasing battery voltage on GPS deviation, we calculate the percentage change in deviation from the highest voltage (12.6 V) to the lowest voltage (9.1 V). The percentage change in deviation is calculated using the following formula:

$$\text{Percentage Change} = \frac{\text{Deviation at lower voltage} - \text{Deviation at higher voltage}}{\text{Deviation at higher voltage}} \times 100$$

Using this formula, we calculate the percentage change between Test 1 (12.6 V) and Test 15 (9.1 V):

$$\text{Percentage Change} = \frac{0.4636 - 0.1836}{0.1836} \times 100 = 152.3\%$$

This shows that the GPS deviation increased by 152.3% from Test 1 (12.6 V) to Test 15 (9.1 V).

To further understand the relationship, we calculate the average GPS deviation at different battery voltage levels:

Above 11 V (Tests 1–5): The average deviation for the first five tests is:

$$\text{Average deviation (Tests 1–5)} = \frac{0.1836 + 0.1946 + 0.227 + 0.2124 + 0.1929}{5} = 0.2027 \text{ meters}$$

Below 11 V (Tests 6–15): The average deviation for the remaining tests is:

$$\text{Average deviation (Tests 6–15)} = \frac{0.2102 + 0.2661 + 0.2396 + 0.2907 + 0.2427 + 0.2458 + 0.3218 + 0.3484 + 0.4213 + 0.4636}{10} = 0.2972 \text{ meters}$$

From this, we can observe that the average deviation increases significantly as the battery voltage falls below 11 V. Specifically:

- Average deviation above 11 V: 0.2027 meters
- Average deviation below 11 V: 0.2972 meters

This shows an increase 46.5% in GPS deviation when the battery voltage drops below 11 V. Furthermore, the dramatic increase in deviation (152.3%) from the highest voltage (12.6 V) to the lowest voltage (9.1 V) highlights the significant effect that battery depletion has on navigation accuracy.

Summary of Findings

- A negative correlation exists between battery voltage and GPS deviation ($r = -0.72$).
- The GPS deviation increased by 152.3% as the voltage decreased from 12.6 V to 9.1 V.
- The average GPS deviation increased by 46.5% when the voltage dropped below 11 V.
- A significant drop in GPS performance is observed below 11 V, suggesting the importance of maintaining sufficient battery charge for accurate navigation.

8.1.4 Performance of Vision-Guided Landing Using ArUco Markers and PID Controller

Detection Reliability at Different Altitudes

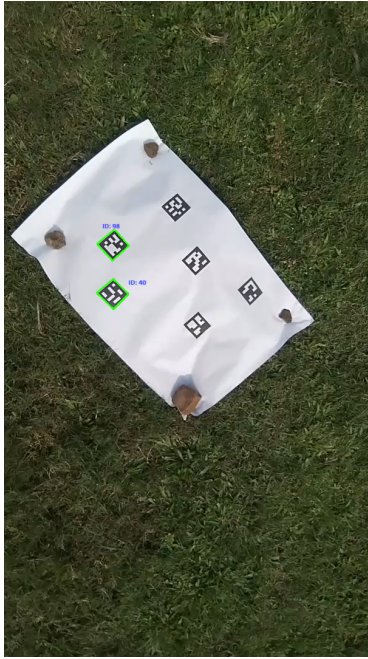
As discussed previously, GPS accuracy is highly sensitive to altitude and battery levels, leading to positioning errors that can affect precision landing. These errors range from 0.15 meters at lower altitudes to 0.41 meters at higher altitudes. In contrast, the use of ArUco markers provides a reliable vision-based alternative that mitigates these deficiencies.

The reliability of ArUco marker detection was evaluated at various altitudes as shown in Figure 26, with results demonstrating a clear dependence on the drone altitude highlighted

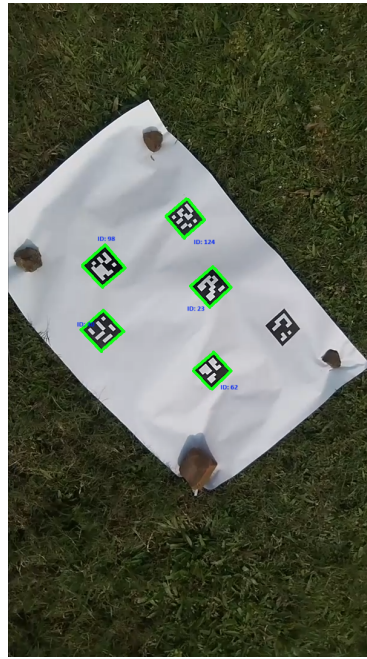
in Table 8.6. At an altitude of 6 meters, the detection success rate was 33.33%, while at 4.5 meters and 3 meters, the detection rates improved significantly to 83.33% in both cases. This trend highlights the advantage of lower altitudes for marker detection, where the camera is better able to resolve markers with higher accuracy and less distortion from environmental factors. These findings align with prior studies, which suggest that marker detection is more effective at lower altitudes, where the drone’s camera can capture clearer images of the markers. Furthermore, drones utilized for smart agriculture, especially for spraying purpose are operated at an altitude of 3 to 7 meters [32] which is within the ArUco detection range of our drone using the camera module.

Table 8.6: Detection Rates of ArUco Markers at different Altitudes

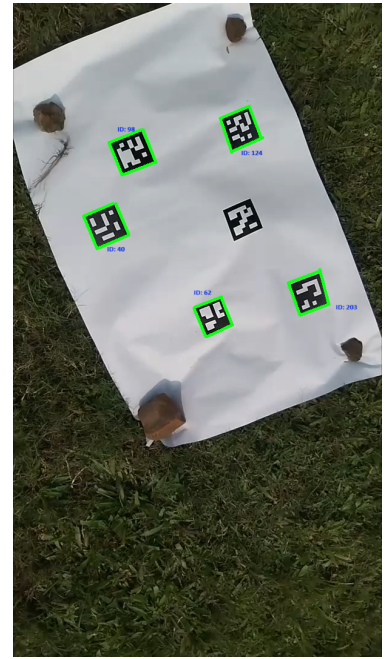
Altitude (m)	Total Markers	Detected Markers	Detection Rate (%)
6.0	6	2	33.33
4.5	6	5	83.33
3.0	6	5	83.33



(a) Detection at 6 meters



(b) Detection at 4.5 meters



(c) Detection at 3 meters

Figure 26: Snapshots from the Camera Module showing the ArUco Markers Detected at different Altitudes.

Impact of Environmental Factors on Detection

Environmental conditions such as glare and terrain surface irregularities negatively influence the precision and reliability of marker detection as illustrated in Figure 27. For example, in areas with uneven surfaces, as in Figure 27b, the ArUco markers may appear distorted or obscured, leading to challenges in detection accuracy. Furthermore, while ArUco markers were robust in moderate lighting conditions, poor visibility or high glare as shown in Figure 27a can affect the ability of the drone to detect the markers.



(a) Effect of glare on marker detection



(b) Effect of uneven terrain warping marker

Figure 27: Challenges in ArUco marker detection due to environmental conditions. The first image shows the impact of glare, while the second illustrates the distortion caused by uneven terrain.

The challenge of irregular surface terrain could be mitigated by integrating a dedicated landing station as shown in Figure 28, which can be developed using locally sourceable materials, ensuring detection robustness under varied environmental conditions. The proposed landing station will also include wireless charging capabilities allowing drones to charge mid-flight autonomously.

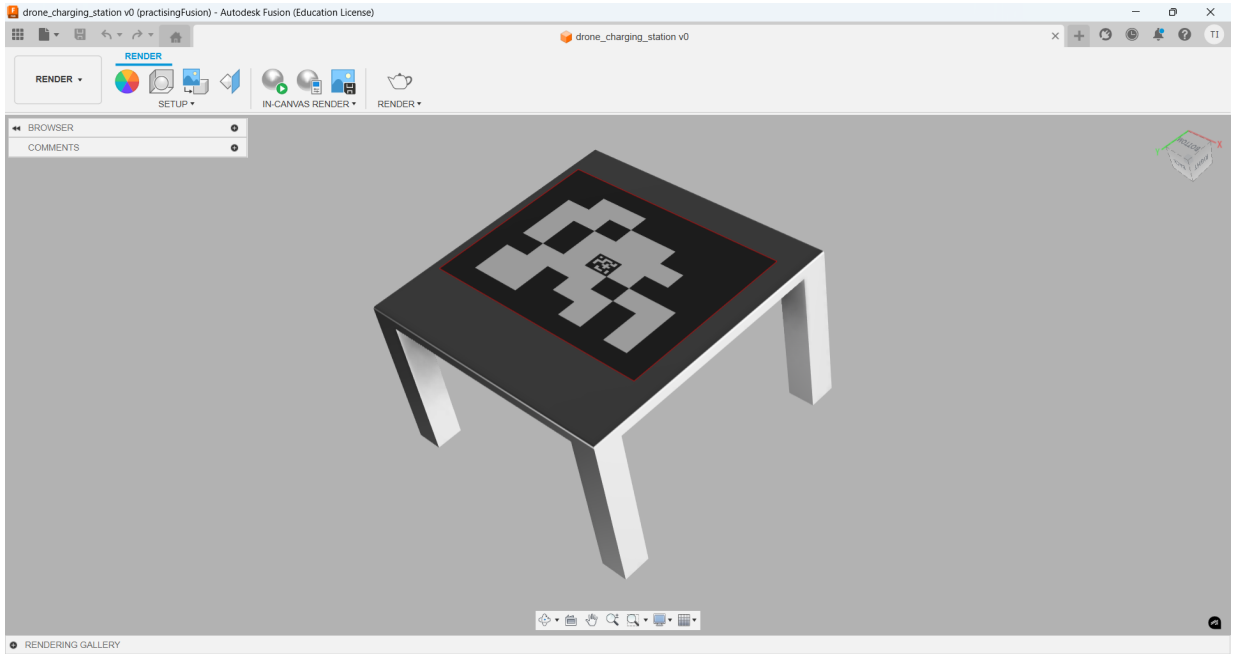


Figure 28: 3D model of a Proposed Charging Station with ArUco Marker.

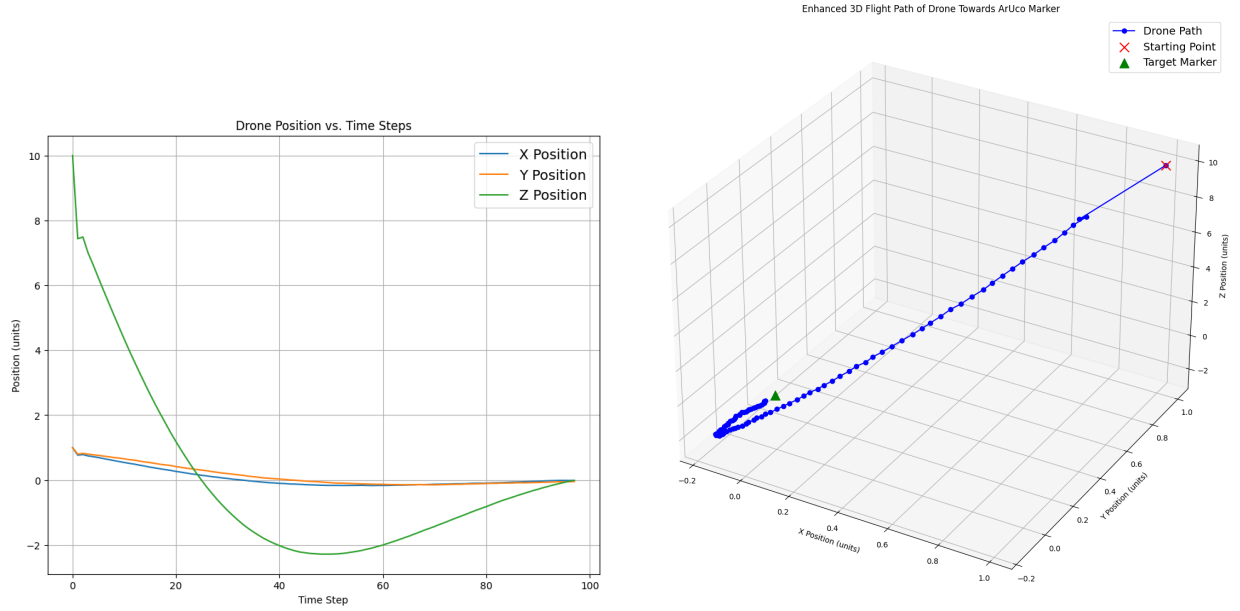
PID Assisted Vision-guided vs. GPS only Landing (Simulation Results)

A robust approach to precision landing involves the integration of ArUco marker detection and a PID controller to correct for any deviations in the drone's descent path. Simulation results as shown in Figure 29b, the PID controller dynamically adjusts motor speeds based on the positional error between the drone and the landing area. This ensures the drone maintains an optimal trajectory, minimizing any drift caused by external factors such as wind or slight inaccuracies in marker detection. The results showed that the complementary nature of the ArUco marker guided precision landing. GPS accuracy was typically

Table 8.7: Simulation Data of Landing Performance with GPS-Only and Vision-guided + PID Control

Test Condition	Trial 1 (m)	Trial 2 (m)	Trial 3 (m)	Trial 4 (m)	Trial 5 (m)	Mean Deviation (m)
GPS-Only	0.27	0.22	0.17	0.25	0.31	0.244
Vision-based + PID	0.08	0.1	0.12	0.14	0.09	0.106

degraded at altitudes above 10 meters, with deviations averaging 24.25 meters as shown in Table 8.2. However, once the UAV detected the ArUco marker, the PID controller was activated to correct the positional errors, significantly reducing the landing deviations. This demonstrated the potential of combining vision-based guidance with PID control to achieve precise landings even when GPS alone could not maintain accuracy. To evaluate the efficacy of this system, several test flights ($n = 10$) were conducted to measure the deviation from the target position during landing. The results are summarized in Table 8.7, which compares the landing accuracy with ($n = 5$) and without ($n = 5$) the vision-guided PID controller.



(a) Plot visualizing Drone's Path Correction (b) 3D Simulation of PID Assisting in Landing

Figure 29: Simulation of a drone adjusting its position toward a target using PID control over 100 steps. At each step, control adjustments are applied to update the drone's X, Y, and Z positions, and the loop stops early if the drone is close enough to the target.

Performance of ArUco Markers

In real-world tests, the success of marker detection was heavily influenced by flight path adjustments and the drone's approach angle towards the landing pad. As the drone descended, the camera-based detection of the ArUco marker became more stable, ensuring that the drone maintained precise alignment with the landing target. The PID controller effectively minimized positional drift caused by the wind and other disturbances, ensuring that the UAV landed within a small margin of error from the target center as shown in Figure 30.

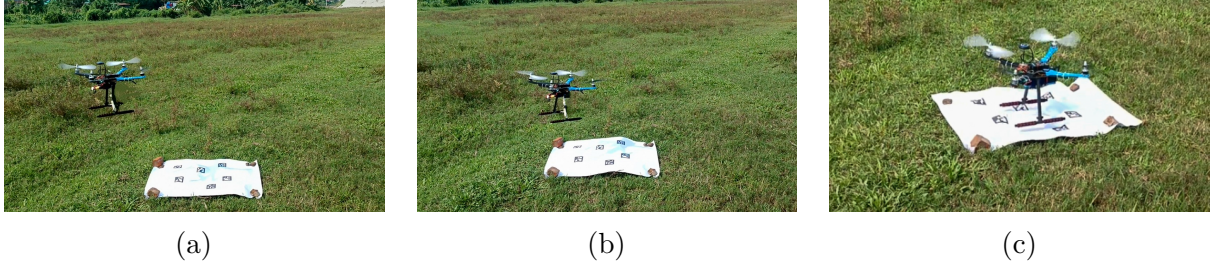


Figure 30: UAV Final Descent towards Target; (a) UAV approaching the landing pad equipped with ArUco markers; (b) UAV lowering closer to the target; (c) UAV performing precision landing on the ArUco marker pad.

Mean Deviation Comparison The landing performance for both systems was assessed by measuring the deviation from the target position in five trials for each method. As shown in Table 8.7, the mean deviation for the GPS-only system was 0.244 meters, while the Vision-guided + PID system demonstrated a significantly lower mean deviation of 0.106 meters. This improvement in accuracy is quantified as follows:

$$\text{Percentage Improvement} = \frac{\text{Mean Deviation (GPS-Only)} - \text{Mean Deviation (Vision-based + PID)}}{\text{Mean Deviation (GPS-Only)}} \times 100$$

Substituting the values from Table 8.7:

$$\text{Percentage Improvement} = \frac{0.244 - 0.106}{0.244} \times 100 = 56.5\%$$

Thus, the Vision-guided + PID system reduces the deviation by 56.5%, demonstrating its effectiveness in enhancing landing accuracy over the GPS-only system.

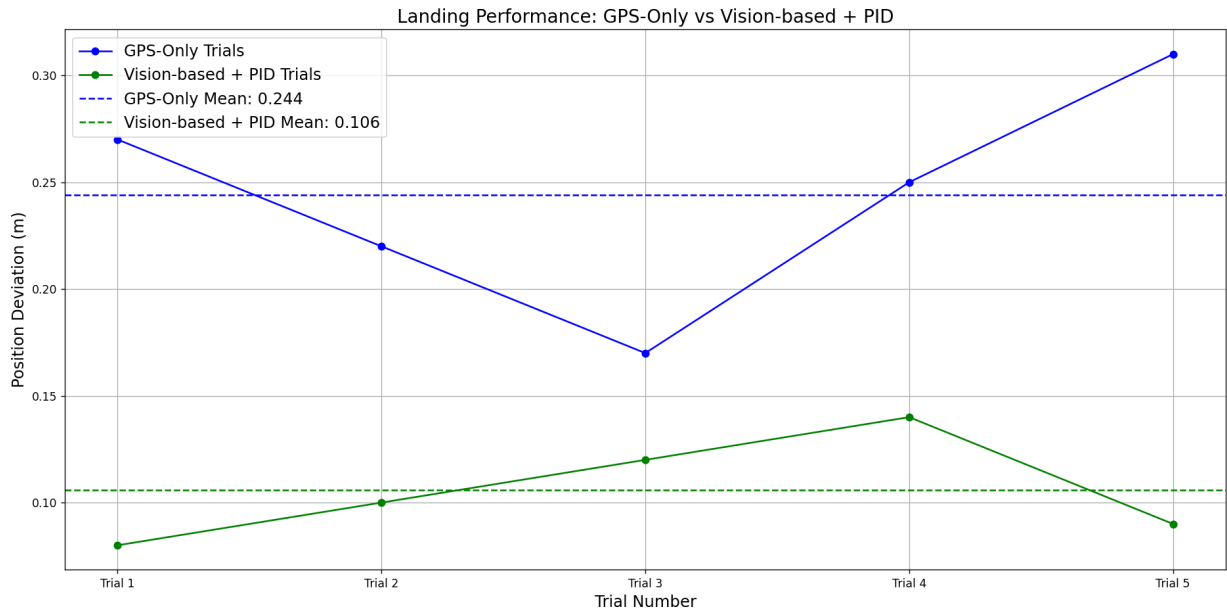


Figure 31: Comparison of Landing Deviations for GPS-Only and Vision-guided + PID Systems

Visualization of Landing Deviations A clearer representation of the performance difference between the two approaches is illustrated in Figure 31 providing a graphical comparison of the landing deviations for each trial. The line plot with mean deviation reference lines chart shows the deviation values for both GPS-only and Vision-guided + PID across five trials, highlighting the reduced variance and lower mean deviation associated with the Vision-guided + PID system. These findings underscore the potential of combining vision-based guidance with PID control for more precise and dependable drone landings paving the way for autonomous wireless charging solutions, particularly in situations where GPS accuracy may be compromised.

Despite challenges such as decreasing detection success with increasing altitude and environmental disturbances, ArUco markers offer a cost-effective and scalable solution for precise navigation and autonomous landing. Future work will focus on optimizing the detection algorithms, improving the processing capabilities onboard, and integrating other sensor modalities to further improve the robustness and reliability of the system under a wider range of operational conditions.

9 Conclusion

In this study, we set out to address the critical challenges of autonomous UAV operations, focusing on precision landing and flight optimization by integrating ArUco marker detection and a PID controller. Our research aimed to overcome the limitations of GPS-only navigation, such as inaccuracies exacerbated by altitude changes, battery depletion, and environmental conditions.

The experimental results demonstrated significant advances. GPS-based autonomous navigation exhibited deviations ranging from **0.15** to **0.33** meters at altitudes of **10** and **25** meters receptively. Deviation inaccuracy is further increased to **152.3%** as battery voltage dropped from **12.6 V** to **9.1 V**. These deviations highlight the inherent limitations of GPS, especially at high altitudes exceeding **25** meters and low battery voltages $\leq 75\%$. In contrast, the incorporation of ArUco markers dramatically improved landing precision, achieving a detection rate of **83.33%** at an altitude of **4.5** meters and reducing deviations to as low as **0.08** meters.

The PID controller was pivotal in refining the descent process, dynamically adjusting the UAV's position to mitigate deviations and ensure a smooth landing. Our tests revealed that the combination of ArUco markers with the PID control system reduced landing deviations by up to **56.5%** compared to GPS-only approaches. This integration paved the way for autonomous docking at intelligent charging stations, facilitating extended UAV operations, and enhancing mission endurance.

Furthermore, our system's ability to handle varying battery levels and environmental factors underscores its robustness and adaptability. By enabling seamless autonomous charging, we pave the way for continuous deployment of UAVs in applications such as precision agriculture, environmental monitoring, and wireless sensor networks.

This research marks a significant step forward in UAV performance optimization, offering a cost-effective and scalable solution to the challenges of autonomous navigation. As we look to the future, incorporating Wireless Power Transfer (WPT) systems and improving IoT communication protocols will further improve UAV autonomy and efficiency. Our findings provide a strong foundation for sustainable, reliable, and precise UAV operations, advancing their applicability in real-world environments.

9.1 Limitations and Challenges

The project encountered several limitations and challenges during the development and testing phases. A primary limitation was the restricted flight time due to battery depletion, which averaged only 25 minutes per full charge. This constraint limited the number of autonomous flight tests and the scope of data collection opportunities. Furthermore, the dependency on clear visual conditions for effective ArUco marker detection posed a challenge, as uneven or sloped terrain often distorted the landing pad, reducing marker detection accuracy and affecting landing precision. The reliability of GPS positioning was

also affected in areas with poor satellite signal strength, leading to significant positional uncertainties during navigation. Furthermore, the computational demands of real-time image processing for marker detection and PID control placed strain on the Raspberry Pi 5 onboard processor, occasionally resulting in delays in control responses. Addressing these challenges will require advancements in energy management, more robust vision algorithms, and improved computational hardware to improve the efficiency and reliability of the system.

9.2 Ethical Considerations

The ethical implications of this project were carefully addressed throughout its development and implementation. Data privacy and security were prioritized to ensure that any data collected during UAV operations comply with local regulations and ethical guidelines. The design and deployment of the UAV system adhered to safety protocols to prevent harm to individuals, property, or wildlife. In addition, measures were taken to avoid misuse of the technology, such as implementing access controls and ensuring the system is used strictly for its intended purposes. Transparency in reporting findings and acknowledging limitations further supports the ethical integrity of this work, fostering trust and accountability in its applications.

9.3 Sustainability Of The Work

Ensuring the sustainability of this work involves several key considerations. Firstly, the use of locally available components during drone assembly was prioritized to minimize costs and dependence on imported parts. The modular design of the drone was another critical focus, allowing easy repair, component replacement, and future expansions, extending its operational lifespan. Furthermore, the promise of integrating energy-efficient wireless charging and the potential for solar-powered stations highlight the project's commitment to environmentally conscious solutions. The design also emphasizes scalability, allowing the drone to adapt to multi-drone operations and diverse applications such as precision agriculture and environmental monitoring. By balancing cost effectiveness, modularity, and energy efficiency, the project sets a strong foundation for long-term usability and environmental consciousness, addressing current challenges while remaining adaptable for future advancements.

9.4 Social and Environmental Effects And Analysis

The development and deployment of an autonomous UAV system that integrates precision landing based on vision-based ArUco markers offer significant social and environmental benefits, as well as considerations for responsible implementation.

Social Effects

- **Enhanced Accessibility and Efficiency:** The system facilitates precision agriculture and environmental monitoring in remote and underserved areas. By enabling autonomous UAV operations, it reduces the need for extensive human intervention, thus increasing accessibility and operational efficiency.

- **Improved Data-Driven Decision Making:** By providing high-precision data collection capabilities, the UAV system empowers stakeholders in agriculture, environmental science, and urban planning to make informed decisions, enhancing productivity and sustainability.
- **Skill Development and Employment Opportunities:** Integration of advanced technologies such as computer vision, IoT, and autonomous systems promotes skill development in these fields, potentially creating job opportunities in the operation, maintenance, and analysis of UAV data.

Environmental Effects

- **Reduction in Carbon Footprint:** The UAV system's energy-efficient operations and integration of the UAV system with wireless solar-powered charging stations minimize fossil fuel dependency. This reduces greenhouse gas emissions compared to traditional monitoring and agricultural methods.
- **Precision Agriculture and Resource Optimization:** The ability to monitor and manage agricultural fields with high precision allows for optimized water use, fertilizers, and pesticides. This not only reduces environmental impact but also conserves valuable natural resources.
- **Support for Biodiversity Conservation:** The UAV's capability to monitor environmental conditions and track wildlife activities can aid in biodiversity preservation by enabling timely interventions to address environmental degradation and habitat loss.

Challenges and Mitigation

While the system offers numerous benefits, potential challenges include:

- **Electronic Waste Management:** The proliferation of UAV systems could contribute to electronic waste. Developing eco-friendly recycling and disposal mechanisms will be critical.
- **Energy Consumption During Manufacturing:** The production of UAV components involves significant energy usage. Incorporating green manufacturing practices can mitigate this impact.
- **Potential for Privacy Concerns:** Deploying UAVs in public or private spaces may raise privacy concerns. Establishing clear regulatory frameworks and adhering to ethical guidelines is essential to ensure responsible use.

9.5 Future Plan

The findings and contributions of this research pave the way for several future advancements and optimizations in autonomous UAV systems. The proposed future work focuses on addressing the current system's limitations and exploring innovative enhancements to expand its applicability:

- **Integration of Advanced Charging Solutions:** To enhance the drone’s operational efficiency and sustainability, we plan to integrate Wireless Power Transfer (WPT) technologies with solar-powered charging stations. This integration aims to enable continuous field deployment and reduce dependence on manual charging systems.
- **Enhanced Marker Detection and Navigation:** Future work includes exploring advanced vision-based techniques, such as deep learning algorithms, to improve the reliability of ArUco marker detection under diverse environmental conditions. Additionally, incorporating 3D LiDAR will allow better terrain mapping, improving navigation and landing accuracy in complex terrains.
- **System Scalability and Fleet Coordination:** Expanding the system for multi-UAV operations is a key focus. This includes implementing decentralized communication protocols using IoT technologies like LoRa and SigFox, which can optimize drone coordination and extend operational capabilities for large-scale applications.
- **Long-term Testing and Usability Studies:** Extended field testing will be conducted to assess the system’s long-term performance and gather feedback from end-users. This iterative process will refine the system’s usability and ensure its alignment with practical requirements.

By addressing these objectives, the future work proposed aims to overcome current limitations, broaden the operational capabilities of UAVs, and contribute to the advancement of autonomous aerial technologies for critical real-world applications.

A Appendix

A.1 Turnitin Originality Report

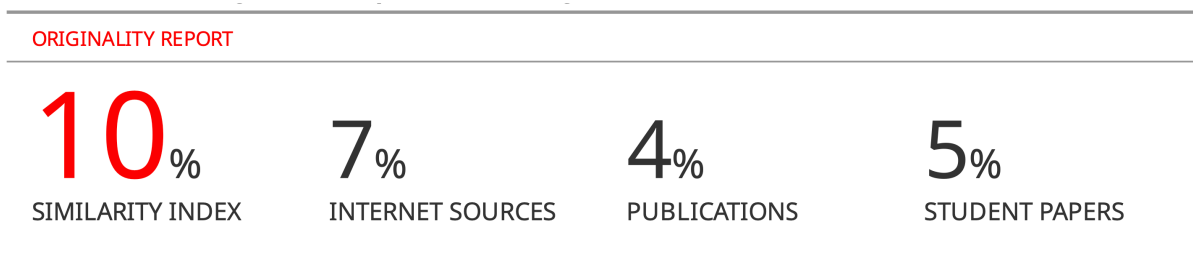


Figure 32: Turnitin Originality Report Indexes

Bibliography

- [1] A. D. Boursianis, M. S. Papadopoulou, P. Diamantoulakis, A. Liopa-Tsakalidi, P. Barouchas, G. Salahas, G. Karagiannidis, S. Wan, and S. K. Goudos, “Internet of things (iot) and agricultural unmanned aerial vehicles (uavs) in smart farming: A comprehensive review,” *Internet of Things*, vol. 18, p. 100187, 2022.
- [2] A. M. Jawad, H. M. Jawad, R. Nordin, S. K. Gharghan, N. F. Abdullah, and M. J. Abu-Alshaeer, “Wireless power transfer with magnetic resonator coupling and sleep-/active strategy for a drone charging station in smart agriculture,” *IEEE Access*, vol. 7, pp. 139839–139851, 2019.
- [3] H. Shakhathreh, A. H. Sawalmeh, A. Al-Fuqaha, Z. Dou, E. Almaita, I. Khalil, N. S. Othman, A. Khreishah, and M. Guizani, “Unmanned aerial vehicles (uavs): A survey on civil applications and key research challenges,” *Ieee Access*, vol. 7, pp. 48572–48634, 2019.
- [4] S. J. Kim, G. J. Lim, and J. Cho, “Drone flight scheduling under uncertainty on battery duration and air temperature,” *Computers & Industrial Engineering*, vol. 117, pp. 291–302, 2018.
- [5] M. Brzozowski and K. Parczewski, “Problems related to the operation of autonomous vehicles in adverse weather conditions,” *Combustion Engines*, vol. 62, 2023.
- [6] J. Springer, G. P. Guðmundsson, and M. Kyas, “A precision drone landing system using visual and ir fiducial markers and a multi-payload camera,” *arXiv preprint arXiv:2403.03806*, 2024.
- [7] G. Balamurugan, J. Valarmathi, and V. Naidu, “Survey on uav navigation in gps denied environments,” in *2016 International conference on signal processing, communication, power and embedded system (SCOPES)*, pp. 198–204, IEEE, 2016.
- [8] Z. Ge, J. Jiang, E. Pugh, B. Marshall, Y. Yan, and L. Sun, “Vision-based uav landing with guaranteed reliability in adverse environment,” *Electronics*, vol. 12, no. 4, p. 967, 2023.
- [9] J. Wang, D. McKiver, S. Pandit, A. F. Abdelzaher, J. Washington, and W. Chen, “Precision uav landing control based on visual detection,” in *2020 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, pp. 205–208, IEEE, 2020.
- [10] I. Lebedev, A. Erashov, and A. Shabanova, “Accurate autonomous uav landing using vision-based detection of aruco-marker,” in *International Conference on Interactive Collaborative Robotics*, pp. 179–188, Springer, 2020.

- [11] G. Badakis, M. Koutsoubelias, and S. Lalis, “Robust precision landing for autonomous drones combining vision-based and infrared sensors,” in *2021 IEEE Sensors Applications Symposium (SAS)*, pp. 1–6, IEEE, 2021.
- [12] R. Morales-Ferre, P. Richter, E. Falletti, A. De La Fuente, and E. S. Lohan, “A survey on coping with intentional interference in satellite navigation for manned and unmanned aircraft,” *IEEE Communications Surveys & Tutorials*, vol. 22, no. 1, pp. 249–291, 2019.
- [13] I. Um, S. Park, H. T. Kim, and H. Kim, “Configuring rtk-gps architecture for system redundancy in multi-drone operations,” *IEEE Access*, vol. 8, pp. 76228–76242, 2020.
- [14] G. Sriram Reddy and A. Nippun Kumaar, “Gps-based path planning algorithm for agriculture drones,” in *Advanced Computational Paradigms and Hybrid Intelligent Computing: Proceedings of ICACCP 2021*, pp. 241–250, Springer, 2022.
- [15] J. Patoliya, H. Mewada, M. Hassaballah, M. A. Khan, and S. Kadry, “A robust autonomous navigation and mapping system based on gps and lidar data for unconstrained environment,” *Earth Science Informatics*, vol. 15, no. 4, pp. 2703–2715, 2022.
- [16] G. Muchiri and S. Kimathi, “A review of applications and potential applications of uav,” in *Proceedings of the Sustainable Research and Innovation Conference*, pp. 280–283, 2022.
- [17] E. Alvarado, “237 ways drone applications revolutionize business,” *Drone Industry Insights*, vol. 540, 2021.
- [18] C. Koparan, A. B. Koc, C. V. Privette, and C. B. Sawyer, “Adaptive water sampling device for aerial robots,” *Drones*, vol. 4, no. 1, p. 5, 2020.
- [19] C. Koparan, A. B. Koc, C. V. Privette, and C. B. Sawyer, “Autonomous in situ measurements of noncontaminant water quality indicators and sample collection with a uav,” *Water*, vol. 11, no. 3, p. 604, 2019.
- [20] F. Rekabi-Bana, J. Hu, T. Krajník, and F. Arvin, “Unified robust path planning and optimal trajectory generation for efficient 3d area coverage of quadrotor uavs,” *IEEE Transactions on Intelligent Transportation Systems*, 2023.
- [21] A. Koubâa, A. Allouch, M. Alajlan, Y. Javed, A. Belghith, and M. Khalgui, “Micro air vehicle link (mavlink) in a nutshell: A survey,” *IEEE Access*, vol. 7, pp. 87658–87680, 2019.
- [22] S. R. Haque, R. Kormokar, and A. U. Zaman, “Drone ground control station with enhanced safety features,” in *2017 2nd International Conference for Convergence in Technology (I2CT)*, pp. 1207–1210, IEEE, 2017.
- [23] P. Kumbhar, M. Attaullah, S. Dhere, and S. Hipparagi, “Real time face detection and tracking using opencv,” *International journal for research in emerging science and technology*, vol. 4, no. 4, pp. 39–43, 2017.
- [24] S. Mishra, V. Verma, N. Akhtar, S. Chaturvedi, and Y. Perwej, “An intelligent motion detection using opencv,” *International Journal of Scientific Research in Science, Engineering, and Technology*, vol. 9, no. 2, pp. 51–63, 2022.

- [25] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez, “Automatic generation and detection of highly reliable fiducial markers under occlusion,” *Pattern Recognition*, vol. 47, no. 6, pp. 2280–2292, 2014.
- [26] “Online ArUco markers generator. (n.d.). <https://chev.me/arucogen/>.”
- [27] M. J. Willis, “Proportional-integral-derivative control,” *Dept. of Chemical and Process Engineering University of Newcastle*, vol. 6, 1999.
- [28] A. L. Salih, M. Moghavvemi, H. A. Mohamed, and K. S. Gaeid, “Modelling and pid controller design for a quadrotor unmanned air vehicle,” in *2010 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR)*, vol. 1, pp. 1–5, IEEE, 2010.
- [29] Z. Zhang, H. Pang, A. Georgiadis, and C. Cecati, “Wireless power transfer—an overview,” *IEEE transactions on industrial electronics*, vol. 66, no. 2, pp. 1044–1058, 2018.
- [30] J. Kwak and Y. Sung, “Autonomous uav flight control for gps-based navigation,” *IEEE Access*, vol. 6, pp. 37947–37955, 2018.
- [31] I. Cohen, Y. Huang, J. Chen, J. Benesty, J. Benesty, J. Chen, Y. Huang, and I. Cohen, “Pearson correlation coefficient,” *Noise reduction in speech processing*, pp. 1–4, 2009.
- [32] C. H. S. Lee, S. K. Phang, and H. K. Mun, “Design and implementation of an agricultural uav with optimized spraying mechanism,” in *MATEC Web of Conferences*, vol. 335, p. 02002, EDP Sciences, 2021.