DEVELOPMENT OF PID CONTROL FOR AUTONOMOUS HORIZONTAL NAVIGATION OF NON-GPS DRONE WITH ONBOARD MICROCOMPUTER

SUBMITTED TO THE FACULTY OF THE
DEPARTMENT OF ELECTRICAL ENGINEERING
COLLEGE OF ENGINEERING AND AGRO-INDUSTRIAL TECHNOLOGY
UNIVERSITY OF THE PHILIPPINES LOS BAÑOS
BY

CHRISTIAN JAKE TAGULINAO MANGABAY

IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE
DEGREE OF

BACHELOR OF SCIENCE IN ELECTRICAL ENGINEERING
(Major in Computer Engineering)

JUNE 2024

This thesis manuscript can be accessed:

By the general public	YES
Only after consultation with the author/adviser	-
Only by those bound by a confidentiality agreement	-

Signature of Student: ____________
Signature of Adviser: ____________

The thesis attached hereto, entitled **"DEVELOPMENT OF PID CONTROL FOR AUTONOMOUS HORIZONTAL NAVIGATION OF NON-GPS DRONE WITH ONBOARD MICROCOMPUTER"**, prepared and submitted by **CHRISTIAN JAKE TAGULINAO MANGABAY** in partial fulfillment of the requirements for the degree of **BACHELOR OF SCIENCE IN ELECTRICAL ENGINEERING**, is hereby accepted.

MELVIN C. ILANG-ILANG
Member
Guidance Committee
14-June-2024
Date Signed

ROCK CHRISTIAN V. TOMAS
Member
Guidance Committee
14-June-2024
Date Signed

LORWIN FELIMAR B. TORRIZO
Member
Guidance Committee
14-June-2024
Date Signed

ROB CHRISTIAN M. CADUYAC
Adviser and Chair
Guidance Committee
14-June-2024
Date signed

RODERICK L. CATRIZ
Chair
Department of Electrical Engineering
14JUNE2024
Date Signed

REX B. DEMAFELIS
Dean
College of Engineering and Agro-industrial Technology

Date Signed

BIOGRAPHICAL SKETCH

Christian Jake T. Mangabay was raised in Los Baños, Laguna, Philippines. He studied elementary school in this town, junior high school at a neighboring town, Bay, and then senior high school at Calamba City. Those places lived his dream to become an inventor.

Mangabay started his college education at University of the Philippines Los Baños as a student in Bachelor of Science in Electrical Engineering in 2019. His major was in Computer Engineering with the field of interest encompassing automated electronic and mechanical systems, programming, artificial intelligence, and 3D models and designs.

Mangabay was a charter member and one of the first executive committee members of UPLB Data Science Guild, serving as the chairperson for Training and Skills Committee in Academic Year 2023-2024. There, he has harnessed basic skills in technical aspects of data science through seminars and practices. He was also a member of the UP Engineering Radio Guild - Los Baños since 2022.

CHRISTIAN JAKE TAGULINAO MANGABAY

ACKNOWLEDGEMENT

Completing this thesis and my degree has been a profound journey, and I am grateful to all who have assisted me along the way. This accomplishment would not have been possible without the guidance, support, and inspirations from numerous individuals and groups.

First and foremost, I would like to express my sincere gratitude to my adviser, Engr. Rob Christian Caduyac. He has welcomed me under his supervision and provided me with steadfast help and thoughtful feedback throughout the conduct of my study despite my shortcomings as an advisee. His expertise and knowledge were the key instruments in finishing this paper.

I am also thankful to the members of the guidance committee, Engr. Melvin Ilang-Ilang, Engr. Rock Christian Tomas, and Engr. Lorwin Felimar Torrizo for their time, constructive critiques, and helpful suggestions which significantly directed the betterment of this work. The critiques were challenges that pushed me to improve, while the praises served as the drive to continue striving for excellence. All the professors I've met at UPLB especially from DEE were part of my great learning.

My journey on my thesis as well as the degree has not been easy. There are many others whom I would like to address my appreciation to for their invaluable input.

To my family, your sacrifices have allowed me to focus wholeheartedly on my thesis and studies. Your tireless commitment and understanding have supplied the foundation I needed to persevere through the challenges. Thank you for believing in me and making innumerable sacrifices so I can pursue my dreams.

To DOST-SEI Undergraduate Scholarship, without your financial aids, I would not have enough resources to fully engage in my studies and attain my academic goals.

To UP Engineering Radio Guild - Los Baños, your contributions to my academic and social life can never be measured. Norbert, Edmel, and Kailah had always been there from the start of the course until the end of my thesis, we crammed it all. And to all the members that I may not be able to mention, I had a lot of fun, learnings, and assistance from our camaraderie during our events, leisure, meetings, and schoolwork.

To UPLB Data Science Guild, the trust and opportunities you have given me have raised my courage and skills. You had started the clarity of direction to my dream in the future.

To Charlie, Alina, and Victor, the moments we've shared as a group have created a collection of sweet memories. From our dinners and hangouts, each conversation is where we celebrate and worry with everyone. Whether it was late-night cramming, eating ice cream on chilly evenings, playing badminton on hot days, having walk trips, or hitting the gym, we've done it all. It's from each of you that I draw my confidence.

To Nani, your contributions were special. Your consistent support has always uplifted me during the setbacks throughout my thesis. Our shared interests and personalities, and the album of our companionship are fuels to my passion and motivation to keep on aspiring. The checklist we made allowed me to finish the overwhelming tasks, especially during the crucial final week that determined this success. Thanks to you and to Tita Bel for the countless encouragements you had granted me.

Ultimately, I am grateful to the Almighty Who has gifted me everything and everyone, from the beginning of my venture until where I am now.

Finally, after five years, I can have the sunflower and sash. All of these will be my treasure forever.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

MANGABAY, CHRISTIAN JAKE T. College of Engineering and Agro-industrial Technology, University of the Philippines Los Baños. June 2024. **Development of non-GPS Drone Using Onboard Microcomputer for Autonomous Horizontal Navigation.**

Major Adviser: Rob Christian M. Caduyac

Unmanned aerial vehicles (UAVs), or drones, have transformed numerous sectors with diverse applications. Traditional remote control methods are prone to human error, and automated flights via GPS can be unreliable in areas with weak signals. Integrating computer vision into onboard microcomputers allows drones to interpret visual data, reducing dependency on GPS and remote controls. This research enhances drone technology by developing a vision-based mission control system using a Raspberry Pi Zero 2 W for autonomous horizontal navigation. Movements were determined by the relative position of ArUco markers within captured images. PID controllers were employed to smooth movements based on the offset magnitude from the setpoint. The Raspberry Pi, attached to the drone, connected to the drone's system via WiFi to send commands and receive sensor data. The drone successfully navigated a path of six markers, spaced an average of 1.2 meters apart, while carrying the Raspberry Pi Zero 2 W, power module, and battery. The setup was evaluated for inference time, travel time, and path error. Variations of PID, specifically PID, PI, and PD, were compared. PD achieved the highest accuracy of 2.375 cm for forward-backward movement but had low accuracy for left-right movement due to a slow start. PI showed opposite results with an overshoot of 13 cm. PID offered balanced performance, with 14 cm left-right accuracy and a 6.5 cm overshoot. PI was the fastest for straight paths, leading PID by 3.6 seconds, but PID excelled in turns, leading by 1.5 seconds due to better path accuracy. Inference times showed minimal differences among the three variations.

1. INTRODUCTION

This provides an overview of the research objectives and significance. It addresses the need for alternative navigation systems in drones, especially without human input and when GPS signals are unreliable. This chapter sets the stage for achieving autonomous horizontal navigation in drones, emphasizing the development of an onboard microcomputer-based navigation system.

1.1. Background of the Study

Unmanned aerial vehicles (UAVs), commonly known as drones, have transformed numerous sectors with versatile applications in surveillance, photography, inspection, and emergency response. They provide cost-effective access to challenging terrains and remote areas without requiring the physical presence or actual physical position and location of a person (Lutkevich, 2019). Controlling drones requires precise and reliable methods to navigate and maneuver. The traditional way of flying drones is through a remote control. Remote control systems allow operators to manually pilot the drone in real-time using a handheld transmitter or controller. However, the interference between the drone and the controller results in delay and more errors. In many cases, these mistakes can lead to flight inefficiencies, such as wasted time and energy, incorrect task execution, and, in the worst-case scenario, even crashes. Moreover, this method is highly dependent on the pilot's skills, responsiveness, and exertion, which introduces the risk of mistakes.

An autonomous flight system is another method for controlling drones. It utilizes onboard or remote sensors, navigation systems, and algorithms to enable drones to operate independently without direct human intervention. This results in increased efficiency, reduced operator workload, and the ability to perform complex tasks with minimal human input. According to Amer (2021), various methods of autonomous flights and navigation include the use of Global Positioning System (GPS), Radio Frequency (RF) and Beacon-based Navigation. GPS is being the most used.

Computer vision is a field of study focusing on enabling computers to interpret, understand, and analyze visual data from the real world by developing algorithms and techniques to extract meaningful information from digital images or video sequences. Implementing this in an onboard microcomputer can be applied to drones by using an onboard camera to capture images instead of relying on transmitted signals from other devices. This has advantages, particularly in scenarios where the other methods, specifically the GPS signals may be inaccurate or inaccessible. Deploying a vision-based method in a microcomputer attached to the drone eliminates the necessity of other external control signals in navigation.

Raspberry Pi Zero W is a low-cost and streamlined version of the Raspberry Pi. It offers a compact and lightweight alternative with integrated Wi-Fi and Bluetooth capabilities. However, due to its smaller size and reduced hardware specifications, the computing power of the Raspberry Pi Zero W is limited in comparison to the higher-end models in the Raspberry Pi family. This brings the necessity of optimization and minimization to be suitable for simpler tasks and projects where a lower processing capability is sufficient.

This research paper focuses on developing a drone mission control using an onboard Raspberry Pi Zero 2 W for autonomous navigation. The setup is intended to be independent of any external signals from other devices. Only ArUco markers will guide the drone to traverse the entire track. The system will be evaluated in real-world scenarios to assess its performance and generalization capabilities. The research aims to contribute to the advancement of autonomous drone technology.

1.2. Significance of the Study

Vision-based mission control of drones can mark a pivotal step forward in the technological evolution of unmanned aerial systems. Primarily, using an onboard microcomputer removes the dependence on radio wave navigation systems, especially GPS and on remote controls. This is particularly crucial as those methods, despite their widespread use in modern drones, comes with inherent disadvantages. These include inaccuracies and susceptibility to signal interferences, especially in areas such as dense urban landscapes with tall buildings, deep valleys or canyons, dense foliage in forests, tunnels, and indoor spaces. The use of RF and Beacon-based Navigation follows a similar premise. Moreover, using remote devices introduces delays. By eliminating reliance on those signals, autonomous drone navigation offers a more versatile and robust solution, enhancing the adaptability of drones across a spectrum of applications and environments.

Eliminating manual control gives an advantage by removing the requirement of external devices' signals and significantly reducing human error. Manual control poses challenges when the operator lacks the necessary skills, focus, or situational awareness, leading to flight inefficiencies and potential accidents. Autonomous flight offers the advantages of increased efficiency, reduced operator workload, and the ability to perform complex tasks with minimal human input. Autonomous navigation thereby enhances the overall reliability, efficiency, and safety of the flight of the drone. This serves as a transformative solution in diverse industries where precision is critical.

In addition, the absence of a dedicated pilot emerges as a secondary but significant benefit. This feature liberates workers, allowing them to redirect their attention to other tasks, effectively optimizing workflow and leveraging human resources for enhanced productivity. Autonomous drone navigation not only streamlines operations but also positions itself as a holistic solution, addressing both the imperative need for precision and the optimization of workforce engagement in various applications.

1.3. Objectives of the Study

The goal of this study is to develop a drone mission control using an onboard Raspberry Pi Zero 2 W for autonomous horizontal navigation.

1. Design a PID controller that will navigate the horizontal path pipeline based on ArUco markers positions relative to the drone.
2. Deploy the PID control algorithm on the Raspberry Pi Zero 2 W that will be attached to the drone.
3. Evaluate the performance of the setup in a real-world environment using inference time, travel time and path error.

1.4. Scope and Limitations

This study is exclusive on navigation along the horizontal plane, omitting the vertical movement or along the y-axis. The flight evaluation only includes the start and the end of the path excluding the take-off and the landing. The image showing ArUco markers outputs solely provides information regarding positions to produce velocities and accelerations.

The navigation analysis is conducted in proximity to an ideal environment, excluding the influence of aerodynamics such as uncontrolled wind and environmental disturbances like atmospheric precipitation. Additionally, automatic obstacle avoidance, such as for avoiding birds or tall structures, is not considered.

This study incorporates the communication between the computer containing the system and algorithm, and the drone, particularly concerning the time-delay. It is crucial for adaptability, synchronizing, and the responsiveness of the control outputs to the image inputs. This offers the adaptability for different machine locations the system will be installed, whether on the onboard processing unit or on a separate computer device.

Lastly, the capability of performing other tasks such as mapping, and surveillance is not a concern in making the algorithm.

1.5. Date and Place of the Study

The study will be conducted at an event hall at Bay, Laguna. This hall has walls at all of the sides with windows that can be closed. The total floor area is about 150 square meters while the ceiling is dome-type ranging from 3 meters up to 7 meters. The specific location and building will not be stated due to privacy reasons. The data gathering involving the actual drone flights will be implemented at the event hall. The development of the system will be undertaken at the researcher’s residence in Los Baños.

2. REVIEW OF RELATED LITERATURES

This chapter explores the existing research and knowledge surrounding unmanned aerial vehicles (UAVs) with a particular focus on navigation systems not reliant on Global Positioning System (GPS) technology. It encompasses a comprehensive review of literature investigating various methodologies and technologies utilized in autonomous navigation, including computer vision techniques. Additionally, recent advancements in microcomputing and their applications in enhancing the autonomy and navigation capabilities of drones in GPS-denied environments are examined.

2.1. Status Quo of Drone Navigation

According to Lesley (2023), most of the drones available in the market today are equipped with GPS. GPS in drones is used for features like returning home, maintaining location, and pilotless flying.

In general, the purposes of drone mission control revolve around monitoring, surveillance, payload carrying and delivery, and change detection on locations and tasks that are difficult, dangerous, or inefficient for human workers to conduct. The following are the studies that concluded about the use of drones in specific areas.

2.1.1 Applications of Drone Navigation

Crop Health Monitoring: Drones equipped with cameras efficiently identify crop diseases, damage, and pest infestations, optimizing crop protection and growth through precise chemical applications based on specific conditions. Their accuracy and reduced waste minimize environmental impact and health risks. Gayathri Devi (2020).

Battlefield Surveillance: According to Bera (2022), drones transform battlefield intelligence gathering by providing real-time visual data for situational awareness. Drones significantly enhance surveillance by navigating hostile areas with stealth and flexibility, aiding in mission planning, target acquisition, and overall operational efficiency.

Delivery of Healthcare Items: Drones, highlighted by Scott (2017) and Flemons (2020), offer a transformative solution for delivering healthcare items to inaccessible locations, ensuring timely delivery, and potentially saving lives in regions with limited traditional transportation methods.

Terrain Analysis and Change Detection: Described by Agarwal (2019), drones revolutionize terrain analysis by efficiently detecting land changes. Equipped with advanced sensors, they capture high-resolution data for precise monitoring of large areas, aiding in identifying erosion patterns, mapping landforms, assessing vegetation health, and detecting natural disasters. Drones' versatility provides cost-effective and timely data collection, eliminating the need for manual surveys or expensive satellite imagery.

2.2. Disadvantages of Manual Flight Control

The largest disadvantage of manual control is the risk of human error which reduces the efficiency of the flight. Most of the factors affecting the probability of errors in the performance of an operator are pilot qualification, physical limitations, multitasking, and fatigue.

Controller Delay. Generally, unmanned aerial vehicles are operated through remote control (RC) devices via a designated wireless channel. The delay value within the control channel stands out as a primary factor influencing the ease and effectiveness of remotely controlling an unmanned vehicle. (Brodnev et. al, 2020)

Pilot Qualification. Pilot qualifications are crucial for efficient drone flights. According to Cervantes (2019), a qualified pilot optimizes operations, understands control inputs, and prevents overshooting by adjusting speed and turn radius. Effective application of skills and experience is essential to avoid these issues.

Physical and Mental Limitations. Mia (2020) and Cervantes (2019) highlight inherent physical limitations in humans, affecting navigation and tasks. These include reaction time, processing time, and vision clarity, leading to delayed responses, processing delays, and prolonged reaction times. Additionally, human concentration declines during navigation, impacting cognitive functions and causing cognitive overload during extended periods.

Necessity for Multitasking. Drone operation diminishes autonomy and multitasking capability as pilots must split focus between navigation and the intended purpose, like surveillance. Cervantes (2019) notes that pilots, responsible for flight control and navigation, face challenges in managing parameters like altitude, speed, and location. Multitasking, especially with objectives like surveillance, divided attention, compromising control and response time. This dual focus impacts task quality, leading to potential oversight and reduced effectiveness in surveillance.

2.3. GPS Drones

GPS drones are unmanned aerial vehicles (UAVs) that are equipped with Global Positioning System (GPS) technology. The integration of GPS allows these drones to receive signals from satellites and determine their precise location on the Earth's surface. In navigation, the built-in GPS system enables drones to autonomously follow a flight path, maintain position, and return to a specific location without constant manual input from the operator.

A study by Brunner (2019) on the application of drones particularly using GPS navigation in delivery of products through the clients' balconies found out that using GPS in urban environments can present several challenges due to the unique characteristics of cities and the dense infrastructure. Signal Interference: Tall buildings, bridges, and other structures in urban areas can obstruct GPS signals or cause multi-path interference. The signals may bounce off buildings and create inaccuracies in positioning. Here, the GPS signal can suffer also from drift, where the drone's reported position may shift slightly from its actual location. This drift and interference can accumulate over time and may lead to navigation errors. Additionally, the precision of GPS has an error range of about 4.9 meters under the best conditions which can be wider in less ideal situations. In cases of applications that need high accuracy, having a mistake of more than 5 meters might be a failure of purpose such as in package deliveries.

Utilizing the images captured by the drone for navigation removes the conflicts with satellite signal interferences. The drone becomes self-reliant on its visual perception of the environment, which enhances its navigational capabilities. Due to this feature, non-GPS drones can be used in navigation which usually have lower costs as compared to their GPS-equipped counterparts.

2.4. Computer Vision

Computer vision is a branch of artificial intelligence focused on teaching machines to interpret visual data. It involves creating algorithms and models for analyzing images and videos, enabling computers to make decisions based on visual information. Applications range from image recognition to autonomous vehicles. In drone navigation, computer vision is crucial, allowing drones to interpret surroundings in real-time for autonomous navigation and obstacle avoidance. This capability enhances safety and efficiency in drone operations, making computer vision integral to maximizing the potential of drone technology.

A study by Arafat, M, et. al (2023) shows the fitness of computer vision in making drones able to see and understand things virtually. It uses smart algorithms to make drones better at avoiding obstacles and figuring out where to go. The study proves that computer vision makes drones work well in different situations, making them more precise and efficient.

2.5. Microcomputers

A microcomputer as described by Dutch et. al . (2018) , is a small-scale computing device that incorporates a microprocessor as its central processing unit (CPU). They are known for their versatility, affordability, and ease of integration into various projects. They often run on lightweight operating systems and support programming in multiple languages. With features like GPIO pins for hardware interaction, microcomputers are widely used for DIY projects, educational purposes, and prototyping. Their small form factor makes them suitable for a range of applications, from embedded systems and robotics to home automation and creative coding endeavors.

2.5.1. Raspberry Pi Zero 2 W

The Raspberry Pi Zero W is a compact and affordable Linux-based single-board computer that offers versatile computing capabilities. It features a Broadcom BCM2835 system-on-chip (SoC) with a single-core ARM1176JZF-S CPU running at 1GHz. The board is equipped with 512MB of RAM, making it suitable for lightweight computing tasks. It has built-in wireless connectivity options, including 802.11n Wi-Fi and Bluetooth 4.0. It provides a microSD card slot for storage, a mini HDMI port for video output, a micro-USB port for power and data, and a 40-pin GPIO header for interfacing with various peripherals. These can be used in pre-interfacing and PID tuning. With its minimal power consumption and compact form factor, it is well-suited for embedded projects on IoT applications and educational purposes.

2.6. ArUco Markers

Augmented Reality (AR) is a technology that combines digital information or virtual content with the real-world environment. It enhances a user's perception by overlaying computer-generated images, graphics, or information onto the physical world. Unlike Virtual Reality (VR), which fully immerses users in a simulated environment, AR allows users to interact with both the virtual and real worlds simultaneously. One of the examples of using AR is by ArUco markers. A study by Avola (2016) demonstrates the enhancement of AR using ArUco markers which makes the identification process faster. Moreover, it is easier to track movements as by identifying a marker, the location and position of the marker also comes with it.

ArUco markers make it easier for the programs to read in a real-world environment. Instead of assessing every pixel for every feature of the 2D images of an object, scene, or location, a marker can be used to represent that unique location together with its angular data. Li (2020) concluded in his study that this marker remarkably lessens the computation power and code lines needed to perform a navigation. These serve as control points on the ground to greatly simplify the mapping and location processes.

2.7. DJI Ryze Tello Drone

DJI Tello is a non-GPS drone. The key feature in this unit is the capability to be controlled and programmed in a separate computer with Wi-Fi as the means of connection and communication. A paper by Ghazi and Voyer (2024) introduces the Tello drone as a useful tool for education primarily focusing on control engineering. This is because its software development kit (SDK) has a wide source for programming. Using python codes, a computer can access the data from its sensors and instruct its motors to produce the desired movements. This enables the image processing and control algorithms contained in a python script to be interactive to the drone's controls and sensors.

Drone Specifications: The DJI Tello drone specifications were provided by Ryzer Robotics on their website. This drone has a 5-megapixel camera equipped with electronic stabilization for photographs and video recordings. The fixed wide-angle camera has a field of view (FOV) of approximately 82.6°. The secondary camera at the bottom of the drone will be used only for stabilization purposes since the scope of this study only covers the horizontal movement in a control environment. The maximum altitude of its flight in a controlled environment is 30 meters while its horizontal range from the controller is about 100 meters.

2.8 Root Mean Squared Error (RMSE)

Root Mean Squared Error (RMSE) is a widely used statistical metric known for its versatility in quantifying error across diverse domains. This metric calculates the average magnitude of differences between observed and expected values. It's instrumental in evaluating accuracy and precision, aiding in various applications such as quality control and spatial analysis. Lower RMSE values indicating better model performance. Its advantages lie in its simplicity, ease of interpretation, and ability to penalize larger errors more significantly, making it particularly useful for assessing the overall performance of regression models.

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}} \qquad \textit{(Equation 2-1)}$$

where:

$RMSE$	is the Root-Mean Squared Error
n	is the number of samples
$\hat{y}_i$	is the ideal value
y_i	is the actual value

RSME will be used in the study to find the path error in the experiment. This will measure the deviation relating to the accuracy of the PID controller based on the actual navigated path and the target path.

2.9 Zeagler-Nichols PID Tuning Method

The Ziegler-Nichols method is a classical technique for tuning PID controllers, particularly in closed-loop systems. This method involves gradually increasing the proportional gain (Kp) until the system exhibits sustained oscillations, identifying the critical gain (Kc) and oscillation period (Tu). From these parameters, the proportional, integral, and derivative gains are calculated for the PID controller. The paper "Tuning Rules for PID Controller Using Ziegler-Nichols Method: A Review" offers a comprehensive examination of the Ziegler-Nichols method for tuning PID controllers. It provides an overview of the method's principles, variations, and applications in control engineering. (Bafkar, 2011)

This method will be used in the study to get the values of the PID constants. This is because there are limitations in the reference for the equations that will predict the movement of the drone that affects the initial dynamics of the original Ryze Tello drone control. The reason for this is the added weight to the drone. These constants will be used throughout the whole experimentation specifically for comparing PID, PI, and PD.

2.9. Feasibility of the Onboard Raspberry Setup for Horizontal Navigation

These are similar studies involving the use of a microcomputer for drones. These will serve as references for the setting up of the onboard microcomputer.

2.9.1 Computing Capabilities of Raspberry Pi Zero W for Image Processing

The paper by Nagpal et. al (2018) introduces a system comprising a Raspberry Pi Zero, Raspberry Pi Camera Module, capacitive touch sensor, and an OLED display. Facial detection is performed using the Haar Cascade classifier to identify any faces within the captured image. Subsequently, facial recognition is executed using the Local Binary Pattern Histogram (LBPH) algorithm, utilizing the system's face database. The image processing workflow is divided into two main segments: Detection, focusing on identifying faces within the image, and Recognition, which involves identifying the recognized face within the image.

The system comprises a Raspberry Pi Zero equipped with a Broadcom BCM2835 SoC and an ARM11 CPU running at 1 GHz, accompanied by 512 MB of

RAM. Connected to this is a Raspberry Pi camera module with a 5-megapixel resolution capable of capturing images up to 2592 x 1944 pixels, as well as a 0.96-inch OLED display with a resolution of 128 x 64. All components are linked to the Raspberry Pi, serving as the central control hub for the entire setup. The results show that the mean time to capture an image, detect a face, and then display it on an OLED screen is between 2 and 3 seconds.

The setup of Nagpal, et. al. (2018) shown the image processing capability of a raspberry pi zero w with the stated specifications that are relatively lower than a raspberry pi zero 2 w. In relation to the drone setup, the image resolution is probably way less than 2592 x 1944 pixels if the target inference time is less than 0.1 seconds even if the hardware of Raspberry Pi Zero 2 W offers more power than its predecessor.

2.9.2 PID Tuning Example Using Progressive Refining

The paper by Xing, et. al. (2018) utilized a microcomputer to instruct the drone's movement. In there, the camera is looking at the drone. In relation to the study, the baseline inference time for capturing image, processing, and sending commands to the drone is 0.1 second or better as derived from the stated results. Moreover, the use of a Pi camera and connecting to the drone via WiFi offers a good system. The only difference to the study is the Pi camera, and the Raspberry Pi Zero 2 w will be onboard.

Additionally, it suggests initial values for Kp and Kd which are 0.22 and 0.08, respectively.

2.9.3 Feasibility of Onboard Microcomputer for Ryze Tello

The paper by Maalouf (2018) demonstrated the feasibility of DJI Tello drone carrying a Raspberry Pi Zero 2 W. The difference is that the microcomputer used in the setup was for object detection and classification. This has also provided the placement of the microcomputer on the drone especially to avoid interfering with the Visual Positioning System (VPS) sensor at its bottom.

2.10 Related Studies

These are the studies related to autonomous drone navigation of non-GPS drones, Raspberry Pi Zero, and ArUco markers. These were gathered to support the idea that the microcomputer and ArUco markers individually can be used to create a system of autonomous drone navigation, without requiring GPS and remote signals.

A paper by Teixeira, B. M., Silveira, L. A., Pereira, G. A. S., and Mendes, R. P. (2018) presented a methodology for autonomous UAV navigation using visual and inertial sensors. The study focused on the integration of these sensors to provide reliable navigation data in GPS-denied environments. The proposed system employed a combination of visual odometry and an inertial measurement unit (IMU) to achieve accurate state estimation. The results demonstrated that the UAV could navigate autonomously with high precision in various indoor environments. This high level of accuracy underscored the robustness of the proposed fusion method in varying indoor conditions. The primary challenge addressed was the reliability of sensor data in dynamic and GPS-denied environments. The assumption was that visual and inertial sensors could provide complementary information to enhance navigation accuracy. The study tackled these challenges by developing robust sensor fusion algorithms and conducting extensive real-world testing to validate the system's performance under different conditions.

A study by Madani, T., & Benallegue, A. (2006) introduced a backstepping control approach for the autonomous navigation of a quadrotor helicopter. The authors designed a nonlinear controller that guaranteed stability and robustness in the presence of external disturbances and model uncertainties. The effectiveness of the proposed controller was validated through both simulations and experimental flights, showing significant improvements in navigation accuracy and stability. Key challenges included handling model uncertainties and external disturbances. The authors assumed the quadrotor dynamics could be accurately represented using their model. They addressed these challenges by employing backstepping techniques that enhanced the robustness and stability of the control system, ensuring reliable performance even under varying conditions.

Research by Mellinger, D., Michael, N., & Kumar, V. (2012) focused on trajectory generation and control strategies for executing precise and aggressive maneuvers with quadrotors. The authors developed a method that combined differential flatness and model predictive control (MPC) to generate feasible trajectories in real-time. The proposed approach enabled the quadrotor to perform complex maneuvers with high precision and minimal computational overhead. The main challenges were real-time trajectory generation and ensuring control accuracy during aggressive maneuvers. Assumptions included accurate modeling of quadrotor dynamics and availability of sufficient computational resources. The study addressed these by leveraging differential flatness for efficient trajectory planning and MPC for adaptive control, validated through both simulations and experimental results.

A paper by Forster, C., Carlone, L., Dellaert, F., & Scaramuzza, D. (2017) presented a novel approach for visual-inertial odometry (VIO) using on-manifold preintegration. The method improved the efficiency and accuracy of VIO by integrating

IMU measurements directly on the manifold, avoiding linearization errors. The approach was tested in various scenarios, demonstrating robust performance and precise state estimation in real-time applications. Challenges included handling noise in sensor data and ensuring real-time performance. Assumptions were that the IMU and visual sensors were well-calibrated and provided accurate data. The study addressed these by developing an on-manifold preintegration technique that minimized errors and computational load, validated through extensive experiments in different environments.

Research by Faessler, M., Kaufmann, E., & Scaramuzza, D. (2018) introduced a differentiable Kalman filter framework for learning optimal sensor fusion strategies in autonomous systems. The proposed method integrated machine learning techniques with traditional Kalman filtering to adaptively learn the best fusion weights from data. The approach was tested on UAV navigation tasks, showing improved accuracy and robustness compared to conventional methods. The main challenges were optimizing sensor fusion in dynamic environments and ensuring generalization across different scenarios. Assumptions included the availability of sufficient training data and accurate sensor models. The study addressed these by combining machine learning with Kalman filtering, allowing the system to learn optimal fusion strategies that enhanced navigation performance, demonstrated through experimental validation.

A paper by Droeschel, D., Nieuwenhuisen, M., Beul, M., Houben, S., & Behnke, S. (2016) explored multilayered mapping and navigation techniques for autonomous micro aerial vehicles (MAVs). The authors developed a system that combined 2D and 3D mapping to enhance navigation capabilities in complex environments. The MAV used onboard sensors to create detailed maps and plan optimal paths, achieving reliable navigation in both indoor and outdoor settings. Challenges included handling diverse and dynamic environments and ensuring real-time mapping and navigation.

Assumptions were that onboard sensors could provide accurate data and that computational resources were sufficient. The study addressed these by integrating multilayered mapping techniques and optimizing the navigation algorithms for real-time performance, validated through extensive experimental flights.

A paper by Lee, H., Choi, H. S., Kim, H. J., & Shim, D. H. (2017) presented a relative localization method for formation flight of quadrotors using moving baseline GPS. The authors developed a system that allowed multiple quadrotors to maintain precise relative positions without relying on fixed ground stations. The approach was tested in various formation flight scenarios, demonstrating high accuracy and robustness in maintaining formation. Key challenges included ensuring reliable relative localization in the presence of GPS errors and maintaining formation in dynamic environments. Assumptions included the availability of moving baseline GPS and that quadrotors had synchronized time. The study addressed these by developing robust relative localization algorithms and conducting extensive experimental validation in different flight scenarios.

Research by Bry, A., & Roy, N. (2011) introduced rapidly-exploring random belief trees (RRBT) for motion planning under uncertainty. The authors developed a novel planning algorithm that accounted for both state and sensing uncertainties, enabling robust motion planning in uncertain environments. The approach was validated through simulations and experiments, showing improved performance in navigating complex and uncertain environments. The main challenges were handling uncertainty in state estimation and sensor data. Assumptions included the availability of accurate models for uncertainty representation. The study addressed these by developing RRBT, which efficiently explored the belief space and planned robust trajectories, validated through both simulations and real-world experiments.

A study from Li, Y., & Zhang, B. (2016) proposed an adaptive robust control strategy for quadrotor helicopters. The authors designed a controller that adapted to changing flight conditions and external disturbances, ensuring stable and accurate flight. The effectiveness of the proposed controller was demonstrated through simulations and experimental flights, showing significant improvements in stability and robustness. Key challenges included dealing with model uncertainties and external disturbances. Assumptions were that the quadrotor dynamics could be accurately modeled and that the environment was relatively known. The study addressed these by developing an adaptive control strategy that adjusted parameters in real-time to maintain stability and performance, validated through experimental results.

Research by Sun, J., Yang, S. X., & Peng, K. (2015) presented a bioinspired approach to multi-UAV formation control, drawing inspiration from natural behaviors such as flocking and schooling. The authors developed a decentralized control strategy that allowed multiple UAVs to form and maintain formations autonomously. The approach was validated through simulations and real-world experiments, demonstrating efficient and robust formation control. The primary challenges included ensuring coordination among multiple UAVs and maintaining formation in dynamic environments. Assumptions included the availability of reliable communication among UAVs and accurate relative positioning. The study addressed these by developing bioinspired control algorithms that mimicked natural behaviors, leading to robust and adaptive formation control, as demonstrated through extensive testing.

Table 2.10. Summary of the Related Studies

Summary of RRL	Environment			Evaluation		Sensor Used		Onboard Control
Author	Known	Unknown	Non-Navigation	Actual	Simulation	Visual	Non-Visual	
This Study								
Teixeira, B. M., Silveira, L. A., Pereira, G. A. S., & Mendes, R. P. (2018).								
Madani, T., & Benallegue, A. (2006)								
Mellinger, D., Michael, N., & Kumar, V. (2012).								
Forster, C., Carlone, L., Dellaert, F., & Scaramuzza, D. (2017).								
Faessler, M., Kaufmann, E., & Scaramuzza, D. (2018).								
Droeschel, D., Nieuwenhuisen, M., Beul, M., Houben, S., & Behnke, S. (2016).								
Lee, H., Choi, H. S., Kim, H. J., & Shim, D. H. (2017).								
Bry, A., & Roy, N. (2011).								
Li, Y., & Zhang, B. (2016).								
Sun, J., Yang, S. X., & Peng, K. (2015).								

2.10.1 Summary of Challenges in the Studies

Sensor Reliability and Integration. A recurring challenge is achieving reliable navigation and state estimation in GPS-denied environments. Integrating multiple sensors (visual, inertial, etc.) to compensate for individual weaknesses, like visual odometry's sensitivity to lighting and IMU's drift over time, is crucial for robust performance. To address this, this study will calibrate the camera to ensure minimum inaccuracy in measurements. To have consistency, the measurement will rely on the visual data.

Nonlinear Dynamics and Control. The complex and nonlinear dynamics of quadrotor helicopters pose significant challenges for control system design. Ensuring stability and robustness in the presence of model uncertainties and external disturbances, such as wind, requires advanced control techniques like backstepping and adaptive robust control. To minimize the effect of non-linear dynamics, the navigation will be limited to horizontal movements. Also, the environment will be controlled excluding the challenges from wind and aerodynamics.

Trajectory Planning and Execution and Uncertainty in Motion Planning. Developing real-time trajectory generation methods that enable precise and aggressive maneuvers while maintaining minimal computational overhead is challenging. Motion planning under uncertainty, especially in environments with significant state and sensing uncertainties, is a complex challenge. To address these challenges, the paths that will be used are known and defined for simplicity.

Real-Time Performance. Achieving real-time performance while maintaining high accuracy in state estimation and sensor fusion is vital for practical applications. Techniques like on-manifold preintegration for visual-inertial odometry and differentiable Kalman filters for sensor fusion aim to address these challenges but require meticulous algorithm design and validation .

Mapping and Navigation in Complex Environments. For autonomous navigation, creating detailed and accurate maps in real-time while navigating through complex and cluttered environments is challenging. Multilayered mapping techniques aim to enhance navigation capabilities, yet ensuring consistent mapping accuracy and reliable obstacle avoidance remains difficult. To simplify this, the path that will be used will have minimal complications focusing on the PID control.

3. MATERIALS AND METHODS

The methodology aims to implement a vision-based autonomous horizontal navigation by attaching a Raspberry Pi Zero 2 W in a drone. The methods to achieve this will be discussed in this chapter.

Materials, Flight Setup and Venue include the actual devices, the software, and subparts of software that will be used in the experiment. The location and the path traces are also added to serve as guide for the pilot in manual controlling of the drone.

3.1. DJI Tello Drone

The drone that will be used is DJI-Ryze Tello. It is a mini drone that has a broad support for programming through open-source libraries. Its maximum flight altitude is 30 meters with slow and fast speed modes.

Figure 3-1. DJI-Ryze Tello drone.

3.2. Camera

The flight altitude for the path will be 2.5 meters. At this altitude, the bottom-view blind spot on the ground is about 3.75 meters from the point on the ground below the drone's camera due to the angle of the drone's camera. The optimal view of the ground is 6 meters starting from the end of the blind spot.

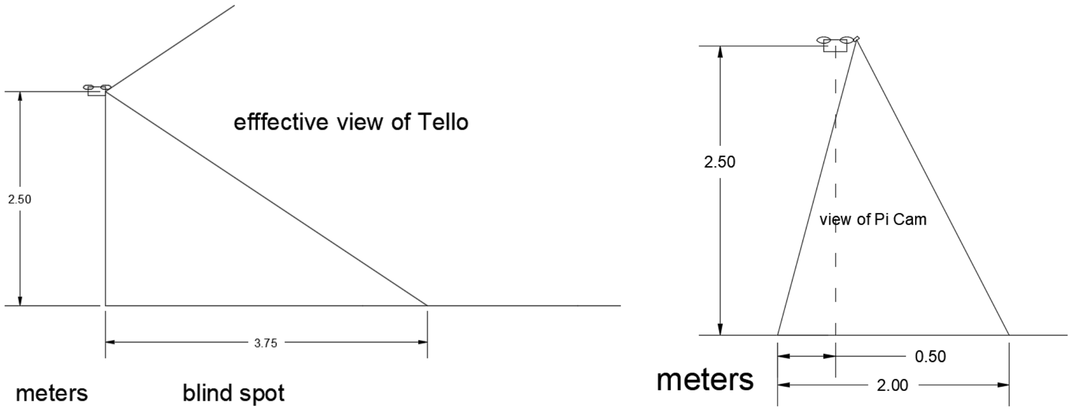

Figure 3-2. Scope of the optimal view of the ground.

Figure 3-3. Raspberry Pi Camera Module v2.

A separate and onboard camera will be used to remove the necessity for the usage of the drone's camera through WiFi. This will eventually reduce unnecessary delays as this will be wired connected. Also, the blind spot disadvantage of the drone camera will be avoided by this addition.

The Raspberry Pi Camera V2 is a compact and versatile camera module designed specifically for Raspberry Pi boards. It has an 8-megapixel sensor for images and 1080p30, 720p60, and VGA90 video captures. Its small form factor and low power consumption will make it ideal for embedded projects. Using this camera will remove the necessity for the usage of the drone's camera through WiFi. This will eventually reduce unnecessary delays as this will be wired connected.

Figure 3-4. Pi Camera 22-pin to 15-pin adaptor.

An adaptor will be used to connect the pi camera with 15 pins for CIS1 to the Raspberry Pi Zero 2 W with CIS2 since this microcomputer has a 22-pin camera slot.

3.3. Raspberry Pi Zero 2 W

Raspberry Pi Zero 2 W will be used as the primary computing device and the center of processing. The processing of the image captured by the camera, the data management, the PID construction, and the control that will be sent to the drone will be implemented with this microcomputer.

Figure 3-5. Raspberry Pi Zero 2 W without pins.

3.4. Power Setup

A rechargeable Li-Po battery will be used to power the microcomputer to avoid draining the battery of the drone quickly. This will also ensure enough current for all the operations. The description of the battery includes 3.7V, 1200 mAh, 50x34x6mm, and its weight is about 30 grams.This is the most suitable for drone attachment due to its shape and weight.

Figure 3-6. LiPo battery.

The required voltage for the Raspberry Pi Zero W is 5V. To regulate the voltage, a 5V step up module will be used. This will also be used for charging.

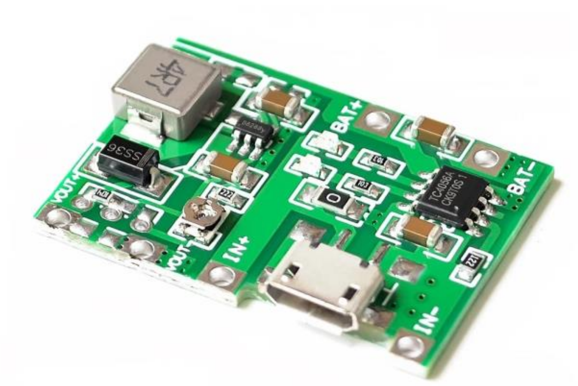

Figure 3-7. 5V step up and charging module.

A rocker switch will be used for turning on and cutting the power from the battery. This will also be used as the initiator for starting the program.

Figure 3-8. Rocker Switch.

3.5. Python Programming Language

Python is a high-level, general-purpose programming language with key features of versatility, open-source, large standard library, and portability. Libraries that will be used for this experiment include OpenCV that will detect ArUco markers and estimate the frame poses, and *tellopy* that will enable the communication between the computer program and the Tello drone. Lastly, its high portability will make it available and compatible to various platforms and operating systems including Linux in Raspberry Pi Zero W. It will be the main facilitator for the whole system.

3.5.1. DJITelloPy Library

The Python library, *djitellopy* will provide a simplified but rich interface for controlling the DJI Tello drone using the computer. With it, the program will connect to the drone and perform various tasks, including taking off, landing, sending rc commands for moving the drone, and querying the altitude and battery percentage of the drone.

3.5.2. OpenCV Library

OpenCV, or Open-Source Computer Vision Library, is a comprehensive computer vision and machine learning software library. It will be used in reprocessing the image received from the drone, detecting the markers, and determining their poses on the frame.

3.5.3. Simple-PID Library

Proportional-Integral-Derivative (PID) control is a widely used feedback control mechanism employed in various engineering applications. This control system relies on three components to regulate the output of a system by minimizing the error between the desired and actual values. This tool will be used to determine the velocities required to move the drone by taking the margin between the desired and actual frame location of the markers. The library that will be used is simple-pid.

3.6. ArUco Markers

The size of the ArUco markers will be 0.24 meter or 24 centimeters since it is the optimal size for a 3-meter altitude flight. The positioning of the markers will be aligned to the flight path. Considering the scope of view in the image, the maximum distance between the markers will be 1.4 meters to ensure that there will be at least one marker at each frame for each segment of the path. To be more readable at longer ranges and to handle possible low image resolution, the dictionary that will be used is 4x4. The materials of each marker consist of an illustration board for the base and the white segments while the black segment will be black felt papers to avoid misreading due to reflections.

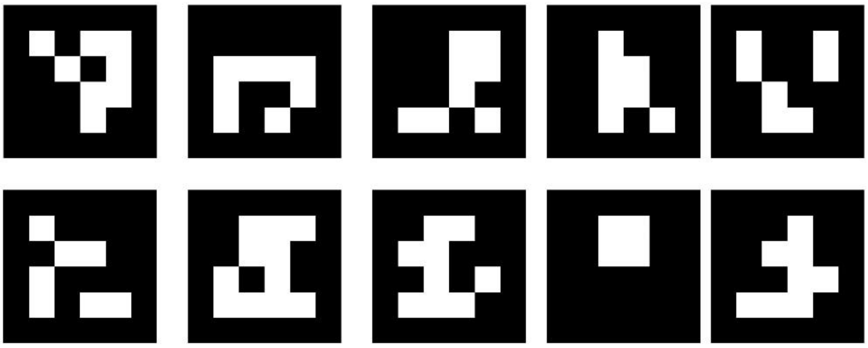

Figure 3-9. 4x4 ArUco markers from IDs 0-9 (top-left to bottom-right).

3.7. Navigation Path Patterns

The navigation will be implemented on a room with a flat floor measuring at least 5 by 6 meters. The maximum angle of the next marker will be 90 degrees from the path line of the previous marker. It is to avoid having no markers at the viewing angle. There will be 2 paths for this experiment. The first one will be made to be simple because it will only be used for PID tuning. The second one will be used to test the setup itself. Too much complication will be avoided because it might pose difficulties in the pilot's control during the manual navigation in training.

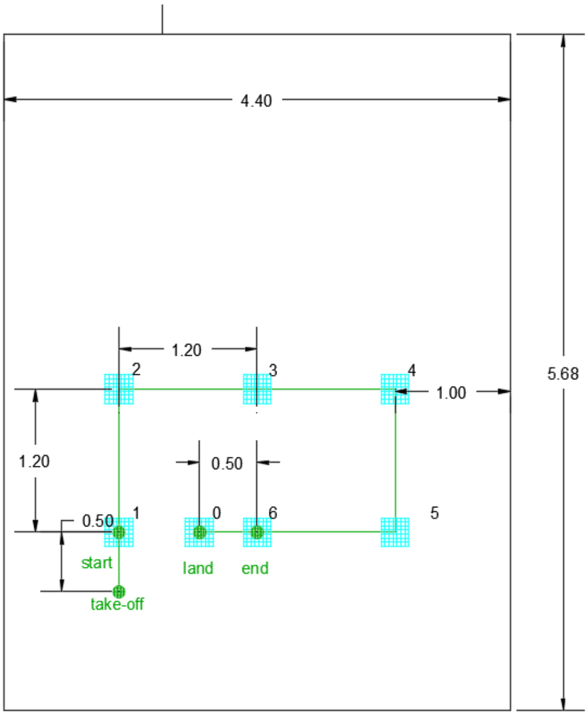

Figure 3-10. Navigation path 1.

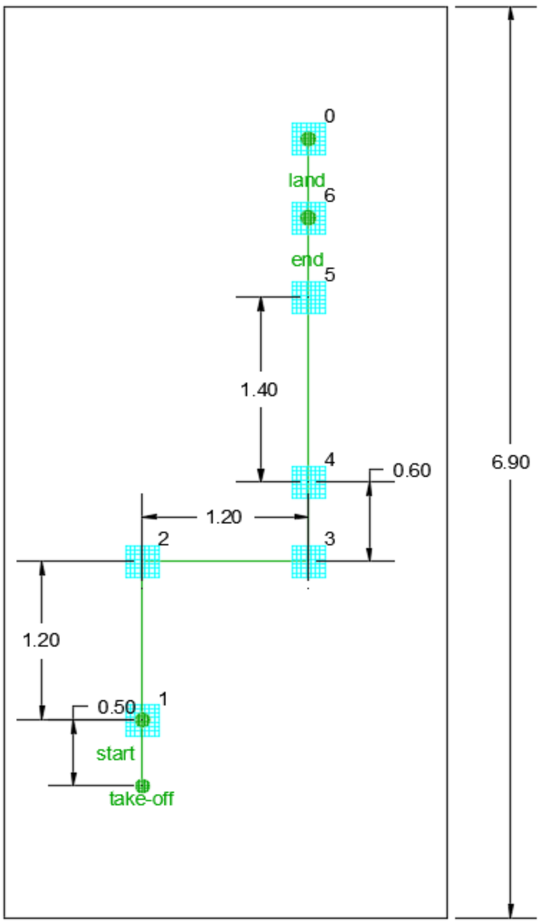

Figure 3-11. Navigation path 2.

3.8. Venues of the Experiment

The preferred venue for experimentation will be indoors since the environment should be a controlled setup. The outdoor factors, primarily the wind, will be avoided. The venue will be the indoor part of an event hall in Bay, Laguna.

3.9. Software and Navigation Setting Up

The setup of software will be discussed in this section. This includes the installation and the pipeline for the navigation where the codes will follow.

3.9.1. Software Installations and Setup

The necessary software and libraries will be installed to Raspberry Pi Zero 2 W primarily including OpenCV, DJITelloPy, and SimplePID. The different versions of each component will be tried and tested to determine their compatibility. The minimum purposes will be checked as part of the functionalities of each library. Peripherals such as monitor and keyboard will be temporarily used for interfacing with the Raspberry Pi Zero 2 W.

The connectivity and controllability of the Tello drone will be tested. These commands will be minimum to taking off, movement by specific velocity, altitude determination, and landing. Getting the battery percentage will also matter to ensure effective flight. The movement by specific velocity will be in terms of individual velocities for x, y, and z axes plus the rotation along the y-axis.

The OpenCV will be ensured to have the functionality of ArUco marker detection and limiting the frame resolution right from the start to maximize the speed capability. Additional features also include the in-frame labeling and annotation to review the flight for better visualization for adjustments.

Bluetooth control via a remote smartphone will be utilized to ensure the safety of the drone by forcing cancellations of commands and emergency landing. It will also be used to monitor the status of the drone flight processes.

A virtual environment will be used for the program. Using this will significantly increase the speed of the system.

3.9.2. Movement Setup

The yaw-pitch-roll-throttle will not be considered as the main definition for movement. Instead, the way of movement of the drone will mimic the manual control via remote control which is based on 2 analog sticks. These sticks were for the upward-downward movement, left-right hover, forward-backward hover, and left-right rotation. The “send_rc_command” of DJITelloPy will take the inputs ranging from -100 to 100 for each axis of movement and rotation. The library will convert these inputs into proper yaw-pitch-roll-throttle intensity and duration to achieve the desired movement. This command will be sent to the drone from the RPi using wifi connection.

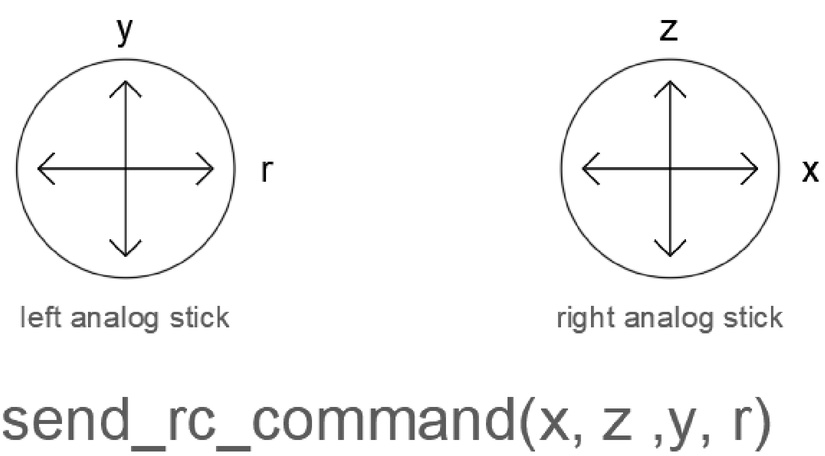

Figure 3-12. Analog stick and rc command.

The direction of the stick determines the velocity or the intensity of speed along the axis. Negative stick direction including the down and left indicates also the negative direction of movement. The center is a (0, 0) value. This will repeat at every frame refresh of the system or at each iteration in the main loop.

3.10. Navigation Setup

The program will start after the raspberry pi is turned on. The bluetooth connectivity with a smartphone will be used to monitor and facilitate the experiment processes such that the raspberry pi will report updates to the smartphone. This will be helpful to the researcher's time in cases where unexpected errors will occur. At any moment during the run of the program, the researcher can send a stop command to break the operation and land the drone.

The main loop in the program will start after the successful takeoff. For each iteration of the loop, an image will be captured and the movement command will be sent to the drone.

The drone will be instructed to first fly to the desired altitude which is 3 meters. This is achievable by sending an upward or downward movement based on the altitude measured by the IR sensor using the "get_height()" command. Afterwhich, the horizontal movement will begin.

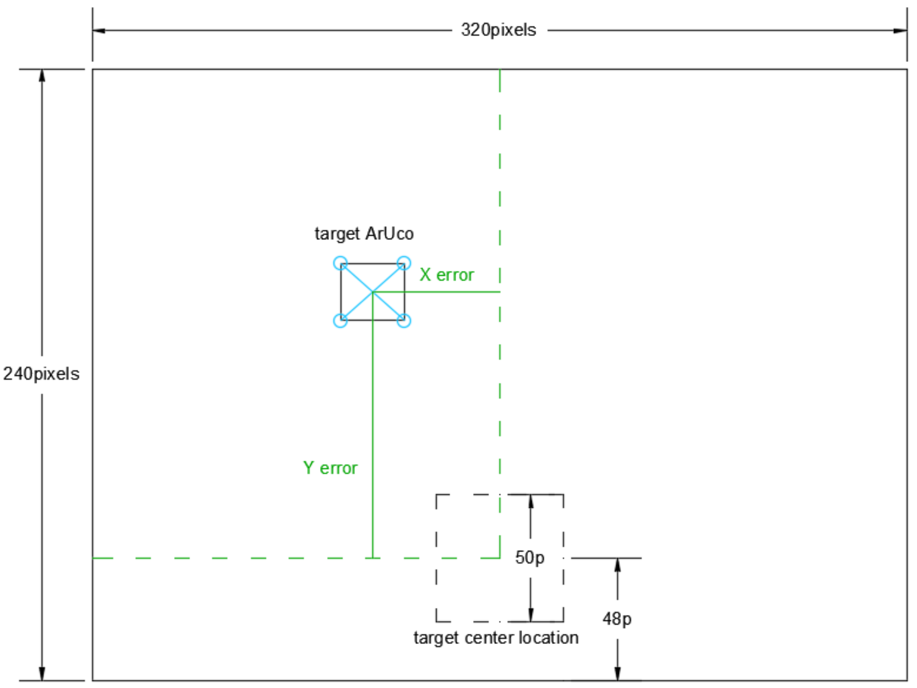

Figure 3-13. Corner center and target.

Markers will be detected at every iteration of the main loop using the *detectMarkers* command. The location of corners of the target marker relative to screen pixels will also be determined using this command. The movement will be done by aiming to relatively move the center of the marker to a desired area in the screen. In Figure 3-10-1, the y-axis refers to the z-axis in actual coordinates. There will be a threshold of 25 pixels in accepting the placement of the target marker.

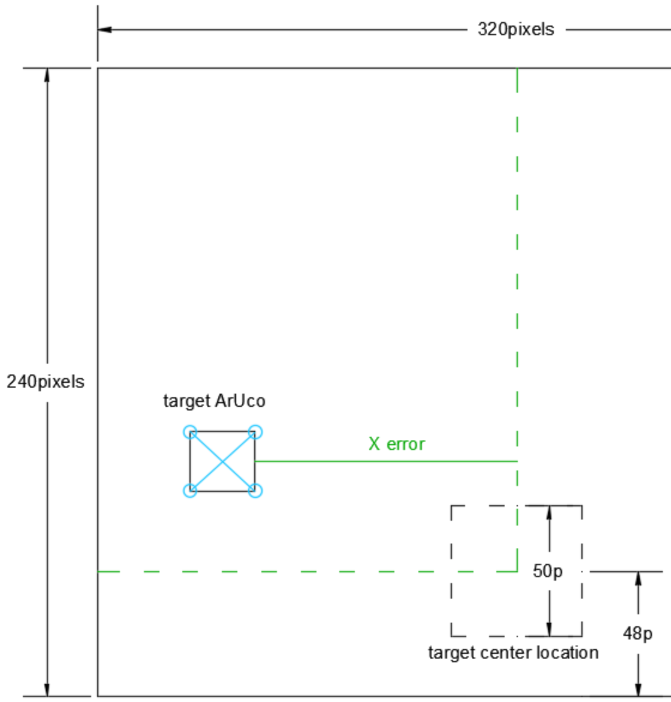

Figure 3-14. Phase 1 Visualization.

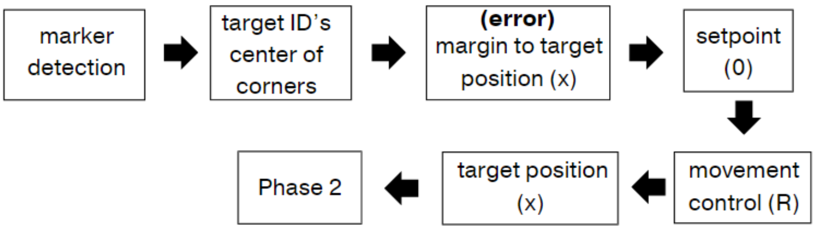

Figure 3-15. Phase 1 Process Flow Chart.

The horizontal movement of the drone towards the target marker will have 2 phases. The first phase is the alignment of the drone to the target marker. The drone will be facing the target marker by rotational movement particularly in the yaw direction. This phase will end after the drone is facing towards the marker along the y-axis.

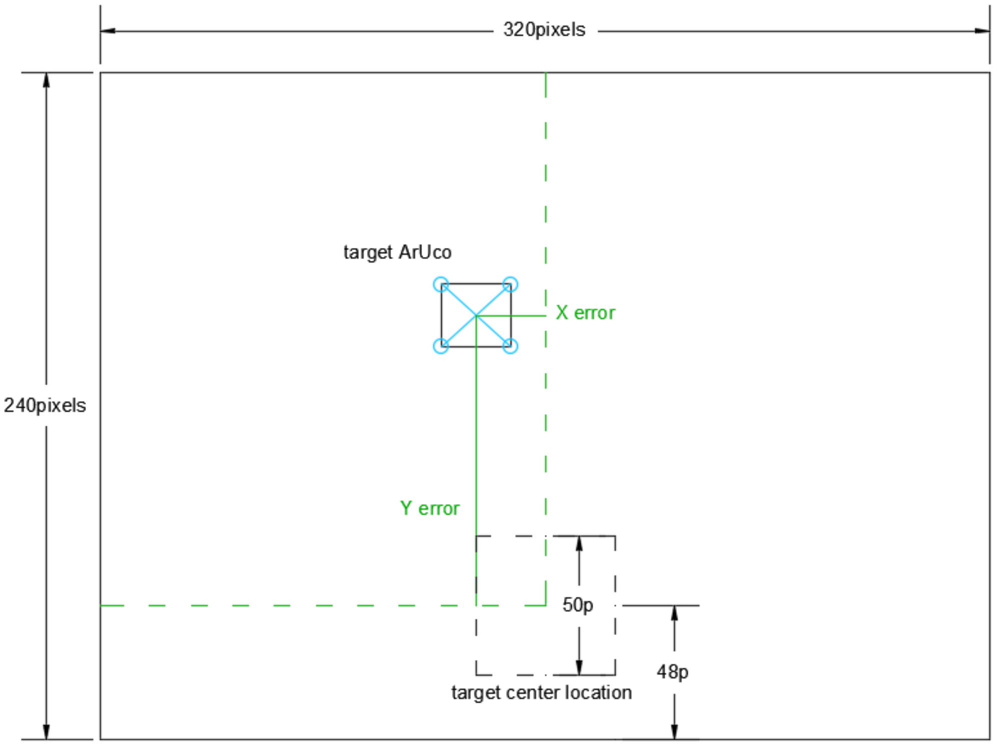

Figure 3-16. Phase 2 Visualization.

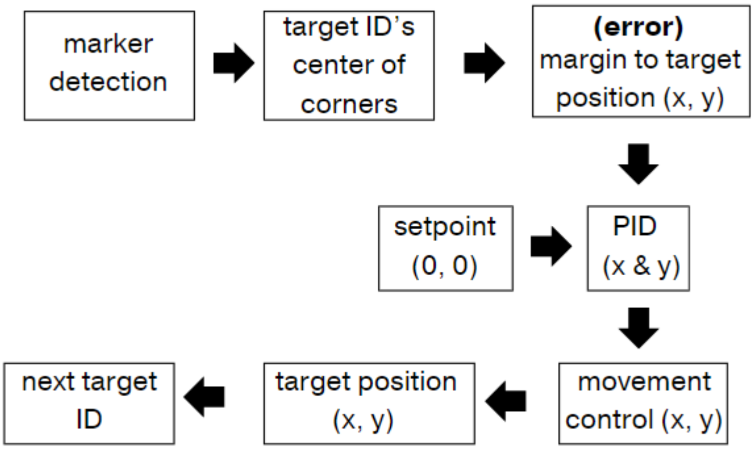

Figure 3-17. Phase 2 Process Flow Chart.

The second phase will include hovering the drone horizontally in terms of forward-backward and leftward-rightward simultaneously. Even though the drone will be ideally facing the marker after the 1st phase, there can be offsets in forward-backward movement which will be corrected by the leftward-rightward movements.

After the marker is placed within the threshold of target location, the phase is reset to 1 for the next target marker.

The PID will be used for smoothening the movements during the 2nd phase. There will be 2 separate PID systems for x (right-left) and y (forward-backward) movements. The errors will be the difference between the target x and y locations and the center of corners of the target marker. The setpoint is 0 which will indicate the same spot of target and actual. The PID will reset every time the marker is placed on the target location. The movement during the 1st phase or the rotation will not have a PID because the rotation of the drone is very responsive and not overshooting.

The horizontal navigation for the experiment starts after the drone reached the marker ID #1. Then, the succeeding ID's will be targeted. The navigation will end at marker ID #6. It will land after reaching ID #0.

3.11. Binding of Raspberry Pi to the Drone

The RPi setup including the switch, the battery, and the camera module will be placed on an illustration board attached on top of the drone as shown in the Figure 3-18 to avoid obstructing the sensors at the bottom of the drone. These sensors are used by the drone for its Visual Position System (VPS) for its stabilization.

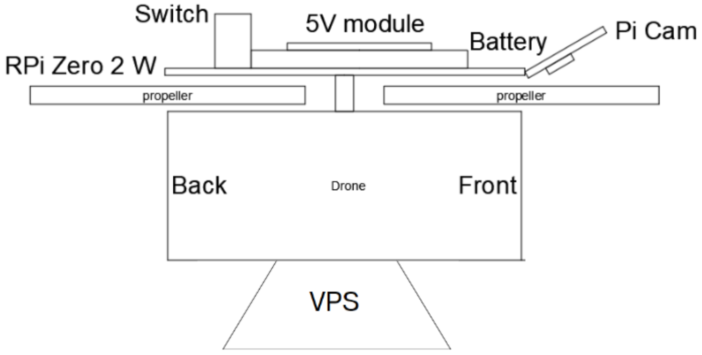

Figure 3-18. Diagram for binding to the drone.

Additional power setup will be installed to provide enough current to the RPi and the camera module and avoid draining the drone camera at shorter time. The power source that will be used is a 3.7-volt LiPo battery with 1200 mAh capacity.

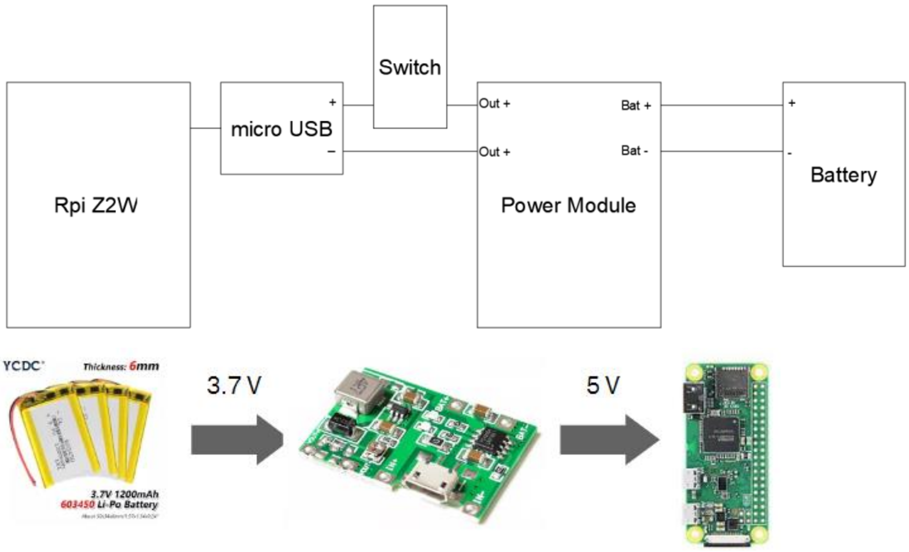

Figure 3-19. Diagrams for Power Connection.

The visual data that will be collected by the camera module will be transferred using a CIS2 adapter connected to the dedicated camera port of the RPi. The images will be processed by OpenCV to extract positions. The PID will translate the positions into movement values. These values will be used by DJITelloPy to send movement controls to the drone connected via WiFi.

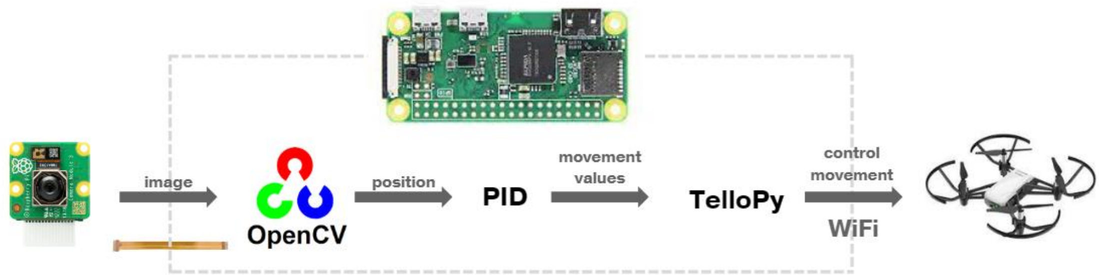

Figure 3-20. Communication between the major components.

3.12. PID Tuning

The tuning of the PID will initially be patterned by the closed loop variation of Ziegler-Nichols Tuning method. It will start by finding the critical value for Kp such that the oscillation will continue infinitely with constant amplitude while setting the values of Ki and Kd to zero. The next step is by selecting the value of Ki for which the oscillation will be removed and the setpoint will be reached at the shortest time. The Kd will be selected for which value the oscillation will be entirely removed.

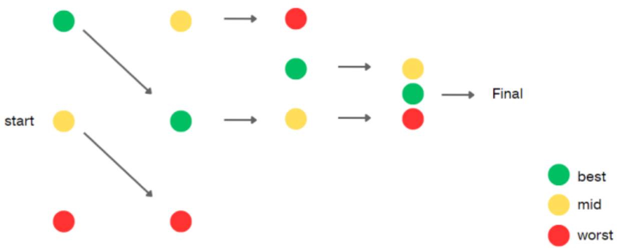

Figure 3-21. Progressive Refinement.

The succession of selection of values to be experimented will be based on progressive refinement. This will be a pattern of finding and minimizing the range by utilizing average. It will start with an initial value. Two points having the same distance from the initial value will be experimented. If the best result among the three points is on an outer value, it will be set as the next initial value with the same range of trial, indicating that the best value may still not be in the current range. If the best value is in the center, this will indicate that the best value may be in the range of the best and the second best value. The next center value will be the average of the best and the second best value with those two values as the range, which will be half of the previous range. This cycle will continue until there will be no significant difference between the results of the best and 2nd best value.

The best value will be determined by the comparison of speed and travel error with the help of visual inspection. The overshooting will be characterized by the overshooting of the drone to the target marker. The infinite oscillation will be characterized by the repetitive going back and forth of the drone with the same forward and backward distance. The starting figures for the proportion, integral and derivative will be 1.0, 0.1, and 0.1 as inspired from the study of Xing, et. al (2018). The gain will start at 1.0 because the maximum speed for scaling will be applied instead for the safety of the drone. Also, this will offer flexibility for different altitudes for future uses.

3.13. Software Optimization and Adjustments

The different optimizations will be experimented to increase the performance of the system in terms of loading and execution time. These will include trying out different libraries for each functionality and trimming the dependencies. The functionalities include capturing the images, optimal resolution and refresh rate, and movement decision making.

3.14. Data Collection

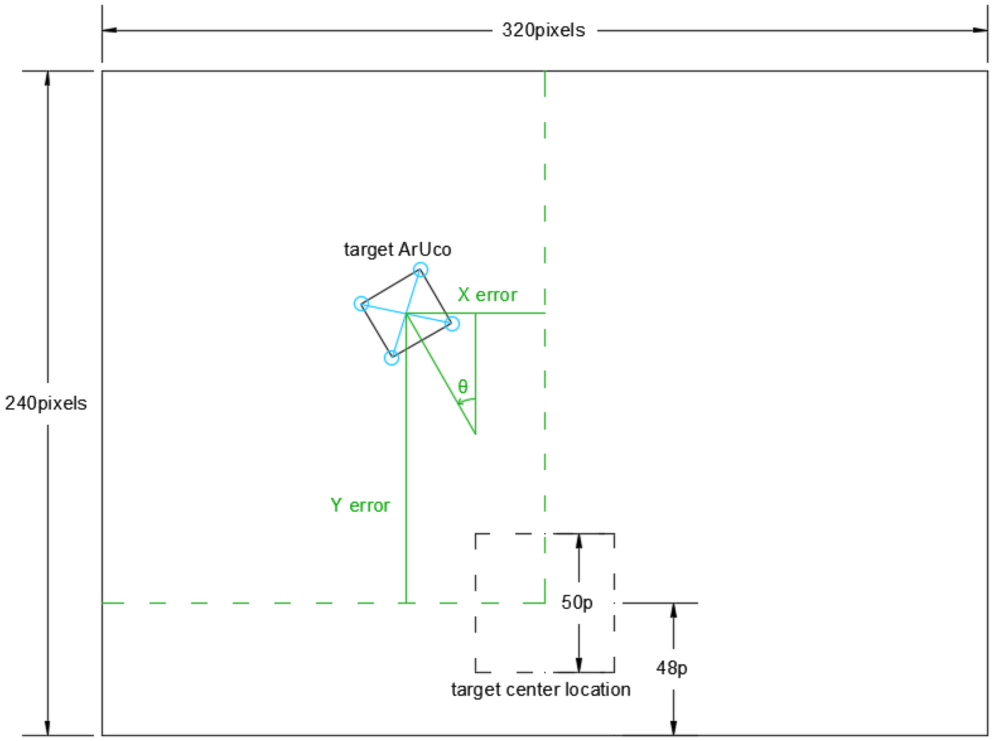

Figure 3-22. Marker orientation and position data gathering.

The data will be collected throughout the flight starting from reaching the marker ID number 1 until reaching the 6th marker. At each frame cycle, the relative position of the drone to the markers through the x and y axes, the angle of the marker relative to the drone, the target marker ID will be recorded, the time interval to the last frame, and the current phase. This will give the distance of the drone from the marker, the orientation of the drone, the location of the drone in the path, the time of flight and instantaneous velocity, and the current phasing, respectively.

Table 3-1. Output CSV File Column Contents.

Frame	X (pixels)	Y (pixels)	Angle (deg)	Time (s)	Target ID	Phase
#1						
#2						
#3						
…						

These outputs will be recorded in a CSV file, with each frame represented in a row and each parameter represented in a column. This will be used for faster data post processing.

3.15. Evaluation

The metrics that will be concerned are the inference time, the travel time, and the offset to path. The inference time will be the time a frame consumes for capturing image, processing movements, and sending commands to the drone. There will be separate time records for image capturing, marker detection, and PID computation. The target minimum inference time will be 0.1 second. The travel time will span from the time the drone reached the 1st marker up to the 6th marker. The travel of the drone from the last marker to the marker ID #0 will be excluded because that will only be for landing. The offset to the path will be measured by recording the location of the markers for each frame and then deriving the position of the drone for each time for mapping. The offset along the x-axis will be measured when the drone is already facing the marker. The offset along the y-axis will be measured once the drone has passed the marker which will be the overshooting.

These will be compared between the different variations of PID including PID, PI, and PD. The best application will be recommended for each metric.

3.16. Flow Chart

Below is the flowchart that will guide the experiment process. This will be used to attain the objectives.

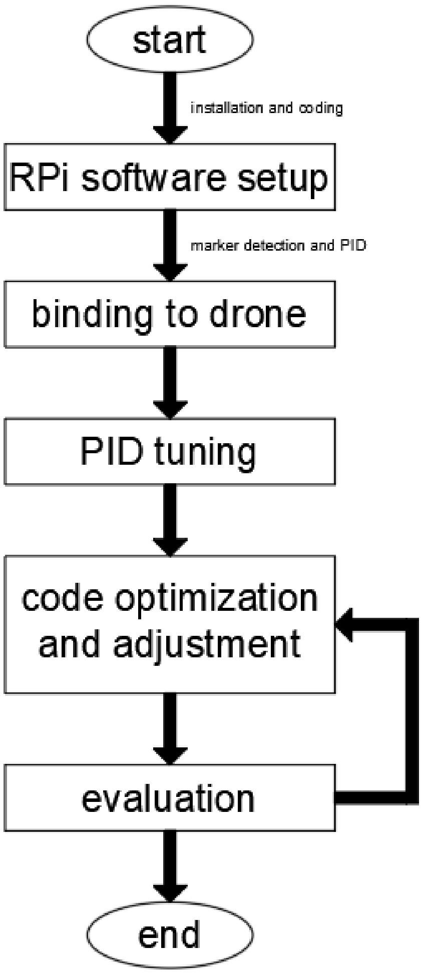

Figure 3-23. Flow Chart of the Study.

4. RESULTS AND DISCUSSION

In this section, the results obtained from the study will be presented and discussed. This will include the design of the PID controller for navigating a horizontal path pipeline based on the ArUco markers positions relative to the drone, the deployment of the PID control algorithm on the Raspberry Pi Zero 2 W attached to the drone, and the evaluation of the performance of the setup in a real-world environment using inference time, travel time, and path error.

4.1. Design of the PID controller

The PID controller was designed by using the relative position of the ArUco markers on the frame of the image. The PID controller was able to generate speed controls for the x and y axes based on the relative position of the markers in the screen. The drone was able to generate controls for itself by only relying on the visual data and not from external signals such as GPS.

The results from the progressive refining of the constants will be discussed in this section. The green cells in the table represent the best value for the respective iteration. The yellow cells represent the second best while the red cells represent the last. For every constant, the drone is initially 1 meter from the marker except for Kpx where the marker is initially 30 centimeters away because that is the distance along the x-axis is greatly lessened after the 1st phase of navigation where the drone is facing the marker. The pixel measurements were converted into centimeters by using height ratio.

Table 4-1. Difference in Oscillation Amplitude for 5 Cycles and Constants for Kpx.

		Initial	1st	2nd	3rd	4th	5th	6th	7th	8th	Final
Kpx	Upper		2	3	2	2	1.75	1.5	1.625	1.5	
	Mid	1	1	2	1.5	1.75	1.5	1.375	1.5	1.4	**1.5**
	Lower		0.01	1	1	1.5	1.25	1.25	1.375	1.375	
Osc Diff (cm)	Upper		23.5	35.2	23.5	23.5	19.7	12.3	14.2	**12.3**	
	Mid		30.7	23.5	12.3	19.7	12.3	13.6	12.3	**13.1**	12.3
	Lower		-	30.7	30.7	12.3	15.4	15.4	13.6	13.6	

The amplitude of the first oscillation is compared to the amplitude of the last oscillation. The trial stopped when the difference of the results of the best value and the second value was less than a centimeter which are 12.3 cm and 13.1 cm. The value of Kpx started at 1.0 and its final value is 1.5.

Table 4-2. Difference in Oscillation Amplitude for 5 Cycles and Constants for Kpy.

		Initial	1st	2nd	3rd	4th	5th	6th	Final
Kpy	Upper		2	2	1.5	1.5	1.25	1.25	
	Mid	1	1	1.5	1	1.25	1	1.125	**1**
	Lower		0.01	1	0.5	1	0.75	1	
Osc Diff (cm)	Upper		19.8	19.8	15.3	15.3	11.2	11.2	
	Mid		8.6	15.3	8.6	11.2	8.6	**9.5**	8.6
	Lower		-	8.6	17.5	8.6	16.7	**8.6**	

The amplitude of the first oscillation is compared to the amplitude of the last oscillation. The trial stopped when the difference of the results of the best value and the second value was less than a centimeter which are 8.6 cm and 9.5 cm. The value of Kpy started at 1.0 and its final value is still 1.0.

Table 4-3. Time For Steady State After the First Half-Oscillation and Constants for Ki.

		Initial	1st	2nd	3rd	4th	5th	6th	7th	Final
Ki	Upper		0.2	0.2	0.15	0.15	0.125	0.125	**0.125**	
	Mid	0.1	0.1	0.15	0.1	0.125	0.1125	0.1188	**0.1219**	**0.12**
	Lower		0.01	0.1	0.05	0.1	0.1	0.1125	**0.1188**	
t before steady st (s)	Upper		10.2	10.2	9.1	9.1	4.4	4.4	4.4	
	Mid		7.8	9.1	7.8	4.4	5.1	3.9	3.4	3.4
	Lower		-	7.8	10.6	7.8	7.8	5.1	3.9	

The duration when the drone first passed the marker until it stopped oscillating was measured in seconds. The trial stopped after the difference of the values of the constant is at the thousandths. The final value of the Ki was rounded off to 0.12.

Table 4-4. Time For Steady State from the Start and Constants for Kd.

		Initial	1st	2nd	3rd	4th	5th	6th	Final	
Kd	Upper		0.2	0.2	0.2	0.175	0.15	0.1625	0.15	
	Mid	0.1	0.1	0.15	0.175	0.15	0.1375	0.15	0.14	**0.15**
	Lower		0.01	0.1	0.15	0.125	0.125	0.1375	0.1375	
t before steady st (s)	Upper		8.9	8.9	8.9	7.5	4.7	5.4	**4.7**	
	Mid		11.7	4.7	7.5	4.7	5.2	4.7	**5.0**	**4.674**
	Lower		-	11.7	4.7	5.9	5.9	5.2	5.2	

The duration from when the drone started moving towards the marker until it stopped was measured in seconds. The trial stopped when the difference between the best result was less than 0.5s.

Table 4-5. Final Constants for PID

	Kpx, Kpy	**Ki**	**Kd**
PID	1.5, 1.0	0.12	0.15
PI	1.5, 1.0	0.12	-
PD	1.5, 1.0	-	0.15

The final values for Kpx, Kpy, Ki, and Kd were 1.5, 1.0, 1.12, and 1.5, respectively. The drone did not oscillate at PI and PD because during the 2nd phases, the drone was shifting to the next target ID after it passes the target location without being necessarily at the steady state and the threshold of 25 pixels for acceptance.

The Figure 4-1-2 and Figure 4-1-3 showed the actual footage of the navigation through the RPi camera. It is shown that the distance along the x and y axes resulted in movement control along the x and y axis. During the 1st phase, the movement was only by rotating as seen in Figure 4-1-2.

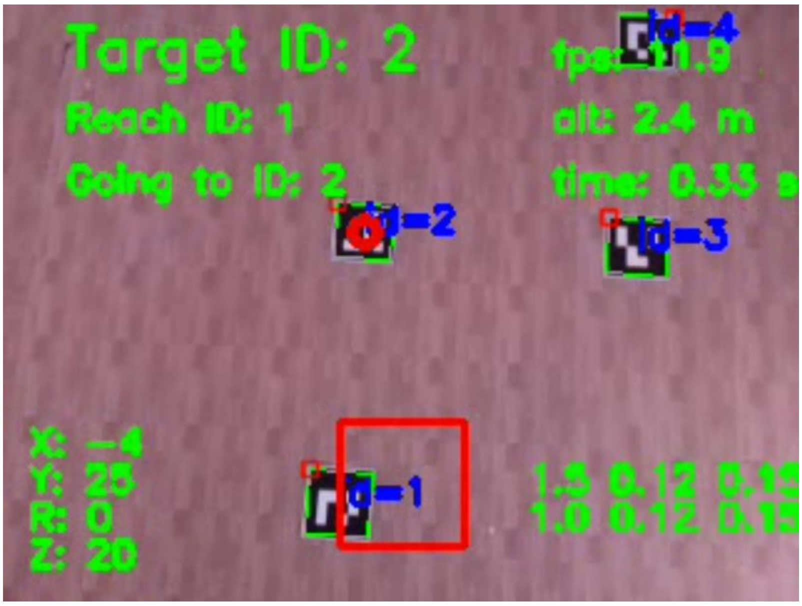

Figure 4-1. Actual Sample Navigation Footage from Drone (Path 2, Phase 2).

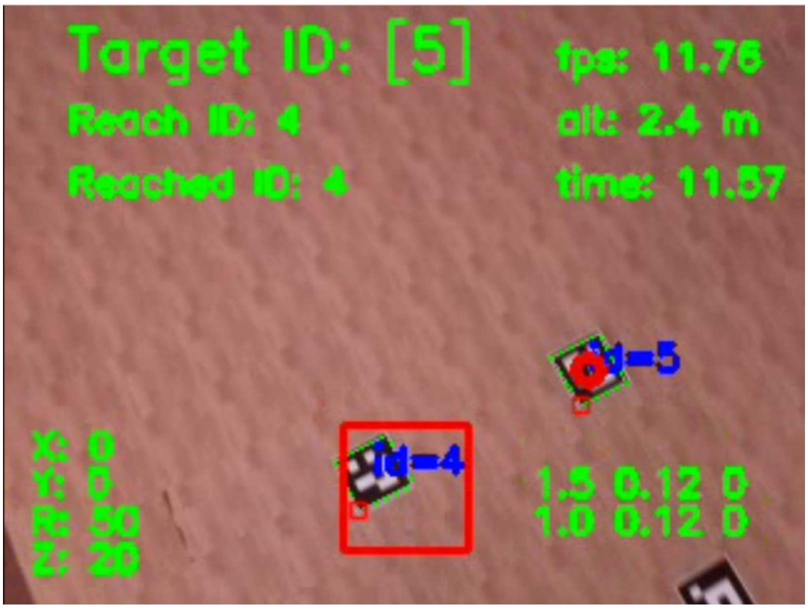

Figure 4-2. Actual Sample Navigation Footage from Drone (Path 1, Phase 1).

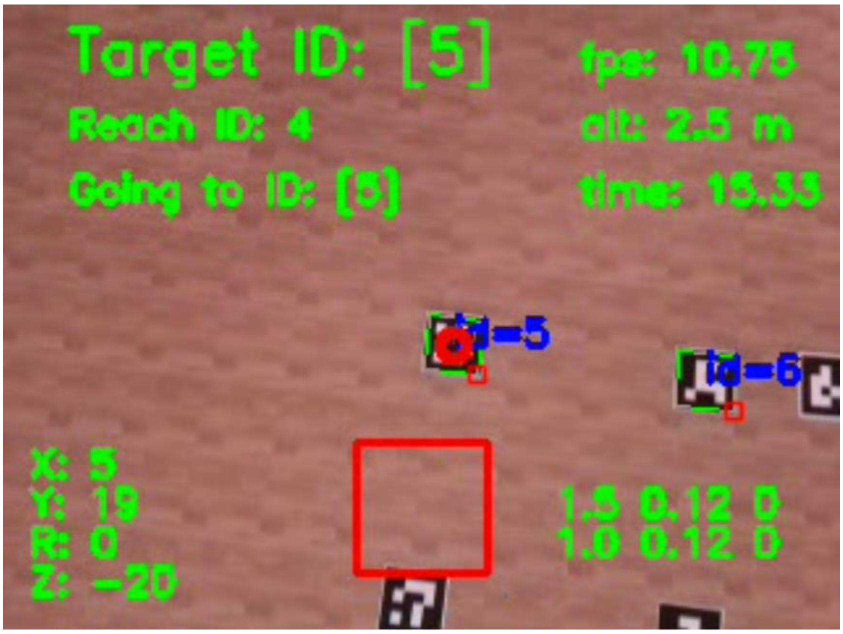

Figure 4-3. Actual Sample Navigation Footage from Drone (Path 1, Phase 2).

4.2. Attachment of Raspberry Pi Zero 2 W to the Drone

The Raspberry Pi Zero 2 W and the power setup has been attached to the drone. The actual whole setup is shown in Figure 4-4.


Figure 4-4. Actual setup of binding the RPi to the drone.

The PID control algorithm has been deployed on the Raspberry Pi attached to the drone. The actual setup of the RPi and the drone is shown in Figure 4-2. It was able to command the drone for movement without needing external computers nor person for decision making. The drone was able to take off and move with balance. The drone carrying the microcomputer and the power setup was able to navigate the path while maintaining the altitude of 2.5 meters. The battery consumption was around 18% per navigation as compared to 8% of no payload flight of the drone. The drone was stable during the flight.

4.3. Evaluation of the Performance of the Setup

This section will focus on the metrics used in the experiment including the inference time, travel time, and path error on a real-world environment. The best gain for x-axis was 1.5 and for y-axis was 1.0. For both axes, the best Ki and Kd was 0.12 and 0.15, respectively.

4.3.1. Inference Time

The results for the inference time are summarized in Table 4-3-1. This shows that the three variations of the PID had almost similar inference times with minimal difference. The mean for the three variations was 82.9 milliseconds. The FPS setting for OpenCV was 12 fps. Higher FPS setting resulted in frame drops which still resulted in 82 ms frame time which was still close to 12 fps.

Table 4-6. Summary of inference time.

Control Algorithm	Inference Time (ms)
PID	83.3
PI	83.2
PD	82.3

The inference time was not comparable to manual flight. However, when the pilot was using the video sent by the drone, there were delays for video input and the control outputs.

The algorithm with fast inference time is suitable for the applications that require higher frame rate processing. These include surveillance and monitoring where the details of the video are important.

4.3.2. Travel Time

The researcher observed that PI was able to gain speed a lot faster than PD but unable to decrease its speed when it comes to stopping. On the other hand, PD took much time to gain speed while it is sensitive when it comes to stopping. PID was in the middle of the two.

The results for the time of travel are summarized in Table 4-3-2. This shows that the PID was the fastest. For path #2, PI was the fastest because there were few turns in there. This means that overshooting in forward direction had less negative effects but rather became an advantage for having momentum for the next marker. PD struggled for path #2 because of the short distances between some of the markers.

Table 4-7. Summary of Travel Time Results.

Control Algorithm	Path 1 Travel Time (s)	Path 2 Travel Time (s)
PID	25.7	23.647
PI	29.3	22.263
PD	27.643 (29.81)	40.061

The time of travel when using manual control was around 21 seconds. The algorithm with the fastest travel time is suitable for applications that require speed of the drone without prioritizing the path. These include payload delivery and search and rescue missions.

4.3.3. Path Error

The path errors were calculated using Root Mean Squared Error (RMSE). The values obtained in pixels were converted into centimeters using the ratio of the actual width to the pixel width for 2.5-meter altitude. The scaling of actual distance in centimeters to pixel distance is 1:1.378

The path errors were relatively large for manual control. This is due to difficult perspective since the pilot was in 3rd person view as compared to the consistent 1st person view of the pi camera. As compared to GPS, the maximum guaranteed accuracy of GPS is within 1 meter. For the height of 2.5 meters, the autonomous drone had a better accuracy of less than half a meter.

The results for the path are summarized in Table 4-3-3. This shows that for left-right navigations, PI was the most accurate while for forward-backward navigation, PID was the most accurate. PID was in the middle.

Table 4-8. Summary of Path Error Results.

PID Variation	Path 1 (cm)		Path 2 (cm)	
	X - error	Y - overshoot	X - error	Y - overshoot
PID	10.12	15.4	4.25	9.41
PI	9.76	17.61	3.89	14.90
PD	12.81	14.91	5.80	5.67

The path navigated by the drone at each frame was recorded and mapped. The markers present, the relative angle, and relative the positions were used to derive the approximate actual location of the drone in the path. The mapping only covered phase 2 because the PID controller was not included in phase 1.

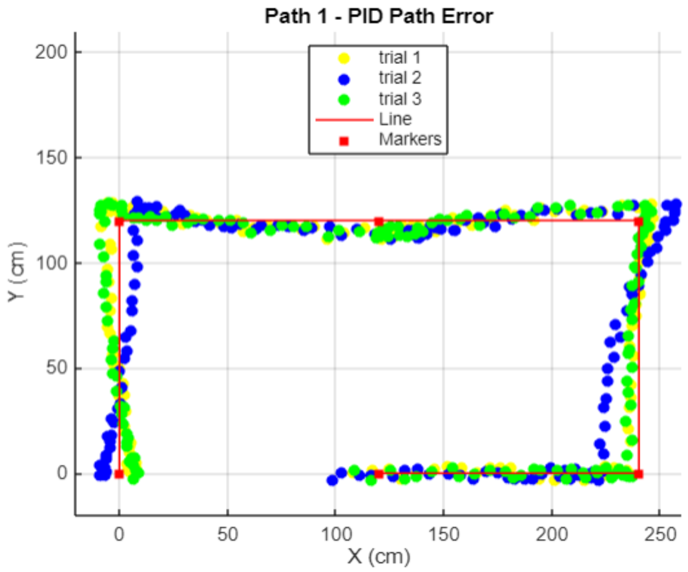

Figure 4-5. PID Mapping for Path 1.

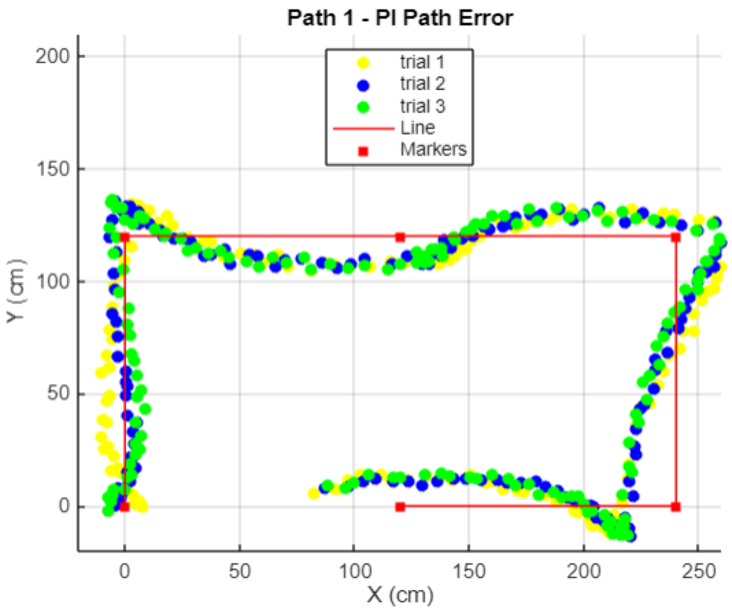

Figure 4-6. PI Mapping for Path 1.

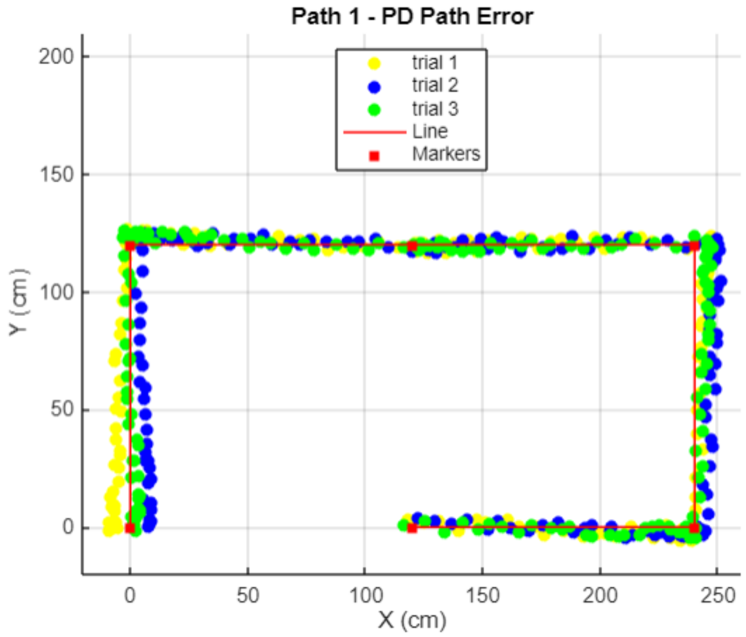

Figure 4-7. PD Mapping for Path 1.

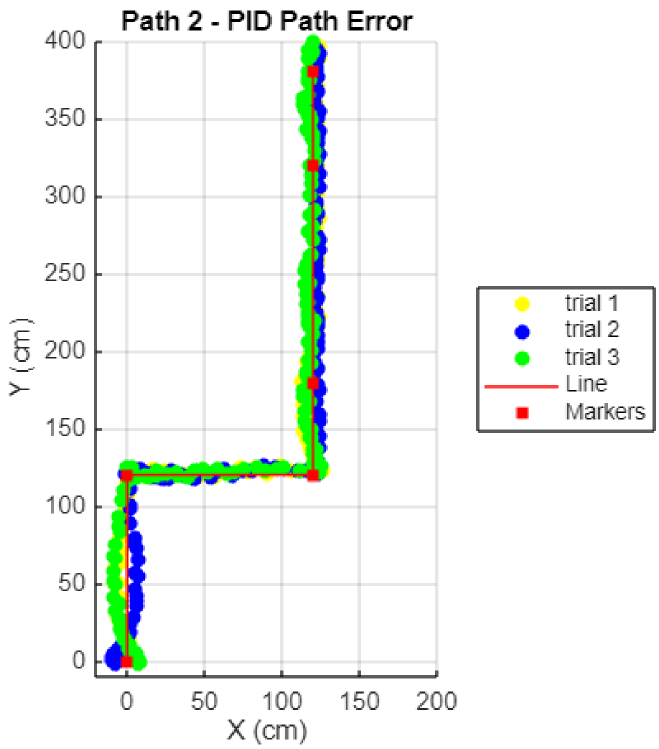

Figure 4-8. PID Mapping for Path 2.

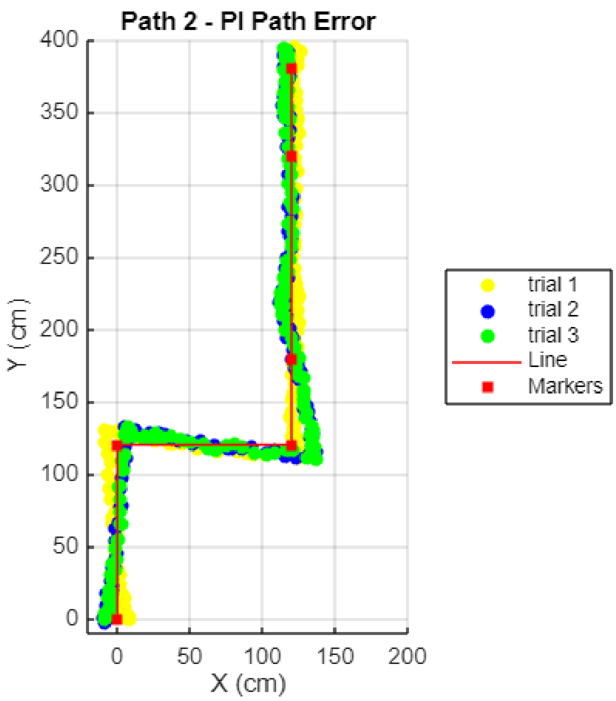

Figure 4-9. PI Mapping for Path 2.

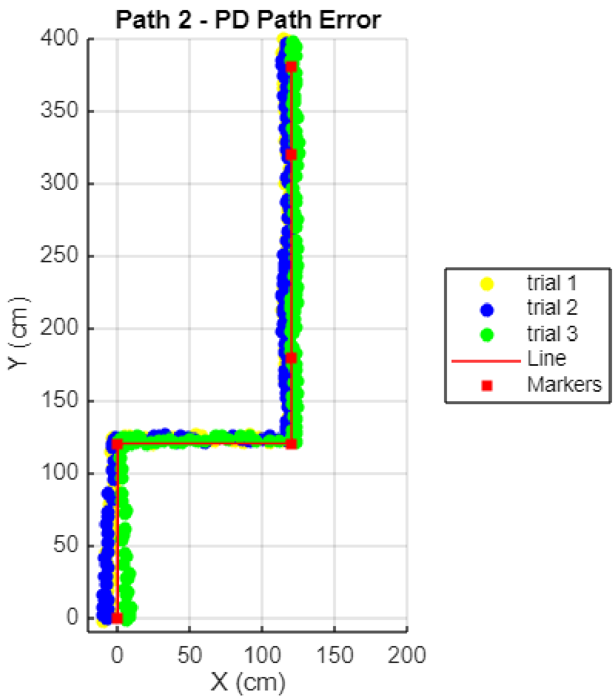

Figure 4-10. PD Mapping for Path 2.

The algorithm with the most accurate path navigation is suitable for applications that require accurate details. These include mapping, surveying, and attachment carrying such as pesticide and fertilizer sprayer which is more for agriculture.

The researcher observed that PD was good for long distances or large values away from the setpoint as it is good for avoiding the overshooting. However, for short distances, it took much time for PD to gain speed that resulted in a large right-left offset. On the opposite, PI was good for short distances due to its sensitivity but it had lower success on stopping itself from overshooting for forward-backward navigation.

5. SUMMARY AND CONCLUSION

This thesis presents an approach to address the challenges of horizontal navigation for non-GPS drones using onboard microcomputers, leveraging image data captured by the drone itself. There were no control signals from other devices to instruct the drone movement along the path aside from the images captured by the Raspberry Pi Zero 2 W setup onboard on the drone. With the recognition of GPS limitations in certain environments, this research embarked on developing an alternative solution centered around ArUco markers and a PID controller implemented on a Raspberry Pi Zero 2 W.

The study's milestones have been reached through experimentation and implementation. First, a PID controller has been developed by leveraging ArUco markers' positional data and facilitated the real-time navigation adjustments. It is aligned with the primary goal of integrating image-based navigation into non-GPS drone systems. Second, the algorithm has been successfully mounted onto the drone's onboard system, particularly utilizing the Raspberry Pi Zero 2 W, which has enabled onboard processing and decision-making while flying. Lastly, the PID, PI, and PD controllers in real-world environments were evaluated considering travel time, inference speed, and path accuracy. This has provided comprehensive insights into the effectiveness of different control strategies, which methods are best for things like making the drone travel faster, deciding quickly, and staying on course.

Several coefficients for PID were tested starting from the recommendations of Xing (2018). The method of comparing the results from raising and lowering the coefficients gave final values of 1.0, 0.12, and 0.15 coefficients for proportion, integral, and derivative, respectively.

The findings showed that the PID controller exhibited an optimal overall performance. Conversely, the PD controller demonstrated its accuracy in distances , ensuring precise and stable path tracking at the expense of time. Lastly, a PI controller emerged as the fastest for longer distances due to its quick gain but slow in adjustments for lowering the speed. In trade, overshooting is more probable.

Ultimately, autonomous navigation proved to be nearly as effective as manual control, even on basic routes. Notably, it outperformed manual control in terms of travel time and path accuracy, indicating its potential superiority. These results highlight the promising capabilities of autonomous navigation systems, even in relatively straightforward situations.

6. RECOMMENDATIONS

These recommendations stem from factors that may have limited the scope and precision of our findings. By addressing these areas in future research, the capabilities and the robustness of non-GPS drone navigation systems can be expanded.

The mild shakiness of the drone while flying midair resulted in small fluctuating readings of the marker's positions. It is recommended to explore advanced image stabilization techniques to further improve navigation accuracy. By ensuring clearer and more stable image capture, the drone's onboard system can better interpret visual data from ArUco markers, leading to more precise navigation.

Incorporating an external system to measure path offset would offer a more accurate assessment of navigation performance. By comparing the drone's perceived path with ground truth data, researchers can refine control algorithms and mitigate deviations from the desired path.

Experimenting with different navigation algorithms beyond straight path-to-marker approaches could yield innovative solutions. Techniques such as path planning algorithms or machine learning-based strategies may offer improved efficiency and adaptability in navigating complex environments.

Scaling up experiments to involve larger drones and more complex paths would provide valuable insights into the scalability and robustness of the navigation system. This expansion would test the system's capabilities in handling varied environmental conditions and navigating through more intricate terrains.

Exploring the integration of additional applications or tasks while navigating could enhance the versatility of the drone system. For instance, incorporating environmental monitoring sensors or payload delivery capabilities alongside navigation tasks could broaden the scope of drone applications and contribute to real-world utility.

By addressing these recommendations in future research endeavors, the capabilities and practicality of non-GPS drone navigation systems can further advance, paving the way for broader adoption and utilization in various industries and applications.

REFERENCES

Arafat, M, et. al (2023). *Vision-Based Navigation Techniques for Unmanned Aerial Vehicles: Review and Challenges*. Recent Advances in UAV Navigation. 7(2), 89. Doi: doi.org/10.3390/drones7020089

Al-Gawda, A. et. al. (2022). *AUTONOMOUS UAV INSPECTION SYSTEM FOR PREVENTIVE MAINTENANCE OF SOLAR FARMS.* Journal of Engineering Science and Technology. https://jestec.taylors.edu.my/Special%20Issue%20SIET2022/SIET2022_09.pdf.

Avola, D. et. al. (2016). *A Practical Framework for the Development of Augmented Reality, Applications by using ArUco Markers.* Proceedings of the 5th International Conference on Pattern Recognition Applications and Methods https://www.scitepress.org/Papers/2016/57558/57558.pdf. Doi: 10.5220/0005755806450654

Bera, B., et. al. (2022). *Access Control Protocol for Battlefield Surveillance in Drone-Assisted IoT Environment*. IEEE Internet of Things Journal. 7(4), pp 2708-2721. Doi: 10.1109/JIOT.2020.3049003

Brodnevs, D. and Kutins, A. "Requirements of End-to-End Delays in Remote Control Channel for Remotely Piloted Aerial Systems," in IEEE Aerospace and Electronic Systems Magazine, vol. 36, no. 2, pp. 18-27, 1 Feb. 2021, doi: 10.1109/MAES.2020.3039853.

Brunner, G., et. al. (2019). *The Urban Last Mile Problem: Autonomous Drone Delivery to Your Balcony.* 2019 International Conference on Unmanned Aircraft Systems (ICUAS). https://ieeexplore.ieee.org/abstract/document/8798337. Doi: 10.1109/ICUAS.2019.8798337

Bry, A., & Roy, N. (2011). Rapidly-exploring random belief trees for motion planning under uncertainty. Proceedings of the 2011 IEEE International Conference on Robotics and Automation (ICRA), 723-730. https://doi.org/10.1109/ICRA.2011.5980398

Cervantes, E. De la Torre, G., and Ramallo, M. (2016). *Workload perception in drone flight training simulators.* Computers in Human Behavior. (64). pp 449-454. Doi: 10.1016/j.chb.2016.07.040

Duch, P. and Jaworski, T. "Enriching Computer Science Programming Classes with Arduino Game Development," 2018 11th International Conference on Human System Interaction (HSI), Gdansk, Poland, 2018, pp. 148-154, doi: 10.1109/HSI.2018.8430994.

De Cubber, G., De Smet, H., and Doroftei, D.(2020). *Reducing Drone Incidents by Incorporating Human Factors in the Drone and Drone Pilot Accreditation*

Process. Advances in Human Factors in Robots, Drones and Unmanned Systems. pp 71–77. Doi: 10.1007/978-3-030-51758-8_10

Droeschel, D., Nieuwenhuisen, M., Beul, M., Houben, S., & Behnke, S. (2016). Multilayered mapping and navigation for autonomous micro aerial vehicles. Journal of Field Robotics, 33(4), 451-475. https://doi.org/10.1002/rob.21612

Eris, H and Cevik, U. (2019). *IMPLEMENTATION OF TARGET TRACKING METHODS ON IMAGES TAKEN FROM UAV (UNMANNED AERIAL VEHICLES*. 2019 IEEE 17th World Symposium on Applied Machine Intelligence and Informatics (SAMI). https://acikbilim.yok.gov.tr/bitstream/handle/20.500.12812/124145/ yokAcikBilim_10216547.pdf. Doi: 10.1109/SAMI.2019.8782768

Faessler, M., Kaufmann, E., & Scaramuzza, D. (2018). Differentiable Kalman filter for learning optimal sensor fusion strategies. Proceedings of Robotics: Science and Systems (RSS). https://doi.org/10.15607/rss.2018.xxiv.033

Flemons, K., et al. (2022). *The use of drones for the delivery of diagnostic test kits and medical supplies to remote First Nations communities during Covid-19*. American Journal of Infection Control. 50 (8), pp 849-856. Doi: 10.1016/j.ajic.2022.03.004

Forster, C., Carlone, L., Dellaert, F., & Scaramuzza, D. (2017). On-manifold preintegration for real-time visual-inertial odometry. IEEE Transactions on Robotics, 33(1), 1-21. https://doi.org/10.1109/TRO.2016.2597321

Gayathri Devi, K., et. al. (2020). *Review on application of Drones for crop health monitoring and spraying pesticides and fertilizers*. Journal of Critical Reviews.7(6). Doi: 10.31838/jcr.07.06.117

Ghazi, G., & Voyer, J. (2024). Use of a DJI Tello Drone as an Educational Platform in the Field of Control Engineering. Proceedings of the Canadian Engineering Education Association (CEEA). Retrieved from https://ojs.library.queensu.ca/index.php/PCEEA/article/view/17061

Hashemi-Pour, C. (2023). backpropagation algorithm. *Tech Target.* https://www.techtarget.com/searchenterpriseai/definition/backpropagation-algorithm

Huang, H., et. al. [2019]. *Deep Reinforcement Learning for UAV Navigation Through Massive MIMO Technique*. IEEE Transactions on Vehicular Technology. doi://doi.org/10.1109/TVT.2019.2952549

Jiang, P., et. al. (2022*). A Review of Yolo Algorithm Developments*. Procedia Computer Science. https://www.sciencedirect.com/science/article/pii/S1877050922001363. Doi: doi.org/10.1016/j.procs.2022.01.135

Lee, H., Choi, H. S., Kim, H. J., & Shim, D. H. (2017). Relative localization for formation flight of quadrotors using moving baseline GPS. IEEE Transactions on Aerospace and Electronic Systems, 53(3), 1279-1291. https://doi.org/10.1109/TAES.2017.2671358

Li, Y., & Zhang, B. (2016). Adaptive robust control for quadrotor helicopters. IEEE Transactions on Industrial Electronics, 63(10), 6765-6772. https://doi.org/10.1109/TIE.2016.2577561

Lutkevich, B., and Earls, A. (2019). *Definition: drone (UAV).* TechTarget. (Online). https://www.techtarget.com/iotagenda/definition/drone

Maalouf, A., Gurfinkel, Y., Diker, B., Gal, O., Rus, D., & Feldman, D. (2023). Deep Learning on Home Drone: Searching for the Optimal Architecture. In 2023 IEEE International Conference on Robotics and Automation (ICRA) (pp. 8208-8215). London, United Kingdom: IEEE. doi: 10.1109/ICRA48891.2023.10160827

Madani, T., & Benallegue, A. (2006). Backstepping control for a quadrotor helicopter. Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, 3255-3260. https://doi.org/10.1109/IROS.2006.282538

Mellinger, D., Michael, N., & Kumar, V. (2012). Trajectory generation and control for precise aggressive maneuvers with quadrotors. The International Journal of Robotics Research, 31(5), 664-674. https://doi.org/10.1177/0278364911434236

Miah, A. (2020). *Regulating Drones. Drones*: The Brilliant, the Bad and the Beautiful. Pp 37-64. Doi: 10.1108/978-1-83867-985-920201003

Nagpal, G. S., Singh, G., Singh, J., & Yadav, N. (2018). Facial Detection and Recognition using OpenCV on Raspberry Pi Zero. 2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN). doi:10.1109/icacccn.2018.8748389

Oxford Languages (dictionary).

Shafiee, M. (2017). Fast YOLO: A Fast You Only Look Once System for Real-time Embedded Object Detection in Video. 2017 Journal of Computational Vision and Imaging Systems 3. https://arxiv.org/abs/1709.05943. Doi: doi.org/10.48550/arXiv.1709.05943

Scott, C., and Scott, J. (2017). *Drone Delivery Models for Healthcare*. Global Health IT Strategies.

Sun, J., Yang, S. X., & Peng, K. (2015). A bioinspired approach to multi-UAV formation control. IEEE Transactions on Automation Science and Engineering, 12(3), 812-819. https://doi.org/10.1109/TASE.2015.2411355

Teixeira, B. M., Silveira, L. A., Pereira, G. A. S., & Mendes, R. P. (2018). Autonomous UAV navigation using visual and inertial sensors. Journal of Intelligent & Robotic Systems, 91(1), 49-68. https://doi.org/10.1007/s10846-018-0834-9

The Science of Machine Learning. (n.d.). Tanh Activation Function. *ML Science.* https://www.ml-science.com/tanh-activation-function

Triple M. (2019). DJI TELLO Drone Full Review - Best Beginner Drone for 2020. (YouTube). https://www.youtube.com/watch?v=-87xEi8vGdI

Walker, A., et. al. [2019]. *A Deep Reinforcement Learning Framework for UAV Navigation in Indoor Environments*. 2019 IEEE Aerospace Conference, Big Sky, MT, USA, 2019, pp. 1-14 doi:10.1109/AERO.2019.8742226.

Wang, P. Fan, E., and Wang, P. (2021). *Pattern Recognition Letters*. Pattern Recognition Letters. https://www.sciencedirect.com/science/article/abs/pii/S0167865520302981. pp 61-67. (141). Doi: 10.1016/j.patrec.2020.07.042

Wubben, J., et al. [2019]. *A vision-based system for autonomous vertical landing of unmanned aerial vehicles*. 2019 IEEE/ACM 23rd International Symposium on Distributed Simulation and Real Time Applications (DS-RT), pp. 1-7, doi: 10.1109/DS-RT47707.2019.8958701.
Xing, B., Zhu, Q., Pan, F., & Feng, X. (2018). Marker-Based Multi-Sensor Fusion Indoor Localization System for Micro Air Vehicles. Sensors, 18(6), 1706. doi:10.3390/s18061706

Yu, J. Jiang, et. al. (2021). *Application of a Vision-Based Single Target on Robot Positioning System.* Sensors and Robotics. Doi: 10.3390/s21051829