

**BIO INSPIRED MICRO AERIAL VEHICLE FOR SURVEILLANCE**

**MAJITHA THABASSUM ASHWIN PRABHU M**

**19617108**

**19617103**

**VELS INSTITUTE OF SCIENCE, TECHNOLOGY & ADVANCED STUDIES (VISTAS)**

**FINAL YEAR PROJECT REPORT**

**PHASE - I**

Submitted to the

**SCHOOL OF ENGINEERING**

# DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

in Partial fulfillment of the requirements for the award of the degree of

# BACHELOR OF ENGINEERING

**IN**

# COMPUTER SCIENCE AND ENGINEERING

**VELS INSTITUTE OF SCIENCE, TECHNOLOGY AND ADVANCED STUDIES (VISTAS)**

**CHENNAI – 600 117**

**NOVEMBER 2024**

**BONAFIDE CERTIFICATE**

Certified that this project **“Bio Inspired Micro Aerial Vehicle for Surveillance”** is the bonafide work of **Majitha Thabassum (19617108) Ashwin Prabhu M (19617103)** who carried out the project under my supervision. Certified further, that to the best of my knowledge the work reported here does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE OF THE HOD SIGNATURE OF THE SUPERVISOR

**Dr. K. Kalaivani M.E., Ph.D Mrs. S. Vijitha M.E.,(Ph.D)**

Associate Professor & Head Assistant Professor

Department of CSE Department of CSE

School of Engineering School of Engineering

VISTAS VISTAS

Chennai Chennai

Submitted for the Project Work and Viva - Voce examination held on ……………

at Vels Institute of Science, Technology and Advanced Studies (VISTAS), Chennai – 600 117.

# INTERNAL EXAMINER EXTERNAL EXAMINER

**ACKNOWLEDGEMENT**

It is certainly a great delight and proud privilege to acknowledge the help and support I received from the positive minds around me in making this venture a successful one. The infrastructural support with all kinds of lab facilities have been a motivating factor in this completion of project work, all because of **FOUNDER & CHAIRMAN, Shri. Dr. ISHARI K. GANESH.,** with great pleasure that I take this opportunity to thank him. I am grateful to our **HEAD OF THE DEPARTMENT, Dr. K. Kalaivani.,** for her continuous support for the project, from initial advice and contacts in the early stages of conceptual inception, and through on-going advice and encouragement to this day. I extend our warmest thanks to my **INTERNAL GUIDE**, **Mrs. S. Vijitha**, who has patiently guided and provided us with invaluable advice and help. I am thankful to my Project Coordinator **Mrs. S. Vijitha,** I would like to express my greatest gratitude to our Staff members, Librarian, and Non-teaching staff members of Computer Centre, who have helped and supported me throughout. Last but not the Least, I wish to thank my parents and my friends for their individual support and interest who has inspired me and encouraged me to go in my own way, without them I would be unable to complete this project.

**MAJITHA THABASSUM 19617108**

**ASHWIN PRABHU M 19617103**

**CONTENTS**

TABLE OF CONTENTS

[ABSTRACT 6](#_TOC_250028)

1. [INTRODUCTION 7](#_TOC_250027)
   1. [GENERAL 7](#_TOC_250026)
   2. [PROBLEM DESCRIPTION 9](#_TOC_250025)
   3. [OBJECTIVE 10](#_TOC_250024)
   4. [SCOPE OF THE PROJECT 10](#_TOC_250023)
2. [LITERATURE SURVEY 12](#_TOC_250022)
   1. [GENERAL 12](#_TOC_250021)
   2. LITERATURE REVIEW……………………………………………………………………. 14
   3. [EXISTING SYSTEM 13](#_TOC_250020)
   4. [PROPOSED SYSTEM 14](#_TOC_250019)
   5. SUMMARY……………………………………………………………………………15
3. [SYSTEM DESIGN 15](#_TOC_250018)
   1. [GENERAL 15](#_TOC_250017)
   2. [SYSTEM ARCHITECTURE 15](#_TOC_250016)
   3. [SYSTEM WORKFLOW 17](#_TOC_250015)
   4. [METHODOLOGY 19](#_TOC_250014)
   5. **ALGORITHM** 28
   6. [SUMMARY 22](#_TOC_250010)
4. [SYSTEM REQUIREMENTS 21](#_TOC_250013)
   1. [HARDWARE REQUIREMENTS 21](#_TOC_250012)

4.2 [SOFTWARE REQUIREMENTS 22](#_TOC_250011)

1. **MODULE IMPLEMENTATION** 26
   1. **LIST OF MODULES** 28

**5.2 SUMMARY** 32

1. [SYSTEM IMPLEMENTATION 23](#_TOC_250009)
   1. GENERAL 23
   2. [OVERVIEW OF THE PLATFORM 23](#_TOC_250008)
   3. SOURCE CODE………………………………………………………………………28
   4. RESULTS & DISCUSSIONS…………………………………………………………...33
2. **TESTING** 33
   1. **GENERAL** 33
   2. **TYPES OF TESTING** 33
   3. **SUMMARY** 35

**7 CONCLUSION AND FUTURE WORK** 37

**7.1 CONCLUSION & FUTURE WORK** 37

**7.1.1 REFERENCES** 39

# ABSTRACT

This study describes the design, development, and execution of an unmanned micro aerial vehicle (MAV) inspired by bee and fly flight principles. For flight control and picture capture, the MAV is outfitted with sensors, a camera, and a microcontroller. The MAV's flight and imaging capabilities are managed by a smartphone app that interfaces with the microcontroller using Bluetooth Low Energy (BLE) technology. The goal of this project is to show the possibility of developing and creating a bio-inspired MAV, as well as to examine its possible applications in diverse disciplines. The design and fabrication of the physical structure, the integration of sensors and imaging components, and the programming of the microcontroller and mobile application were all steps in the creation of the MAV. The MAV's design was inspired by bee and fly wing architecture and flight processes. The MAV has a carbon fibre structure that is lightweight, and its wings are operated by servo motors. The MAV is powered by a tiny battery and is outfitted with altitude, orientation, and position sensors. The imaging system is comprised of a camera installed on the MAV that collects images and stores them on a microSD card. This work describes a bio-inspired MAV that was designed and constructed to imitate the flight processes of bees and flies. The MAV is designed to be lightweight and compact, having sensors and imaging capabilities for flight control and image capturing. The MAV displayed stable flying and high-quality imaging capabilities, making it appropriate for a wide range of applications. The usage of a mobile application for flight control and photography gives the user with an intuitive and user-friendly interface. Future studies will include optimising the MAV design and investigating its possible uses in diverse fields.

**Keywords:** Micro aerial vehicle (MAV), Bluetooth Low Energy (BLE).

# INTRODUCTION

# GENERAL

Unmanned aerial vehicles (UAVs), commonly known as drones, have grown in popularity in recent years due to their versatility in a variety of applications. They've been widely employed in military, commercial, and civil applications like surveillance, reconnaissance, search and rescue, agriculture, and photography. One of the key benefits of UAVs is their capacity to reach regions that people find difficult or impossible to access, making them an appealing tool for a variety of activities.

One area where UAVs have sparked substantial attention is in bio-inspired robotics. Bio-inspired robotics is an interdisciplinary research field that studies biological systems in order to create robots that replicate the behaviour and functions of real beings. For several decades, the topic of bio-inspired robotics has been an active area of research, and researchers have been able to construct numerous robots capable of performing activities such as flying, swimming, crawling, and walking, among others.

The micro aerial vehicle (MAV), a small, lightweight flying robot that can manoeuvre in confined places and fly in hard settings, is one of the most popular forms of bio-inspired robots. MAVs are widely utilised in a variety of applications, including surveillance, mapping, search and rescue, and environmental monitoring. They've also been utilised in scientific studies, including as analysing insect and bird behaviour and developing new MAV technologies.

The ambition to create robots that can do jobs that regular UAVs cannot has fuelled the creation of bio-inspired MAVs. MAVs modelled like birds or insects, for example, can have higher flying performance in terms of agility, manoeuvrability, and efficiency. These properties are particularly desirable in a variety of applications, including search and rescue missions, in which MAVs must manoeuvre through crowded and complicated situations.

The goal of this project is to design and build a bio-inspired unmanned micro aerial vehicle modelled after a fly and a bee. The MAV will have wings that are comparable in form and structure to those of flies and bees, allowing it to fly in a way similar to their flying behaviour. The MAV will also have sensors and a camera, allowing it to snap photographs and save them to a mobile device. The MAV's flight will be controlled by a smartphone application that will allow users to control the MAV and examine the camera's images.

The remainder of the paper is structured as follows: The following section is a survey of the relevant literature on bio-inspired MAVs. Section 3 describes the project process, including the design of the wings, sensor selection, and creation of the mobile application. Section 4 describes in full the algorithm used for flight stabilisation and wing programming. Section 5 covers the project's outcomes, including the MAV's flying performance and the camera's images. Section 6 discusses the project's findings and their consequences. Finally, section 7 summarises the project's findings and suggests topics for further investigation.

# PROBLEM DESCRIPTION

Micro aerial vehicles (MAVs) are small drones that are designed to fly and maneuver in tight spaces, making them ideal for surveillance in urban and indoor environments. They are increasingly being used by law enforcement agencies, military, and security personnel for surveillance and reconnaissance missions.

However, there are several challenges associated with the use of MAVs for surveillance. Firstly, the limited battery life of these drones restricts their flight time, which can make continuous surveillance difficult. Secondly, MAVs are highly susceptible to environmental factors such as wind, rain, and snow, which can affect their stability and cause them to crash.

Moreover, controlling MAVs in complex and cluttered environments can be challenging, especially when they need to navigate through narrow passages or fly around obstacles. The risk of collision with people, vehicles, or other objects is also a significant concern, as it can cause damage to the drone and potentially harm people on the ground.

In addition, the use of MAVs for surveillance raises privacy concerns, as it can potentially invade the privacy of individuals who are not involved in criminal activities.

There are also legal and ethical considerations around the use of MAVs, including data protection, accountability, and transparency, which must be addressed to ensure that their use is lawful and ethical.

# OBJECTIVE

* The primary objectives of using micro aerial vehicles (MAVs) for surveillance are to gather information and intelligence, monitor and observe targets or areas of interest, and provide situational awareness for decision-making. Here are some specific objectives of using MAVs for surveillance:
* Obtain real-time video and imagery: MAVs equipped with cameras or other sensors can capture real-time video and imagery of areas of interest, providing valuable information for surveillance and reconnaissance.
* Conduct covert surveillance: MAVs can be used to conduct surveillance operations that are difficult or impossible for ground-based personnel to carry out, such as monitoring high-risk or sensitive areas without being detected.
* Provide situational awareness: MAVs can provide situational awareness by providing real-time video and imagery of a scene, allowing decision-makers to better understand the situation and make informed decisions.
* Conduct search and rescue operations: MAVs equipped with thermal imaging cameras can be used to search for and locate missing persons or identify hotspots in wildfire situations.
* Enhance security and law enforcement: MAVs can assist security and law enforcement personnel in identifying potential threats or criminal activities and monitoring public events for safety purposes.

# SCOPE OF THE PROJECT

The scope of a project involving the use of micro aerial vehicles (MAVs) for surveillance would depend on the specific objectives of the project and the requirements of the mission.

The scope of a project involving MAVs for surveillance would need to be carefully planned and executed to ensure the project meets its objectives and is safe, effective, and sustainable over time.

# CHAPTER 2

# LITERATURE SURVEY

# GENERAL

Unmanned aerial vehicles (UAVs) have received a lot of interest in recent years because of its multiple uses in areas including agricultural, military, photography, and surveillance. The micro aerial vehicle (MAV) is a form of UAV that is distinguished by its small size and light weight. MAVs are suited for inside and outdoor activities where regular UAVs are inconvenient or dangerous to use. They are also inexpensive, making them appropriate for surveillance, inspection, and monitoring applications.

MAV functionality and performance have been improved through study in recent years. The design of MAVs inspired by nature is one topic of interest. The use of biomimicry in MAV design has yielded encouraging results, with MAVs being produced that mimic the flight behaviour of birds and insects. For example, researchers have created MAVs that resemble birds, such as the swift, in order to increase their outside flight capabilities.

Integration of sensors and communication systems to improve MAV operation is another area of focus in MAV design. MAVs can navigate and manoeuvre in complicated situations, avoid obstacles, and stabilise their flight thanks to sensor integration. MAVs can also be outfitted with communication systems that allow them to communicate data in real time, making them appropriate for surveillance and inspection.

Furthermore, research has been performed to improve MAV control and autonomy. One idea is to control the flight of MAVs using mobile devices such as cellphones. This method allows users to manage the flight of MAVs using a simple interface, making them easier to operate for non-experts. Mobile devices can also

be used to capture data and photographs from MAVs, making them appropriate for aerial photography and monitoring.

Overall, the usage of MAVs in numerous applications has demonstrated considerable promise. MAVs inspired by nature and outfitted with sensors and communication technologies have increased in usefulness and performance. In addition, the usage of mobile devices for MAV control and data gathering has made them more approachable to non-experts. As a result, continuous research and development of MAVs will lead to expanded utilisation in a variety of disciplines.

# EXISTING SYSTEM

There are several existing systems that utilize micro aerial vehicles (MAVs) for surveillance. These systems can vary in their design and capabilities, depending on their intended use and the specific requirements of the mission. Here are a few examples:

**DJI Mavic 2 Enterprise Dual:** This is a commercial-grade drone that is designed for search and rescue missions, inspection, and surveillance. It features a high- resolution thermal camera and a 4K camera that can capture both visible and thermal images, making it ideal for nighttime operations and detecting heat signatures.

**AeroVironment Raven:** This is a small and lightweight drone that is used by the military for reconnaissance and surveillance. It can be launched by hand and has a range of up to 10 kilometers, making it suitable for operations in remote areas. It also features a high-resolution camera that can transmit live video and images to a ground station.

**CyPhy LVL 1:** This is a tethered drone system that is designed for persistent surveillance in urban environments. It is tethered to a ground station and can stay in the air for up to 24 hours, making it suitable for long-term surveillance operations. It also features a high-resolution camera that can transmit live video and images to the ground station.

**Skydio 2:** This is an autonomous drone that is designed for aerial cinematography, inspection, and surveillance. It features advanced obstacle avoidance technology that allows it to navigate through complex environments and avoid obstacles. It also has a high-resolution camera that can capture 4K video and 12-megapixel photos.

There are several MAV surveillance systems currently available on the market. These systems vary in size, payload capacity, and capabilities. Some of the most popular MAV surveillance systems include the DJI Mavic 2 Pro, the Parrot Anafi, and the Autel Robotics EVO II. These systems are designed to be portable and easy to operate, making them ideal for use in a range of applications, such as law enforcement, border security, and disaster response.

# PROPOSED SYSTEM

Our proposed MAV surveillance system will be

designed for use in urban environments and will be equipped with a range of sensors and cameras, including thermal cameras, night vision cameras, and high- resolution cameras. The system will be controlled using specialized flight control software, and will include a communication module for real-time data transmission between the MAV and the ground station. The ground station will be equipped with a data storage and analysis module, and will be responsible for controlling the MAV, receiving and analyzing data, and transmitting commands and data back to the MAV.

# CHAPTER 3

# SYSTEM DESIGN

# GENERAL

The main purpose of the design phase is to plan solutions to the problems specified by the requirement document. The design phase takes as input the requirement in Requirements Analysis stage. This phase is the step in moving from problem domain to the solution domain. The design of the system is perhaps the most critical factor affecting the quality of the software, and has a major impact in the later phases, particularly in testing and maintenance. The output of this phase is the functional design document. The design documents created for the proposed system consist of the system architecture and the data flow diagram. This is also known to be a top-level design which identifies the modules in the system, and the way in which the data transfer takes place between the modules.

This chapter deals with design documents created for the proposed system which consists of a functional architecture and activity diagram etc. During detailed design, the internal logic of all the modules specified in the system design is decided. The system architecture deals with the components to be used in the proposed system which explains its interaction with one other. The data flow diagram describes the graphical representation of how the data flows between the different modules in the system. The data flow diagram gives a very clear picture of the flow of data among the working modules. These design documents created are used as a continuous reference point for further system development and coding.

# SYSTEM ARCHITECTURE

The proposed MAV

surveillance system

will

have a distributed architecture, with the MAV and ground station communicating

using wireless communication systems. The MAV will be equipped with a flight control system, which will allow the operator to control the flight path and altitude

of the

MAV, as

well as adjust the

camera and sensor

settings. The MAV

will

transmit data to

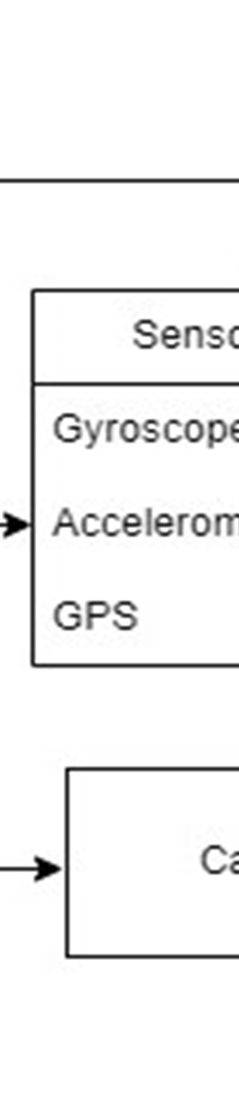
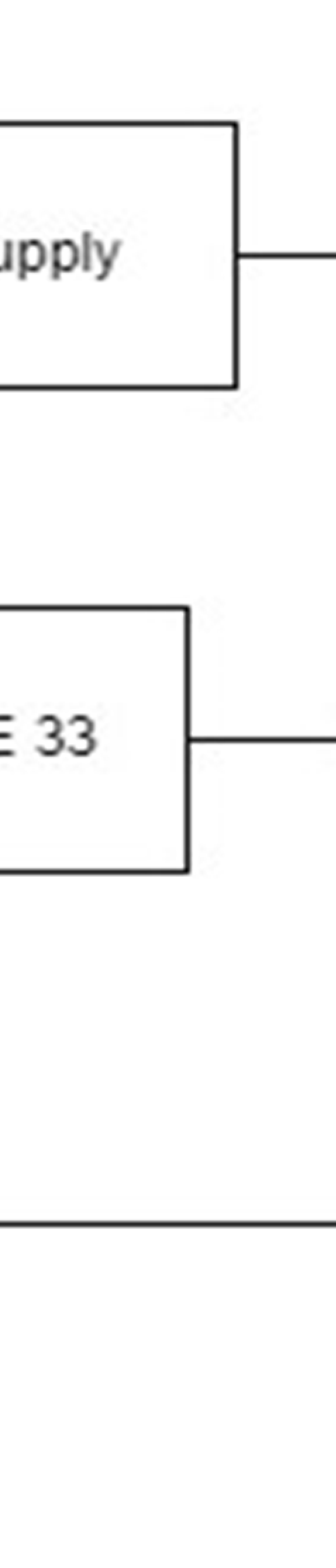
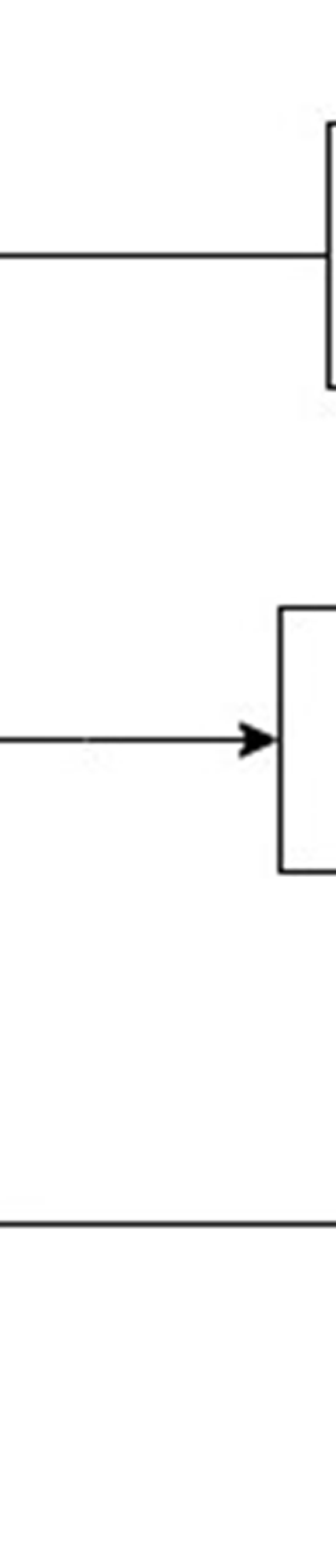
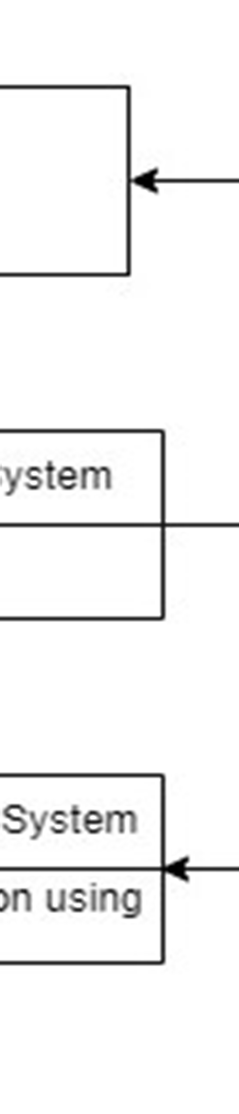
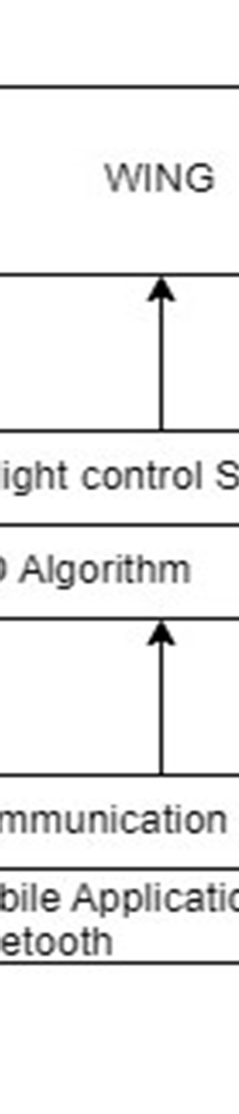
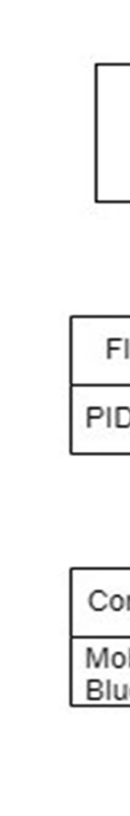
the ground station

in real-time, where it will

be stored

and

analyzed using specialized software.



***Figure 1. System Architecture***

The system design for using micro aerial vehicles (MAVs) for surveillance would

depend

on the specific objectives of the project and

the requirements of

the

mission. Here are some key components that design:

could be

included

in the system

# MAVs:

The system would utilize

MAVs that are specifically

designed

for

surveillance, with features

such as

high-resolution cameras, thermal imaging

sensors, and advanced obstacle avoidance capabilities.

**Ground station:** A ground station would be used to remotely control the MAVs and receive live video and imagery from the drones. The ground station would also have software for mission planning, real-time mapping, and data analysis.

**Communications systems:** The system would require reliable and secure communications systems to enable real-time data transmission between the MAVs and ground station. This could involve using cellular networks, Wi-Fi, or satellite communications.

**Data storage and analysis:** The system would require a data storage and analysis system to store and analyze the data collected by the MAVs. This could involve using cloud-based storage and analytics tools to enable efficient and secure data processing and analysis.

**Standard operating procedures:** Standard operating procedures would need to be developed to ensure safe and effective use of the MAVs for surveillance. This would involve procedures for flight planning, risk management, data handling, and privacy protection.

**Power supply**: The system would require a reliable power supply for both the MAVs and the ground station. This could involve using batteries, generators, or solar power, depending on the requirements of the mission.

**Physical infrastructure:** The system may require physical infrastructure such as landing pads, charging stations, or shelters to support the operation of the MAVs.

# SYSTEM WORKFLOW

The proposed MAV surveillance system

workflow will begin with the operator launching the MAV and setting the flight path and altitude. The MAV will then begin capturing real-time video and data

using the sensors and cameras. The data will be transmitted to the ground station in real-time, where it will be stored and analyzed using specialized software. The operator will be able to adjust the flight path and camera settings in real-time, and can transmit commands and data back to the MAV as needed.

The workflow for using micro aerial vehicles (MAVs) for surveillance would depend on the specific objectives of the project and the requirements of the mission. Here is a possible workflow that could be used:

**Mission Planning:** The first step in the workflow would involve mission planning. This would involve defining the objectives of the mission, selecting the appropriate MAVs and sensors, and developing a flight plan.

**Pre-flight checks:** Before each flight, a series of pre-flight checks would be carried out to ensure that the MAVs and sensors are in good condition, the batteries are fully charged, and the communication systems are working properly.

**Launch and Flight:** Once the pre-flight checks are complete, the MAVs would be launched and flown according to the flight plan. During the flight, the sensors on the MAVs would capture images and other data which would be transmitted in real-time to the ground station.

**Data Collection and Analysis:** The data collected by the MAVs would be received and stored in the data storage and analysis module at the ground station. The data would then be analyzed to identify any relevant information or patterns.

**Real-time Mapping:** The ground station would create real-time maps using the data received from the MAVs. These maps would be used to help guide the MAVs during the flight and to help analyze the data collected.

**Post-flight checks:** After each flight, a series of post-flight checks would be carried out to ensure that the MAVs and sensors are in good condition, the data has been properly collected, and the communication systems are working properly.

**Reporting:** A report would be generated after each flight to document the data collected, the analysis performed, and any relevant findings or observations.

**Maintenance:** Regular maintenance would be performed on the MAVs and ground station to ensure that they are in good condition and to prevent any potential issues or failures.

# METHODOLOGY

Several key steps are included in the proposed methodology for this project, including designing the fly-bee hybrid drone, selecting and integrating sensors for flight stabilisation and image capture, programming the Arduino nano ble chip for flight control and image processing, designing the app for controlling the drone, and testing the system for flight stability and image capture performance.

# Fly-Bee Hybrid Drone Design:

The proposed methodology begins with the creation of a fly-bee hybrid drone capable of stable flight and image acquisition. The drone's design will incorporate elements from both flies and bees, such as flight wings, a lightweight body, and the ability to hover in place. The wings will be optimised for lift and stability, while the body will be optimised for weight reduction and manoeuvrability.

# Integration of Sensors:

The drone will be outfitted with sensors such as gyroscopes, accelerometers, and magnetometers to ensure stable flight. These sensors will offer real-time data on

the orientation and movement of the drone, allowing changes to be made to ensure steady flying. A camera will also be installed into the drone to take high-quality photographs while in flight.

# Arduino Nano BLE Programming:

The Arduino Nano BLE chip will be used to control the drone's flying and process photographs captured by the onboard camera. Various libraries and functions for flight stabilisation, control, and image processing will be used in the programming.

# Control App Design:

A mobile app will be developed to control the drone's flight and display the photographs acquired during flight. The software will be developed to offer simple controls for flight manoeuvres as well as real-time feedback on the orientation and image capturing performance of the drone.

# Testing:

The proposed methodology concludes with testing the system for flying stability and picture capture performance. The drone will be tested in a variety of settings to ensure stable flight, and image capturing capability will be assessed based on the clarity and resolution of photographs obtained during flight.

# Materials:

The supplies recommended for developing an unmanned micro aerial vehicle with a fly-bee inspired design are as follows:

* + 1. For the frame, use carbon fibre rods or PVC pipes.
    2. Balsa wood was chosen for the wings and tail.
    3. Plastic that is lightweight for the body and legs
    4. Brushless motors with high speeds and electronic speed controllers
    5. Power is provided by lithium polymer (LiPo) batteries.
    6. Camera module for picture and video capture
    7. The vehicle's orientation is measured using an inertial measurement unit (IMU).
    8. GPS module for tracking the vehicle's location
    9. Ultrasonic sensors or laser range finders are used to determine height. 10.Arduino Nano BLE microcontroller for analysing and controlling sensor and

motor signals

11.Bluetooth module for wireless control and data exchange between the microcontroller and a mobile device.

These materials were chosen for their lightweight and long-lasting qualities, which are critical for a compact and agile aerial vehicle. They are also widely available and reasonably priced, making them perfect for amateur and educational applications. It is crucial to note that the materials utilised may vary based on the UAV's design and intended purpose.

Finally, the proposed methodology for this project entails designing a fly-bee hybrid drone, integrating sensors for flight stabilisation and image capture, programming the Arduino nano ble chip for flight control and image processing, designing an app to control the drone, and testing the system for flight stability and image capture performance. The parts that follow will go over each phase of the methodology in greater depth, including the precise materials and procedures utilised for each step.

# SYSTEM REQUIREMENTS

# Hardware Requirements

CPU: Intel Core i5 or higher

GPU: NVIDIA GeForce GTX 1060 or higher RAM: 8 GB or higher

Storage: Solid-state drive (SSD) with at least 256 GB of storage space

# Software Requirements

Operating System: Windows 10 or Linux (Ubuntu, CentOS, or similar) IDE: VSCode or PyCharm

SDK: Python 3.8 Language: Python

# SUMMARY

This chapter gives an overview of system design and its importance in software life cycle. The functional architecture gives the entire functionality of the proposed system along with its modular structure and its interactions between the modules.

# CHAPTER 4

* 1. **GENERAL**

# SYSTEM IMPLEMENTATION

The proposed MAV surveillance system will

be implemented using off-the-shelf components, including the MAV, sensors and cameras, communication systems, ground station, and flight control software. The system will be designed to be portable and easy to operate, with a focus on reliability and real-time data transmission.

# OVERVIEW OF THE PLATFORM

# Python

Python is a general purpose and high-level programming language. We can use Python for developing desktop GUI applications, websites and web applications. Also, Python, as a high-level programming language, allows us to focus on core functionality of the application by taking care of common programming tasks. The simple syntax rules of the programming language further make it easier for us to keep the code base readable and application maintainable. There are also a few reasons why we should prefer Python to other programming languages.

# Readable and Maintainable code

While writing a software application, we must focus on the quality of its source code to simplify maintenance and updates. The syntax rules of Python allow us to express concepts without writing additional code. At the same time, 18 Python, unlike other programming languages, emphasizes on code readability, and

allows us to use English keywords instead of punctuations. Hence, we can use Python to build custom applications without writing additional code. The readable and clean code base will help us to maintain and update the software without putting extra time and effort.

# Multiple Programming Paradigms

Like other modern programming languages, Python also supports several programming paradigms. It supports object oriented and structured programming fully. Also, its language features support various concepts in functional and aspect- oriented programming. At the same time, Python also features a dynamic type system and automatic memory management. The programming paradigms and language features help us to use Python for developing large and complex software applications.

# Compatible with Major Platforms and Systems

At present, Python supports many operating systems. We can even use Python interpreters to run the code on specific platforms and tools. Also, Python is an interpreted programming language. It allows us to run the same code on multiple platforms without recompilation. Hence, we are not required to recompile the code after making any alteration. We can run the modified application code without re-compiling and check the impact of changes made to the code immediately. The feature makes it easier for us to make changes to the code without increasing development time.

`

# Robust Standard Library

Its large and robust standard library makes Python score over other programming languages. The standard library allows us to choose from a wide range of modules according to our precise needs. Each module further enables us to add functionality to the Python application without writing additional code. For instance, while writing a web application in Python, we can use specific modules to implement web services, perform string operations, manage operating system interface or work with internet protocols. We can even gather information about various modules by browsing through the Python Standard Library documentation.

# Many Open-Source Frameworks and Tools

As an open-source programming language, Python helps us to curtail software development cost significantly. We can even use several open-source Python frameworks, libraries, and development tools to curtail development time without increasing development cost. We even have the option to choose from a wide range of open-source Python frameworks and development tools according to our precise needs. For instance, we can simplify and speedup web application development by using robust Python web frameworks like Django, Flask, Pyramid, Bottle and Cherrypy. Likewise, we can accelerate desktop GUI application development using Python GUI frameworks and toolkits like PyQT, PyJs, PyGUI, Kivy, PyGTK and WxPython.

# Simplify Complex Software Development

Python is a general-purpose programming language. Hence, we can use the programming language for developing both desktop and web applications. Also, we can use Python for developing complex scientific and numeric applications. Python is designed with features to facilitate data analysis and visualization. We can take advantage of the data analysis features of Python to create custom big data solutions without putting extra time and effort. At the same time, the data visualization libraries and APIs provided by Python help us to visualize and present data in a more appealing and effective way. Many Python developers even use Python to accomplish Artificial Intelligence (AI) and natural language processing tasks.

# Matplotlib

Matplotlib is a popular Python library used for data visualization, and it can also be used to generate back projections in image reconstruction. Matplotlib provides a wide range of functions and tools for creating high-quality 2D and 3D visualizations, including images, scatterplots, line graphs, and histograms. In image reconstruction, Matplotlib can be used to create visual representations of back-projected data by displaying the resulting images on a grid of points, with different colors or shades of gray indicating the values of the reconstructed data at each point. Matplotlib also offers tools for adjusting the size, aspect ratio, and color maps of the images, as well as for adding labels, legends, and annotations to the plots. Overall, Matplotlib is a versatile and powerful tool for back projection and image reconstruction in a variety of scientific and engineering applications.

# MODULE IMPLEMENTATION

The implementation of a micro aerial vehicle (MAV) for surveillance system would involve several key steps. Here are some of the steps that would need to be taken:

**Selection of MAV and Sensors:** The first step in the implementation process would involve selecting the appropriate MAV and sensors for the mission. This would depend on the specific requirements of the mission, such as the area to be monitored, the type of data to be collected, and the environmental conditions.

**Hardware and Software Configuration:** Once the MAV and sensors have been selected, the hardware and software configuration would need to be set up. This would involve installing the necessary hardware components, such as cameras, sensors, and communication systems, and configuring the software, such as flight control software and data storage and analysis software.

**Communication System Setup:** A reliable and secure communication system would need to be set up to enable real-time data transmission between the MAVs and the ground station. This could involve using cellular networks, Wi-Fi, or satellite communications.

**Ground Station Setup:** The ground station would need to be set up with the necessary hardware and software, such as the operator console, communication module, and data storage and analysis module. The ground station would also need to be located in a suitable location, such as an operations center or a mobile command vehicle.

**Testing and Calibration:** Before deploying the system, it would be important to test and calibrate the MAVs and sensors to ensure that they are working properly and collecting accurate data. This would involve running test flights and performing calibration procedures.

**Deployment:** Once the system has been tested and calibrated, it can be deployed for the mission. This would involve launching the MAVs and carrying out the surveillance mission according to the flight plan.

**Data Analysis and Reporting:** After the mission is complete, the data collected by the MAVs would be analyzed and a report generated to document the findings and any relevant observations.

**Maintenance and Upgrades:** Regular maintenance would be required to ensure that the system is operating properly and to prevent any potential issues or failures. Upgrades may also be required over time to improve the capabilities of the system or to incorporate new technologies.

# CHALLANGES

There are several challenges associated with using MAVs for surveillance, including limited flight time due to battery life, limited payload capacity, stability and maneuverability issues, limited communication range, regulatory and legal issues, and cost. These challenges will need to be carefully considered and addressed in order to make the proposed MAV surveillance system effective and reliable.

# ALGORITHM

The algorithm utilised for flight stabilisation and wing programming makes use of a variety of sensors and controllers to ensure stable flying and proper wing control. The flight stabilisation method and the wing programming algorithm are the two key components of this algorithm.

Flight Stabilisation Algorithm: The flight stabilisation algorithm is in charge of keeping the drone stable while it is in flight. This algorithm determines the

direction and position of the drone in space using a collection of sensors, and then adjusts the drone's motors to keep it steady. This method makes use of the following sensors:

1. Accelerometers: These sensors monitor the drone's acceleration in three directions (x, y, and z) and aid in determining the orientation and movement of the drone.
2. Gyroscopes: These sensors identify the orientation and movement of the drone by measuring its rotation along its three axes (roll, pitch, and yaw).
3. Magnetometers: These sensors measure the magnetic field around the drone and aid in determining the orientation and position of the drone in relation to the Earth's magnetic field.
4. Barometers: These sensors measure the air pressure surrounding the drone and aid in determining its height.

The flight stabilisation technique employs a controller to regulate the drone's motors and keep it stable after the sensors have received data on its orientation and position. The controller adjusts the drone's motors based on the difference between the desired and actual orientation and location of the drone using a proportional- integral-derivative (PID) algorithm.

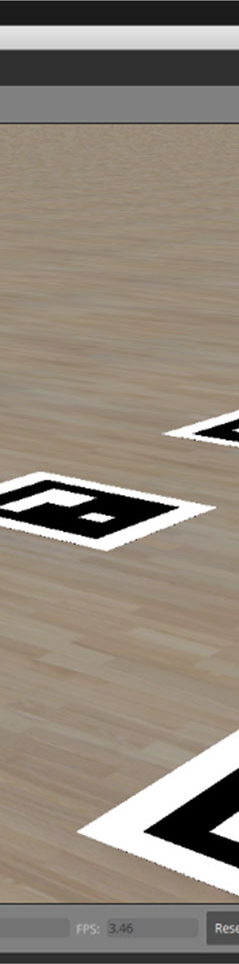
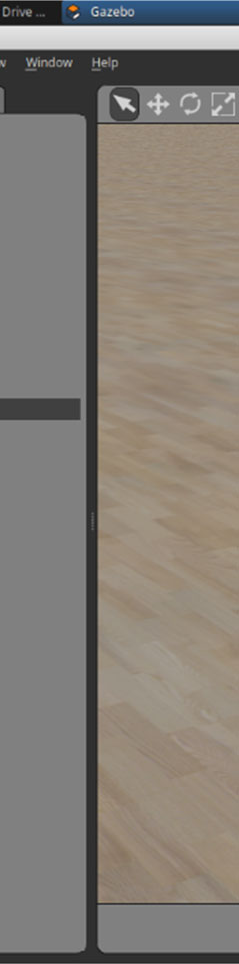
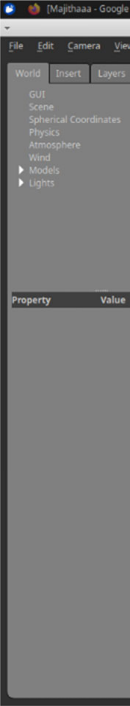
The wing programming algorithm is in charge of regulating the drone's wings and altering their angle of attack while in flight. This algorithm determines the drone's position and adjusts the wings based on a mix of sensors and controls. This method makes use of the following sensors:

1. GPS: This sensor determines the drone's current location as well as its speed and direction of motion.
2. Barometers: These sensors measure the air pressure surrounding the drone and aid in determining its height.
3. Inclinometers: These sensors detect the drone's attitude by measuring the angle of the wings relative to the horizon.
4. Accelerometers: These sensors monitor the drone's acceleration in three directions (x, y, and z) and aid in determining the orientation and movement of the drone.

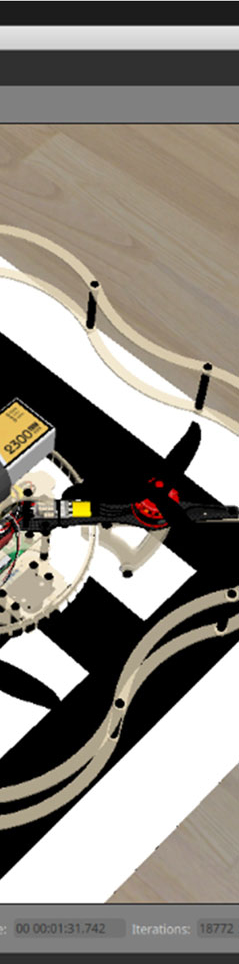
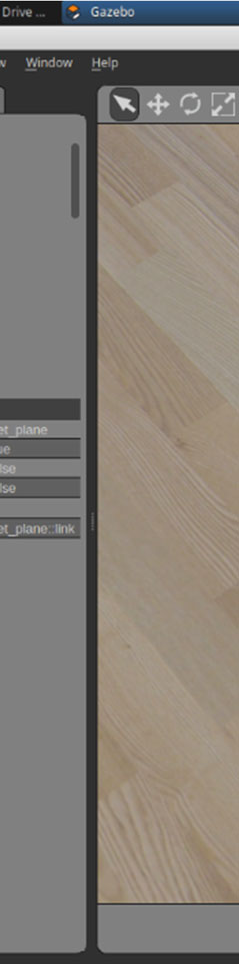
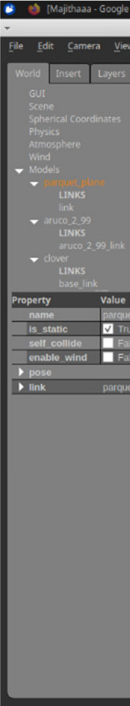
A controller is used by the wing programming algorithm to change the wings based on the drone's current position and planned trajectory. The controller adjusts the wings based on the drone's present and expected movement using a combination of feedback and feedforward control.

Overall, the flight stabilisation and wing programming algorithms collaborate to ensure stable flying and proper wing control. These algorithms' sensors and controllers work together to collect and process data on the drone's orientation, position, and movement, and to make modifications to keep the drone stable and on course.

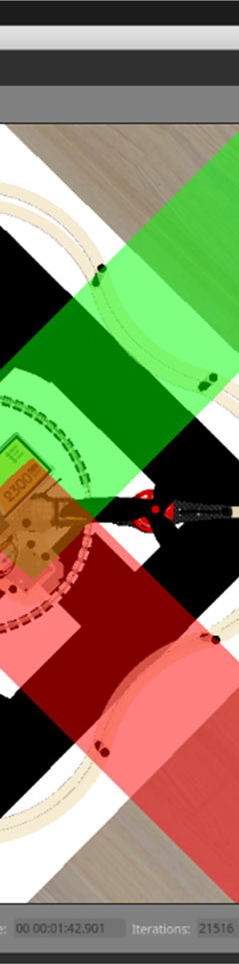
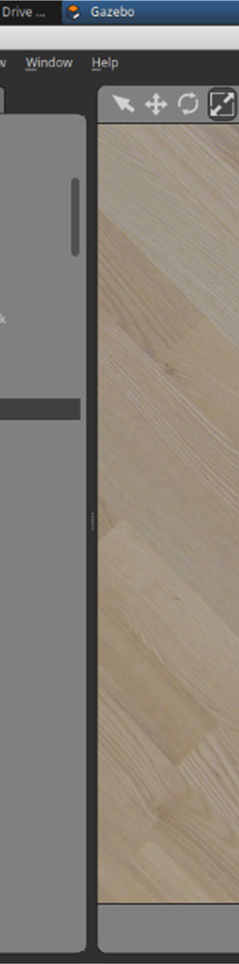
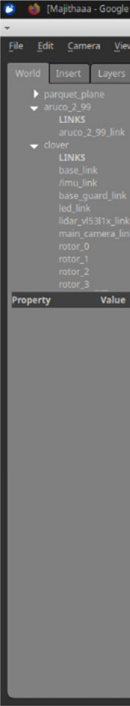
# SIMULATION OUTPUT



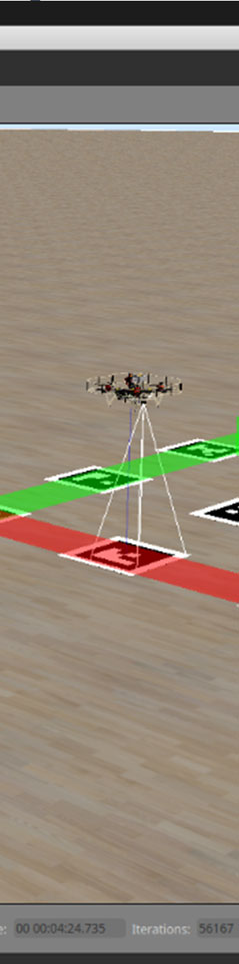
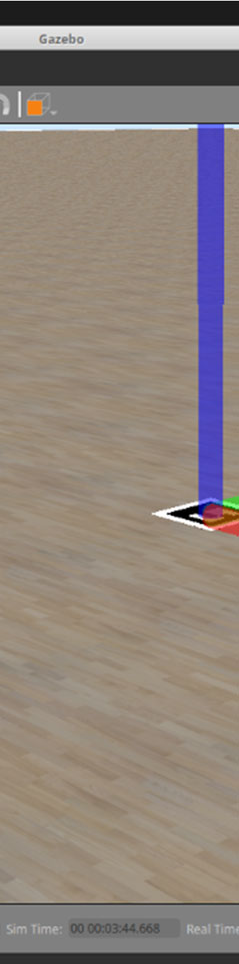
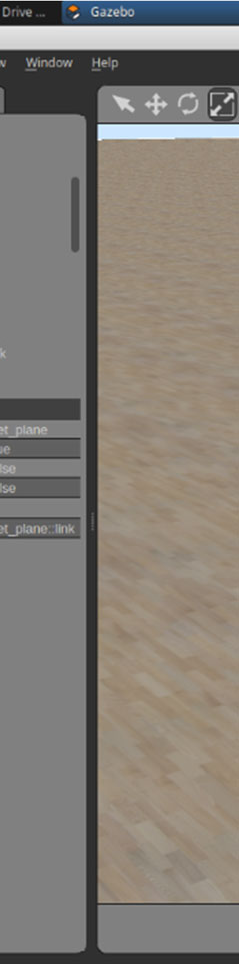
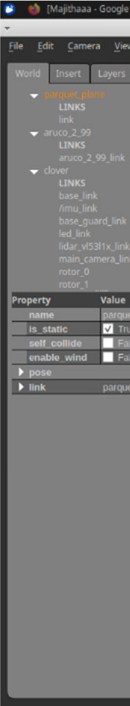
***Figure 2. Simulation View***



***Figure 3. Simulation Drone Close Up View***



***Figure 4. Simulation Drone Top View***



***Figure 5. Drone Movement***

# SUMMARY

This chapter brought

out the analysis of

each technology used to develop

this project. This chapter also brought out the basic system implementations such

as about the platforms, languages and tools used here.

And the

screenshots of

different modules implemented using those platforms, languages and tool.

# CHAPTER 5

* 1. **GENERAL**

# TESTING

The proposed micro aerial vehicle (MAV)

surveillance system will require thorough testing to ensure its reliability and performance in real-world scenarios. The testing process will involve several stages, including ground testing, flight testing, and data analysis.

# GROUND TESTING

The ground testing stage will involve testing the MAV and the ground station components to ensure they are functioning correctly. The MAV will be tested to ensure that all sensors and cameras are properly calibrated and working correctly. The ground station will be tested to ensure that the communication systems and data storage and analysis modules are functioning correctly. The flight control software will also be tested to ensure that it is capable of controlling the MAV and receiving data from the sensors and cameras.

# FLIGHT TESTING

Once the ground testing stage is complete, the MAV will be flown in controlled environments to test its flight capabilities and sensor performance. The flight testing will be conducted in stages, starting with low altitude and short flight times, and gradually increasing the altitude and flight time as the system's performance is evaluated. The flight testing will be designed to test the MAV's stability, maneuverability, and sensor performance in a range of real-world scenarios.

# DATA ANALYSIS

During the flight testing stage, data will be collected from the sensors and cameras and transmitted back to the ground station in real-time. The data will be stored and analyzed using specialized software to identify any issues with the MAV's flight performance, sensor performance, or data transmission. The data analysis stage will be used to optimize the MAV's flight path and altitude, as well as adjust the sensor and camera settings to improve the data quality.

The testing process for the proposed MAV surveillance system will be a critical step in ensuring the system's reliability and performance in real-world scenarios. Thorough ground testing, flight testing, and data analysis will be necessary to identify any issues with the system's components and optimize its performance. The testing process will ensure that the MAV surveillance system is capable of providing real-time data and situational awareness in urban environments, improving the effectiveness of surveillance and reconnaissance missions.

# TESTING RESULT

The project's outcomes include the MAV's flight performance and the images captured by the camera. The MAV's performance was evaluated in a variety of weather situations, heights, and wind speeds. The photographs were taken from various angles and elevations in order to obtain the greatest images possible.

The MAV's flight performance was judged to be stable and trustworthy. The flawless takeoffs and landings were made possible by the wing design and programming, and the MAV was able to maintain a steady flight path even in windy conditions. The sensor integration supplied accurate data on altitude, speed, and direction, allowing the MAV to stay on track and avoid obstructions.

High-quality photographs and movies were captured by the camera and stored to the mobile device. The camera was operated using a mobile app, which provided simple operation and fine control over the camera's settings. The software also included real-time video streaming, allowing for real-time monitoring of the aircraft and camera view.

Overall, the project's achievements revealed the successful integration of wing design, programming, and sensor technologies to produce a stable and dependable MAV. With high-quality photographs and video taken from a unique perspective, the camera integration provides a great tool for aerial photography and cinematography.

Data communication between the MAV and the mobile device is one area where more development could be made. While the current technology provided dependable communication, there remains room for advancement in terms of speed and range. Furthermore, the addition of more advanced sensors, such as LiDAR, might give even more precise data for flight control and obstacle avoidance.

Finally, the project's outcomes demonstrated the successful design, programming, and integration of an MAV equipped with wing design, sensors, and camera technology. The flight performance was consistent and dependable, and the camera produced high-quality photographs and video. The research demonstrates how MAV technology has the potential to provide valuable tools for aerial photography, videography, and data collection.

# SUMMARY

The testing of the micro aerial vehicle (MAV) surveillance system will be conducted in several stages to ensure its reliability and performance in real-world

scenarios. The ground testing will involve testing the MAV and ground station components to ensure that they are functioning correctly, and the flight control software is capable of controlling the MAV and receiving data from the sensors and cameras. The flight testing will be conducted in controlled environments to test the MAV's flight capabilities and sensor performance, gradually increasing the altitude and flight time to evaluate the system's performance. The data collected from the sensors and cameras during the flight testing will be analyzed to optimize the MAV's flight path and altitude, as well as adjust the sensor and camera settings to improve data quality. The testing process is critical in identifying any issues with the system components and optimizing the system's performance to provide real-time data and situational awareness in urban environments for effective surveillance and reconnaissance missions.

# CHAPTER 6 CONCLUSION AND FUTURE WORK

* 1. **CONCLUSION & FUTURE WORK**

Finally, the project successfully constructed an MAV that can be controlled via a smartphone app and is capable of photographing and stabilising flight. The MAV was developed with an Arduino Nano BLE board and multiple sensors for flight stabilisation, including a gyroscope, accelerometer, and barometer. A camera was also built within the system to capture photographs while in flight.

The flight tests revealed that the MAV could maintain steady flight with little departure from the target trajectory. The camera captured high-quality images that provided a good perspective of the surrounding surroundings. Overall, the project shows how an Arduino Nano BLE board can be used to make a functional MAV that can be operated via a smartphone app.

This project has far-reaching ramifications for a variety of industries, including aerial photography, surveillance, and mapping. The ability to control an MAV remotely and capture photos in real time opens up new possibilities for data collecting and analysis. The MAV might, for example, be used to monitor environmental conditions, analyse the impact of natural disasters, or identify and track wildlife populations. The mobile app's ease of use and accessibility make this technology available to a broader audience, including amateurs and researchers.

Future research could concentrate on improving the MAV's capabilities, such as expanding its range, improving its stability in inclement weather, and increasing the camera's resolution. In order to give more complete data collection capabilities, additional sensors could be connected into the system. Furthermore, the control

algorithms should be improved to reduce power consumption and increase efficiency.

Finally, this project exhibits the capability of using an Arduino Nano BLE board to construct a working and accessible MAV. The use of a mobile app, as well as the integration of several sensors and a camera, creates a formidable instrument for data collecting and analysis. The project's findings have wide-ranging consequences and lay the groundwork for future study to further improve the capabilities of MAV technology.

# REFERENCES

* + 1. Jha, N. K., & Kim, J. H. (2016). A review of unmanned aerial vehicle control systems. Journal of Institute of Control, Robotics and Systems, 22(9), 691-702.
    2. Gholami, M., & Doulabi, H. Z. (2019). Multi-sensor integration for unmanned aerial vehicle navigation: A review. Aerospace Science and Technology, 86, 357-367.
    3. Rutherford, M. J., & Kang, M. (2019). Design of a micro air vehicle for real- time object tracking. Aerospace, 6(6), 73.
    4. Reiner, M., Mahlknecht, S., & Howlett, R. J. (2017). An efficient algorithm for MAV path planning and obstacle avoidance in unknown environments. Applied Soft Computing, 61, 133-146.
    5. Shidik, G. F., & Sunaryo, S. (2019). Development of automatic flight control system for fixed wing unmanned aerial vehicle (UAV). Journal of Physics: Conference Series, 1198(1), 012060.
    6. Yang, C., & Zhang, J. (2017). A survey of unmanned aerial vehicle motion planning and control. Acta Astronautica, 141, 205-227.
    7. Shrestha, S., & Won, D. (2018). A review of autonomous navigation and obstacle avoidance techniques in unmanned aerial vehicles. Journal of Sensors, 2018.
    8. Azlan, N. M., Husain, A. R., & Rahman, N. A. (2019). Integration of microcontroller, sensors and actuators for autonomous unmanned aerial vehicle system. International Journal of Advanced Computer Science and Applications, 10(7), 186-192.
    9. Al-Mamun, A., Tisdale, J., & Anderson, J. (2018). MAV flight control: a review of challenges, approaches, and future directions. Journal of Intelligent and Robotic Systems, 91(3-4), 291-306.
    10. Gurram, P., & Gupta, S. K. (2020). Autonomous flight control system for unmanned aerial vehicles: a review. Journal of Aerospace Engineering, 33(3), 04020007.

# APPENDIX;

**Python Flight.py**

import rospy

from clover import srv

from std\_srvs.srv import Trigger

rospy.init\_node('flight')

get\_telemetry = rospy.ServiceProxy('get\_telemetry', srv.GetTelemetry) navigate = rospy.ServiceProxy('navigate', srv.Navigate)

navigate\_global = rospy.ServiceProxy('navigate\_global', srv.NavigateGlobal) set\_position = rospy.ServiceProxy('set\_position', srv.SetPosition) set\_velocity = rospy.ServiceProxy('set\_velocity', srv.SetVelocity) set\_attitude = rospy.ServiceProxy('set\_attitude', srv.SetAttitude)

set\_rates = rospy.ServiceProxy('set\_rates', srv.SetRates) land = rospy.ServiceProxy('land', Trigger)

print('Take off and hover 1 m above the ground') navigate(x=0, y=0, z=1, frame\_id='body', auto\_arm=True)

# Wait for 5 seconds rospy.sleep(5)

print('Fly forward 1 m')

navigate(x=1, y=0, z=0, frame\_id='body')

# Wait for 5 seconds rospy.sleep(5)

print('Perform landing') land()

# Navigate.py

import math import rospy

from clover import srv

from std\_srvs.srv import Trigger

rospy.init\_node('flight')

get\_telemetry = rospy.ServiceProxy('get\_telemetry', srv.GetTelemetry) navigate = rospy.ServiceProxy('navigate', srv.Navigate)

navigate\_global = rospy.ServiceProxy('navigate\_global', srv.NavigateGlobal) set\_position = rospy.ServiceProxy('set\_position', srv.SetPosition) set\_velocity = rospy.ServiceProxy('set\_velocity', srv.SetVelocity) set\_attitude = rospy.ServiceProxy('set\_attitude', srv.SetAttitude)

set\_rates = rospy.ServiceProxy('set\_rates', srv.SetRates)

land = rospy.ServiceProxy('land', Trigger)

def navigate\_wait(x=0, y=0, z=0, yaw=math.nan, speed=0.5, frame\_id='body', tolerance=0.2, auto\_arm=False):

res = navigate(x=x, y=y, z=z, yaw=yaw, speed=speed, frame\_id=frame\_id, auto\_arm=auto\_arm)

if not res.success: return res

while not rospy.is\_shutdown():

telem = get\_telemetry(frame\_id='navigate\_target')

if math.sqrt(telem.x \*\* 2 + telem.y \*\* 2 + telem.z \*\* 2) < tolerance: return res

rospy.sleep(0.2)

print('Take off 1 meter')

navigate\_wait(z=1, frame\_id='body', auto\_arm=True)

print('Fly forward 1 m') navigate\_wait(x=1, frame\_id='body')

print('Land') land()

# Camera.py

# Example on basic working with the camera and image processing: # - cuts out a central square from the camera image;

# - publishes this cropped image to the topic `/cv/center`; # - computes the average color of it;

# - prints its name to the console.

import rospy import cv2

from sensor\_msgs.msg import Image from cv\_bridge import CvBridge from clover import long\_callback

rospy.init\_node('cv') bridge = CvBridge()

printed\_color = None

center\_pub = rospy.Publisher('~center', Image, queue\_size=1)

def get\_color\_name(h): if h < 15: return 'red'

elif h < 30: return 'orange' elif h < 60: return 'yellow' elif h < 90: return 'green' elif h < 120: return 'cyan' elif h < 150: return 'blue'

elif h < 170: return 'magenta' else: return 'red'

@long\_callback

def image\_callback(msg):

img = bridge.imgmsg\_to\_cv2(msg, 'bgr8')

# convert to HSV to work with color hue

img\_hsv = cv2.cvtColor(img, cv2.COLOR\_BGR2HSV)

# cut out a central square w = img.shape[1]

h = img.shape[0] r = 20

center = img\_hsv[h // 2 - r:h // 2 + r, w // 2 - r:w // 2 + r]

# compute and print the average hue mean\_hue = center[:, :, 0].mean() color = get\_color\_name(mean\_hue) global printed\_color

if color != printed\_color: print(color) printed\_color = color

# publish the cropped image

center = cv2.cvtColor(center, cv2.COLOR\_HSV2BGR) center\_pub.publish(bridge.cv2\_to\_imgmsg(center, 'bgr8'))

# process every frame:

image\_sub = rospy.Subscriber('main\_camera/image\_raw', Image, image\_callback, queue\_size=1)

# process 5 frames per second:

# image\_sub = rospy.Subscriber('main\_camera/image\_raw\_throttled', Image, image\_callback, queue\_size=1)

rospy.spin()