DEVELOPMENT OF AMBULANCE DRONE FOR DELIVERING EMERGENCY AIDS

The
BRITISH UNIVERSITY
IN EGYPT

By

Ahmed Hatem	**213157**
Carmen Maher	**207179**
Kareem Alaa	**207155**
Laila Ali	**209413**
Miral Ashraf	**212811**

A Thesis Submitted to the Faculty of Engineering,
The British University in Egypt, (**BUE**)

In Partial Fulfilment of the Requirements for
The **Bachelor of Engineering (BEng)** Degree in

Robotics Engineering

June 2025

DEVELOPMENT OF AMBULANCE DRONE FOR DELIVERING EMERGENCY AIDS

Thesis Approved:

Thesis Advisor Name	**Signature / Date**
Dr. Abdullah ElGammal	**----------**

Internal Committee Member	**Signature / Date**
Dr. Mostafa Abdelaziz	**----------**

External Committee member	**Signature / Date**
Dr. Tamer Khaleel	**----------**

Head of Mechanical Engineering Department

Faculty of Engineering. The British University in Egypt

Name: Dr. Ayman Abbas

Signature: _____ Date: _____

Declaration of Authorship

We hereby declare that this work titled “Development of Ambulance Drone for Delivering Emergency Aids” submitted to the Mechanical Engineering department of the British University in Egypt is an original piece of work and has not been published or showcased anywhere else in the past. This work was carried out under the supervision of Dr. Abdullah ElGammal.

Name: Ahmed Hatem

Signature / Date: Ahmed Hatem 13/2/2025

Name: Carmen Maher

Signature / Date: Carmen Maher 13/2/2025

Name: Kareem Alaa

Signature / Date: Kareem Alaa 13/2/2025

Name: Laila Ali

Signature / Date: Laila Ali 13/2/2025

Name: Miral Ashraf

Signature / Date: Miral Ashraf 13/2/2025

Acknowledgements

I would like to express my deep gratitude to Dr. Abdullah ElGammal, our graduation design project supervisor and our mentor, for his patient guidance, enthusiastic encouragement and useful critiques of this design project.

I would also like to thank Eng. Saifeldeen Radwan for their support while carrying out the lab work, for working alongside me for the better part of a year and being there to brainstorm ideas and solve problems encountered during this project especially during the lab work.

I am also very grateful to the British University of Egypt for providing the opportunity to carry out this graduation project and the resources necessary to carry it out successfully.

Finally, I would like to thank my parents for always supporting and encouraging my education. Their hard work and patience is the reason I am where I am today.

Abstract

The swift transfer of medical supplies during emergencies is critical to save lives, however traditional road-based ambulances tend to be stuck in traffic as well as accessibility restrictions. This project conceptualizes an autonomous quadcopter drone that will deliver emergency medical aid effectively in urban and rural environments. Quadcopter with light X-shaped carbon fibre structure (600 mm diagonal) for 1.5 kg payload, servo gripper for safe carry of medical box, and temperature-controlled payload box with Arduino sensors. Power-driven by a 4S 5200mAh LiPo battery and Sunny Sky 800kV motors, with a thrust-to-weight ratio that reaches 3.19 and a flight time up to 20 minutes. Autonomous flight is endorsed by a Pixhawk 2.4.8 autopilot, Raspberry Pi 4B, GPS, IMU, Barometer and vision systems, which are Aruco markers to enable precise landing and obstacle avoidance algorithms. Design included SolidWorks modelling, ANSYS structural analysis, Gazebo simulation, and real-world testing in order to confirm the system's stability, navigation accuracy, and payload release. Flight tests showed stable yaw, pitch, and roll, GPS readings verifying accurate trajectory tracking. Despite of limited flight time, the prototype provides an expandable response for emergency medical logistics, decreasing response times and resolving worldwide challenges in healthcare delivery. Future upgrades include longer battery life and BVLOS compliance to further enhance operation capabilities.

Table of Contents

List of Figures

List of Tables

Nomenclature

m	Mass of the quadrotor UAV
I	Moment of inertia matrix
ω	Angular velocity (body frame)
τ	Torque acting on the body
F	Force vector (sum of all external forces)
R	Rotation matrix (body frame to inertial frame)
T	Transformation matrix (for angular velocity conversion)
v	Linear velocity
a	Acceleration
g	Gravitational acceleration vector
U	Total thrust force
η	External disturbances (e.g., wind forces)
J	Jacobian matrix (used in control/dynamics modeling)
b	Arm length (distance from rotor to center)
d	Drag coefficient
K_t	Thrust coefficient
K_d	Drag torque coefficient
θ,ϕ,ψ	Euler angles: pitch, roll, and yaw respectively
ρ	Air density
$vmax$	Maximum forward velocity
TWR	Thrust-to-Weight Ratio
m	Mass of the quadrotor
I	Moment of inertia matrix
ω	Angular velocity vector
R	Rotation matrix from body to inertial frame
T	Transformation matrix for velocity mapping
F	Force vector (external or thrust)
τ	Torque vector
ϵ	Strain (stretch ratio - 1)
λ	Stretch ratio

Abbreviations

UAV	Unmanned Aerial Vehicle
EMS	Emergency Medical Services
IMU	Inertial Measurement Unit
GPS	Global Positioning System
ESC	Electronic Speed Controller
FEA	Finite Element Analysis
PWM	Pulse Width Modulation
PLA	Polylactic Acid (3D printing material)
PID	Proportional-Integral-Derivative
ROS	Robot Operating System
SCARA	Selective Compliance Articulated Robot Arm
BVLOS	Beyond Visual Line of Sight
HDOP	Horizontal Dilution of Precision
GCS	Ground Control Station
PDB	Power Distribution Board
LiPo	Lithium Polymer (battery type)
TWR	Thrust-to-Weight Ratio

Chapter (1) Introduction

1.1. Background

Within the last couple of years, drones, also known as unmanned aerial vehicles (UAVs) that can operate remotely or autonomously, have revolutionized industries in numerous ways including redefining how products, materials, and even individuals are transported while likewise advancing capabilities in surveillance and area mapping. Drones have proven to be a great asset for humans with their many advantages and innovative features one being their versatility and efficiency that grants them the advantage of being well-suited for numerous tasks. Furthermore, some classes of drones can manoeuvre vertically and hover in position, thereby gaining access to challenging environments. A class of drones that is capable of vertical manoeuvrability is multi-rotors, such as quadcopters [1]. A class of drones that is capable of vertical manoeuvrability is multi-rotors, such as quadcopters or quadrotor, a type of drone that is propelled by four rotors [2]. Their size serves as an advantage especially for complex, congested spaces, making them very suitable for urban deliveries where space is limited, and navigation is intricate. One of the applications where quadcopters can be adopted to enhance the efficiency of the operation is in the field of emergency medical services.

Today, traditional emergency medical systems EMS depend much on ground ambulances. While ambulances may be effective in many aspects, it normally incurs delays due to congestion in traffic, limited access routes, and infrastructural constraints. Delays in such urgent cases could be fatal because the lack of timely intervention might cause extreme consequences or death. Thus, drones have the potential to be novel solutions in emergency response in the medical sector, where every second response delay determines life or death outcomes for ambulance delays have been one of the main contributors to preventable deaths worldwide as more efficient and immediate medical services can be lifesaving, with a marked 50% reduction in mortality if victims receive treatment within the "golden hour" following an injury [3].

1.2. Aim and Objectives

Although quadcopters might seem like the perfect alternative for ambulances, current drones are lacking in some respects. Therefore, this project aims to develop a quadcopter capable of navigating very adverse environments and delivering medical supplies to people in distress, with an emphasis on robust mechanical and electrical design all while improving the control systems and stability dynamics. To be able to achieve the desired outcome, the following objectives where set

1. Obtain a lightweight yet robust quadrotor design that has optimized load capacity, energy-efficient components, well-balanced weight distribution, and enhanced aerodynamic efficiency.
2. Incorporate into the quadrotor a compatible autonomous flight system that employs AI-driven sensors to detect and avoid obstacles in real time and navigates with high precision to enhance stability and improve overall flight performance.
3. Establish real-time feedback system to track the flight performance and battery health and ensure system reliability.
4. Fabricate a working quadrotor prototype that follows the obtained design specifications to test its robustness and efficiency in various real-life scenarios and challenging environments.

1.3. Design Methodology

The methodology employed for this project can be separated into five stages, each stage playing a critical role in the development of a fully functional, optimized quadrotor. First, the quadrotor design, which includes model visualization on SolidWorks, analysis for structural robustness and aerodynamic efficiency using ANSYS, and components selection, was obtained. After theoretically validating and finalizing it, it was experimentally validated through a prototype that includes the smooth integration of the system's main components such as motors, batteries, sensors, and systems of payload delivery onto the plates of the prototype drone frame. Following

the assembling of the prototype comes the manual flight experimentation using a transmitter to test basic flight dynamics, stability, and control responsiveness before implementing autonomous function. Obstacle avoidance and precision landing algorithms were then integrated to help further increase the flight performance and decrease unwanted structural damages. To optimize the overall system efficiency, system health monitoring and IoT-based were included for autonomous capability with real-time feedback. Several controlled experiments have been performed for flying stability, obstacle avoidance capability, and payload delivery. Finally, field tests were conducted to mimic actual emergency scenarios allowing for the investigation of the quadcopter's capability to navigate through urban and rural areas.

Chapter (2) Design Review

This chapter demonstrates the design considerations, technological advancements and different applications of drones while shedding light on medical delivery and emergency response. Moreover, it explores the historical developments, classifications, and the core components to show the possible challenges that the drone faces when it is exposed to a complex real environment. By exploring those limitations, future trends like advanced propulsion, artificial intelligence, swarm technology and telemedical integration are discussed to find more efficient solutions and to create more sustainable drones.

2.1. Introduction

Autonomous drones emerged as a novel technology with components that provides them with more efficiency as they incorporate advanced systems such as artificial intelligence, sensor fusion, and navigation technologies to ensure precision and reliability. Modern drones have facilitated the performance of some critical challenges, particularly time related functions, in dynamic environments [4]. Additionally, the timeline of the drone's evolution is showing rapid movement that started from simple devices used for militarily surveillance to highly advanced systems with the capabilities for independent operations [5]. With their low operative cost and high benefit, they have become an essential resource, to perform tasks that were usually by helicopters or vehicles before, such as medical emergency response. As autonomous drones become more widely used, the demand for a better understanding of their capabilities and limitations grows. Thus, this chapter presents a detailed overview of quadcopters, mentioning design analysis, classification and main components. It further points out the number of challenges and limitations, such as battery life, regularity constraints and safety concerns. On the bright side, it provides insight into emerging trends and innovative solutions [6], [7].

2.2. Overview of Autonomous Drones

2.2.1. Drone Classifications

Drones are classified by size, range, altitude, and configuration. Configuration-based, including single-rotor, fixed-wing, hybrid, and multirotor UAVs, comes with its set of unique characteristics, advantages, and limitations. Single Rotor UAVs, resembling helicopters in design, have one large rotor for lifting and a tail rotor for stabilizing. This configuration gives them more efficiency compared to multi-rotors, especially during long-duration flights and with heavy payloads [8], [9]. Transitioning from efficiency to endurance, fixed-wing UAVs outshine in long range deliveries, covering wide areas with minimal energy thus making them ideal for delivering critical medical supplies, such as vaccines or medications [9]. However, they are unable to hover or execute vertical take-offs and landings and thus require large open spaces for deployment and recovery [8], [10]. Hybrid UAVs merge the features of multirotor and fixed wings by enabling vertical take-off and long flight durations. Finally, multirotor UAVs are the most used configuration for their manoeuvrability, simplicity, and accessibility. With multiple rotors, these drones can hover, make vertical take-offs, and fly steadily in confined spaces [9]. Among multi-rotors, quadcopters particularly balance payload capacity and stability, making them a perfect choice for medical delivery applications. Although their short flight times and small payload capacities limit their use for large-scale deliveries, quadcopters remain a key component in rapid-response medical logistics [8], [9], [10].

2.2.2. Key Components of Drones

Upon closer investigation, it becomes evident that a drone's capacity to fly and operate properly is dependent on the flawless collaboration of its numerous interrelated pieces. The frame supports the drone’s structure. It is arms support motors, enabling manoeuvring and propulsion. To maintain stability and smooth operation, a flight controller, for instance Pixhawk or DJI, coordinates movement using sensor data and human commands. Pixhawk is a flexible platform that allows integration with many sensors and communication modules for drone navigation and stable flight. In terms of orientation, sensors such as IMUs, GPS, barometers, and cameras provide stability,

navigation, and also object recognition techniques. Communication systems allow and regulate real-time data exchange for remote operations.

2.3. Autonomous Navigation and Control Systems

2.3.1. UAV Core Components

One of the quadcopter's most appealing features is its capability of autonomous operation including navigation. For a drone, GPS always offers it the exact position, feeding important information back to the pilot or flight control software. Sánchez illustrated the features that the GPS offers to the drone which includes accurate navigation as it allows the drone to move with exactitude while maintaining its position [11]. In case of an emergency, it offers a safe return to home feature when the contact is lost, or battery is running low.

In addition, Inertial Navigation Systems (INS) board is that estimates the angular velocity and linear acceleration of the robot; hence, position can be estimated as well by knowing starting point, velocity and orientation. Likewise, the inertia measurement unit (IMU), according to Petritoli *et al.*, contains a navigation computer that performs the changes according to the purpose and sends the attitude, speed, altitude, and depth data considering different errors among the various transformations[12].

Simultaneous Localization and Mapping (SLAM) is the technical mapping procedure to create maps and locate its position in that map. Rauf *et al.* illustrated that this technique achieves both at the same time by identifying landmarks and figuring the position of the vehicle in relation to them, then explores the assigned area until it has a sufficient number of landmarks to make a map, enabling to simultaneously map a place and position itself within it [13].

Another important system is the communication systems that provides real-time monitoring, coordination, and execution of commands through interchanging of data between the drone and the ground control. Huan Lv *et al.* explored methods of monitoring drones by analysing their radio frequency communication, enabling the detection, and monitoring of drone activities above an area for security enhancement [14]. The findings hint at a feasible approach whereby RF communication patterns could be one of many techniques useful in drone detection systems toward more effective solutions in airspace monitoring and management.

Additional features include artificial intelligence (AI) and machine learning (ML) systems that can analyse sensors data to achieve optimum flight paths, predicts when drone requires maintenance and make decisions when it detects sudden environmental changes. Many research papers focus on battery and power management for it is considered the biggest limitation in the UAVs as the Lithium Polymer (LiPo) batteries are the heaviest payload. LiPo batteries are commonly used in drones because of their high energy density, which enables them to store more energy in a compact form, but their weight adds substantial load to the UAV, reducing the possible flight time and limiting payload capacities for additional equipment. This is particularly critical for medical delivery drones where precise optimization to balance battery weight with these payloads is a necessity. The systems require consistent and sufficient power supply, making distribution even more complicated. Besides, operations require real-time power monitoring especially during long flights or operation in uncomfortable conditions. These challenges are being addressed through research and development into lighter, more energy-dense batteries and enhanced power management systems that optimize energy usage.

Finally, Arafat *et al.* focused on utilizing vision-based systems to enhance the performance of autonomous UAV navigation systems as it aids it in avoiding obstacles [15] . It mentions two types of cameras; stereo and monocular vision, which can enable a drone to detect and avoid obstacles in real time. The authors have indicated the advantages of vision-based systems, their ability to work in areas where GPS is not available. They have also pointed out some challenges, like the need for robust algorithms that can handle dynamic and complex scenarios. The study has

emphasized that a vision system should be combined with other sensors to enhance the capacity of drones to recognize obstacles and avoid them.

2.3.2. Software

Ground Control Station (GCS) software is essential for drone operation since it serves as an interface between the operator and the drone's autopilot system. These platforms allow for mission planning, parameter configuration, and real-time telemetry monitoring. For drones that use modern navigation systems like GPS, LiDAR, and camera-based navigation, the GCS software serves as the command centre, allowing for smooth communication and control and improves safety by giving operators real-time data on battery status, flight route changes, and obstacle recognition.

Mission Planner and QGroundControl are two popular open-source GCS solutions for providing full functioning of autopilot systems such as ArduPilot and PX4, and both are compatible with Raspberry Pi. Mission Planner, which was built specifically for ArduPilot, excels at delivering extensive capabilities for detailed mission scripting, parameter adjustment, and diagnostics. However, it is only compatible with Windows, limiting its portability[16]. On the other hand, QGroundControl provides a cross-platform solution that works with Android, iOS, Linux, macOS, and Windows. Its mobility makes it especially useful for field operations, and its sleek, user-friendly interface simplifies mission planning with features like geofence construction and 3D visualization [17].

2.3.3. Beyond Visual Line of Sight (BVLOS)

Long-range flights will be necessary for flights over rural or underserved areas, given the capability of Beyond Visual Line of Sight (BVLOS) operations to fly a drone farther than what the operator can see. Davies, L. *et al.* emphasizes conformity to the legal criteria on the aspects of BVLOS in terms of safety measures, autonomous flight systems, and remote monitoring, which also emerge as significant since such aspects would directly relate to certification and operating

approval procedures of drones [18]. As part of the process, the regular assessment of vision systems, radar, and LiDAR makes sure that drones will be able to navigate through difficult situations autonomously and avoid obstacles. Communication technologies are also considered in the BVLOS flight certification procedure where real-time data is exchanged between a drone and a ground station. Compliance with BVLOS laws has to be done for public protection and confidence in drone technology.

2.4. Payload Delivery Mechanisms

2.4.1. Design Considerations

Payload is considered to be a major section for any type of delivery drone for significantly impacts the stability of the drone affecting the choice of motors, propellers, and frame measurements. It includes the weight of the components and the varying delivery payload. Thus, designing a stable mechanism involves weight distribution, stabilizing the centre of mass for the drone, and using algorithms to modify flight dynamics [19]. In addition, the aerodynamics drag force can significantly impact its flying range and efficiency as drone speed increases, drag force on motors increases which cause higher loads. Designers tend to simplify delivery payload to minimize drag force and achieve the required payload [20].

Another factor to consider while designing the payload delivery mechanism is how to load it and unload it without causing distribution issues. This may involve using specific features like clamps, hooks, rectangle cargo compartments or grippers that can be either controlled by the user or by a program integrated into the drone. Designers tend to choose materials like carbon fibre and lightweight metals for durability and structural integrity while reducing weight [21]. Moreover, payloads ought to be designed for extreme weather conditions, such as strong winds, and include impermeable casing and waterproof coatings [22].

2.4.2. Medical Control

Medical payloads, including equipment, medicine, and vaccines, require specific conditions to ensure their effectiveness including temperature regulation. Drones with active cooling systems or thermally insulated compartments are used to deliver these items without jeopardizing quality [23]. Designing cargo compartments for shock absorption and vibration resistance is essential [5]. Another aspect to be considered is the sterilization of medical equipment and ensuring environmental stress during the delivery while protecting the payload from contamination [24].

2.4.3. Drop of Mechanisms

The drop of mechanism is a crucial part of unloading a payload with a variety of techniques used depending on the payload type and efficiency. One common mechanism is winching drop, which allows the drone to drop its load in the air by using a cable that is attached to lower the payload to the ground as shown in figure (1). However, the technology renders the drone more substantial and complicated, which makes it less effective at flying [25].

Figure 1 Winch Drop

Another method being used when the drone is able to land first when it arrives at this district destination then releases its package using a gripper as shown in figure (2). However, it requires smooth landing which makes it less suitable for uneven or blocked terrain that can result in damage for the drone as well as extending delivery time and consuming more energy [26].

Figure 2 Handoff Landing Using Gripper

Lastly, parachute dropping is another method where a parachute is released to gently lower the payload to the ground as seen in figure (3). Where the parachute is released by lowering the load down to the ground when the destination is reached, this saves energy for the drone to land. Also, this method is frequently utilized in isolated or rural locations but less precise in windy conditions [27].

Figure 3 Parachute Dropping

2.5. Challenges and Future Trends

Autonomous UAVs have a high potential in facing technical, safety, and infrastructural challenges. The technical challenges include low battery capacity and exposure to harsh flight environments,

and susceptibility to hacking attacks [7], [28]. Safety and security are critical issues, as hackers can target drones using signal jamming, eavesdropping, GPS spoofing, Denial of Service (DoS) assault, and hijacking techniques [7]. For the drone services to gain more reliability, it would require the use of new battery technologies, better safety measures, and a transparent regulation process.

When talking about future trends, it is important to note that the new highly developed propulsion technologies, including hydrogen fuel cells and solar-powered drones, have a huge effect in transforming drone technology by offering longer flight times and reducing dependency on fossil fuels [28], [29]. AI and machine learning have brought to the market with the latest technology drone the real-time route optimization, avoiding collisions, and enabled the machines to make decisions as to which path will be the best to follow alone hence those drones are more efficient in use in urban settings and precision agriculture [30] . These companies e.g. UPS and Amazon are among the companies that are currently using AI-enabled drones to deliver items faster and cheaper (thus more economically) while advanced autonomous drones are used in the inspections of infrastructural issues in order to reduce human risk.

2.6. Conclusion

Quadcopters' technological development brought new dimensions in the various fields like health sectors, disaster management, and logistical operations. The applications of drones range from navigating through complex areas with GPS and other sensor systems to accomplishing payload release mechanisms that require precision for life-saving medical supplies, hence ensuring their usefulness to solve practical challenges. On the other hand, series challenges remained over limited battery capacity, safety susceptibilities, regulatory barriers and infrastructural constraints. Therefore, development and research emerged trends to improve battery technologies, AI-driven navigation systems, and adaptive payload mechanisms that promised to surmount many of the limitations found today. By aligning technological innovations with practical applications, UAVs played a transformative role in the creation of safer, more efficient, and sustainable solutions for

critical sectors. As research and development continue forging ahead, drones will increasingly become key tools in addressing global challenges by closing the gap between technological capabilities and real-world needs.

Chapter (3) Mechanical Design

3.1. Introduction

The mechanical design of the quadcopter is an important feature of its development. It makes sure that its structure is robust and vigorous, has aerodynamic efficiency, and key components function together harmoniously. A good mechanical framework improves stability, tolerates payload, and is durable enough, which are essential factors for the quadcopter to operate as an emergency medical transport.

The objectives of the design are to optimize the distribution of the structure's weight and to enhance the structural strength, as well as to ensure efficient propulsion integration. The most important technical problems are the vibrations that might affect sensors onboard, stability when the quadcopter is under different payload circumstances, and the aerodynamics that should be optimized to reduce the drag and improve the flying efficiency. Such problems were solvable through the choice of material, computational fluid dynamics simulations for aerodynamics refinement, and some iterative design enhancements.

The technique of the design included the conceptualization part, where different structures based on aerodynamics, weight, and manufacturability were taken into consideration. The design that was chosen was polished through CAD modelling, finite element analysis (FEA) for deformation analysis, and strenuous material selection, which was implemented carefully to balance weight and strength efficiency. Structural testing of the real world was also carried out to verify the behaviour of the material under operational conditions.

This chapter describes the mechanical design procedure, which begins with the exploration of the conceptual design, then it proceeds with structural optimization and tail validation processes. The results of this phase are the most crucial in the development of a reliable quadcopter that can fly and deliver medical supplies effectively.

3.2. Conceptual Design

3.2.1. Body

In order for the medical drone to effectively fulfil its function of delivering medical kits, its design is one of the most critical aspects. The design must prioritize high speed, accuracy, and precision to ensure the drone can reach its destination swiftly and save lives. One of the key components to consider in the design is the drone's base. The base serves as the foundation, carrying all the essential components of the drone, from the structural parts to the navigation and mobility systems. The base can take on various shapes and forms, which are selected based on specific criteria and the number of components each drone requires.

3.2.2. Arms

For the drone to achieve flight, the propellers and motors must be positioned correctly. This is where the arms play a fundamental role, as they connect the body of the drone to the motor mounts. Each arm is designed to distribute thrust as efficiently as possible, ensuring balanced lift and precise control during flight. Together, the base and the arms are two essential components of the drone's design, each playing a crucial role in maintaining stability, manoeuvrability, and overall functionality. In addition, its being orientated in the true X shaped. Utilizing symmetrical arm placement at 90-degree angles, as the chosen design ensures consistent load across all motors and balanced thrust distribution, improving performance and stability. As its symmetry facilitates easier control, increases yaw control, and improves flight dynamics.

3.2.3. Gripper

The mechanism that is responsible for carrying the packages is usually not the main concern of most researchers as they prioritize the flight time, having a battery efficiency system and reducing the overall weight of the drone. Not only this but also each research paper addresses different scenarios for the drone that lead to the various designs seen today.

Different techniques that were done so far raised concerns, like dropping a kit using a parachute [31] having a gripper that holds a hanger attached to the kit or having a box that carries the package inside it [32], [33]. Thus, all these methods face the risk of accidental collision into obstacles, package sliding, and medical supplies that get damaged while landing.

Hence, this paper investigates the importance of delivering the medical box safely by designing a secure gripper that ensures a firm grip of the kit. Erlingsson *et al.* [34] developed a current robotic claw used for a five-axis CRS robotics A255 arm for achieving compact, lighter weight and more precise grasp. It is actuated by a servo motor and a simple mechanism integrated into the design to provide an optimal gripping force that would make the manipulation solution ever more capable and effective.

3.2.4. Landing Gears

Landing gears are considered a load-carrying part of the drone where it is very critical to safeguard the air frame to protect it against any kind of damage. Small to medium UAVs undergo different scenarios such as endurance, poor performance of landing, and high rate of failure. Thus, selection of the right landing gear for all kinds of situations will decrease all these limitations [35].

This project designs a multipurpose delivery drone that transports medical care from one hospital to another or to any location that contains patients in serious need for the medical care so to avert the road traffic, drones can be sent instead of an ambulance. Therefore, the drone is expected to land in complex and bumpy roads, hence adaptive landing gears were found suitable in this situation.

Adaptive landing gears contain joints to adapt to uneven ground. Researchers utilized sensors along with AI systems [36], servo motors [37] and actuating cylinders [38]. These techniques

offered many benefits to the concept; however, they were not selected for the aims of this project which include achieving a battery efficient system, quick landing and decreasing the weight.

Baker *et al.* [39] explores the application of a passive landing gear system that would absorb the landing forces by means of connected mechanical links. Systems without active components can be designed to increase stability and stress absorption during the landing of a vehicle, resulting in weight reduction and less maintenance. For this reason, it inspired this project to design passive landing gears for safe and quick landing on uneven roads.

3.3. Detailed Design

The following figure (4) showcases the final design of the autonomous quadrotor.

Figure 4 Final Drone Design.

While the subsequent figure (5) illustrated parts of design in more detail.

1 & 17: Frame

2: Arms

3: Rod

5 & 6 & 10 & 11: Landing Gears

14: Motor Mount

14: Propeller Guards

16: Carbon Fiber rods

20: Gripper

21: Propellers

24 & 25: Raspberry Pi Camera Setup for obstacle avoidance

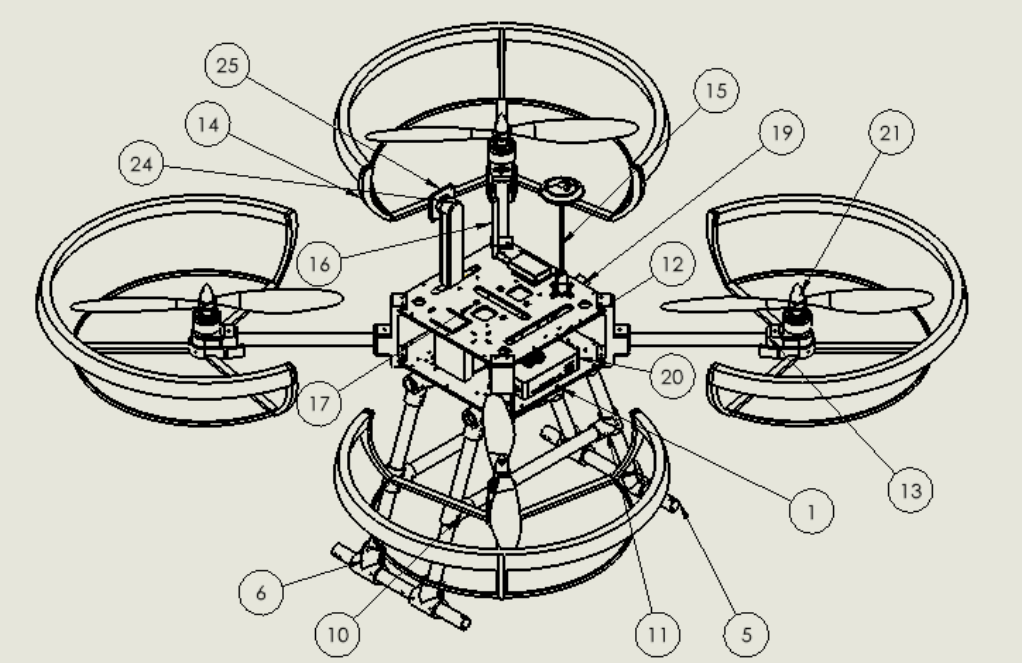


Figure 5 Drawing Sheet assembly of only the main parts.

3.3.1. Frame

The swift distribution of the required supplies for emergency medical conditions might mean the difference between life and death. A game-changing innovation, medical drones provide quick and dependable delivery of vital medical supplies to isolated or disaster-affected locations. A lightweight construction for longer flying hours must be maintained while speed, precision, and stability are given top priority in the drone's design to ensure optimum performance.

A critical aspect of this design is the frame, which serves as the foundation for all components, including navigation and mobility systems. due to its rigidity and optimal weight distribution, the True X frame distinguishes out among other framework configurations and is especially well-suited for medical delivery drones. A number of significant benefits are offered by this asymmetrical design, which has arms arranged at 90-degree angles to create a "X." A key component of the drone's design, the True X frame provides optimal thrust distribution and consistent mass distribution, greatly improving the aircraft's stability, manoeuvrability, and power efficiency. To guarantee dependable operation during medical delivery missions, these qualities are essential. To optimize the drone's payload capacity, battery life, and overall flying efficiency, weight reduction strategies are also used. Lightweight materials, such carbon fiber or premium aluminium, are used to reduce bulk without compromising the frame's structural integrity. The frame's design strategically relieves weight, reducing superfluous bulk without sacrificing durability. Additionally, simplified electronics, effective batteries, and small motors are integrated to guarantee peak performance and energy efficiency. Strength, efficiency, and agility are balanced in these design concepts to ensure the drone's efficacy in providing emergency medical supplies in a timely and dependable manner [40], [41], [42], [43].

To begin the design process, the first step was to determine the overall load, which was calculated to be 1.5 kg after accounting for all navigation components, the payload of the medical kit, and all designed parts. The frame size was then determined by calculating the diagonal distance between the centres of two opposite motor mounts, which resulted in a 600 mm frame size. To calculate the required thrust, the total weight of the drone was multiplied by gravitational acceleration (9.81 m/s^2), yielding a required thrust of 14.75 N. To ensure stable flight, a thrust-to-weight ratio of at least 2:1 was considered, meaning the selected motors needed to generate a combined thrust of 29.43 N. Distributing this thrust across four motors resulted in approximately 7.36 N per motor. When divided by gravitational acceleration, the required thrust per motor corresponded to 0.75 kg. Based on these calculations, the most suitable motor was an 800 kV brushless motor, paired with 13-inch propellers to generate the necessary lift efficiently.

3.3.2. Base

Regarding the structural design, the drone's two bases were designed in a square shape to accommodate all required components while maintaining compatibility with the X-shaped frame with 90-degree arm positioning. Each base was equipped with designated enclosures to securely hold components during flight. The bases were cut out from cladding material, ensuring durability while keeping weight to a minimum. Each base had a diagonal length of 200 mm, maximizing space utilization. To further optimize component arrangement, both sides of the bases were utilized to house different elements, as shown in Figures (6). For the total length of the drone ought to be roughly 600mm diagonally.

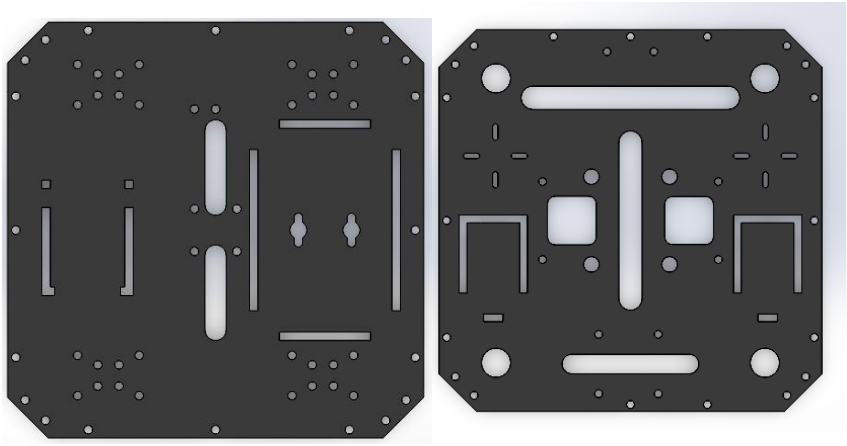


Figure 6 Lower Plate and Upper Plate.

.

3.3.3. Arms

For stability and structural integrity, the arms and bases were connected using bracket-shaped parts, which secured the arms to the frame with screws (Figure 7). These brackets were also made of ABS material, ensuring efficient wire management, and reducing complexity. Considering the distance of the two base plates, its dimensions are designed to be around 88mm.

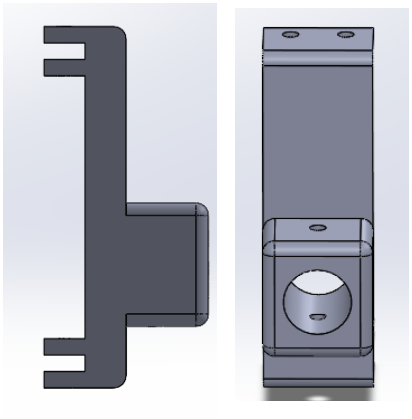


Figure 7 Brackets.

The arms were constructed from carbon fiber tubes, providing a balance of strength and lightweight properties. Each cylindrical arm had an outer diameter of 16 mm, an inner diameter of 14 mm, and a length of 200 mm. Finally, the motor mounts, responsible for holding both the motors and propellers, were designed and manufactured from ABS material. These mounts were securely attached to the carbon fiber arms using screws, as illustrated in Figure (8), in addition to a cut slot at the bottom for the wires to be inserted.

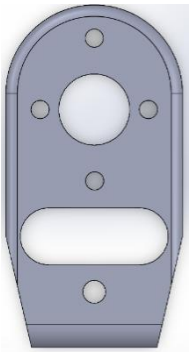


Figure 8 Motor Guards.

The combination of lightweight, durable materials and an efficient assembly design ensured that the drone could achieve high speed, stability, and precise control while delivering medical kits efficiently when assembling the arm as shown in Figure (9).

Figure 9 Arm Assembly.

3.3.4. Gripper

The gripper mechanism is designed to provide a stable and secure grip on the medical kit. Therefore, the gripper is engineered to carry the entire box and is controlled by a 360-degree rotating servo motor, which only releases its grip upon the drone's safe landing to ensure reliable handling. Hence, the following figure shows the design of the gripper while considering the dimensions of the medical box's size while also accounting for the vacancy imposed by the adaptive landing gears that can carry weight and height of medical kit around 19.6 x 8 cm, as shown in figure (10).

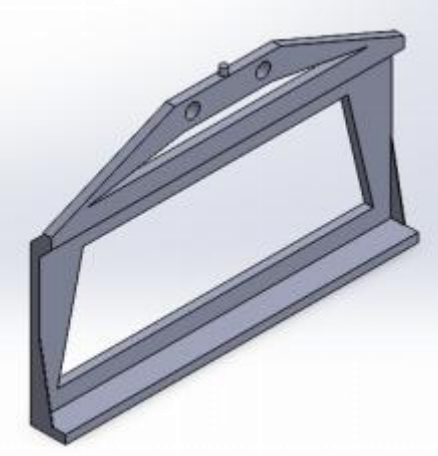


Figure 10 Gripper Arm.

The servo motor is housed within the base shown below, which is designed based on the motor's dimensions. The selection of the servo motor is influenced by both its weight and the weight of the medical box where it carries the exact dimensions of the servo motor utilized in this case 40x32.94 mm, as shown figure (11).

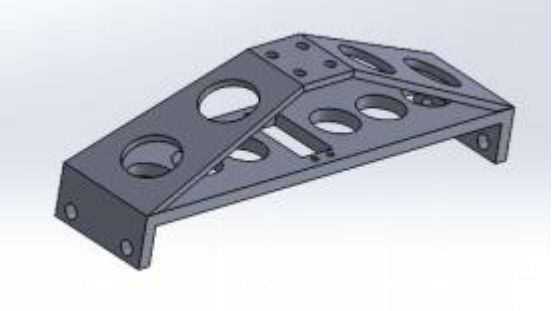


Figure 11 Servo Base.

A cylindrical structure where the gripper arm slides, allowing it to firmly grasp the medical box. That is designed to be 200mm Figure (12).

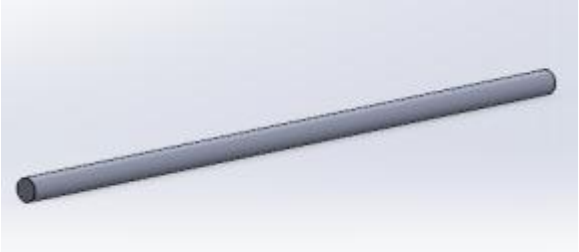


Figure 12 Cylinders.

The subsequent figures are the two links utilized. Figure (13,14) connects the gripper arm to the servo, enabling controlled motion. While the other, links the servo to the supporting structure, ensuring a secure and efficient transmission of motion.

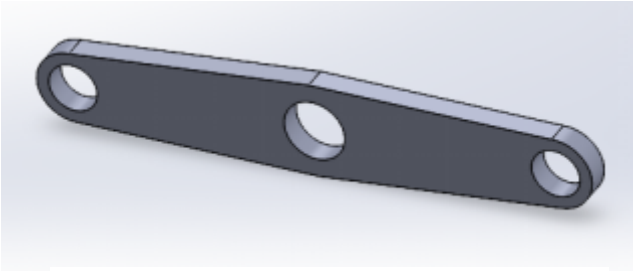


Figure 13 Gripper Arm Link.

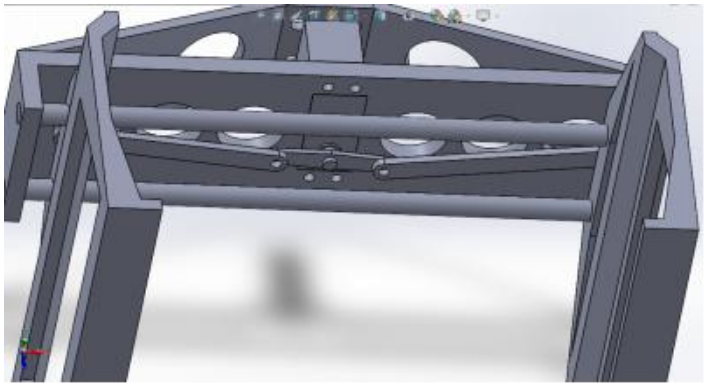


Figure 14 Servo Link.

The following figure shows the complete gripper mechanism, integrating all components to achieve secure grip and controlled release figure (15).

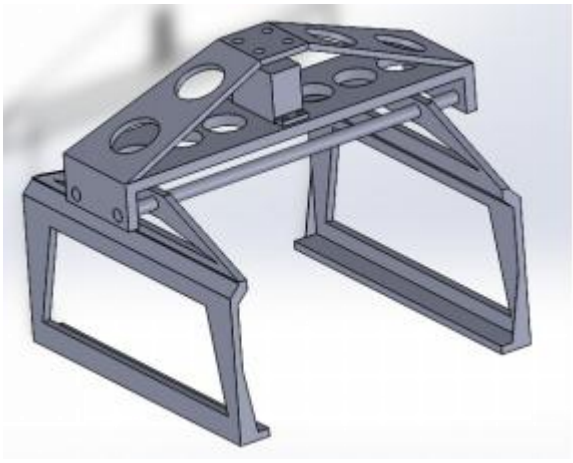


Figure 15 Gripper.

Despite all that, the gripper design had to be further simplified to reduce the overall weight. The figure below illustrates the final design of the new gripper mechanism that was ultimately adopted. This mechanism also employs a servo motor, which is connected to two mechanical links. Upon the drone's safe landing, the controller sends a signal to the servo motor, prompting it to release the two links. These links then slide open, allowing the handles of the medical box to be dropped smoothly, figure (16).

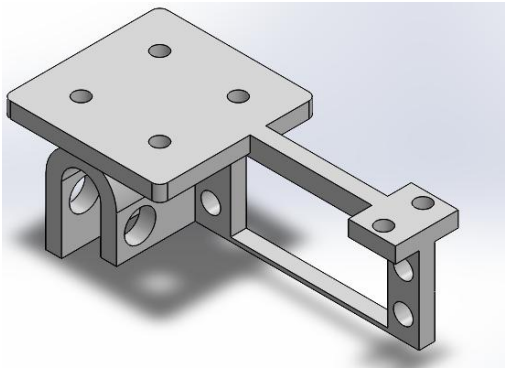
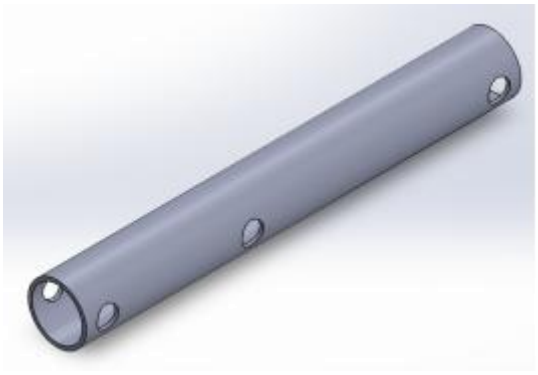


Figure 16 Final Gripper Design (Dropping Mechanism).

3.3.5. Landing Gears

Adaptive landing gears enhance the drone ability to land in uneven terrain by incorporating lightweight and flexible mechanical links and structure. The design sheds light on the efficiency and adaptability to ensure stability during landing. Carbon fiber rods are once again selected for their excellent strength-to-weight ratio, making them ideal for supporting the drone's landing gears. As demonstrated in Figure (18), these rods provide essential structural integrity while maintaining a height that remains above the gripper to prevent interference during landing and take-off.



Figures (19), (20) and (21) are the total links utilized in the design; each serving a distinct function. The first one is responsible for connecting and supporting the fiber rods to ensure structural stability. Link 2 is utilized as a guiding mechanism that restricts movement to provide the controlled motion for the adaptive landing gears. Last, link 3 facilitates the mechanism by incorporating a spring that moves the pin within the hollow sections in both links 2 and 3 Figures (22).

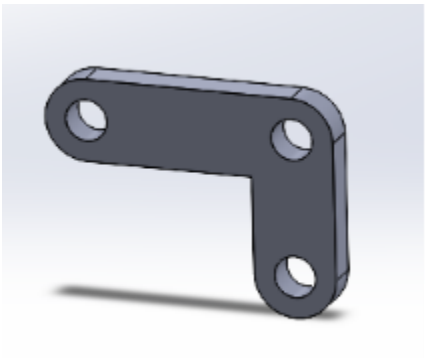


Figure 19 Link 1

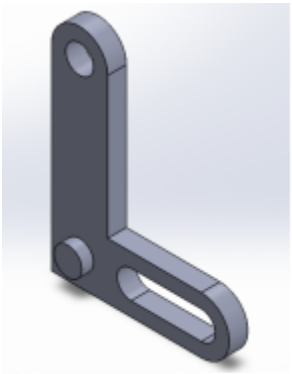


Figure 18 Link 2

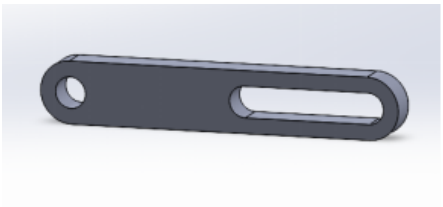


Figure 17 Link 3

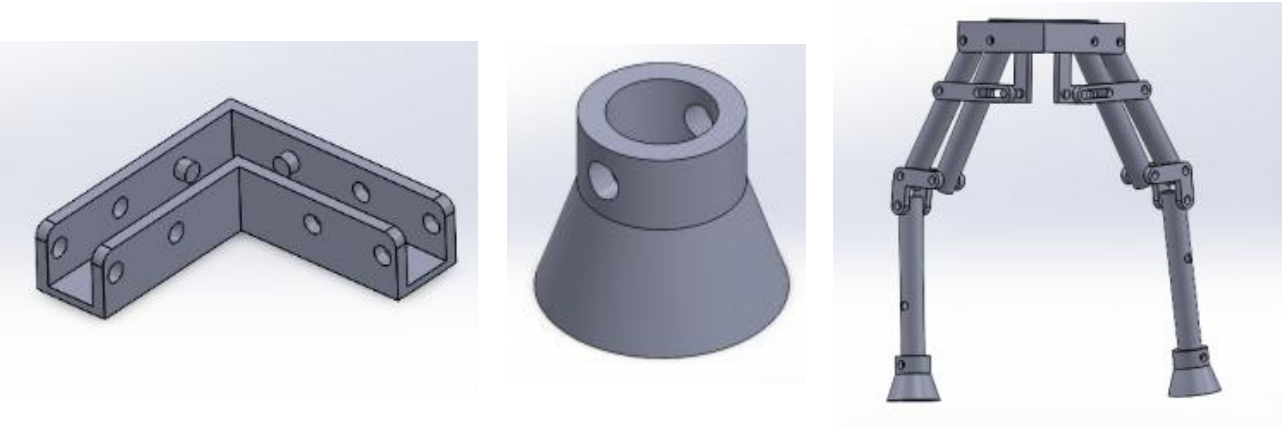


Figure 20 Base, Rubber Leg, Landing Gear.

Figure (23) showcases the base that serves as the attachment point between the base of the drone as previously stated and the assembly of the landing gears. It ensures alignment as it holds the rods and link 3.

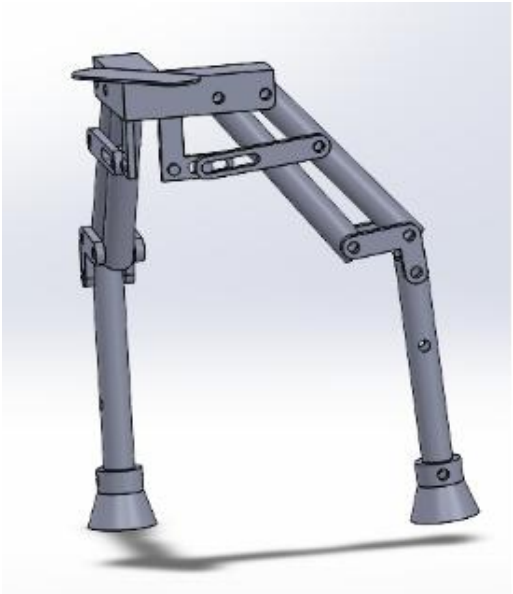


Figure 21 Landing Gears.

The final component is the leg of the drone, which makes the first contact with the ground during landing. This part is constructed from rubber and friction pads, allowing it to absorb impact forces and endure sudden collisions, enhancing durability and overall landing safety.

Although the features that was stated previously had a great contribution to the landing mechanism as it will adapt to landing on uneven trances and step environment, it has increased the payload weight of the drone. That is why a simpler lighter designed was integrated in this design where the base of the second plate is designed with sot for integrating regular shaped tubes connecting to a landing base rod like shaped and in order to modify weight and mention the stability and is coated with foam like material to smoothen the landing process and absorb the landing impact as to protect the drone from any sudden force.

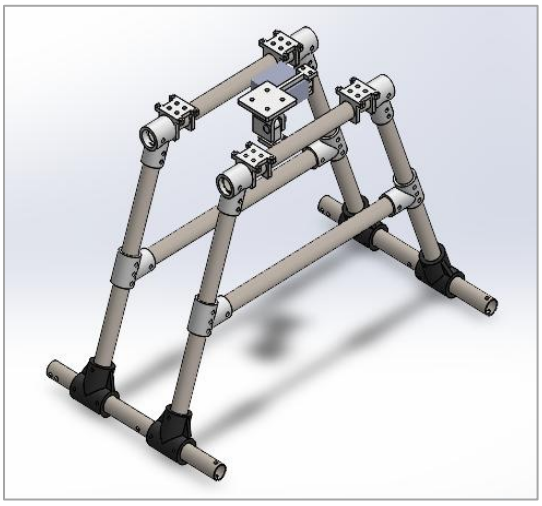


Figure 22 Finalized Landing Gears.

3.3.6. Box

In order to facilitate the transportation of vital medical supplies like blood units, vaccinations, or first aid kits, a custom-designed payload box was created to be integrated with the medical drone. The box is designed to hold up to 1 kilogram of cargo and has dimensions of 190 x 140 x 70 mm. The dispatcher or an authorized online interface transmits the password, which the user must enter to activate its secure locking mechanism. The box has built-in temperature and humidity sensors to provide ideal storage conditions. A fan is automatically turned on to control the environment if the interior temperature rises over the safe level. Furthermore, an integrated weight measuring system continually checks the load and alerts the user if the payload above the 1-kilogram limit. The micro controller that was used in this box is the Arduino nano. This structure guarantees the dependability and safety of delicate medicinal delivery.

3.3.7. Propellers guards

Since, one of a drone's vital parts is its propulsion system, which includes the propellers and motor mounts. The implementation of preventive measures was necessary due to the potential for impact or side collisions during operation, particularly in unanticipated or emergency conditions. That is why A straightforward but efficient guard was created to surround and protect the engine mounts and propellers in order to solve this issue. Due to the drone's longevity and dependability during missions are improved by this protection, which reduces the possibility of harm from unintentional touch or collisions. Figure (25) shows the design in action.

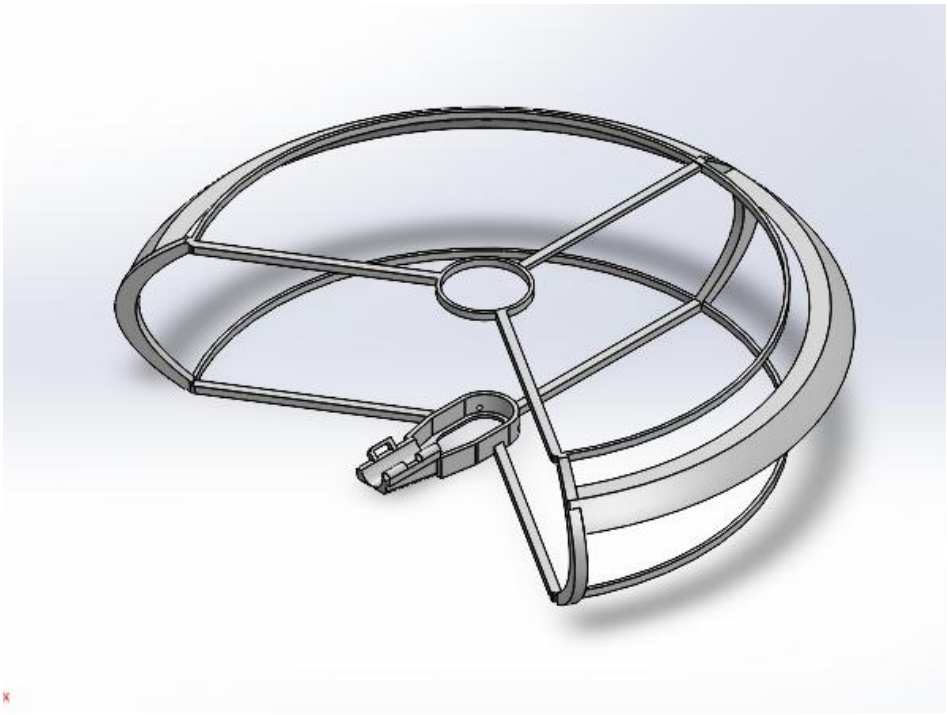


Figure 23 Propeller Guards

However, it proved to be heavy weighting around one kilogram when selecting a durable material as ASA. Hence, the following figure shown the final guard’s design.

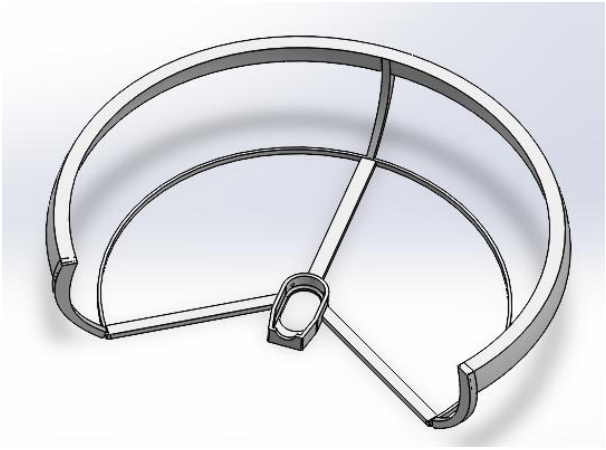


Figure 24 Finalized Propeller Guards

3.3.8. Battery holder

A specific construction for the battery holder was created to meet the design objective of attaining maximum adaptability. therefore, this holder is made to firmly support the drone's battery while it is in flight and fit snugly in the middle of the drone's frame. making it adaptable, user-friendly mechanism makes it simple to insert and remove the battery, greatly streamlining the process of charging and replacing it. This modular strategy will result in lowers mission downtime and improves operational efficiency, Figure (27).

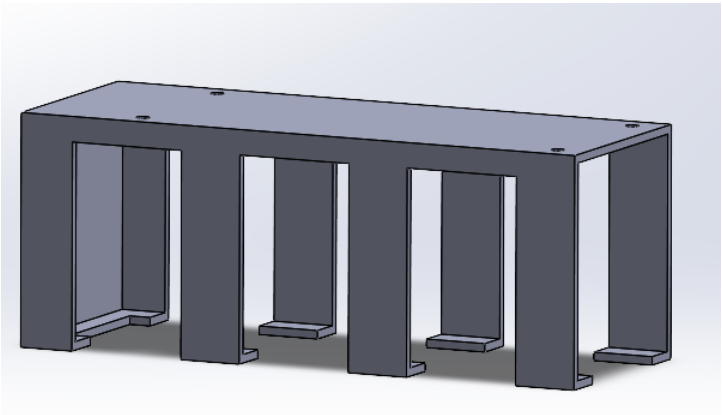


Figure 25 Battery Holder

3.3.9. Sides

To shield the drone from lateral collisions, bracket-shaped side panels were created as part of the protective elements built into the drone's design. These side brackets are made to firmly clip between the frame's top and lower plates, creating a strong structural barrier around the sides. In the case of side accidents or hard handling, this arrangement helps absorb shocks and prevents damage to internal components, increasing the drone's overall longevity and dependability figure (28).

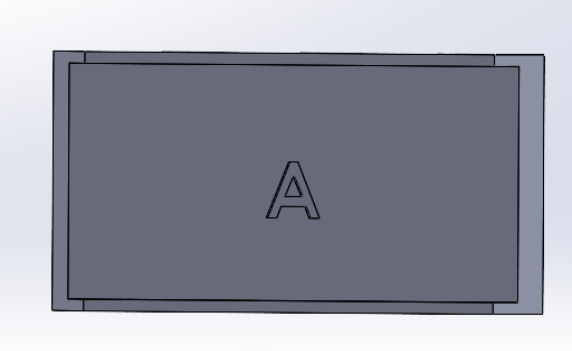


Figure 26 Sides Cover.

3.3.10. Camera Holders

As the drone will depend on the visualization for both the navigation and the landing techniques camera holders was designed to be placed on the lower plate offset with a specific angle with defend distance. the camera holder is required not only for holding the camera but to help with stabilizing its orientation in the air acting as a gamble, figure (29,30). Not only this but also, figure 31 illustrates the RaspberryPi camera setup used for obstacle avoidance where it is set in the front and up high for the propellers guard not to be in the way.

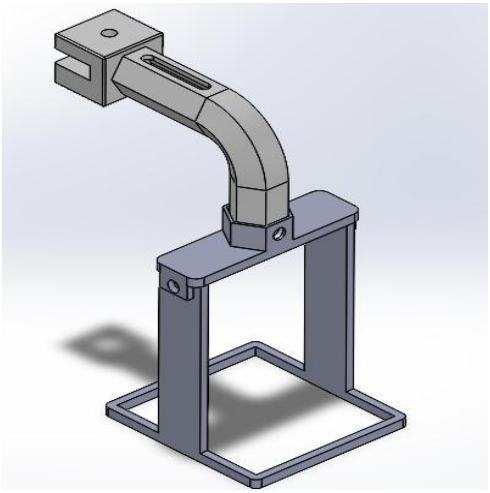


Figure 27 Camera Setup Attached to the Base for illustration.

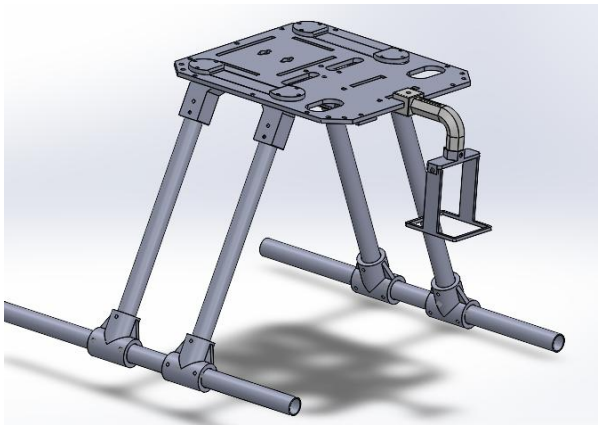


Figure 28 Camera Setup for Landing Feature.

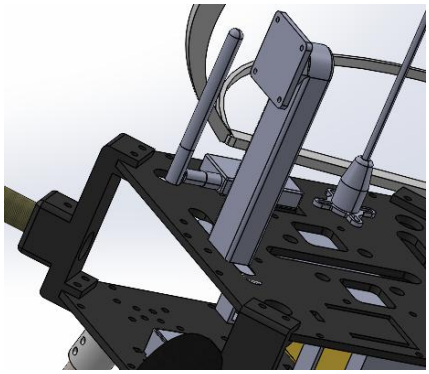


Figure 29 Raspberry Pi Camera Setup for Obstacle Avoidance

3.3.11. Material Selection and Component Specifications

We designed the drone considering the mounting of each component, so it provides the best placing with respect to other components, so it is organized in a way that provides good ventilation for each component and the shortest wiring possible. The components are distributed on two plates upper and lower one. The microcontroller, GPS and telemetry are mounted on the top plate while the battery and distribution board are placed on the bottom of the upper plate. While the companion computer and the receiver are placed on the lower plate. Finally, the landing gears and gripper are connected to the lower plate. Our design also provides space for any new components that might be added in advance. Taking into consideration the materials to provide the required strength while maintaining low weight as each extra gram counts. Carbon fibber was perfect for the drone arms

and the landing gears as it provides high strength with ratio to its light weight. Regarding the upper and lower plates aluminium cladding was chosen as it is lighter than pure aluminium and provides solid strength. The other links and the gripper will be 3d printed from PLA plus material that withstands weights despite its light weight.

3.3.12. Quadrotor Structural Analysis

A vital step in ensuring robustness, stability, and functionality in operational conditions, particularly for an autonomous emergency drone where safety is paramount, is the static structure analysis of the quadcopter frame. Three analyses were conducted using ANSYS: total deformation which measures how much the structure displaces under load and indicates the extent to which components bend or stretch; equivalent elastic strain assesses the stress distribution and identifies areas where the material may yield or fail; and the distribution of von Mises stress that illustrates deformation in relation to the original dimensions, offering insights into the material's elastic behaviour under stress [44] Figure (30). The structure of the frame is composed of carbon fiber rods and SLA (Stereolithography)-3D-printed plate connectors, selected for their unique set of mechanical properties. Carbon fiber was selected for its high strength-to-weight ratio, high rigidity, and resistance to deformation, making it a material of suitability for drone frames in that weight needs to be minimized without loss of strength [45]. The SLA material was selected for their precision and capability to produce complex geometry to enable accurate assembly and fitment. In that SLA resin is generally weaker compared to carbon fiber, though, stress concentration points require careful design consideration [46].

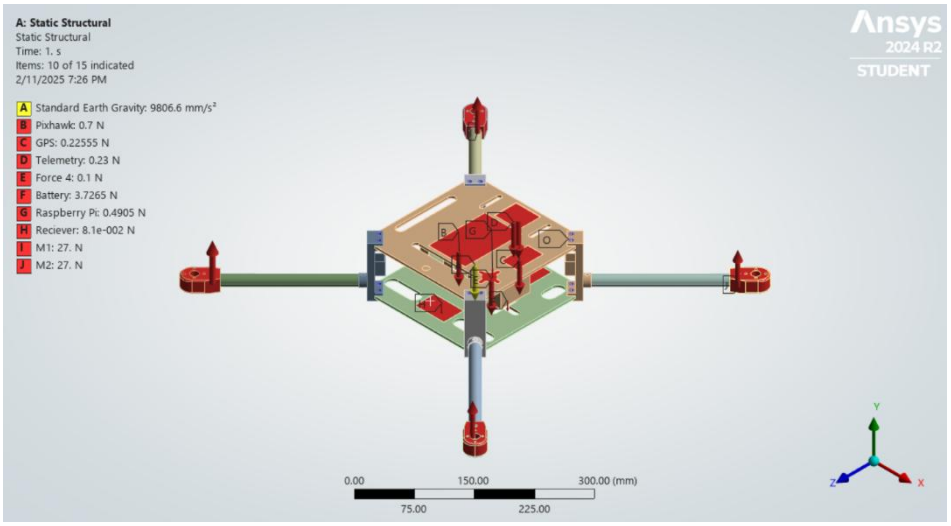


Figure 30 Forces Applied to Drone Frame

For realistic modelling of actual working conditions, as shown in figure (31), an upward force was applied to the motor mounts to simulate the thrust force created by the motors during maximum thrust (full throttle). This is to enable analysis to account for the drone's worst condition, in that the frame is exposed to maximum stress and maximum deformation. The total deformation analysis demonstrated, indicated a maximum of 4.558 mm of deformation, concentrated largely in the motor mounts. This is to be expected, in that maximum force is created by the motors, and such areas carry maximum loads. The carbon fiber rods, though, displayed low bending, proving their efficacy in ensuring structure of the frame is upheld in maximum-throttle situations.

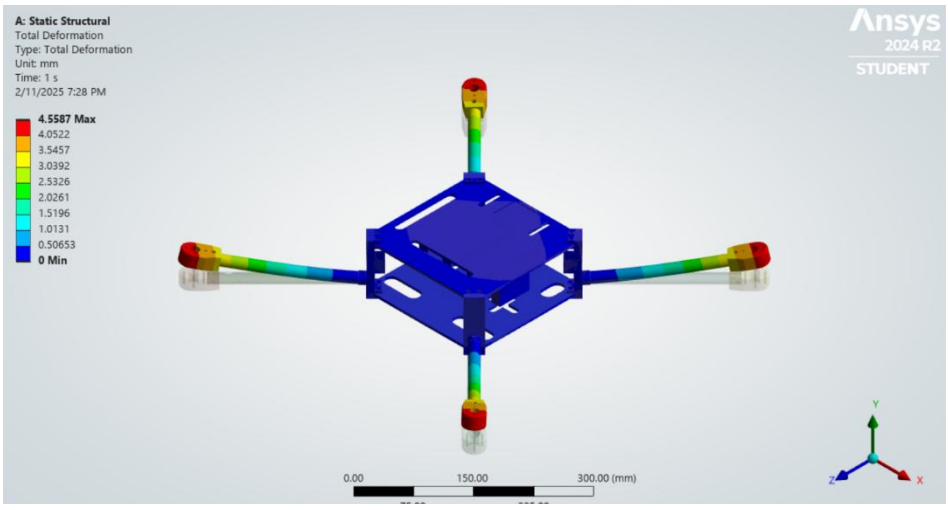


Figure 31 Total Deformation of Quadrotor Frame

The equivalent elastic strain analysis came up to a maximum of 0.0046061 mm/mm of strain as represented in figure (32), predominantly in SLA connectors. The connectors exhibit higher compliances that lead to higher strain levels compared to carbon fiber rods. Flexibility is advantageous in damping small vibrations, though excessive strain would lead to material failure or material fatigue over time.

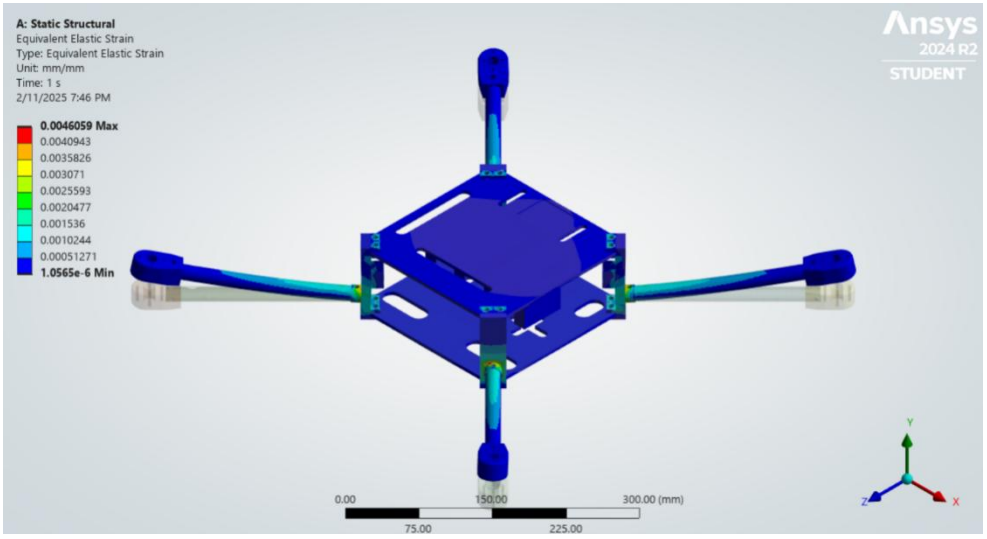


Figure 32 Equivalent Elastic Strain Analysis of Drone Frame.

Figure (33) shows the maximum stress in von Mises analysis came up to 108.56 MPa, maximum stress concentration around joints of connectors and around bolts, given that these are load-bearing components of the structure of the frame. Such stress levels, though high, remain in their material tolerable limits, keeping the structure of the frame in place. The maximum observed strain is 0.00090958 mm/mm as shown in figure 30, a low one that indicates material is elastically responding to applied load.

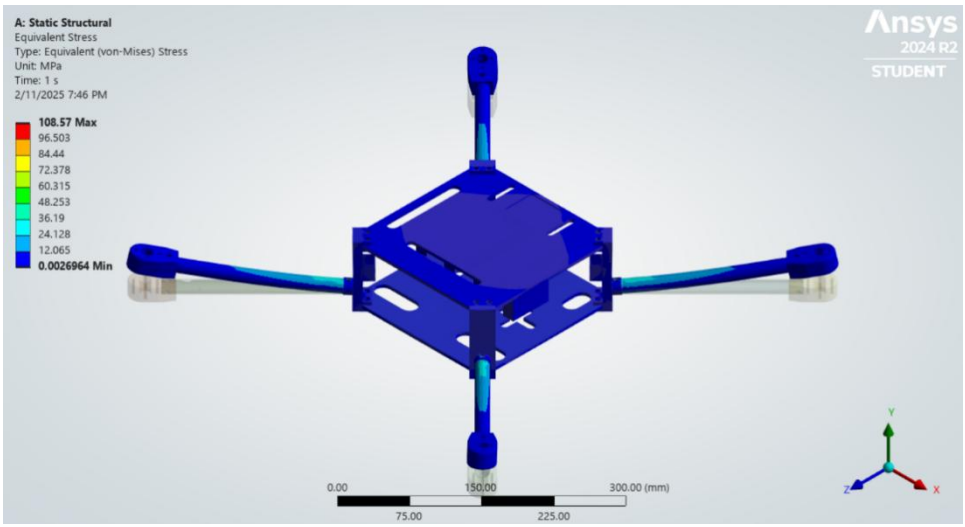


Figure 33 Equivalent Stress of Drone Frame.

3.4. Discussion and Conclusions

The static analysis of quadcopter structure frame and structure of gripper mechanism verifies material and structure adequacy. For carbon fiber quadcopter structure rods exhibited high strength of stiffness and low deformation, ensuring flight stability in maximum throttle condition. Furthermore, the plates holding all the vital components demonstrated minimum deformation and experienced stresses, ensuring a good base for the components be configured. On the other hand, the SLA 3D printing connectors, dimensionally accurate though weightless, exhibited high concentration of stress and strain in joints and motor mounts, suggesting strengthening requirements in such areas. The analysis of the gripper mechanism verified that SLA printing structure is sufficient to carry a one-kilogram medical relief box safely with deformation and stress way below material limits. The low stress and von Mises stress levels verify reliability of the gripper in working loads, ensuring sustainability without material failure risk. Likewise, the low

readings of stress confirm that the design is sufficient to hold working loads. The results in total verify that selected material and structure design is up to specifications for function. However, stress-prone areas in quadcopters and grippers can be minimized using strengthened material or design improvement to provide more durability. Such improvement would make the emergency drone system more resilient and more reliable in actual rescue missions, ensuring sustainability and safety in the long term in high-stake applications.

Chapter (4) Electric and Electronic Design

4.1. System Overview and Objectives

4.1.1. Role of Electronics in the Drone Platform

Electronic systems integrate several systems of an UAV including power, propulsion, navigation and communication, flight control, and payload systems which enable the key functions of flight, navigation, and communication. For a quadcopter to deliver medical aid, designing the electronic systems ought to have four primary design considerations: affordability, components compatibly for fast development, instantaneous data, and easily operated [47].

In terms of flight, the power system of the UAV is the power supply to the electronic system as a whole where UAVs, particularly quadrotors, frequently use battery power for its simplicity and adaptability. The power system, as seen in figure 1, feeds the propulsion system, whose primary function is to transform electrical energy into the mechanical power that drives the UAV through a propulsion unit [48]. To enable smooth navigation, which includes obstacle avoidance and precise landing, the navigation system determines the position, orientation, and direction of movement of the drone to ensure a controlled stable flight and allow the drone to follow a route, return to a starting point, and hold a position [49] Figure (34).

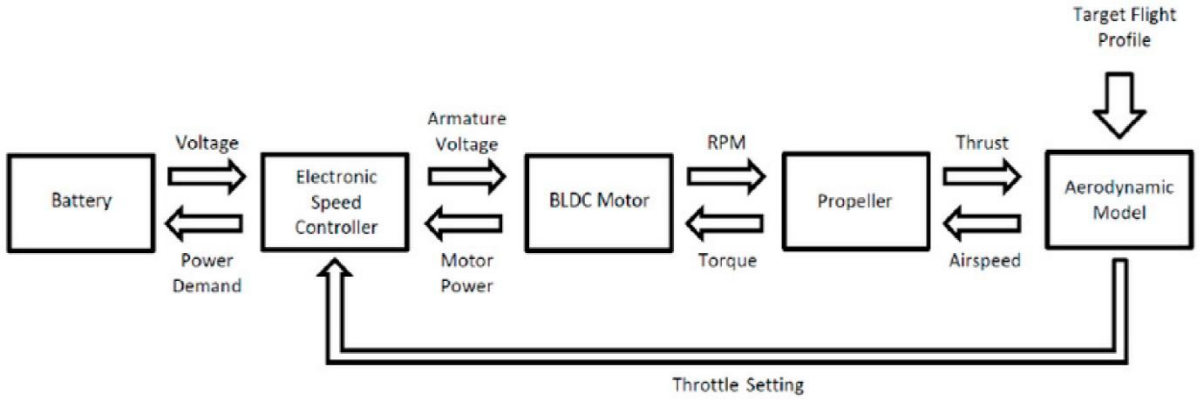


Figure 34 Power and Propulsion Systems' Block Diagram [48]

The flight control system as shown in figure (35) is what takes drone parameters and adjustments them to allow the drone to reach the setpoint. For instance, it can change thrust and attitude to reach a goal position while respecting vehicle-specific restrictions as force commands are converted into motor commands, an operation enabled by the mixer.

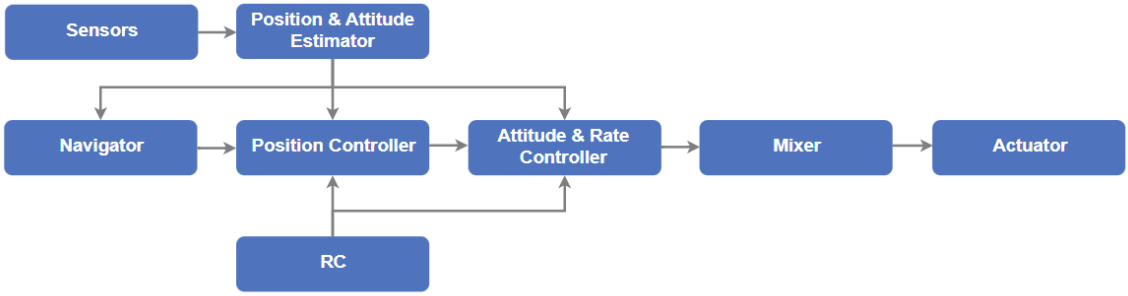


Figure 35 Power and Propulsion Systems' Block Diagram [48]

Lastly, the UAV ought to have a ground station for the controller to follow the state of the drone and to be able to send and receive data through telemetry, such as position, battery level, speed, and altitude, to the drone in case of any uncertainties. Furthermore, during missions, onboard cameras and video transmitters offer real-time visual feedback that improves operational control and situational awareness [50].

4.1.2. Design Objectives and Requirements

The quadrotor ought to have sensors and controllers to achieve a safe, stable, and task-oriented flight making it necessary to select compatible sensors and controllers. Figure 34 emphasizes the components and functions required to obtain the desired output. In order to maximize performance, efficiency, and flight endurance, factors including battery capacity, flight range, navigation systems, safety features, and payload handling mechanisms were carefully examined [51]. However, to reach inaccessible places fast, the drone size ought to be a small-sized drone that does not exceed 6 kg. It is also required to consider low-powered consumption components to allow the drone to have a long travel time range as shown in Figure 34.

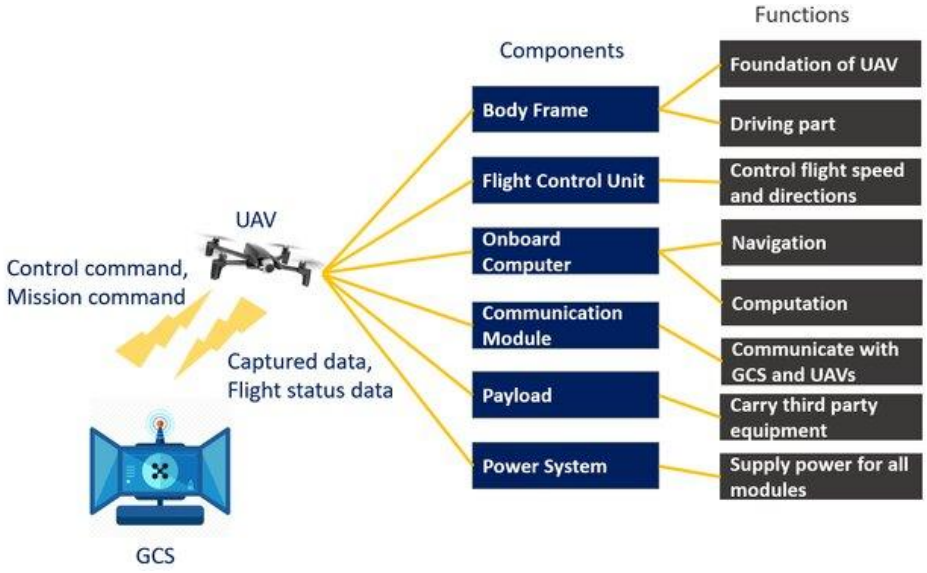


Figure 36 UAV Components and Functions.

4.2. Component Selection and Specification

4.2.1. Power Supply System and Motor Selection

The power supply system of the UAV is considered the most critical part due to its responsibility of supplying all of the other components. UAVs in general have many power supply systems including electrical, engines, or hybrid. Since the selected UAV is small in size, it's powered by a battery because it's cost-effective and provide appropriate flying duration. For small UAVs, lithium-ion (LiPo) batteries are recommended because of their low weight and comparatively high specific energy [52].

LiPo batteries are made up of cells, each with a nominal voltage of 3.7 volts. These cells are connected in series to form a battery and supply the required voltage to the components. The capacity, amount of current delivered per hour before being fully discharged, of a LiPo battery is measured in milliampere-hour (mAh). Increasing the capacity increases flight time and battery weight, thus it must be carefully calculated.

To choose the battery, the motor must be selected first, as voltage directly affects the motor's speed. Likewise, selecting the motor requires knowing the total weight of the vehicle, which is 3-5 kg, to ensure the motor can lift the UAV safely. A motor is characterized by four key properties: kV rating (motor's speed constant), thrust, efficiency, and current. A lower kV indicates higher torque and lower speed [53].

$$Thrust = total\ weight\ of\ drone\ \times acceleration\ due\ to\ gravity$$

$$5\ \times 9.81 = 49.05\ N \tag{1.1}$$

To ensure stability during take-off, the thrust to weight ration ought to be 2:1

$$49.05\ \times 2 = 98.1\ N$$

$$Thrust\ per\ motor = \frac{98.1}{4} = 24.525\ \cong 25N = 2.5\ Kg \tag{1.2}$$

Therefore, Sunnysky x2820 800 kV motor paired with 13×6.5 propellers and 60 Amp electronic speed controller (ESC), recommended from the motor's specs, were selected as it satisfies the 2.5 kg at 50% thrust. ESCs interpret signals to enable stable flight dynamics, manoeuvrability, and motor speed adjustment. The maximum current required for one motor is 46 Amp, meaning $46\ \times 4\ (no.of\ motors) = 184\ Amp$ is the draw current from the battery Figure 35.

Figure 37 SunnySky x2820 800kV and its compatible ESC.

From the motor's datasheet, a 4S which has 14.8 V is recommended. The required flight time can be given by:

$$Flight\ Time\ (\text{min}) = \frac{Battery\ Capacity\ (mAh) \times 60}{1000\ \times Total\ Current\ Draw\ (A)} \tag{1.3}$$

Assuming that the required time is 2 minutes:

$$Capacity = \frac{184\ \times 2\ \times 1000}{60} = 6133.333\ \cong 5200\ mAh \tag{1.4}$$

Since lightweight was prioritized; therefore, a lower capacity was selected. To ensure that the battery can handle the current C-rating, for a 5200mAh, 4S battery Figure 36, its c-rating is 50C.

$$Max\ Continous\ Cureent =\ Capacity\ (Ah) \times C - Rating$$

$$5.2\ \times 50 = 260\ A\ (\ sufficient) \tag{1.5}$$

Figure 38 4S 5200mAh LiPo Battery

Moreover, a power distribution board (PDB) Figure (37) is a vital component in UAV systems, responsible for safely and efficiently managing and distributing electrical power from the main battery to all other onboard components. Many PDBs also incorporate voltage regulators to protect sensitive electronics from voltage fluctuations caused by motor operation. Additionally, by reducing the amount of wiring required, it lowers the overall weight and improves the space efficiency of the UAV, enhancing flight endurance and performance and makes maintenance and troubleshooting easier [54].

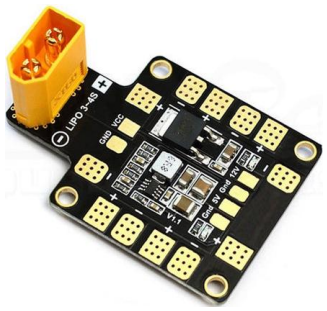


Figure 39 Power Distribution Board Module.

4.2.2. Processing Units and Communication Modules

Selecting suitable microcontrollers and communication modules is crucial to guarantee optimal performance and ensure task completion. Two controllers, STM32 microcontroller and Pixhawk. The STM32 family boasts high processing power and real-time capabilities, making it ideal for precision and rapid response. However, Pixhawk 2.4.8 (Figure 38) was chosen for its availability, strong support for autonomous flight, extensive community backing, and compatibility with ground control software such as QGroundControl and Mission Planner [55].

Figure 40 Pixhawk 2.4.8 Flight Controller

An essential facet of the system integration was setting up dependable remote communication. A Wi-Fi connection was set up between a Raspberry Pi and Pixhawk, facilitating rapid data transfer via MAVLink protocol. Functioning as a companion computer, the Raspberry Pi handled

supplementary tasks such as image recognition. This configuration enabled ground operators to wirelessly access mission data, issue commands, and monitor system health from a distance through ground control stations connected [56] ,Figure 39.

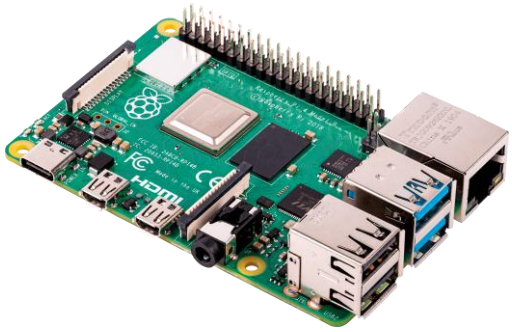


Figure 41 Raspberry Pi 4

4.2.3. On-Board Sensors and Actuators

Sensors like GPS and cameras are essential for improving a delivery UAV's functionality and independence. A GPS module allows the drone to follow predetermined routes and precisely arrive at delivery sites, allowing real-time positioning through data provided by the GPS. On the other hand, the camera allows precision landing, which enables the drone to identify targets or landing markers visually. This visual input guarantees safe delivery positioning and improves landing precision. Additionally, the Pixhawk counts a compass. It also employs an internal magnetometer that measures the Earth's magnetic field across three axes and gives crucial information for yaw orientation and navigation, Figure 40 .

Figure 42 Radiolink m10 GPS Module

Furthermore, at the core of the Pixhawk lies two inertial measurement units (IMU) [49]. Each IMU integrates a 3-axis accelerometer and a 3-axis gyroscope that capture linear acceleration and angular velocity in real-time, enabling exact approximation of the drone's position, orientation, and movement dynamics. Altitude estimation is handled by the onboard barometric pressure sensor that detects changes in atmospheric pressure and convert it to altitude readings. After precisely landing in the required position, the gripper mechanism releases the package at the intended target. Therefore, a servomotor is selected to release the package safely. When combined, these elements greatly enhance the drone's effectiveness, precision, and autonomy in actual delivery situations, Figure 41.

Figure 43 Servo Motor Employed

4.3 Circuit Design and Prototyping

The heart of the drone's electrical system is its LiPo battery, which connects to a PDB module that powers all the key components. To keep things balanced, both the battery and PDB are cleverly positioned at the center of the lower plate. From there, power flows to four ESCs, each linked to a brushless motor. These ESCs get their PWM signals from the Pixhawk 2.4.8 flight controller, which is mounted on vibration-dampening pads at the front of the upper plate to help reduce the impact of motor vibrations on the sensors. A PPM encoder gathers signals from the receiver, which is mounted on the lower plate, and sends these signals to the Pixhawk's RC input. The Raspberry Pi 4 is also placed on the lower plate and powered separately by its own power bank. It connects directly to the Pixhawk via USB for MAVLink communication and works with two cameras and

a Wi-Fi dongle for wireless transmission and processing. To allow for good signal reception, the GPS module, telemetry board, and one of the cameras are placed on the top plate, away from electrical noise. A servo-operated gripper is placed at the bottom of the drone to handle dropping medical supplies. The motors and ESCs are symmetrically positioned to offer balanced thrust and minimize yaw imbalance. The component mounting also allows for natural air flow to keep things cool. Zip ties and spiral wraps are employed to hold wiring in place.

Chapter (5) Control System

5.1. Role and Objectives of Control Systems in Drones

The control system of the drone is a fundamental part to guarantee stable and controllable flights. It works by processing live data collected from onboard components for the drone to reach a desired orientation, altitude, and speed and sustain a preferred flight state [57]. The processing and responding to data in a timely manner is vital for effective use of drones as inaccurate or delayed response can cause instability such as crashes or loss of control. Similarly, control system allows the drone to hover and perform coordinated yaw/pitch/roll movements and turns and execute a set flight path and react accurately to user or autonomous pilot decisions [58].

Drone systems have three main control goals during flight: stability, trajectory tracking and adaptability [59]. To maintain stable flights in the case of multirotor drones, the controller should continuously react to errors from the desired condition. This is best controlled by using Proportional-Integral- Derivative (PID) controllers due to their simplicity and capability of reducing the error between the desired position and the actual position very efficiently.

Avoiding obstacles and precise landing are two other important goals that are achieved by employing computer vision to be able to identify dynamic obstacles, replan paths on the fly, and guarantee a safe, precise landing. Also, the system needs to be smooth and predictable in order not to be aggressive in motion in urban/indoor scenarios. Such predictability has also benefited in terms of flight safety, energy conservation, and increasing assurance of autonomous missions.

5.2. Website Interface for Drone Target Location

The MediDrone Dispatch Center interface functions as the main supervisory interface for an autonomous drone delivery network which allows clients to request for drones and authorized drone dispatchers to control the medical emergency drones through web-based remote access. It

was designed to serve as a user-friendly frontend with DroneKit-Python backend technology to unify user command controls and MAVLink protocol-based low-level flight controller operations. The high-level system interface simplifies direct drone communication through its core design so dispatchers can operate the system through basic buttons and dynamic maps. Each dispatcher button, such as Connect Drone, Arm, or Takeoff, initiates backend functions that create official MAVLink commands, including MAV_CMD_COMPONENT_ARM_DISARM, MAV_CMD_NAV_TAKEOFF, and MAV_CMD_DO_SET_MODE, among others. The Pixhawk onboard the drone executes the received commands through serial or UDP connections to perform motor arming, altitude climb, and navigation operations. Flight Controls through the DroneKit vehicle API let dispatchers control primary flight operations including drone arming and disarming as well as take-off initiation and landing commands.

5.2.1 Client Page

The MediDrone Client Portal – Request a Drone Page functions as the primary user interface which permits patients or their caregivers to start medical drone dispatch either for emergencies or scheduled situations. The platform operates through the web to handle backend complexity while sending real-time actionable requests to the MediDrone dispatcher system. A selection at the top of the interface determines which mission type exists between Medical Emergency or Medical Delivery. The interface automatically modifies form elements according to user selection. The system requires emergency users to identify whether the situation involves themselves or someone else while selecting their emergency type from a dropdown menu to inform the dispatcher about both urgency level and nature of the request.

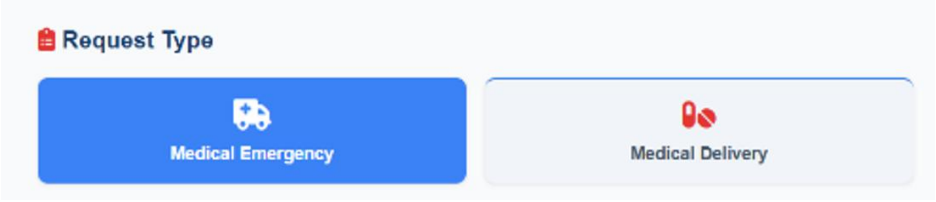


Figure 44 Request Types

The system requires users to provide an Emergency Location through manual input or by selecting the "Use My Location" button which retrieves their present GPS coordinates. The interactive

OpenStreetMap-based map presents the location to users while maintaining spatial precision which reduces critical miscommunication. The map feature enables dispatchers to see the delivery target area which allows them to assess flight routes and delivery capabilities. The "Send" button at the bottom functions as the submission control for requests. Pressing this button activates a backend workflow which combines all user inputs including emergency type details with medical records and location information to send them through a protected API connection to the dispatcher's panel.

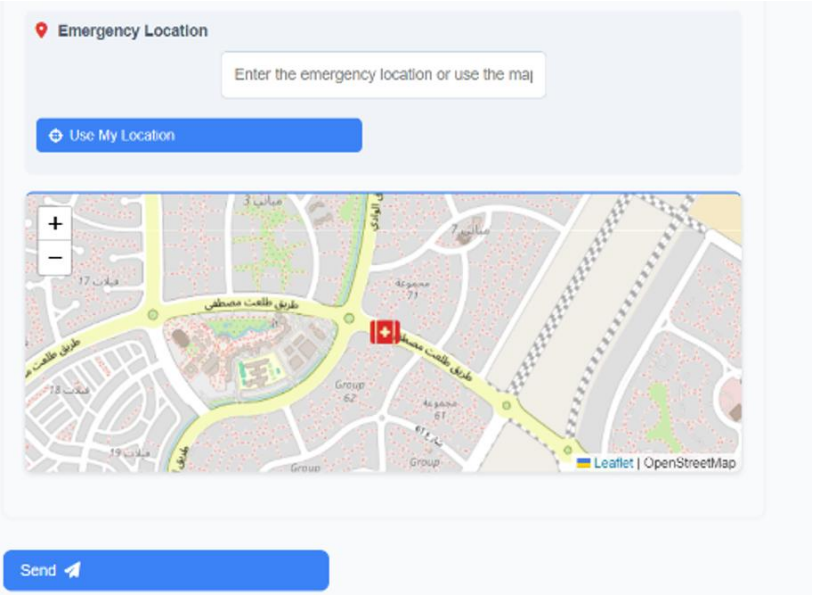


Figure 45 Patient Location Specification

The backend operates without DroneKit and MAVLink during this phase but these technologies activate once the dispatcher receives the request and selects a drone for assignment. The client-side interface serves as the main trigger for starting the command sequence that ultimately transmits MAVLink commands to the flight controller. The left sidebar enables users to switch between three main modules which include Client Information, Request History and Medical History for improving request personalization and traceability. The SOS Emergency button enables immediate priority routing in cases of life-threatening emergencies by skipping standard request validation for quick dispatch. This page delivers an easy-to-use yet powerful interface

which enables users to start urgent drone interventions while connecting human requirements with autonomous system abilities.

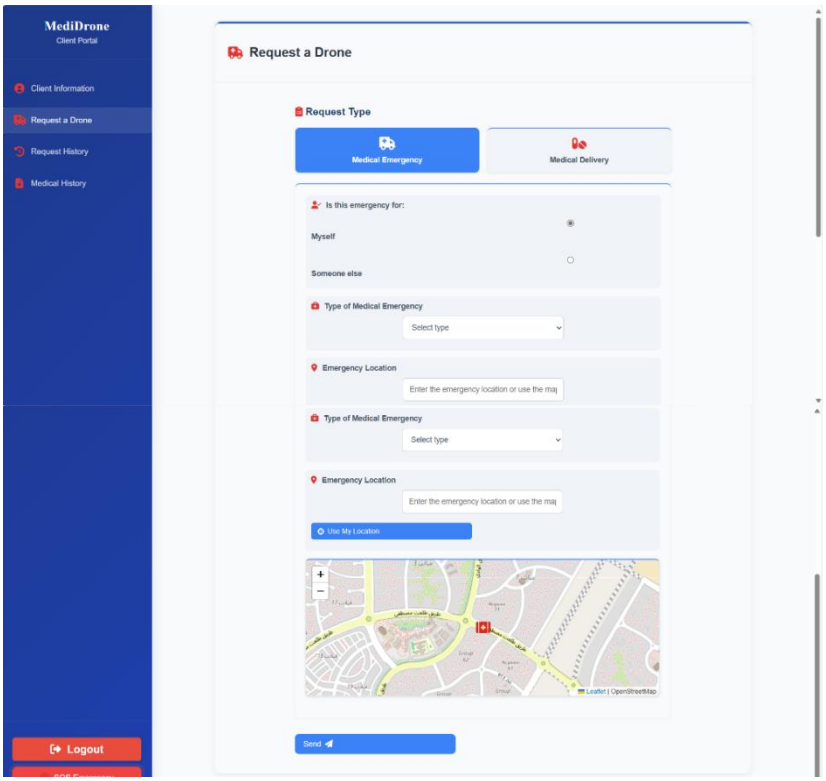


Figure 46 Complete Request a Drone Page

5.2.2 Dispatcher Page

Now that there is a request for a drone, an alert is sent to the dispatcher where they either accept or reject the request. A dispatcher then initiates take-off through the "Take-off" button which triggers “vehicle.simple_takeoff(target_altitude)” command to start the altitude ascent process and check MAVLink telemetry states for mode changes and state tracking. The system provides advanced automation while maintaining complete human control. The Emergency Actions panel provides dispatchers with override functions which activate during abnormal and mission-critical situations. The Emergency Land button activates instant descent commands that use MAV_CMD_NAV_LAND while the “Emergency Stop” activates either failsafe mode or throttle cut depending on the flight mode settings present on the controller. These functions are vital in

mission-critical environments where human decision-making must interrupt autonomous flight safely.

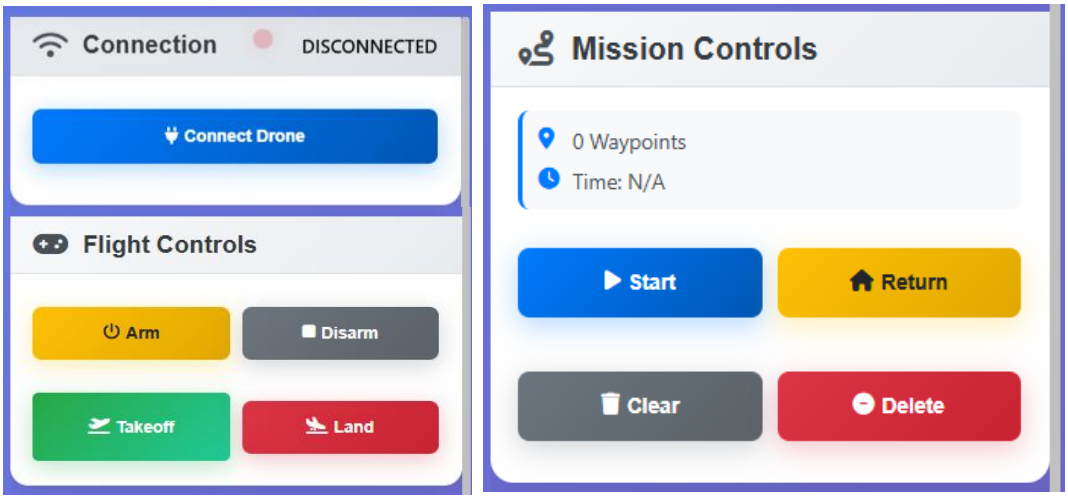


Figure 47 Drone Commands on Dispatcher Screen

The Mission Map shows real-time location data received through MAVLink GPS messages which improves situational awareness. The Drone Status panel together with the Mission Map enables the dispatcher to monitor battery status and velocity and fix status which together provide complete operational awareness of drone flight status and system condition and environmental situation. DroneKit functions as middleware between message parsing and telemetry subscriptions and event callbacks which operate in the background. The system receives updates about vehicle condition while running mission steps and confirming commands from the flight controller. This dispatcher

interface controls drone behavior which provides a comprehensive high-level control framework for executing missions making decisions and handling emergencies through a web-based interface

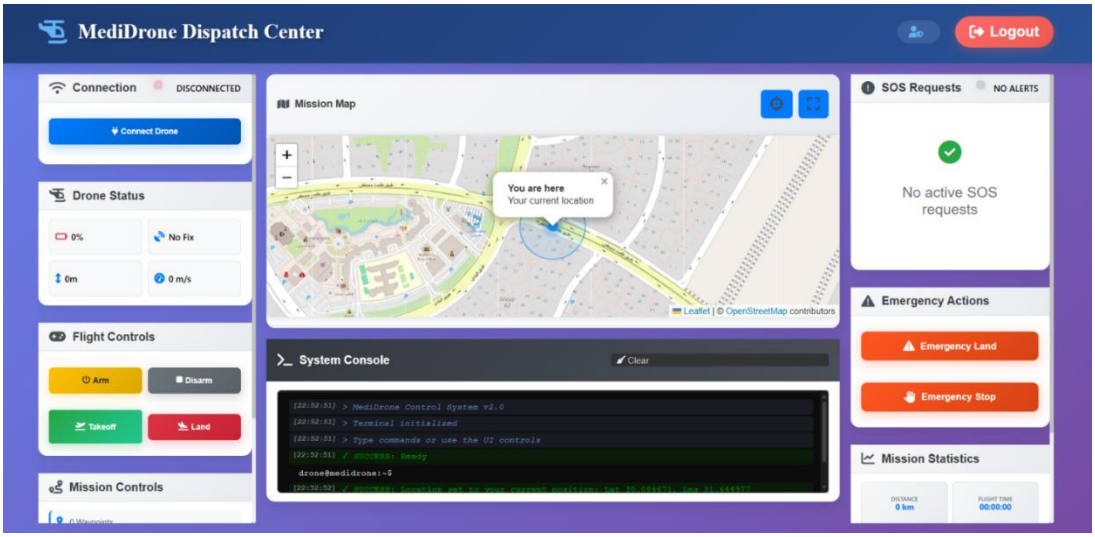


Figure 48 Complete Dispatcher Page

5.2. Mathematical Modelling of the Drone System

5.2.1. Euler Angles and Position Vectors

Euler angles are rotational angles used to describe the orientation of a rigid body in 3D space relative to a fixed coordinate system. The angles typically represent sequential rotations about specific axes (e.g. XYZ) and represent a sequence of three elemental rotations, Figure 42.

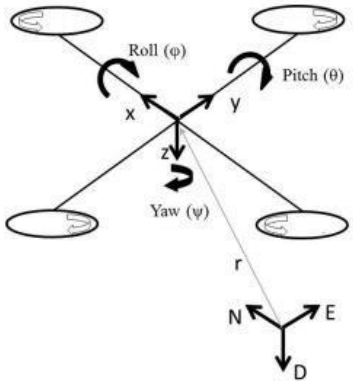


Figure 49 Visualization of Roll, Pitch, and Yaw.

$$R_x(\emptyset) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\emptyset) & -sin(\emptyset) \\ 0 & sin(\emptyset) & cos(\emptyset) \end{bmatrix}, R_y(\theta) = \begin{bmatrix} cos(\theta) & 0 & sin(\theta) \\ 0 & 1 & 0 \\ -sin(\theta) & 0 & cos(\theta) \end{bmatrix}, R_z = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (1.6)$$

By multiplying these matrices, a relation between the inertial and body frame, two reference frames used to describe the motion of a rigid body, can be obtained:

$$R_{zyx}(\emptyset, \theta, \psi) = R_z(\psi) \cdot R_y(\theta) \cdot R_x(\emptyset) \quad (1.7)$$

$$\begin{bmatrix} cos(\theta)cos(\psi) & sin(\emptyset)sin(\theta)cos(\psi) - cos(\emptyset)sin(\psi) & cos(\emptyset)sin(\theta)cos(\psi) + sin(\emptyset)sin(\psi) \\ cos(\theta)sin(\psi) & sin(\emptyset)sin(\theta)sin(\psi) + cos(\emptyset)cos(\psi) & cos(\emptyset)sin(\theta)sin(\psi) - sin(\emptyset)cos(\psi) \\ -sin(\theta) & sin(\emptyset)cos(\theta) & cos(\emptyset)cos(\theta) \end{bmatrix}$$

The body frame determines the local linear and angular velocities:

$$[x \quad y \quad z \quad \emptyset \quad \theta \quad \psi]^T \quad (1.8)$$

The inertial frame displays the global location and orientation:

$$[u \quad v \quad w \quad p \quad q \quad r]^T \quad (1.9)$$

The body frame velocities are converted into the inertial frame using a rotation matrix R in order to connect the velocities between these frames:

$$v = R\, v_B$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = R \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (1.10)$$

The inertial frame velocities are converted into the body frame using matrix T:

$$\omega = T\, \omega_B$$

$$\begin{bmatrix} \dot{\emptyset} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = T \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (1.11)$$

$$T = \begin{bmatrix} 1 & \sin(\emptyset)\tan(\theta) & \cos(\emptyset)\tan(\theta) \\ 0 & \cos(\emptyset) & -\sin(\emptyset) \\ 0 & \sin(\emptyset)/\cos(\theta) & \cos(\emptyset)/\cos(\theta) \end{bmatrix}$$

The kinematic equations can be found by solving these equations:

$$\begin{cases} \dot{x} = u[c(\theta)c(\psi)] - v[c(\emptyset)s(\psi) - s(\emptyset)s(\theta)c(\psi)] + \omega[s(\emptyset)s(\psi) + c(\emptyset)s(\theta)c(\psi)] \\ \dot{y} = u[c(\theta)s(\psi)] + v[c(\emptyset)c(\psi) + s(\emptyset)s(\theta)s(\psi)] - \omega[s(\emptyset)c(\psi) - c(\emptyset)s(\theta)s(\psi)] \\ \dot{z} = u[s(\theta)] + v[s(\emptyset)c(\theta)] + \omega[c(\emptyset)c(\theta)] \\ \dot{\emptyset} = p + q[s(\emptyset)t(\theta)] + r[c(\emptyset)t(\theta)] \\ \dot{\theta} = q[c(\emptyset)] - r[s(\emptyset)] \\ \dot{\psi} = q\left[{s(\emptyset)}/{c(\theta)}\right] + r\left[{c(\emptyset)}/{c(\theta)}\right] \end{cases} \quad (1.7)$$

5.2.2. Translational Motion of a Quadrotor

The translational dynamics of a quadcopter are described by Newton's second law when applied to its body frame:

$$m(\ (\omega_B \times v_B) + \dot{v_B}) = f_B \quad (1.13)$$

where m is quadrotor mass, ω_B is body frame angular velocity $[p \quad q \quad r]^T$, v_B is body frame linear acceleration $[u \quad v \quad w]^T$, $\dot{v_B}$ is linear acceleration $[\dot{u} \quad \dot{v} \quad \dot{w}]^T$, and f_B is sum of all forces acting on body frame $[f_x \quad f_y \quad f_z]^T$. All external forces' contributions are included, including both translational acceleration and rotation-induced Coriolis effects.

$$(\ (\omega_B \times v_B) + \dot{v_B}) = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} u \\ v \\ w \end{bmatrix} + \dot{v_B}$$

$$= \begin{bmatrix} qw - rv \\ ru - pw \\ pv - qu \end{bmatrix} + \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} \dot{u} + qw - rv \\ \dot{v} + ru - pw \\ \dot{w} + pv - qu \end{bmatrix}$$

$$\begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} = m \cdot \begin{bmatrix} \dot{u} + qw - rv \\ \dot{v} + ru - pw \\ \dot{w} + pv - qu \end{bmatrix} \quad (1.14)$$

5.2.3. Rotational Motion of a Quadrotor

The rotational dynamics are governed by Euler's equations:

$$(I \cdot \dot{\omega_b}) + \omega_b \times (I \cdot \omega_b) = m_B \tag{1.15}$$

Where ω_b is body frame angular velocity $[\dot{p} \quad \dot{q} \quad \dot{r}]^T$, m_B is total torque $[\tau_\emptyset \quad \tau_\theta \quad \tau_\psi]^T$, and I is moment of inertia matrix. The cross-product term accounts for the gyroscopic effects of the spinning body.

$$I \cdot \dot{\omega}_b = \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} \cdot \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} I_x \dot{p} \\ I_y \dot{q} \\ I_z \dot{r} \end{bmatrix}$$

$$I \cdot \omega_b = \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} \cdot \begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} I_x p \\ I_y q \\ I_z r \end{bmatrix}$$

$$\omega_b \times (I \cdot \omega_b) = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} I_x p \\ I_y q \\ I_z r \end{bmatrix} = \begin{bmatrix} (I_z - I_y) qr \\ (I_x - I_z) pr \\ (I_y - I_x) pq \end{bmatrix}$$

$$(I \cdot \dot{\omega_b}) + \omega_b \times (I \cdot \dot{\omega}_b) = \begin{bmatrix} I_x \dot{p} \\ I_y \dot{q} \\ I_z \dot{r} \end{bmatrix} + \begin{bmatrix} (I_z - I_y) qr \\ (I_x - I_z) pr \\ (I_y - I_x) pq \end{bmatrix} = \begin{bmatrix} I_x \dot{p} - I_y qr + I_z qr \\ I_x pr + I_y \dot{q} - I_z pr \\ -I_x pq + I_y pq + I_z \dot{r} \end{bmatrix}$$

$$\begin{bmatrix} \tau_\emptyset \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} I_x \dot{p} - I_y qr + I_z qr \\ I_x pr + I_y \dot{q} - I_z pr \\ -I_x pq + I_y pq + I_z \dot{r} \end{bmatrix} \tag{1.16}$$

Thus, the dynamic model of quadrotor in the body frame is obtained as:

$$\begin{cases} f_x = m(\dot{u} + qw - rv) \\ f_y = m(\dot{v} + ru - pw) \\ f_z = m(\dot{w} + pv - qu) \\ \tau_\emptyset = I_x \dot{p} - I_y qr + I_z qr \\ \tau_\theta = I_x pr + I_y \dot{q} - I_z pr \\ \tau_\psi = -I_x pq + I_y pq + I_z \dot{r} \end{cases} \tag{1.17}$$

5.2.4. External Forces in the Body Frame

The external forces in the body frame can computed by the following equation with variables representing mass, gravity, rotation matrix, unit vector $\widehat{e_{Iz}}$ inertial frame, total thrust force U, unit vector $\widehat{e_{Bz}}$ body frame, and f_w is external disturbances for instance wind $[f_{wx} \quad f_{wy} \quad f_{wz}]^T$:

$$f_b = mgR^T \cdot \widehat{e_{Iz}} - U\widehat{e_{Bz}} + f_w \tag{1.18}$$

$mgR^T \cdot \widehat{e_{Iz}}$ refers to inertial frame gravitational force. Using the rotation matrix's transpose, the gravitational force vector is transformed from the inertial to the body frame. Similarly, $-U\widehat{e_{Bz}}$ describes the thrust force generated by the rotors down the body frame's z-axis (downward).

$$R^T = \begin{bmatrix} c(\theta)c(\psi) & c(\theta)s(\psi) & -s(\theta) \\ s(\theta)s(\emptyset)c(\psi) - c(\emptyset)s(\psi) & s(\emptyset)s(\theta)s(\psi) + c(\emptyset)c(\psi) & s(\emptyset)c(\theta) \\ c(\emptyset)s(\theta)c(\psi) + s(\emptyset)s(\psi) & c(\emptyset)s(\theta)s(\psi) - s(\emptyset)c(\psi) & c(\emptyset)c(\theta) \end{bmatrix}$$

$$R^T \cdot \widehat{e_{Iz}} = R^T \cdot \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} -s(\theta) \\ s(\emptyset)c(\theta) \\ c(\emptyset)c(\theta) \end{bmatrix}$$

$$U_1\widehat{e_{Bz}} = U_1 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ U_1 \end{bmatrix}$$

$$f_b = mg \begin{bmatrix} -s(\theta) \\ s(\emptyset)c(\theta) \\ c(\emptyset)c(\theta) \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ U_1 \end{bmatrix} + \begin{bmatrix} f_{wx} \\ f_{wy} \\ f_{wz} \end{bmatrix} = \begin{bmatrix} -mgs(\theta) + f_{wx} \\ mgs(\emptyset)c(\theta) + f_{wy} \\ mgc(\emptyset)c(\theta) - U_1 + f_{wz} \end{bmatrix}$$

$$f_b = \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} = m \cdot \begin{bmatrix} \dot{u} + qw - rv \\ \dot{v} + ru - pw \\ \dot{w} + pv - qu \end{bmatrix}$$

$$f_b = m \cdot \begin{bmatrix} \dot{u} + qw - rv \\ \dot{v} + ru - pw \\ \dot{w} + pv - qu \end{bmatrix} = \begin{bmatrix} -mgs(\theta) + f_{wx} \\ mgs(\emptyset)c(\theta) + f_{wy} \\ mgc(\emptyset)c(\theta) - U_1 + f_{wz} \end{bmatrix}$$

$$\begin{bmatrix} m(\dot{u} + qw - rv) = -mgs(\theta) + f_{wx} \\ m(\dot{v} + ru - pw) = mgs(\emptyset)c(\theta) + f_{wy} \\ m(\dot{w} + pv - qu) = mgc(\emptyset)c(\theta) - U_1 + f_{wz} \end{bmatrix} \tag{1.19}$$

5.2.5. External Moments in the Body Frame

The external moments in the body frame are given by:

$$m_B = \tau_b - g_a + \tau_\omega \tag{1.20}$$

Where τ_b is the control torques generated by the rotors, g_a gyroscopic moments cause by the rotation of the rotors, τ_ω aerodynamic torques are caused by disturbances such wind $\tau_w = [\tau_{wx} \quad \tau_{wy} \quad \tau_{wz}]^T$

$$m_B = (I \cdot \dot{\omega_b}) + \omega_b \times (I \cdot \omega_b) = \begin{bmatrix} I_x\dot{p} - I_yqr + I_zqr \\ I_xpr + I_y\dot{q} - I_zpr \\ -I_xpq + I_ypq + I_z\dot{r} \end{bmatrix} = \tau_b - g_a + \tau_\omega$$

$$m_B = \begin{bmatrix} I_x\dot{p} - I_yqr + I_zqr = \tau_x + \tau_{wx} \\ I_xpr + I_y\dot{q} - I_zpr = \tau_y + \tau_{wy} \\ -I_xpq + I_ypq + I_z\dot{r} = \tau_z + \tau_{wz} \end{bmatrix} \tag{1.21}$$

Complete dynamic model in the body frame:

$$\begin{cases} m(\dot{u} + qw - rv) = -mgs(\theta) + f_{wx} \\ m(\dot{v} + ru - pw) = mgs(\emptyset)c(\theta) + f_{wy} \\ m(\dot{w} + pv - qu) = mgc(\emptyset)c(\theta) - U_1 + f_{wz} \\ I_x\dot{p} - I_yqr + I_zqr = \tau_x + \tau_{wx} \\ I_xpr + I_y\dot{q} - I_zpr = \tau_y + \tau_{wy} \\ -I_xpq + I_ypq + I_z\dot{r} = \tau_z + \tau_{wz} \end{cases} \tag{1.22}$$

5.2.6. Actuator Dynamics

The forces and torques produced by a quadrotor's four propellers are expressed in terms of rotational speeds by the following equations: $\Omega_1, \Omega_2, \Omega_3, \Omega_4$. The overall thrust is calculated by adding the thrust values of all propellers. The arm length b and drag coefficient d scale the thrust differences produced by particular propeller pairs, which determine the τ_x, τ_y, and τ_z, which come from the mechanics of the quadrotor, which specify the relationship between propeller speed and the forces and torques generated:

$$\begin{cases} U = b\left({\Omega_1}^2 + {\Omega_2}^2 + {\Omega_3}^2 + {\Omega_4}^2\right) \\ \tau_x = bl\left({\Omega_3}^2 - {\Omega_1}^2\right) \\ \tau_y = bl\left({\Omega_4}^2 - {\Omega_2}^2\right) \\ \tau_z = d\left(-{\Omega_1}^2 + {\Omega_2}^2 - {\Omega_3}^2 + {\Omega_4}^2\right) \end{cases} \tag{1.23}$$

Thus, the dynamic model of body frame is obtained:

$$\begin{cases} m(\dot{u} + qw - rv) = -mgs(\theta) + f_{wx} \\ m(\dot{v} + ru - pw) = mgs(\emptyset)c(\theta) + f_{wy} \\ m(\dot{w} + pv - qu) = mgc(\emptyset)c(\theta) - b\left({\Omega_1}^2 + {\Omega_2}^2 + {\Omega_3}^2 + {\Omega_4}^2\right) + f_{wz} \\ I_x\dot{p} - I_yqr + I_zqr = bl\left({\Omega_3}^2 - {\Omega_1}^2\right) + \tau_{wx} \\ I_xpr + I_y\dot{q} - I_zpr = bl\left({\Omega_4}^2 - {\Omega_2}^2\right) + \tau_{wy} \\ -I_xpq + I_ypq + I_z\dot{r} = d\left(-{\Omega_1}^2 + {\Omega_2}^2 - {\Omega_3}^2 + {\Omega_4}^2\right) + \tau_{wz} \end{cases} \tag{1.24}$$

5.3. Controller Design and Implementation

The control architecture of the quadrotor is hierarchical and cascaded loop that is inspired and based on ArduCopter principles. The framework refines high level mission goals to low level motor outputs sequentially through a set of coupled control steps: position, velocity, attitude, rate control, and motor mixing. At the top level, the position controller calculates the required velocity command to reduce the position error between the desired and current coordinates of the quadrotor. The proportional control law is:

$$v_{des} = P_{pos}(p_{des} - p_{act}) \tag{1.25}$$

Where P_{pos} is the proportional gain, v_{des} is the desired velocity, and p_{des} is the desired position. In order to prevent abrupt or unexpected accelerations, the system uses a Square Root Controller, in which the proportional gain is scheduled as a function of the position error magnitude:

$$P_{pos} = K_{pos}\sqrt{|p_{err}|} \tag{1.26}$$

Moreover, Leash Length mechanism is also implemented as it limits the maximum allowable position error, thus stabilizing the quadrotor during immense deviations:

$$p_{err,max} = \frac{a_{act}}{2P_{pos}^2} + \frac{v_{act}^2}{2a_{act}} \tag{1.27}$$

This maintains the position controller predictable and smooth, specifically during high-speed manoeuvres, Figure 43.

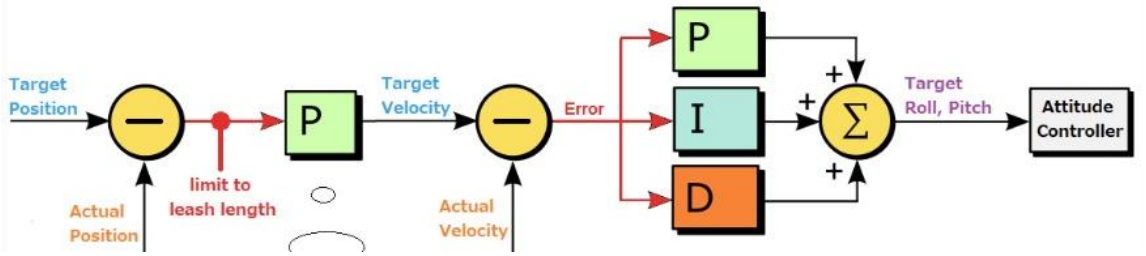


Figure 50 PID Position Control.

After the position control, the velocity controller maps the desirable velocity vector to suitable tilt orders, target pitch (θ) and roll ($\emptyset$) angles. PID controller function on the velocity error:

$$\{\theta, \emptyset\} = PID(v_{des} - v_{act}) \tag{1.28}$$

By controlling the tilt of the body, the system would produce the required accelerations to steer towards the target velocity.

The attitude controller acts as a mediator, which makes the orientation of the quadrotor consistent with the generated desired angles by the velocity controller. It then determines the desired angular rates($\dot{\theta_{des}}, \dot{\phi_{des}}$) proportional to the attitude error

$$\dot{\theta_{des}} = P_{att}(\theta_{des} - \theta) \quad , \quad \dot{\phi_{des}} = P_{att}(\phi_{des} - \phi) \tag{1.29}$$

This layer delivers fast convergence of the rotational dynamics of the drone to the desired orientation in Figure 44.

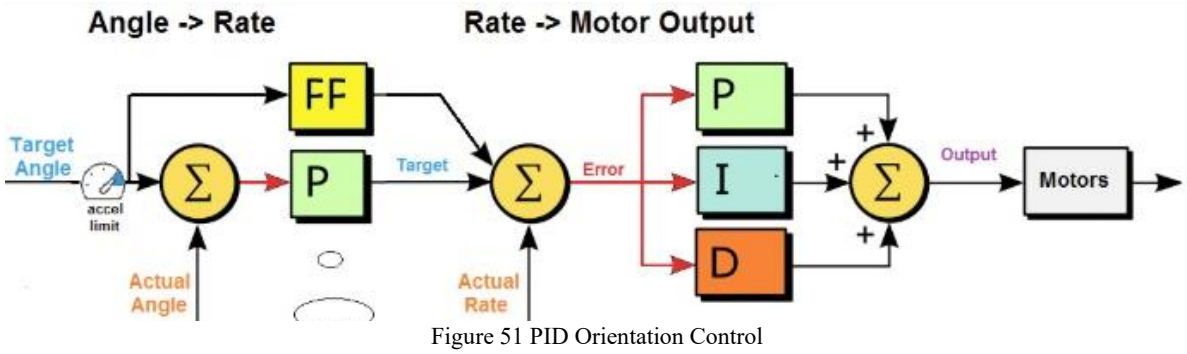


Figure 51 PID Orientation Control

The most inner loop is the rate controller, which directly controls the angular velocities. It utilizes a full PID controller to minimize the error between the desired and actual rates, thus producing commands for the torque:

$$\tau = \mathrm{PID}\left(\dot{\theta}_{\mathrm{des}} - \dot{\theta}\right) \tag{1.30}$$

These torques are then transformed to thrust value for each motor individually by the motor mixer, which is specific to the quadcopter frame geometry.

Trustworthy state feedback to every control loop is provided by using an Extended Kalman Filter (EKF), which fuses measurements from the IMU, GPS, and barometer, thus filtering out noise and precisely estimating position, velocity, and orientation. Supporting the flight controller, a Raspberry Pi 4B that acts as a companion computer runs high-level operations such as real-time obstacle avoidance and Aruco marker-based precision landing, providing dynamic navigation updates during flight. This organized, multi-loop control methodology, with gain scheduling, leash constraints, and sensor fusion, ensures stability, accuracy, and robustness under real operating stresses required for emergency medical deployment in Figure 45.

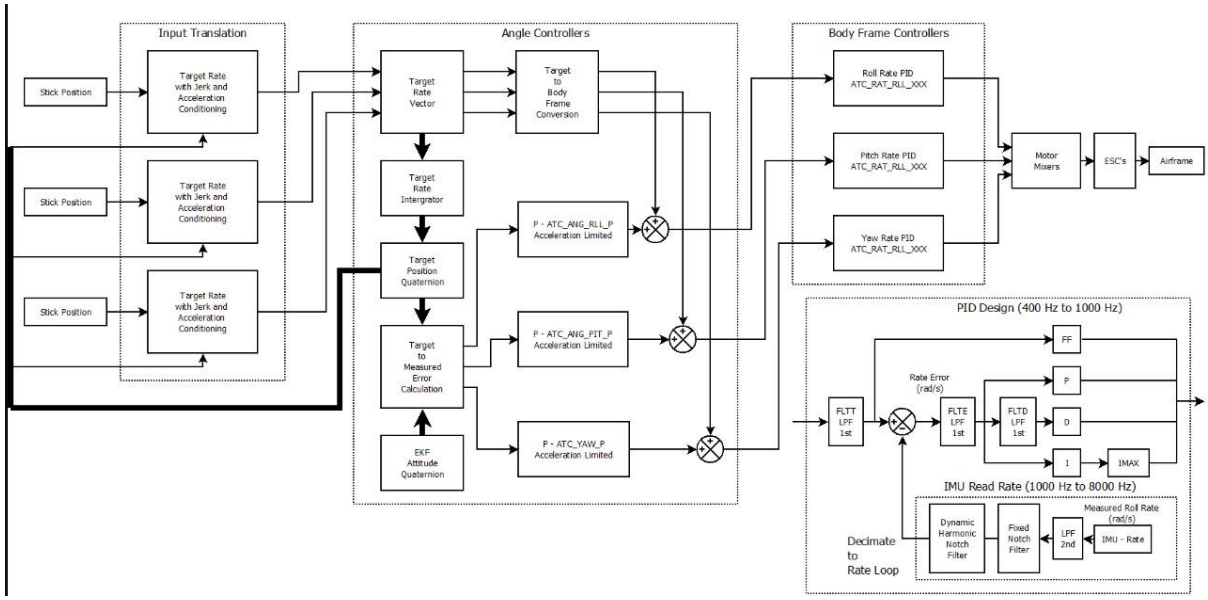


Figure 52 Multi-Loop Control.

Chapter (6) Visualization

6.1. Visualization

Since the drone will be flying autonomously from the take of station to the determinate location by the drone dispatcher, its essential to integrate a visualization system so it will help the drone in identifying objects and recognition, as it serves a critical rule in taking decisions on how to avoid obstacles in dynamic environments, improving safety and operational effectiveness in complex missions. Even more, another task that the drone will have to induce is the landing step where it will be using the Aruco marker Feature to land where the visitation system is essential. Those concepts will be further explained in the following sections.

6.2. Camera Calibration

For the drone to be able to land accurately using the Arcuo marker, it is fundamental to precisely derive three-dimensional (3D) spatial information from two-dimensional (2D) images. As the cameras serve as the eyes of the robots, real-world coordinates for tasks such as navigation, object manipulation, and scene understanding are translated based on visual inputs. However, in computer vision geometric distortions are found and do not necessarily provide metric information.

To narrow this gap, both intrinsic parameters like focal lengths, principal point offsets, and lens distortion coefficients and extrinsic parameters like rotation and translation relative to a world coordinate frame are estimated using camera calibration. Without detailed calibration, regular errors can compound into major deviations in real-world applications, and they will affect pose-estimation and 3D reconstruction [60].

Zhang's camera calibration algorithm presents a fascinating balance between practicality and accuracy. Zhang's method utilizes multiple images from different viewpoints of a simple 2D pattern (commonly a checkerboard) which saves the use of specialized three-dimensional

calibration rigs. The algorithm detects similarities between known pattern coordinates and their pixel projections; through a combination of homographs estimation and nonlinear optimization the algorithm solves for the camera matrix and distortion parameters. Although, real-world challenges such as lens irregularities, lighting nonuniformities, and partial obstruction of calibration patterns can reduce the accuracy. This project Used Zhang's algorithm into an Aruco marker-based pose-estimation pipeline on a single, low-cost camera.

The first building block of Zhang's approach lies in Finding the relationship between a 3D planar pattern and its 2D image via a homograph matrix. From each captured view, you can generate a homography, and from that homography derive linear equations constraining the camera's intrinsic matrix.

After the first linear solution, extrinsic rotations, and translations alongside distortion coefficients modelling both radial bending of light rays (barrel or pincushion effects) and tangential warping (caused by lens decentring) are estimated. At last, all parameters are refined by nonlinear least-squares optimization jointly through minimizing the total reprojection error across all pattern corners.

Three radial coefficients (K1, K2, K3) and two tangential coefficients (P1, P2) are used by Lens distortion models. Straight lines appear in the image curved due to Radial distortion, the margins of the image it the major affected area by this distortion. Tangential distortion introduces asymmetric warping when the lens does not rest perfectly parallel to the imaging sensor. Correcting these distortions is crucial for accurate corner detection, as well as accurate 3D pose estimation.

ArUco markers are small square labels with black-and-white patterns that encode a unique ID using Hamming codes, it is like a QR Code. They can still be spotted if they are tilted or partly hidden due to their built-in error checking and high contrast. And you can calculate its precise

rotation and position relative to a calibrated camera the moment you find the four corner points in the image and know the marker's real-world size [61]. Considering the previous stated the code will be uploaded into the Raspberry Pi OS using Python.

6.3.1. Calibration Methodology

1. Image Acquisition

A Set of images of the checkerboard were captured from a different distances and angles within approximately 0.3 m to 1.2 m of the camera. All internal corners remain visible within the field of view because the pattern was oriented multiple times. Good lightening panels eliminated harsh shadows and glare, making sure that there is consistent corner detection across images as shown in Figure (39).

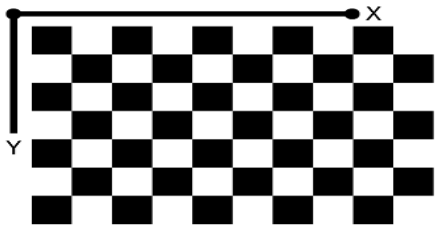


Figure 53 Checkerboard pattern

2. Corner Detection and Sub-Pixel Refinement

This function was "cv2.findChessboardCorners()" used in the code to obtain Initial corner estimates. To improve measurement precision, these estimates were modified to be more refined to sub-pixel accuracy by using "cv2.cornerSubPix()" function, this step is important for minimizing the bias introduced by coarse corner localization and this function terminates after a maximum of 100 iterations or an ε threshold of 0.001 pixels in Figure 40.

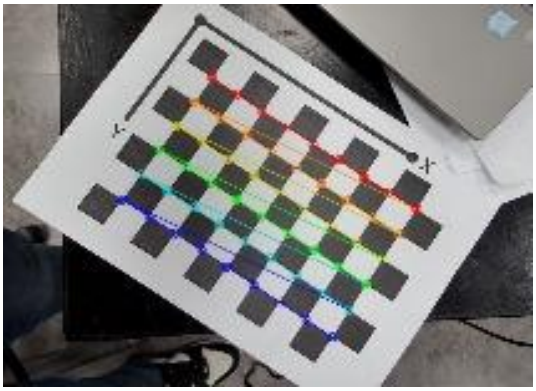


Figure 54 Corner Detection

3. Intrinsic and Extrinsic Parameter Estimation

To compute a homography matrix HH each view's refined corner set and known planar coordinates were used. From multiple homographies, A linear system provided a first estimate of the intrinsic matrix.

$$Camera\ Matrix = \begin{bmatrix} F_x & S & C_x \\ 0 & F_y & C_y \\ 0 & 0 & 1 \end{bmatrix}$$

$$Camera\ Matrix = \begin{bmatrix} 488.839935 & 0.381558597 & 323.456116 \\ 0 & 489.566559 & 239.481735 \\ 0 & 0 & 1 \end{bmatrix} \quad (1.30)$$

At the same time, the five distortion coefficients were obtained by initial guesses.

$$Distortion\ parameters = [K_1 \quad K_2 \quad P_1 \quad P_2 \quad K_3]$$

$$Distortion\ parameters = [0.149996236 \quad -0.5 \quad 0.000571434211 \quad -0.00149008352 \quad -0.0265201256] \quad (1.31)$$

Extrinsic parameters were also derived from homography factors.

4. Nonlinear Optimization

via reducing the sum of squared reprojection errors we could improve all parameters that were computed. OpenCV's Levenberg–Marquardt solver continuously updated all variables until convergence, resulting in a mean reprojection error of 0.018 pixels, which indicate that high-accuracy results were calibrated.

5. Accuracy Metric in Data Processing

A technique is used in computer vision called "Pose estimation" to determine the 3D spatial coordinates of a point, including coordinates on the x, y, and z axes.

$$Accuracy\ pose\ estimation = 100\% - Error\ Relative \tag{1.32}$$

6.3. Aruco Marker

Aruco markers are used for UAV landing because they offer a reliable, low-cost method for precise localization using only a camera, making them ideal in environments where GPS is unreliable or unavailable. Their strong visual structure and resistance to detection errors make them particularly well-suited for critical operations like autonomous descent and touchdown, figure 46. An Aruco marker is a square visual tag composed of a thick black border surrounding a grid of black and white cells. Each marker has a unique binary pattern that encodes an ID and orientation. The bold outer border helps the system quickly locate potential markers in an image, while the structured internal grid ensures that each marker can be uniquely identified, even under challenging visual conditions such as motion blur or partial occlusion.

The detection process begins with the UAV's onboard camera capturing a video frame as it descends toward the landing area. This frame is converted to a grayscale image to simplify processing and focus purely on brightness contrasts, which are critical for detecting the marker's black border. The system then scans the image for contours—specifically looking for closed, four-sided shapes that resemble squares. Each contour is checked to see if it meets geometric criteria, such as having four corners, approximately equal side lengths, and internal angles close to 90 degrees, Figure (41).

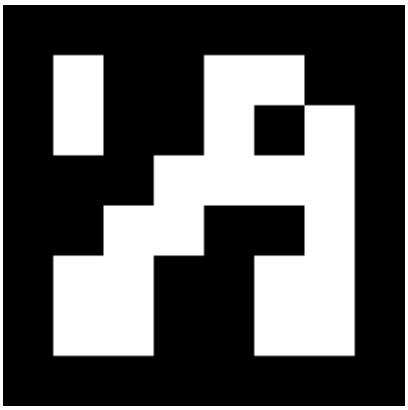


Figure 55 Arcuo Marker.

Once a potential square is found, it is isolated and geometrically corrected using a perspective transformation. This flattens the view of the square, as if looking at it straight from above, regardless of the UAV's angle. Within the square, the system samples the grid of black and white cells to extract the marker's internal binary pattern. Each cell is read based on brightness values, and the resulting pattern is compared to a library of known markers in Figure 42.

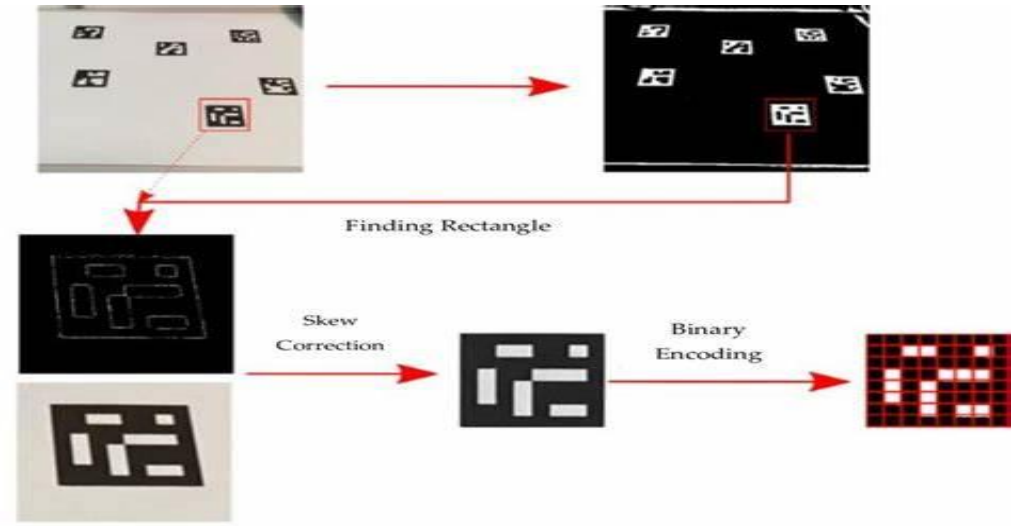


Figure 56 Aruco Marker Working Principle

If a match is found, the system identifies the marker's unique ID and determines its orientation by analyzing the position of key reference cells in the pattern. At this stage, the exact positions of the four marker corners in the image are recorded. These corner points become the basis for understanding how the UAV is positioned and aligned relative to the marker. The system repeats

this process continuously, allowing the drone to track its position in real time as it approaches the landing zone.

After successful identification, the system examines the position and shape of the marker in the image to infer the UAV's relative position. If the marker appears centered and symmetrical, the UAV is properly aligned; if it appears distorted, rotated, or off-center, this indicates that the drone needs to correct its position or heading. The perceived size of the marker in the frame also gives a strong cue about altitude—the closer the drone, the larger the marker appears.

By continuously analyzing the marker's appearance during descent, the UAV receives constant visual feedback that enables it to adjust its trajectory and maintain alignment, Figure 48. This visual guidance system allows for highly accurate, smooth landings, reducing the risk of hard impact or drift. The robust design and detection method of ArUco markers make them an effective and practical solution for vision-based autonomous UAV landings in Figure (43).

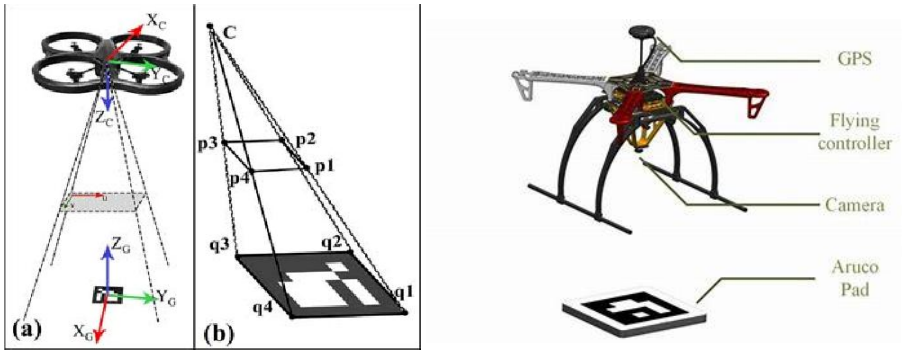


Figure 57 Landing Using Arcuo Marker

6.4. Obstacle Avoidance

Recent studies highlight the advancements in sensor-based and vision-based methods, real-time decision-making, and coordination between 2D and 3D environments [62]. However, it has been proven that utilizing many sensors all around the drone requires more current withdrawal from the

battery which reduced the flight time. Complementing this, lightweight object detection network like YOLO have been developed to improve detection accuracy while minimizing computational demands, making it suitable for real-time UAV applications [63].

This section emphasis how vision-based systems are essential for drones to effectively detect and avoid obstacles during flight. While many researchers rely on distance-measuring sensors due to their simplicity and effectiveness, covering the forward-faced view to provide clear view in the drone's direction of movement. This camera is integrated with an AI-based object detection system using YOLOv5, which runs on a Raspberry Pi to identify obstacles and send avoidance commands to the motors. The model is trained to recognize common aerial obstacles such as trees, buildings, and birds that are key threats during medical delivery missions.

6.5. Conclusion

In short, the addition of a visualization system is essential in allowing the drone to function autonomously and securely in challenging, real-world environments. The technology is required for accurate landing and obstacle avoidance since it senses objects and provides real-time visual feedback. The drone can navigate challenging terrain while having exact localisation based on features like Aruco marker tracking and camera-based obstacle detection. In addition to making the drone more operable, these functionalities are also vital to the success of the mission, especially in emergency medical delivery. Further elaboration on the application of Aruco vision-based obstacle avoidance and marker-based landing will be discussed in subsequent sections.

Chapter (7) Experimental and Simulation

7.1. Introduction

The following chapter will be discussing the results of the prototype and the experimental validation of a medical quadrotor drone designed for emergency situations is presented in this chapter. Furthermore, the drone's performance was evaluated in both simulated and real-world settings, with an emphasis on autonomous capabilities, cargo handling, control responsiveness, and stability. A fully assembled drone with a Pixhawk 2.4.8 flying controller, gripper mechanism, and specially made medical payload box was used for field testing. Flight duration, hover stability, payload delivery accuracy, and obstacle avoidance effectiveness were all evaluated in physical trials. To simulate the drone's dynamics and control system behavior, validate theoretical models, test control algorithms, and compare expected performance with actual results, a comprehensive simulation was constructed in MATLAB/Simulink.

7.2. Experimental Setup

In order to validate the overall performance of the drone, a structured experimental setup was chosen to evaluate the hardware integration with the software interface and the sensor calibration, therefore in the following table 1 quick review on the physical component utilized with its desired layout will be shown in addition to, the layout of the flight-testing environment as well as the sensor calibration procedures, specially the IMU and GPS modules.

Table 1: Main Components.

Subsystem	Component	Description
Flight Controller	Pixhawk 2.4.8	Serves as the primary autopilot system, responsible for attitude and position stabilization.
Companion Computer	Raspberry Pi 4	Executes high-level processing tasks including vision-based navigation, ARuco marker detection, and obstacle avoidance algorithms.
Motors and Propulsion	SunnySky X2820 800KV Motors with 13" Propellers + 60A ESCs	Four brushless DC motors provide lift and maneuverability, powered through electronic speed controllers.
Power Supply	4S 5200mAh LiPo Battery	Supplies electrical power to all onboard components, supporting stable flight with a 1 kg payload.
Frame Structure	Custom X-configuration Carbon Fiber Frame	Lightweight frame design ensures optimal thrust distribution and structural integrity, integrated with landing gear and gripper.
Payload System	Medical Transport Box + Arduino Nano	Enclosed payload box equipped with temperature and humidity sensors, digital lock, and a load monitoring system.

7.2.1. Mission Planner

For the drone to be controlled through ground control operators Mission Planner was utilized for its compatibility with the PixHawk through the telemetry providing reliable real time flight information and firmware tuning. While a separated Wi-Fi communication protocol was induced by the Raspberry Pi to act as remote console for running Python scripts, and activating image-based landing logic using Aruco markers.

7.2.2. IMU Calibration

Another factor that must be considered within the setup of the drone is the IMU calibration and integration with the Pixhawk due to it sever role in identifying the drone's orientation and angular velocity as well as linear acceleration, that is why the calibration is required to eliminate any sensor bias or drifts. To archive this the Mission planner was a major element in the process [64].

The following steps were performed using Mission Planner:

1. **Level Horizon Setup**

   In order to determine a neutral reference orientation, the drone was positioned on a level, sturdy surface.

2. **Six-Point Orientation Calibration**

   For instance, for the flight controller to capture sensor data across several axes, the drone turned through all six faces (top, bottom, left, right, front, and rear) as instructed.

3. **Vibration Compensation**

   The IMU was mounted using vibration-dampening foam pads. Real-time vibration metrics were monitored during motor idle and flight using the onboard log viewer to ensure they remained within acceptable thresholds.

4. **Compass/Magnetometer Alignment**

   As IMU data was synchronized with the internal magnetometer to ensure heading stability.

7.2.3. GPS Fix Validation and Compass Calibration

Additionally, to make sure that the GPS module will not be affected with the electromagnetic interference that may be induced by the high-power components, it was mounted. As well as Poor satellite lock, excessive Horizontal Dilution of Precision (HDOP) can cause high rate of inaccuracy, that is why it’s essential to make sure that before arming the drone, a 3D GPS fix with 8 or more satellites and an HDOP of less than 1.5 to guarantee dependability. Furthermore, the Pixhawk system's compass (magnetometer) and GPS are closely interwoven, necessitating magnetic declination alignment and compass calibration for accurate heading calculation [65, 66].

The following step will explain the steps:

1. **Position Fix Check**

   Initiating autonomous missions requires a minimum of eight satellites. To guarantee positional accuracy (<1.5), HDOP (Horizontal Dilution of Precision) measurements were tracked.

2. **Compass Calibration**

   Mission Planner was used to progressively rotate the drone around all three axes. In this way, local magnetic distortions were somewhat compensated for.

3. **Magnetic Declination Update**

   Aligns magnetic and true north values automatically using the most recent GPS coordinates.

4. **Failsafe Configuration**

   Before the drone could be armed, a GPS lock was necessary. RTL was set up to kick in in the event that the GPS signal was lost while the aircraft was flying on its own.

To guarantee reliable and consistent navigation, these calibration steps were carried out once more before to every testing session. The drone's smooth flight, safe delivery, and dependable autonomous operation were all made possible by the IMU, and GPS being successfully calibrated. Furthermore, for more easier user interface the website can be utilized to control the whole mission.

7.3. Performance metrics

Performance metrics is crucial to know the capabilities of any drone and how efficient will it perform its objective. Performance metrics enable global cooperation and scientific advancement in aeronautics engineering by providing a consistent system to assess how well a drone satisfies mission-specific criteria, that vary from cargo to endurance to maximum altitude limitations. Performance indicators give the research community and engineers tools to downscale design, comply with regulatory constraints, and maximize reliability in a variety of environments based

on measured data analysis by using system models within theoretical boundaries. The quadcopter examined weighs 3. 5 kg and has been outfitted with 60A BLHeli ESCs, 1365 propellers, 5200 mAh 6S batteries, and four SunnySky 2820 800kV motors, each has the capacity for 2.79 kg of thrust (total thrush = 11.16 kg). The performance indicators presented below have been derived using these specifications:

Thrust-to-Weight Ratio (TWR)

This crucial measure of lift and manoeuvrability is calculated as

$$\text{TWR}=\frac{total\ thrust}{drone\ weight}=\frac{11.16\ kg}{3.5\ kg}\approx 3.19 \tag{1.33}$$

demonstrating excellent efficiency appropriate for a range of operating scenarios as min TWR should be 2:1.

Flight Time

This measure, which indicates the overall operating time, is obtained from the energy capacity and power consumption of the battery. Having a battery capacity of 5200 mAh × 6 ×3.7v= 115.2 Wh and have a power of consumption of 296 W× $4 \times 0.8 = 947\ W$, and flight time is calculated by

$$\frac{Battery\ capacity}{Power\ consumption}\times\ 60=\frac{115.2\ Wh}{947\ w}\approx 7.2\ minutes. \tag{1.34}$$

Maximum Altitude

Determined by utilizing the surplus thrust and climb rate, this parameter shows the highest height that can be reached. It is calculated by the following equation:$\Delta h=\frac{(excessthrust\times g\times t)}{weight}$, where the surplus thrust =11.16 kg$-3.5\ kg = 7.66\ kg, g = 9.81\ m/s^2$,and t=432 s (7.2 minutes). As a result,

$$\Delta h=\frac{7.66\ kg\times 9.81\ m/s^2\times 432\ s}{3.5\ kg}\approx 9{,}323\ m\ (9.3\ km\ theoretically). \tag{1.35}$$

However, because of battery drain and the fact that air density decreases with height, practical altitude is limited to 1-2 km.

Forward Speed

This detects the maximum horizontal speed and is calculated using

$$v=\sqrt{\frac{2\times excess\ power}{drag\ coefficient\times air\ density\times frontal\ area}}. \quad (1.36)$$

with excess power calculated by 296 W$\times 4 - 947\ W = 1184\ W - 947\ W = 237\ W$, a frontal area of 0.1 m², an air density of 1.225 kg/m³, and a drag coefficient of 0.5,

$$v=\sqrt{\frac{2\times 237\ W}{0.5\times 1.225\ kg/m^3\times 1.4\ m^2}} \approx \sqrt{\frac{474}{0.8575}} \approx 21\ m/s. \quad (1.36)$$

The TWR calculated at 3.19 is above normal and shows versatility and payload adaptation capability. Although the flight and hover times were reasonable (7.2 minutes) for the battery capacity (5200 mAh), higher battery capacity is required to perform longer flights. Real-world limits will keep the theoretical ceiling of 9.3 km well below attainable; 1-2 km is more likely. The 15-20 m/s range is consistent with previous iterations but forward speed estimate should be improved with additional inputs (propeller efficiency/frontal area). Although measurement needs a validation step that will take into consideration component-specific and environmental variables it can still provide and level of confidence towards additional testing and quality improvements [67].

7.4. Simulation

Constructing simulation for a drone for testing system performance, dependability, and accuracy of the GPS was required to have an overview about overall performance and the error in trajectory. Gazebo was chosen for its sophisticated physics engine and it supports multirotor model interface. Iris quadrotor model was chosen to simulate our flight and test our autonomous commands generated by python code. This simulation aimed to test the functionality of commands created through the Drone Kit library which is based in Python and supports communications with drones through the MAVLink protocol. Using Gazebo, a virtual environment was built where the Iris drone could execute standard procedures: take off, waypoint guided navigation, altitudinal adjustments, landing, and return-to-launch (RTL).

Working through these scenarios made it possible to resolve numerous potential issues ahead of time for physical testing. For example, command and timing delays, as well as GPS coordinates, to mainly see the variance in GPS readings. This greatly streamlined the debugging process while simultaneously protecting physical hardware from damage. Additionally, the real-time feedback from the drone's position, velocity, and orientation enhanced the command-and-control system logic backend structure.

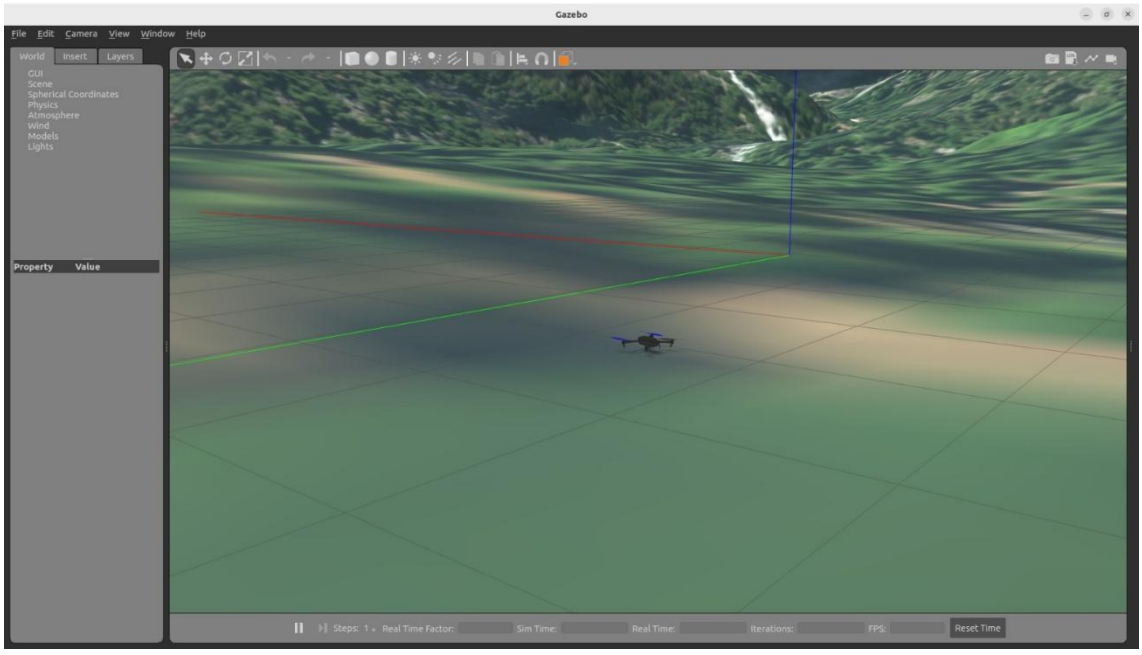


Figure 58 Iris Drone Simulated in Gazebo

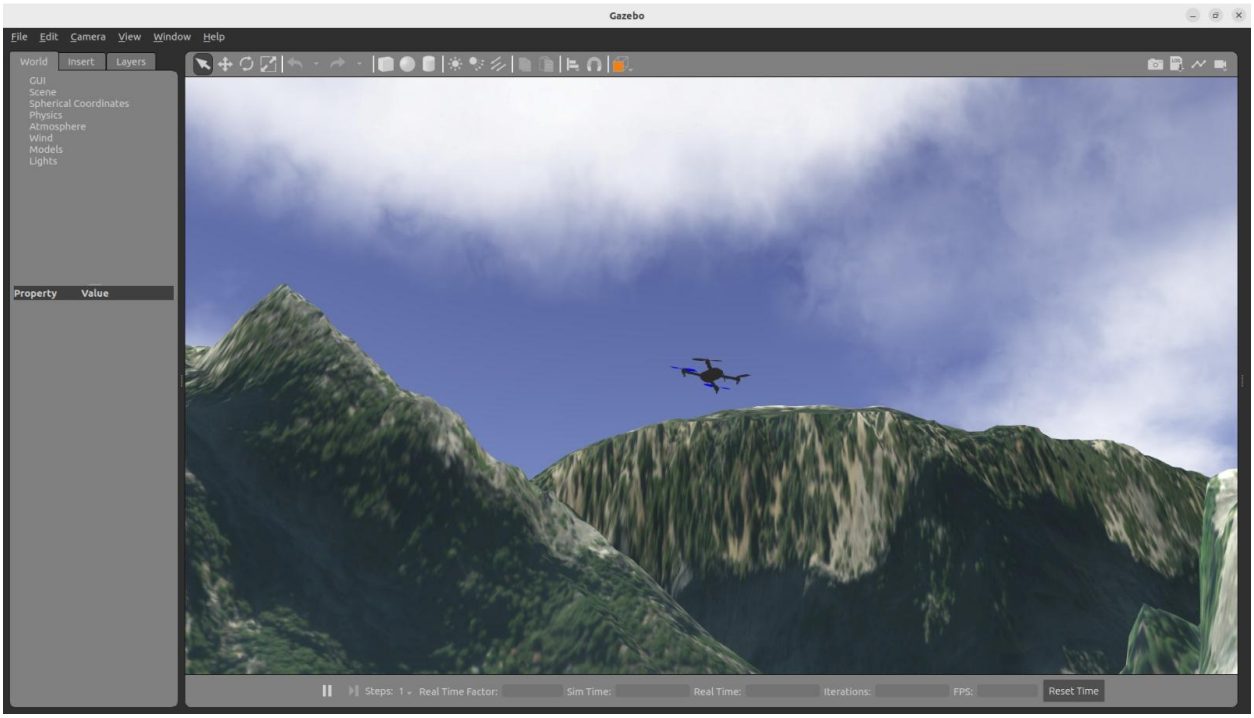


Figure 59 Iris Drone Flying in Gazebo

To sum up, utilizing the Gazebo simulator as a part of the Iris drone model was crucial in testing the drone control system's primary functions before actual implementation. It was a risk-free environment to conduct thorough testing of Drone Kit commands, test system dependability, and guarantee that the drone would function properly when connected to the live web interface. This phase of simulation boosted the safety and effectiveness of the entire system development lifecycle and gave us feedback about Gyro and GPS sensors and that their reading variance.

Chapter (8) Analysis and Discussion

8.1. Introduction

In the upcoming part the results of the flying attempts will be discussed in more details, starting with the measurements of the yaw, pitch, roll to the IMU readings, As the main aim of this research is to analyze environmental stability during flight operations while pinpointing essential maneuvers and examining how the aircraft performs, and outline routes creates a complete mission. In addition to analyze flight performance metrics while determining future research directions that could enhance operational outcomes.

8.2. Results and analysis

8.2.1. Axis Roll Yaw Pitch Data

The following data was extracted from the flight the drone have completed, Understanding the aircraft's orientation, or attitude data, is crucial to comprehending how it is flying. The first three parameters that will be investigated is the YAW, PITCH and ROLL. The first one to start with is the yaw, in the following Figure (46) the aircraft's orientation is shown by the yaw graphs, which span a length of time and primarily remain constant with a few minor oscillations. At one point, there is a noteworthy jump that most likely indicates a purposeful turn or a correction for wind.

Figure 60 Yaw desired VS Yaw actual.

The following graph in Figure (47) represents the Pitch orientation to indicate the difference between the desired pitch and the actual pitch as it indicates how the nose of the drone tilts up or down. At the start the line remain stable where both the parameters are actually zero witch is considered the hovering state meaning the plane's cruising straight and level. But along the trajectory somewhere in the middle, the drone takes a sharp drop in the actual pitch line—as if the plane suddenly tilts down more than it's supposed to. After that quick dip, the graph slowly drifts back up toward the baseline. Thought that sudden downward movement could be intentional, like the pilot or autopilot lowering the aircraft for a closer look at something below, or maybe it was a quick reaction to turbulence or a control input glitch. That can be seen with the fact that the pitch then recovers smoothly suggests the flight control system kicked in and corrected the attitude, bringing the plane back to a stable position. Also, since there are no other spikes in different parameters during this event, it seems like it was just a quick vertical adjustment rather than some broader instability or issue on the plane.

The following data was extracted from the flight the drone have completed, Understanding the aircraft's orientation, or attitude data, is crucial to comprehending how it is flying. The first three parameters that will be investigated is the YAW, PITCH and ROLL. This change probably means the aircraft briefly lost its balance, but the system quickly fixed it, or it could have been a banking turn, maybe to follow the curved path on the map. The roll went up and down before settling, suggesting the plane was flying a tricky route, like circling a target area. The eventual return to a

near-neutral position shows the flight control system did a good job stabilizing the roll and handling side-to-side movement, Figure (48).

Figure 61 Roll Desired VS Roll actual

8.2.2. IMU Gyroscope and Accelerometer Data

- **Gyroscope Graphs (IMU0 GyroX, GyroY, GyroZ):**

  In the following Figure (49) it can be clearly seen That gyroscope graphs show small changes in the aircraft's rotational speed along its axes, hinting at ongoing small adjustments to its orientation. A noticeable peak in the middle of the timeline, which lines up with sudden shifts in yaw, pitch, and roll, suggests the plane made a coordinated turn. The way the oscillations slowly calm down after this peak shows the aircraft was getting back to a steady flight, likely responding to commands from the pilot or autopilot. The steady baseline readings outside this event, as shown in the Figure (49), indicate the sensor suite is well-calibrated and working reliably.

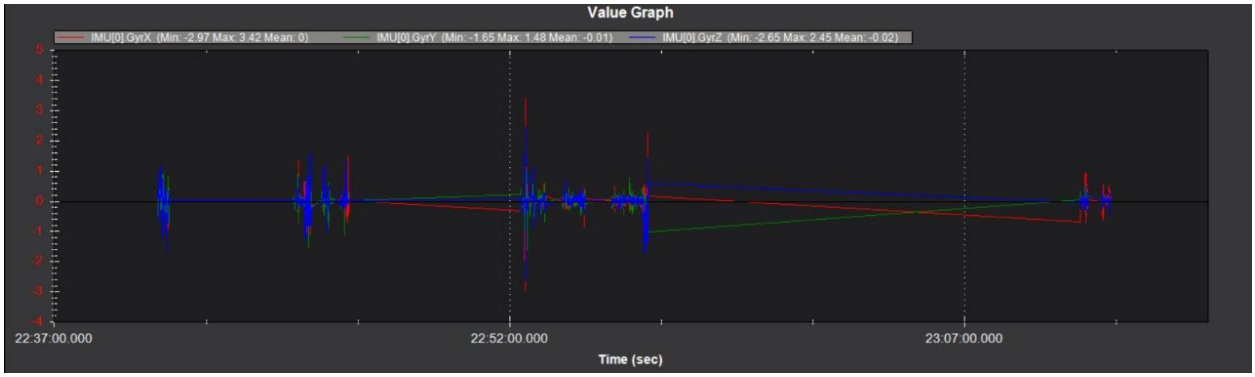


Figure 62 IMU 0 in all axis.

- **Accelerometer Graphs (IMU0 AccX, AccY, AccZ):**

  A varied profile of an aircraft's vertical and horizontal motions is shown by the accelerometer graphs. The data shows a consistent baseline for AccX and AccY, with occasional spikes indicating lateral changes or minor shocks. The AccZ axis shows a noticeable change halfway through the timeframe, dropping below the baseline and then rising again. This suggests a landing impact, a shift in height, or even a dip and then an ascension. Furthermore, alignment with the pitch drop supports a vertical movement, and subsequent stability indicates that the aircraft has adapted to a new equilibrium condition, as shown in the following Figure (50).

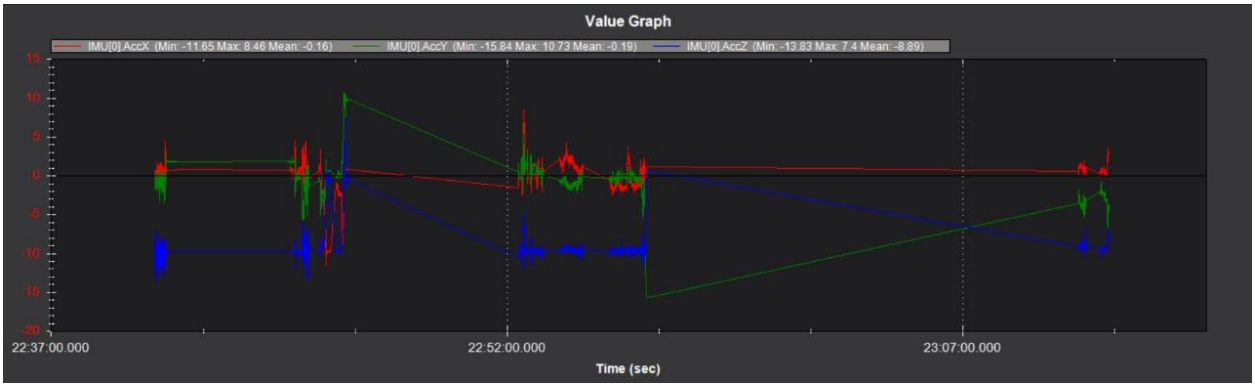


Figure 63 IMU Acceleration in X, Y, and Z Axis.

8.2.3. Flight Path GPS Data

the upcoming part shows the GPS data on satellite images provides a comprehensive flight route, displaying a complex trajectory with many twists and a center pivot point. This suggests that the aircraft was travelling along a predetermined route, maybe for mapping or inspection, as shown in Figure (51). The smooth turn transitions and the alignment with sensor data peaks demonstrate that the flight control system is effectively steering the drone along the designated course. The central pivot point, which may indicate a hover or loiter phase, significantly improves the mission's structure.

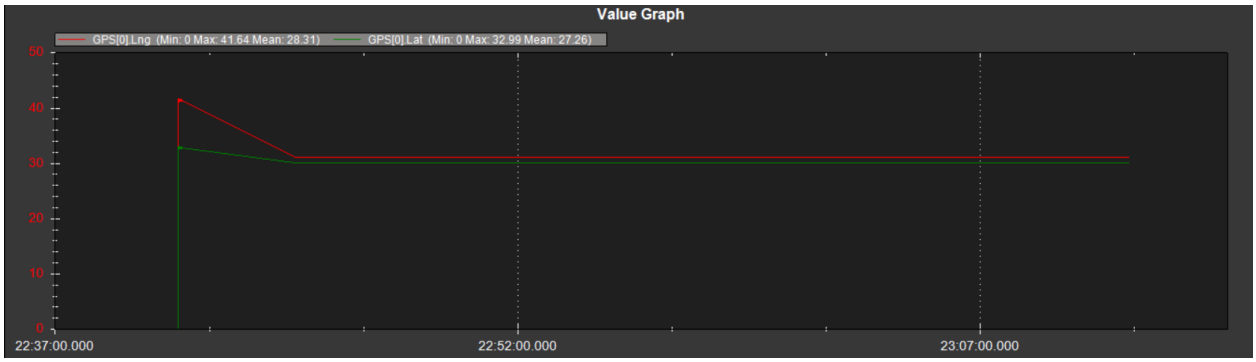


Figure 64 GPS Data.

Chapter (9) Conclusion

The thesis cultivated an autonomous quadrotor UAV that enhance the delivery of medical emergencies, thus tackling essential delays in the traditional ambulance services. With combinations of advanced mechanical, electric, and control systems, the drone shows a proficient solution for rapid medical supply delivery in urban and rural environments. This chapter depicts the most promising achievements, assesses the importance of the design, and recognizes its weaknesses, serving as a base for further advancement in drone-based emergency logistics.

9.1. Conclusion

Ambulance drone development was successful in fulfilling its main objective of developing a secure delivery system for medical products in emergency situations. The light yet strong and enduring structure of the design, autonomous flight modes, and secure payload method are significant advances in tackling real-world healthcare challenges. The study proves the effectiveness of drones in revolutionizing emergency response systems to provide swifter and more accessible healthcare.

The innovation of this design comes from how and what it uses in the technology of autonomous UAVs. With its ability to provide real-time delivery of life-saving goods, the drone is able to lower death rates by limiting prolonged emergency care, especially in densely populated or rural locations. It shatters existing drone technology through the supplement of a temperature-controlled payload chamber and also precision landing abilities, thus assisting in mass-scale use in medical logistics. The thesis also creates footpaths for research in next-generation propulsion and administrative or regulatory systems to improve it for greater efficiency.

Although successful, the design has some limitations that needs to be addressed. Limitations point out areas of improvement and future research direction:

- Limited Flight Time: 20 minutes limits operational range, and there is a need to improve battery life or other power sources such as fuel cells.
- Simplified Landing Gears: Moving towards lighter landing gears may compromise irregular terrain adaptability, thus altering landing security in challenging and difficult environments.
- Regulatory Challenges: BVLOS regulation compliance is yet to be completed, limiting long-range operation in actual or wide scale usage.
- Payload Capacity: The 1.5 kg payload capacity, even for small medical kits, may prove inadequate for larger or multiple supplies for certain emergencies.

These limitations mean that while the prototype represents progress, a further refinement is required in order to introduce it into general use within emergency medical services.

9.2. Future Work

To improve the efficiency and dependability of the medical UAV drone system, various aspects of future development are suggested. Integrating advanced AI and computer vision will enhance autonomous navigation and obstacle avoidance in complicated urban settings. Improving real-time data transfer and secure communication protocols will guarantee the safe distribution of critical medical supplies, particularly in emergency situations. Moreover, adding environmental sensors may enable drones to modify their flight routes according to weather conditions, enhancing safety and effectiveness.

Hybrid power systems, like solar-supported LiPo batteries, need to be investigated to prolong flight duration and enable long-distance missions. Additional component miniaturization and payload enhancement might allow for quicker deployment while maintaining the integrity of essential medical supplies. Creating coordination systems for swarm drones can be beneficial for extensive disaster relief, enabling numerous drones to operate together and decrease delivery times.

Connecting with hospital and healthcare IT systems will enhance logistics and guarantee smooth end-to-end monitoring. Finally, testing in the field and working together with medical experts and emergency responders is crucial to ensure that drone capabilities meet actual healthcare requirements. These improvements will lead to a more resilient, scalable, and life-saving medical UAV system that can respond quickly and dependably in various situations.

References

[1] B. Vergouw, H. Nagel, G. Bondt, and B. Custers, "Drone Technology: Types, Payloads, Applications, Frequency Spectrum Issues and Future Developments," 2016, pp. 21–45. doi: 10.1007/978-94-6265-132-6_2.

[2] R. Amin, L. Aijun, and S. Shamshirband, "A review of quadrotor UAV: control methodologies and performance evaluation," 2016.

[3] Randhawa Sonali and Hashmi Gulfam, "Death by delay: Revamping India's ambulance services could save lives lost to road crashes," scroll.in.

[4] A. Rauf, M. Irshad, W. Muhammad, S. Rasheed, N. Aziz, and H. Taj, "Comparative Study of SLAM Techniques for UAV," *Engineering Proceedings*, vol. 12, p. 67, Jan. 2022, doi: 10.3390/engproc2021012067.

[5] F. Betti Sorbelli, "UAV-Based Delivery Systems: A Systematic Review, Current Trends, and Research Challenges," *ACM Journal on Autonomous Transportation Systems*, vol. 1, no. 3, pp. 1–40, Sep. 2024, doi: 10.1145/3649224.

[6] S. Vertouk, "Can fuel cell powered drones overcome the challenge of long endurance flight? | Intelligent Energy Limited," https://www.intelligent-energy.com/news/can-fuel-cell-powered-drones-overcome-the-challenge-of-long-endurance-flight/.

[7] G. K. Pandey, D. S. Gurjar, H. H. Nguyen, and S. Yadav, "Security Threats and Mitigation Techniques in UAV Communications: A Comprehensive Survey," *IEEE Access*, vol. 10, pp. 112858–112897, 2022, doi: 10.1109/ACCESS.2022.3215975.

[8] A. Gholami, "Exploring drone classifications and applications: a review," May 2024, *Murat Yakar*. doi: 10.26833/ijeg.1428724.

[9] P. S. Ramesh and J. V. Muruga Lal Jeyan, "Comparative Analysis of Fixed-Wing, Rotary-Wing and Hybrid Mini Unmanned Aircraft Systems (UAS) from the Applications Perspective," *INCAS Bulletin*, vol. 14, pp. 137–151, 2022, doi: 10.13111/2066-8201.2022.14.1.12.

[10] G. Singhal, B. Bansod, and L. Mathew, “Unmanned Aerial Vehicle Classification, Applications and Challenges: A Review,” Nov. 2018. doi: 10.20944/preprints201811.0601.v1.

[11] “▷ GPS in Drones: What is it for and when to use it.” [Online]. Available: https://umilesgroup.com/en/gps-in-drones-what-it-is-for-and-when-to-use-it/

[12] E. Petritoli, F. Leccese, and G. S. Spagnolo, “Inertial Navigation Systems (INS) for Drones: Position Errors Model,” *2020 IEEE 7th International Workshop on Metrology for AeroSpace (MetroAeroSpace)*, pp. 500–504, Jun. 2020, doi: 10.1109/MetroAeroSpace48742.2020.9160304.

[13] “(PDF) Comparative Study of SLAM Techniques for UAV.” [Online]. Available: https://www.researchgate.net/publication/357624934_Comparative_Study_of_SLAM_Techniques_for_UAV

[14] H. Lv, F. Liu, and N. Yuan, “Drone Presence Detection by the Drone’s RF Communication,” *J Phys Conf Ser*, vol. 1738, p. 12044, Jan. 2021, doi: 10.1088/1742-6596/1738/1/012044.

[15] M. Y. Arafat, M. M. Alam, and S. Moh, “Vision-Based Navigation Techniques for Unmanned Aerial Vehicles: Review and Challenges,” Feb. 2023, *Multidisciplinary Digital Publishing Institute (MDPI)*. doi: 10.3390/drones7020089.

[16] D. W. Johnson and R. T. Johnson, “Cooperative Learning: The Foundation for Active Learning,” in *Active Learning - Beyond the Future*, IntechOpen, 2019. doi: 10.5772/intechopen.81086.

[17] H. Shakhatreh *et al.*, “Unmanned Aerial Vehicles (UAVs): A Survey on Civil Applications and Key Research Challenges,” 2019, *Institute of Electrical and Electronics Engineers Inc.* doi: 10.1109/ACCESS.2019.2909530.

[18] L. Davies, R. C. Bolam, Y. Vagapov, and A. Anuchin, “Review of Unmanned Aircraft System Technologies to Enable beyond Visual Line of Sight (BVLOS) Operations,” in *2018 10th International Conference on Electrical Power Drive Systems, ICEPDS 2018 - Conference Proceedings*, Institute of Electrical and Electronics Engineers Inc., Dec. 2018. doi: 10.1109/ICEPDS.2018.8571665.

[19] P. E. I. Pounds, D. R. Bersak, and A. M. Dollar, “Stability of small-scale UAV helicopters and quadrotors with added payload mass under PID control,” *Auton Robots*, vol. 33, no. 1, pp. 129–142, Aug. 2012, doi: 10.1007/s10514-012-9280-5.

[20] Y. Wu, D. Liu, T. Wang, and A. Zhao, “Research of a novel aerodynamic evaluation method for fixed-wing UAV,” *J Phys Conf Ser*, vol. 2633, p. 12001, Nov. 2023, doi: 10.1088/1742-6596/2633/1/012001.

[21] G. Liu, Q. Liu, H. Guo, M. Xiang, and J. Sang, “Optimization of ‘vehicle-UAV’ joint distribution routing for cold chain logistics considering risk of epidemic spreading and green cost,” *PLoS One*, vol. 19, no. 6, p. e0306127, Jun. 2024, doi: 10.1371/journal.pone.0306127.

[22] N. M. Wereley, Y. T. Choi, and H. J. Singh, “Adaptive energy absorbers for drop-induced shock mitigation,” *J Intell Mater Syst Struct*, vol. 22, pp. 515–519, Apr. 2011, doi: 10.1177/1045389X10393767.

[23] M. Eichleay, E. Evens, K. Stankevitz, and C. Parker, “Using the Unmanned Aerial Vehicle Delivery Decision Tool to Consider Transporting Medical Supplies via Drone,” *Glob Health Sci Pract*, vol. 7, p. GHSP–D, Dec. 2019, doi: 10.9745/GHSP-D-19-00119.

[24] T. Afrin, N. Yodo, A. Dey, and L. G. Aragon, “Advancements in UAV-Enabled Intelligent Transportation Systems: A Three-Layered Framework and Future Directions,” Oct. 2024, *Multidisciplinary Digital Publishing Institute (MDPI)*. doi: 10.3390/app14209455.

[25] J. Zhang, J. Campbell, D. Sweeney II, and A. Hupman, *Energy Consumption Models for Delivery Drones: A Comparison and Assessment*. 2020.

[26] “Design of Fully Automatic Drone Parachute System with Temperature Compensation Mechanism for Civilian and Military Applications - AL-Madani - 2018 - Journal of Advanced Transportation - Wiley Online Library.” [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1155/2018/2964583?msockid=39b626a543d163240f7e346d426662c2

[27] B. Madani, M. Ndiaye, and S. Salhi, “Hybrid truck-drone delivery system with multi-visits and multi-launch and retrieval locations: Mathematical model and adaptive variable neighborhood

search with neighborhood categorization," *Eur J Oper Res*, vol. 316, pp. 100–125, Jul. 2024, doi: 10.1016/j.ejor.2024.02.010.

[28] "Can fuel cell powered drones overcome the challenge of long endurance flight? Intelligent Energy Limited." [Online]. Available: https://www.intelligent-energy.com/news/can-fuel-cell-powered-drones-overcome-the-challenge-of-long-endurance-flight/

[29] G. R. April 2021, "Tactical drone, powered by solar panels and hydrogen fuel cell, flies 24h." [Online]. Available: https://www.flightglobal.com/military-uavs/tactical-drone-powered-by-solar-panels-and-hydrogen-fuel-cell-flies-24h/143358.article

[30] N. Saini, "AI-Powered Smart Drones: Cutting-Edge Features Explained," Oct. 2024. [Online]. Available: https://www.hashstudioz.com/blog/how-ai-makes-drones-smarter-a-deep-dive-into-cutting-edge-features/

[31] R. Kumar *et al.*, "Parachute deployment system for safe recovery of a drone," *Mater Today Proc*, 2023, doi: 10.1016/j.matpr.2023.04.114.

[32] "Unpowered MIP gripper allows drones to passively perch and grasp," https://newatlas.com/drones/mip-gripper-drones-perch-grasp/.

[33] M. Simon, "Would Delivery Drones Be All That Efficient? Depends Where You Live," https://www.wired.com/story/would-delivery-drones-be-all-that-efficient/.

[34] B. F. Erlingsson, I. Hreimsson, P. I. Pálsson, S. J. Hjálmarsson, and J. T. Foley, "Axiomatic Design of a Linear Motion Robotic Claw with Interchangeable Grippers," in *Procedia CIRP*, Elsevier B.V., 2016, pp. 213–218. doi: 10.1016/j.procir.2016.07.006.

[35] Z. Wang, "Landing gear design for emergency take-off and landing of drones on unconventional landings," SPIE-Intl Soc Optical Eng, Apr. 2024, p. 27. doi: 10.1117/12.3027186.

[36] Y. S. Sarkisov, G. A. Yashin, E. V. Tsykunov, and D. Tsetserukou, "DroneGear: A novel robotic landing gear with embedded optical torque sensors for safe multicopter landing on an uneven surface," *IEEE Robot Autom Lett*, vol. 3, pp. 1912–1917, Jul. 2018, doi: 10.1109/LRA.2018.2806080.

[37] N. Çabuk, “Design and Kinematic Analysis of Proposed Adaptive Landing Gear for Multirotor UAV,” *El-Cezeri Journal of Science and Engineering*, vol. 9, pp. 159–170, 2022, doi: 10.31202/ecjse.952728.

[38] M. Huang, H. Nie, M. Zhang, X. Wei, and S. Yue, “2247. Design of mission adaptive landing gear for near space travel lander,” *Journal of Vibroengineering*, vol. 18, pp. 4949–4963, Dec. 2016, doi: 10.21595/jve.2016.17078.

[39] S. Baker, D. Soccol, A. Postula, and M. V Srinivasan, “Passive Landing Gear using Coupled Mechanical Design.”

[40] L. Oscar, “What to Consider in FPV Drone Frames and Top Recommendations - Oscar Liang,” https://oscarliang.com/fpv-drone-frames/.

[41] C. Harika, A. Sai Kumar, and M. V. Raghavendra Rao, “Comparative Study on Effect of Material on structural Performance of a Quadcopter Drone with ‘X- Frame,’” in *Journal of Physics: Conference Series*, Institute of Physics, 2024. doi: 10.1088/1742-6596/2837/1/012099.

[42] J. Flynt, “How to Choose the Right Quadcopter Drone Frame,” https://3dinsider.com/quadcopter-frames/.

[43] J. Corbyn, “FPV Drone Frames: Building the Foundation for Thrilling Flights,” https://droneracinghub.com/fpv-drone-frames/.

[44] A. Valiavalappil, “STRESS, EQUIVALENT STRAIN AND DEFORMATION FOR MACHINING OPERATION UTILIZING EXPLICIT DYNAMIC ANALYSIS FOR VARIOUS MATERIALS,” *Yanbu Journal of Engineering and Science*, vol. 21, no. 1, Apr. 2024, doi: 10.53370/001c.94740.

[45] N. Chowdhury, A. Khan, G. M. Faysal, H. Rahman, T. Islam, and F. Yeasmin, “A Study on Mechanical Properties of Carbon Fiber Reinforced Polymer Composite,” 2021. [Online]. Available: https://www.researchgate.net/publication/349176504

[46] J. M. Dulieu-Barton and M. C. Fulton, “Mechanical Properties of a Typical Stereolithography Resin,” *Strain*, vol. 36, no. 2, pp. 81–87, May 2000, doi: 10.1111/j.1475-1305.2000.tb01177.x.

[47] L. Sørensen, L. Jacobsen, and J. Hansen, “Low Cost and Flexible UAV Deployment of Sensors,” *Sensors*, vol. 17, no. 1, p. 154, Jan. 2017, doi: 10.3390/s17010154.

[48] Y. N. Saravanakumar *et al.*, “Power Sources for Unmanned Aerial Vehicles: A State-of-the Art,” *Applied Sciences*, vol. 13, no. 21, p. 11932, Oct. 2023, doi: 10.3390/app132111932.

[49] K. S. Hatamleh, O. Ma, A. Flores-Abad, and P. Xie, “Development of a Special Inertial Measurement Unit for UAV Applications,” *J Dyn Syst Meas Control*, vol. 135, no. 1, Jan. 2013, doi: 10.1115/1.4007122.

[50] E. Yanmaz, S. Yahyanejad, B. Rinner, H. Hellwagner, and C. Bettstetter, “Drone networks: Communications, coordination, and sensing,” *Ad Hoc Networks*, vol. 68, pp. 1–15, Jan. 2018, doi: 10.1016/j.adhoc.2017.09.001.

[51] B. Antaki, G. Elasar, H. Arif, and K. Yıldırım, “Design of a UAV for Medical Supply Delivery”, doi: 10.13140/RG.2.2.35127.50085.

[52] M. Nadir Boukoberine, Z. Zhou, and M. Benbouzid, “A critical review on unmanned aerial vehicles power supply and energy management: Solutions, strategies, and prospects A critical review on unmanned aerial vehicles power supply and energy management: Solutions, strategies, and prospects A Critical Review on Unmanned Aerial Vehicles Power Supply and Energy Management: Solutions, Strategies, and Prospects,” *Appl Energy*, vol. 255, p. 113823, 2019, doi: 10.1016/j.apenergy.2019.113823ï.

[53] P. Bogusz, M. Korkosz, A. Powrozek, J. Prokop, and P. Wygonik, “An analysis of properties of the BLDC motor for unmanned aerial vehicle hybrid drive,” in *2015 International Conference on Electrical Drives and Power Electronics (EDPE)*, IEEE, Sep. 2015, pp. 458–464. doi: 10.1109/EDPE.2015.7325338.

[54] K. Mohta *et al.*, “Fast, autonomous flight in GPS-denied and cluttered environments,” *J Field Robot*, vol. 35, no. 1, pp. 101–120, Jan. 2018, doi: 10.1002/rob.21774.

[55] N. Gupta, R. Chauhan, and S. Chadha, “Unmanned Aerial Vehicle (UAV) for Parcel Delivery,” *International Journal of Engineering Research and Technology*, vol. 13, no. 10, p. 2824, Oct. 2020, doi: 10.37624/IJERT/13.10.2020.2824-2830.

[56] T. Q. Khoi, N. A. Quang, and N. K. Hieu, "Object detection for drones on Raspberry Pi potentials and challenges," *IOP Conf Ser Mater Sci Eng*, vol. 1109, no. 1, p. 012033, Mar. 2021, doi: 10.1088/1757-899X/1109/1/012033.

[57] S. Bouabdallah, A. Noth, and R. Siegwart, "PID vs LQ control techniques applied to an indoor micro quadrotor," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, IEEE, 2004, pp. 2451–2456 vol.3. doi: 10.1109/IROS.2004.1389776.

[58] M. Faessler, F. Fontana, C. Forster, E. Mueggler, M. Pizzoli, and D. Scaramuzza, "Autonomous, Vision-based Flight and Live Dense 3D Mapping with a Quadrotor Micro Aerial Vehicle," *J Field Robot*, vol. 33, no. 4, pp. 431–450, Jun. 2016, doi: 10.1002/rob.21581.

[59] E. Yanmaz, S. Yahyanejad, B. Rinner, H. Hellwagner, and C. Bettstetter, "Drone networks: Communications, coordination, and sensing," *Ad Hoc Networks*, vol. 68, pp. 1–15, Jan. 2018, doi: 10.1016/j.adhoc.2017.09.001.

[60] J. Herdiansyah et al., "Implementation of Zhang's Camera Calibration Algorithm on a Single Camera for Accurate Pose Estimation Using ArUco Markers," *J. Fuzzy Syst. Control*, vol. 2, no. 3, pp. 176–188, 2024.

[61] OpenCV Documentation, "Camera Calibration and 3D Reconstruction," https://docs.opencv.org/4.x/dc/dbb/tutorial_py_calibration.html.

[62] Merei, Ahmad, et al. "A Survey on Obstacle Detection and Avoidance Methods for UAVs." *Drones*, vol. 9, no. 3, Mar. 2025, p. 203. *DOI:* https://doi.org/10.3390/drones9030203.

[63] Du, Zongdong, et al. "A Lightweight UAV Visual Obstacle Avoidance Algorithm Based on Improved YOLOv8." *Computers, Materials & Continua*, vol. 81, no. 2, 2024, pp. 2607–27. *DOI:* https://doi.org/10.32604/cmc.2024.056616

[64] P. Rudol and P. Doherty, "Human body detection and geolocalization for UAV search and rescue missions using color and thermal imagery," in *Proc. IEEE Aerospace Conf.*, Mar. 2008, pp. 1–8. doi: 10.1109/AERO.2008.4526389.

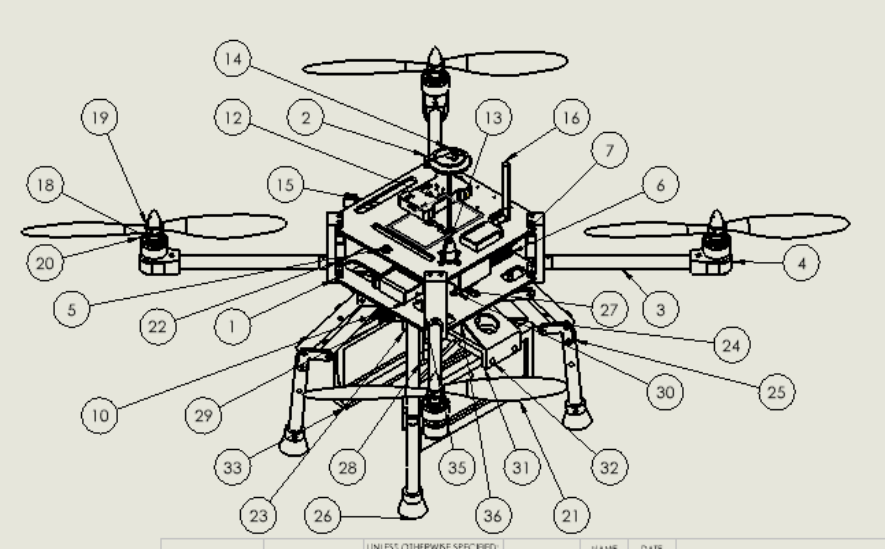
[65] M. A. Achtelik, A. Bachrach, R. He, S. Prentice, and N. Roy, “Autonomous navigation and exploration of a quadrotor helicopter in GPS-denied indoor environments,” *Robotics Research*, vol. 70, pp. 85–103, 2011.

[66] Lorenz Meier, et al., “PX4: A node-based multithreaded open-source robotics framework for deeply embedded platforms,” in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, May 2015, pp. 6235–6240. doi: 10.1109/ICRA.2015.7140074.

[67] QuadPartPicker. (2024, May 17). How to estimate and calculate drone flight characteristics. QuadPartPicker News & Intel. Retrieved June 11, 2025.

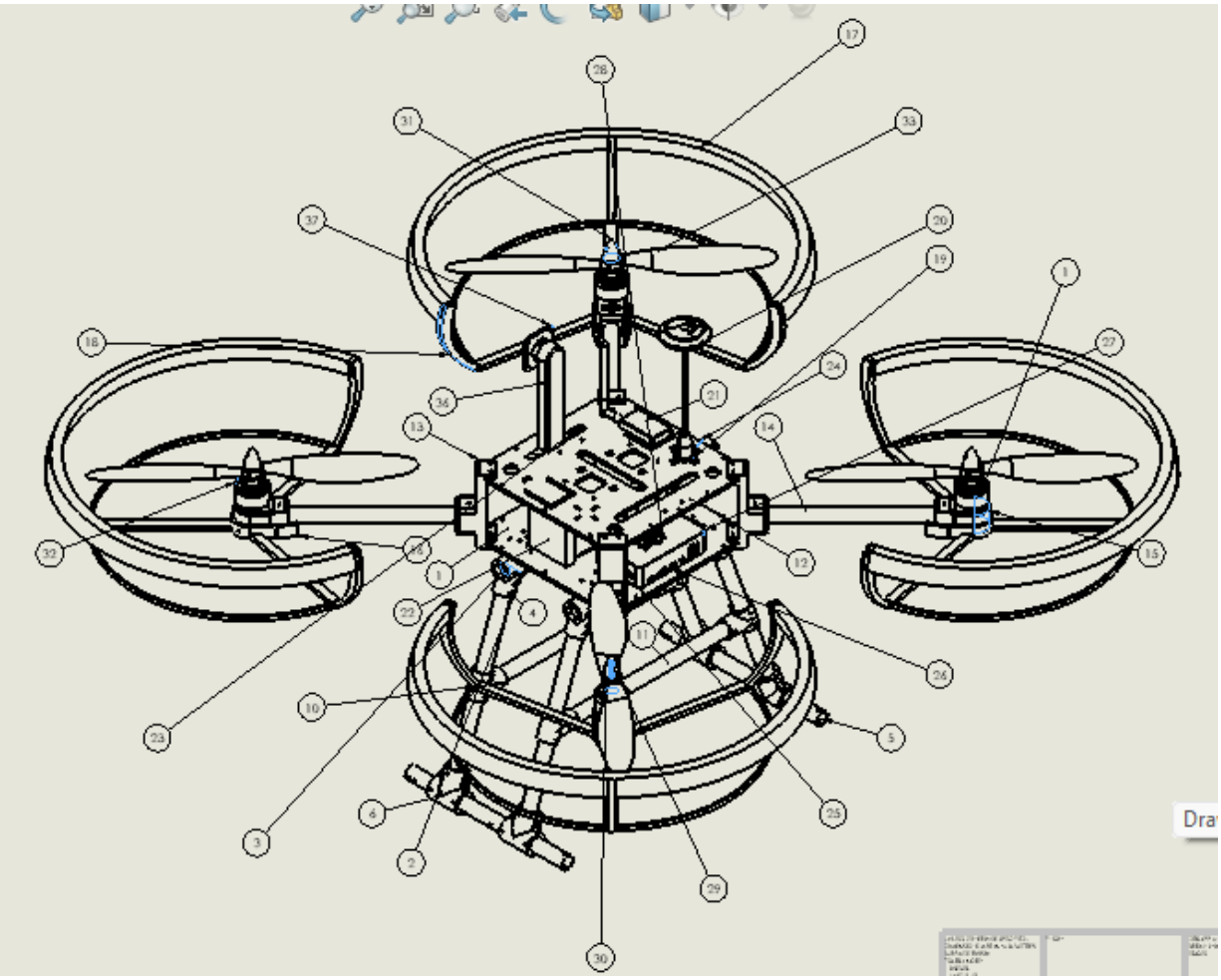
Appendix 1

1- Complete Assembly and Components Drawing Sheet (Design 1)



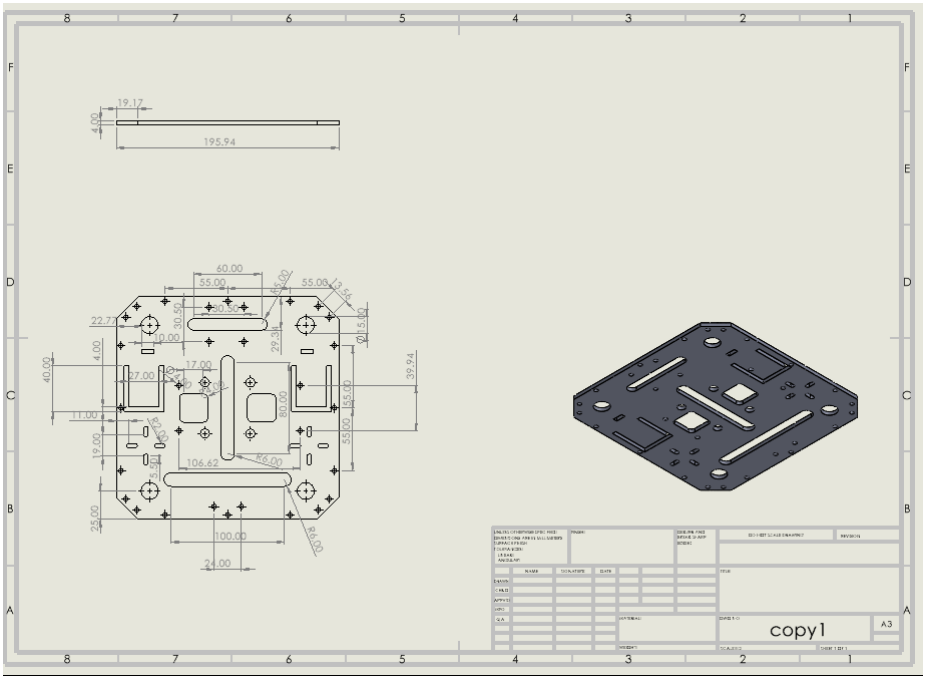
ITEM NO.	PART NUMBER	DESCRIPTION	QTY.
1	base 0 withcomponents 02023		1
2	armassem		4
3	base 1 with components 02023		1
4	raspberrypi with case		1
5	reciver	NOT SPECIFIED	1
6	PDB		1
7	Pixhawk		1
8	GPS_MN8_ASM_STEP		1
9	Battery		1
10	TELEMETRY		1
11	Pixhawkbase	NONE	1
12	Spinner M6		4
13	ESC2023 (1)		4
14	Assem		2
15	Servo base		1
16	Sliding Cylinder		3
17	Gripper arm		3
18	servo link		1
19	servo		1
20	gripper link		2

2- Complete Assembly and Components Drawing Sheet (Finalized Design)

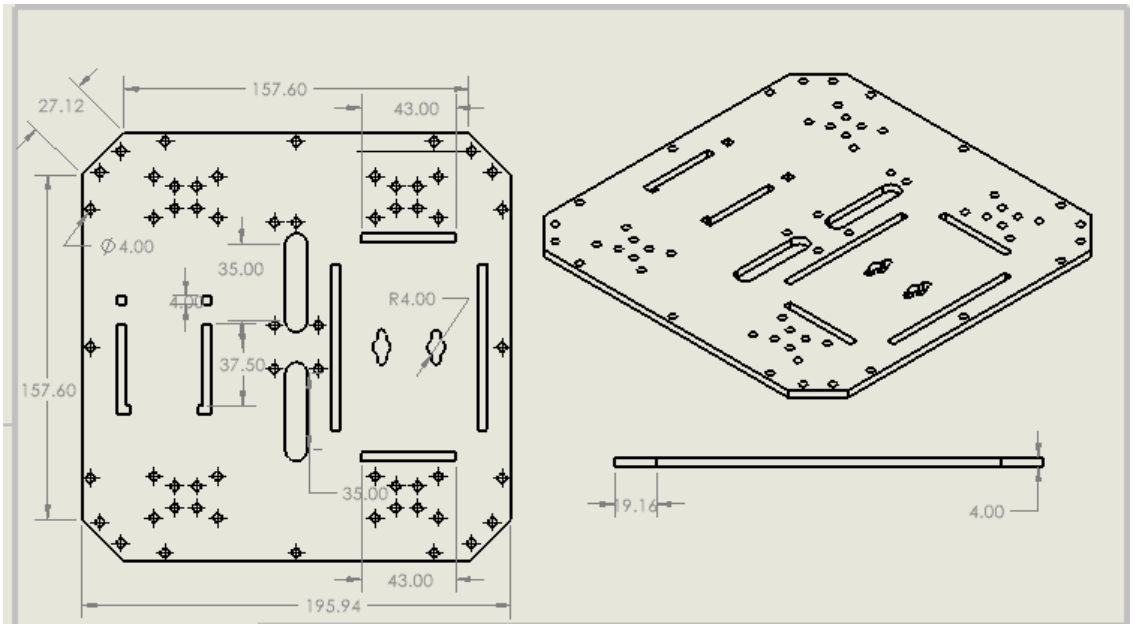


ITEM NO.	PART NUMBER	DESCRIPTION	QTY.
1	Lower_Plate		1
2	Landing_Gears_Support		4
3	Upper_Horizontal_Rods		2
4	Connector_Between_HorizonalRods_and_Main_Landing_Rods		4
5	Main_Landing_Gears		6
6	Landing_Gears_Legs		4
7	gripper		1
8	Gripper_Link		1
9	Box_Handle		1
10	LandingGears_Box_Supporter		4
11	HorizontalRods_Box		2
12	base 1 with components 020231		1
13	armassem		4
14	PropellerGuards		4
15	GPS_MN8_ASM_STEP		1
16	TELEMETRY		1
17	BatteryHolder		1
18	PDB		1
19	Battery		1
20	raspberrypi with case		1
21	Spinner M6		4
22	reciver	NOT SPECIFIED	1
23	servo		1
24	obstrasp		1
25	cameraraspb		1

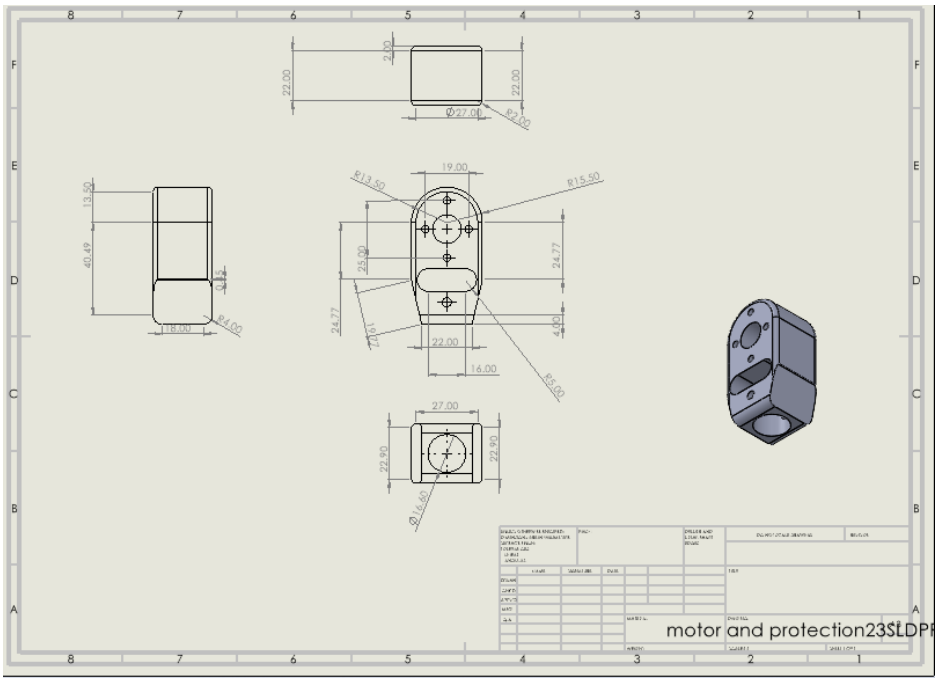
3- Upper Plate



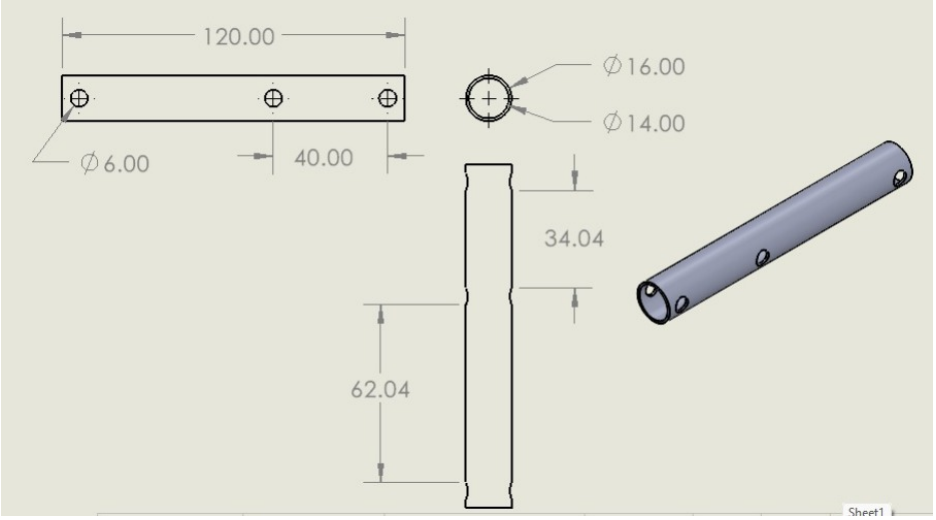
4- Lower Plate



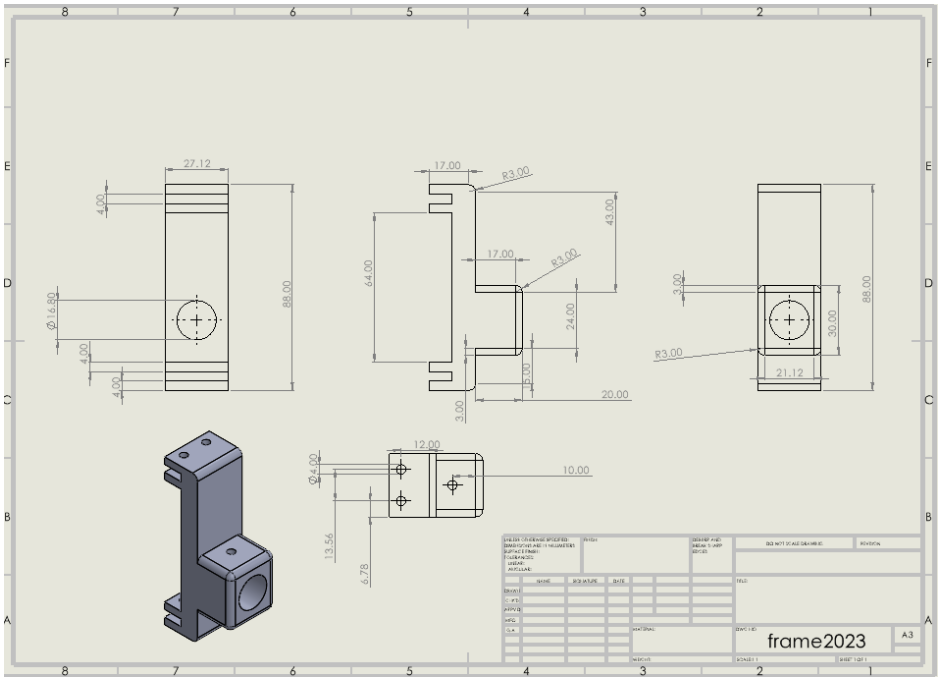
5- Motor Mount



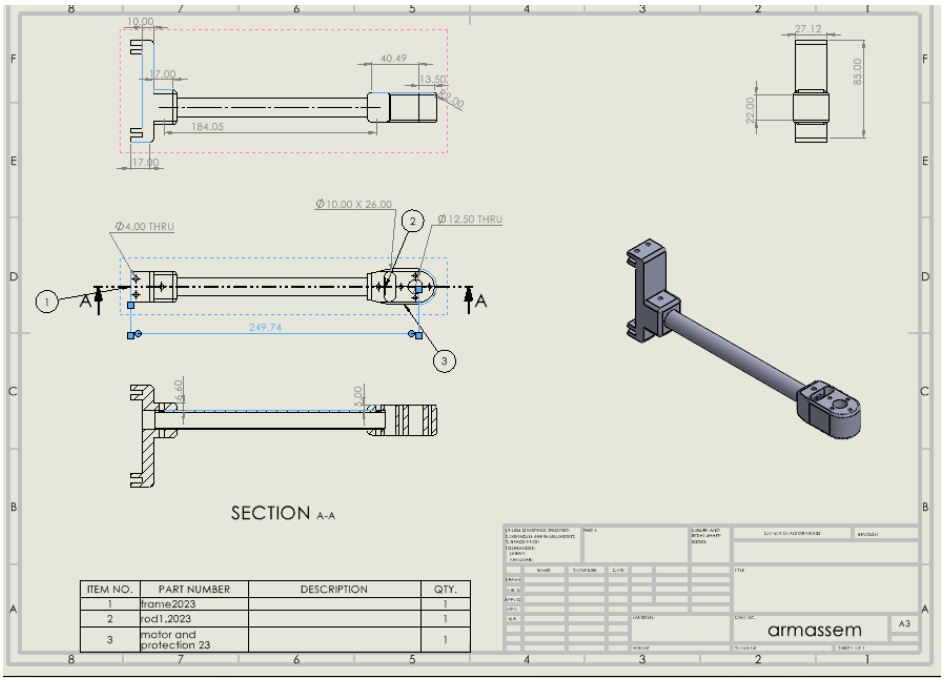
5- Carbon Fiber Quadrotor Arm



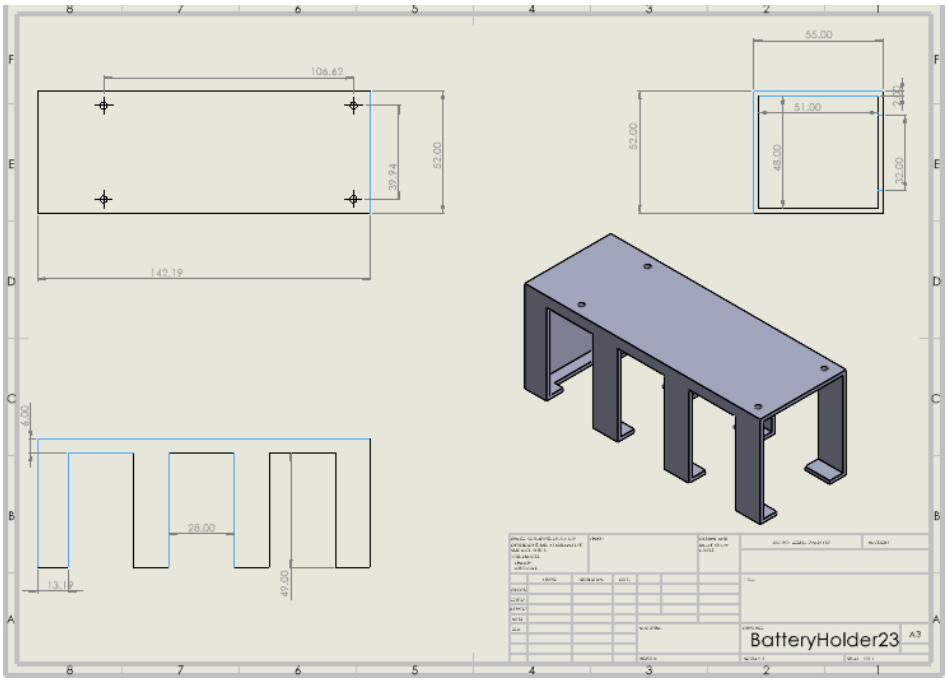
6- Base Holders



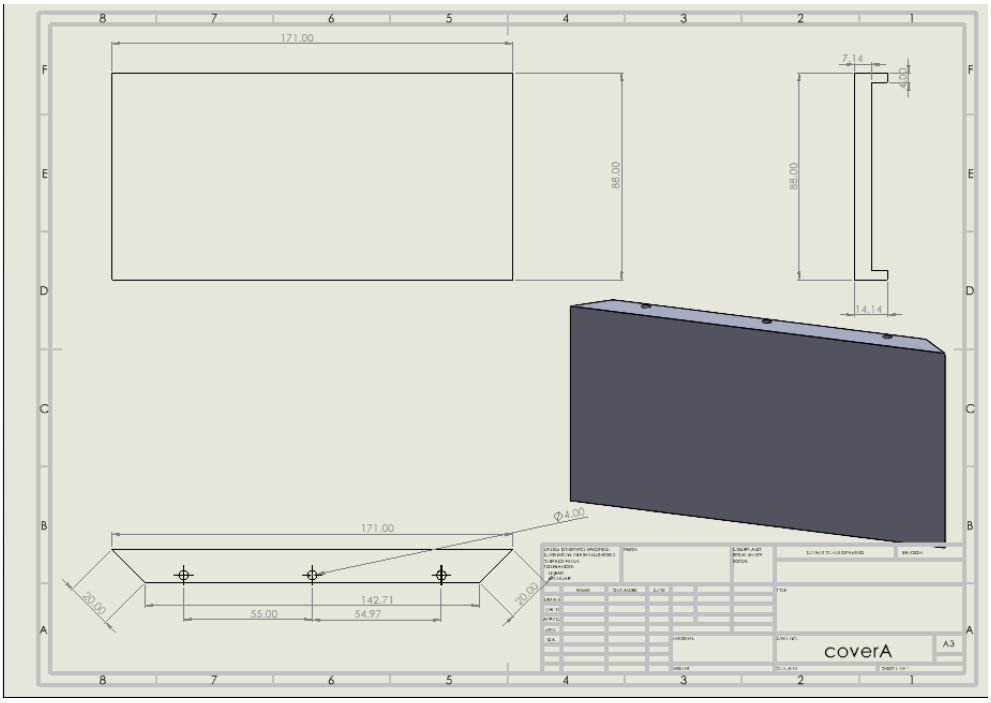
7- Full Arm assembly



8- Battery Holder

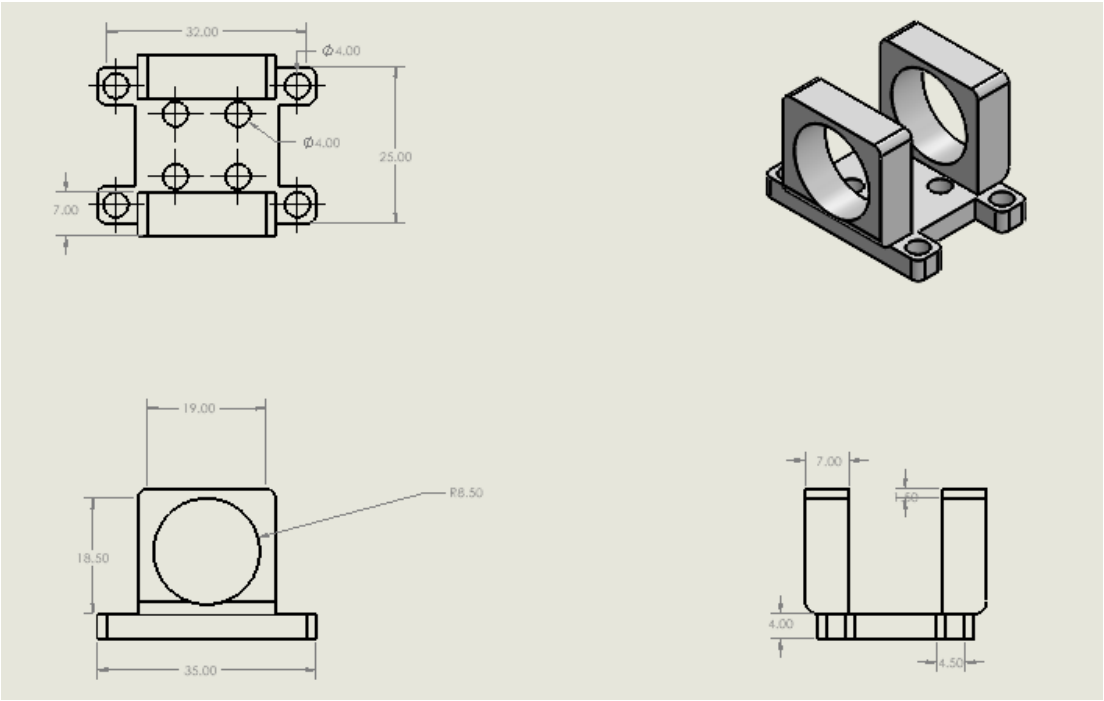


9- Sides Covers

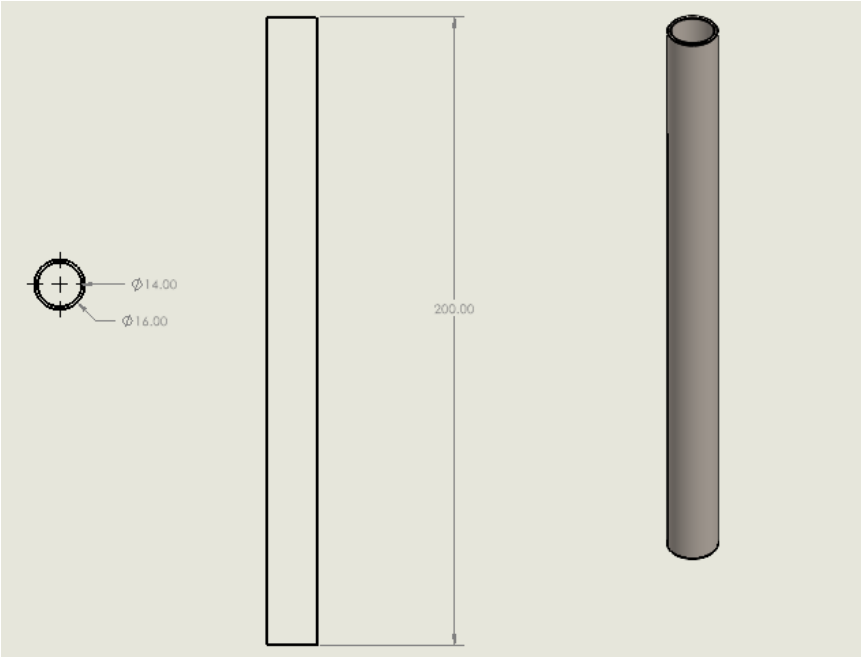


10-Landing Gears

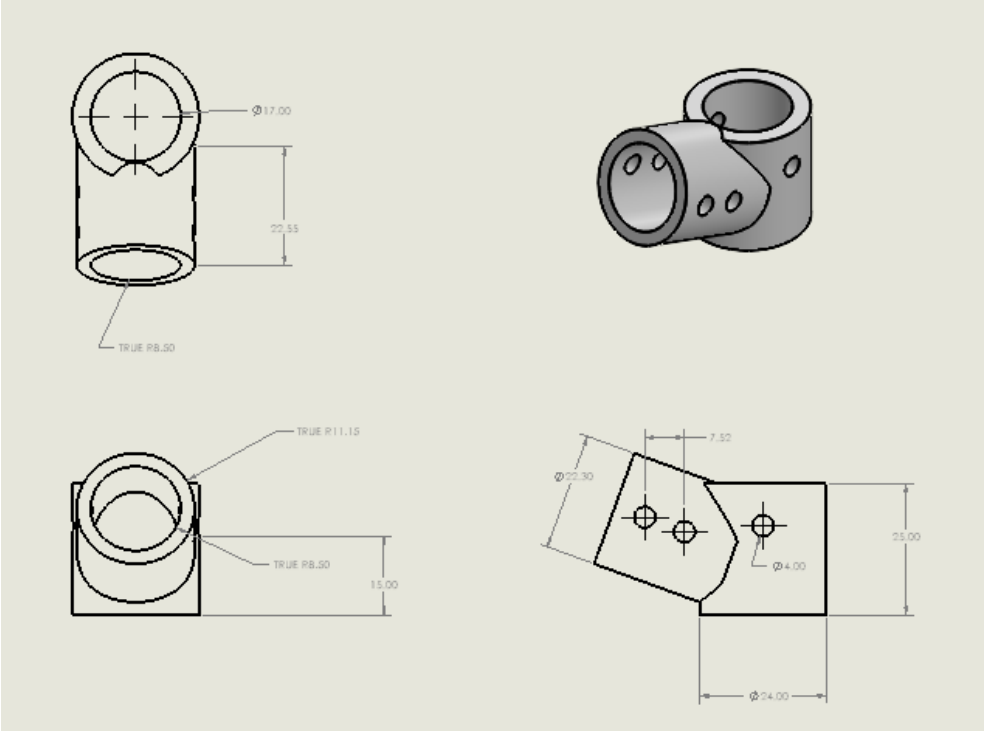
10.1 – Landing Gears (Supports)



10.2- Landing Gears Horizontal Rods



10.3- Connector Between Horizontal Rods and Main Landing Gear Rods



Appendix 2

1- Box Arduino Code:

```
#include <Keypad.h>                // Include the Keypad library to interface with the 4x3 keypad for password input
#include <LiquidCrystal_I2C.h>     // Include the LiquidCrystal_I2C library to control the 16x2 LCD via I2C interface
#include <Servo.h>                 // Include the Servo library to control the servo motor for locking/unlocking the door
#include <DHT.h>                  // Include the DHT library to read temperature and humidity from the DHT11 sensor

// ------------------ Pin Definitions ------------------
#define SERVO_PIN      9          // Define the pin for the servo motor (pin 9 on Arduino Nano)
#define RED_LED_PIN    11         // Define the pin for the red LED (pin 11 on Arduino Nano)
#define GREEN_LED_PIN  12         // Define the pin for the green LED (pin 12 on Arduino Nano)
#define FAN_PIN        10         // Define the pin for the fan control (pin 10 on Arduino Nano)
#define DHT_PIN        A3         // Define the pin for the DHT11 sensor (analog pin A3 on Arduino Nano)
#define DHT_TYPE       DHT11      // Define the type of DHT sensor (DHT11 in this case)

// ------------------ Objects ------------------
DHT dht(DHT_PIN, DHT_TYPE);       // Create a DHT object to interface with the DHT11 sensor, using the defined pin and type
Servo lockServo;                  // Create a Servo object to control the servo motor for locking/unlocking the door
LiquidCrystal_I2C lcd(0x27, 16, 2);  // Create a LiquidCrystal_I2C object for the LCD, with I2C address 0x27, 16 columns, and 2 rows

// ------------------ Keypad ------------------
const byte ROWS = 4;              // Define the number of rows in the keypad (4 rows)
const byte COLS = 3;              // Define the number of columns in the keypad (3 columns)
char keys[ROWS][COLS] = {         // Define the keymap for the 4x3 keypad, mapping each button to a character
```

```
  {'1','2','3'},                // Row 1: keys '1', '2', '3'
  {'4','5','6'},                // Row 2: keys '4', '5', '6'
  {'7','8','9'},                // Row 3: keys '7', '8', '9'
  {'*','0','#'}                 // Row 4: keys '*', '0', '#'
};
byte rowPins[ROWS] = {8, 7, 6, 5}; // Define the Arduino pins connected to the keypad rows (pins 8, 7, 6, 5)
byte colPins[COLS] = {4, 3, 2};    // Define the Arduino pins connected to the keypad columns (pins 4, 3, 2)
Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, ROWS, COLS); // Create a Keypad object with the defined keymap, row pins, column pins, rows, and columns

// ----------------- Password & Timer -----------------
String inputPassword;           // Declare a String variable to store the user-entered password
String displayPassword;         // Declare a String variable to store asterisks (*) for displaying on the LCD (hides actual password)
const String correctPassword = "1234"; // Define the correct password ("1234") that unlocks the system
bool doorUnlocked = false;        // Declare a boolean to track the door's locked/unlocked state (false = locked initially)
unsigned long unlockTime = 0;     // Declare a variable to store the time (in milliseconds) when the door was unlocked
const unsigned long unlockDuration = 180000; // Define the duration (in milliseconds) the door stays unlocked (180,000 ms = 3 minutes)

// ----------------- Function Declarations -----------------
void lockDoor();                // Declare the lockDoor() function to lock the door and update indicators
void unlockDoor();              // Declare the unlockDoor() function to unlock the door and update indicators
void monitorDHTandFan();        // Declare the monitorDHTandFan() function to read DHT11 data and control the fan

void setup() {                    // The setup() function runs once when the Arduino starts
  Serial.begin(9600);             // Initialize serial communication at 9600 baud rate for debugging and monitoring
  lcd.init();                     // Initialize the I2C LCD communication
  lcd.backlight();                // Turn on the LCD backlight for visibility
```

```
  dht.begin();                  // Initialize the DHT11 sensor to start reading
temperature and humidity
   lockServo.attach(SERVO_PIN);    // Attach the servo motor to the defined
SERVO_PIN (pin 9) for control
  pinMode(RED_LED_PIN, OUTPUT);  // Set the red LED pin as an output to control
the LED state
   pinMode(GREEN_LED_PIN, OUTPUT); // Set the green LED pin as an output to
control the LED state
  pinMode(FAN_PIN, OUTPUT);      // Set the fan pin as an output to control the
fan state
  digitalWrite(FAN_PIN, LOW);    // Initially turn off the fan by setting the
fan pin to LOW

  lockDoor();                     // Call the lockDoor() function to lock the
door and set initial indicator states

  lcd.setCursor(0, 0);           // Set the LCD cursor to column 0, row 0 (top-
left corner)
   lcd.print("Enter Password");    // Display "Enter Password" on the LCD to
prompt the user
  Serial.println("LCD: Enter Password"); // Print to Serial Monitor to confirm
the LCD message

   inputPassword.reserve(16);       // Reserve 16 bytes of memory for the
inputPassword String to optimize memory usage
   displayPassword.reserve(16);     // Reserve 16 bytes of memory for the
displayPassword String to optimize memory usage
}

void loop() {                   // The loop() function runs continuously after
setup()
  if (doorUnlocked && millis() - unlockTime < unlockDuration) { // Check if the
door is unlocked and the unlock duration (3 minutes) has not expired
      monitorDHTandFan();              // Call monitorDHTandFan() to read
temperature/humidity and control the fan
  } else if (doorUnlocked && millis() - unlockTime >= unlockDuration) { // Check
if the door is unlocked but the unlock duration has expired
    lockDoor();                   // Call lockDoor() to lock the door and reset
indicators
    lcd.clear();                  // Clear the LCD display
    lcd.setCursor(0, 0);          // Set the LCD cursor to column 0, row 0
     lcd.print("Enter Password"); // Display "Enter Password" on the LCD to
prompt for a new password
```

```
    Serial.println("LCD: Enter Password"); // Print to Serial Monitor to confirm
the LCD message
     doorUnlocked = false;         // Set the doorUnlocked flag to false (door
is now locked)
    inputPassword = "";           // Clear the inputPassword String
    displayPassword = "";         // Clear the displayPassword String
  }

  char key = keypad.getKey();     // Read the key pressed on the keypad (returns
0 if no key is pressed)
  if (key) {                      // Check if a key was pressed (non-zero value)
     Serial.print("Keypad: ");     // Print "Keypad: " to Serial Monitor to
indicate a key press
    Serial.println(key);          // Print the pressed key to Serial Monitor for
debugging

    if (key == '*') {             // Check if the '*' key was pressed (used to
reset the password input)
      inputPassword = "";         // Clear the inputPassword String
      displayPassword = "";       // Clear the displayPassword String (asterisks)
      lcd.clear();                // Clear the LCD display
      lcd.setCursor(0, 0);        // Set the LCD cursor to column 0, row 0
      lcd.print("Enter Password"); // Display "Enter Password" on the LCD
       Serial.println("LCD: Enter Password"); // Print to Serial Monitor to
confirm the LCD message
    }
    else if (key == '#') {        // Check if the '#' key was pressed (used to
submit the password)
      lcd.clear();                // Clear the LCD display
      if (inputPassword == correctPassword) { // Compare the entered password
with the correct password
        lcd.setCursor(0, 0);      // Set the LCD cursor to column 0, row 0
        lcd.print("Access Granted"); // Display "Access Granted" on the LCD to
indicate successful unlock
         Serial.println("LCD: Access Granted"); // Print to Serial Monitor to
confirm the LCD message
         unlockDoor();            // Call unlockDoor() to unlock the door and
update indicators
        unlockTime = millis();    // Record the current time (in milliseconds)
when the door was unlocked
        doorUnlocked = true;      // Set the doorUnlocked flag to true (door is
now unlocked)
      } else {                    // If the entered password is incorrect
```

```
        lcd.setCursor(0, 0);     // Set the LCD cursor to column 0, row 0
        lcd.print("Wrong Password"); // Display "Wrong Password" on the LCD
         Serial.println("LCD: Wrong Password"); // Print to Serial Monitor to
confirm the LCD message
         delay(2000);              // Wait for 2 seconds to allow the user to
read the message
        lcd.clear();             // Clear the LCD display
        lcd.setCursor(0, 0);     // Set the LCD cursor to column 0, row 0
        lcd.print("Enter Password"); // Display "Enter Password" on the LCD to
prompt for a new attempt
      }
      inputPassword = "";        // Clear the inputPassword String
      displayPassword = "";      // Clear the displayPassword String
    }
    else if (doorUnlocked && key == '0') { // Check if the '0' key was pressed
while the door is unlocked (manual lock)
      lockDoor();                // Call lockDoor() to lock the door and reset
indicators
      lcd.clear();               // Clear the LCD display
      lcd.setCursor(0, 0);       // Set the LCD cursor to column 0, row 0
      lcd.print("Enter Password"); // Display "Enter Password" on the LCD
       Serial.println("LCD: Enter Password"); // Print to Serial Monitor to
confirm the LCD message
       doorUnlocked = false;      // Set the doorUnlocked flag to false (door
is now locked)
      inputPassword = "";        // Clear the inputPassword String
      displayPassword = "";      // Clear the displayPassword String
    }
    else if (!doorUnlocked) {    // Check if the door is locked and a key (not
'*' or '#') was pressed
      inputPassword += key;      // Append the pressed key to the inputPassword
String
       displayPassword += '*';    // Append an asterisk to the displayPassword
String to mask the password
       lcd.setCursor(0, 1);        // Set the LCD cursor to column 0, row 1
(second row)
       lcd.print("Input: ");      // Display "Input: " on the LCD to indicate
the password being entered
      lcd.print(displayPassword); // Display the asterisks corresponding to the
entered password
       Serial.print("LCD: Input: "); // Print to Serial Monitor to confirm the
LCD message
```

```
      Serial.println(displayPassword); // Print the asterisks to Serial Monitor
for debugging
    }
  }
}

// ------------------ Door Control ------------------
void lockDoor() {                // Define the lockDoor() function to lock the
door and update indicators
  lockServo.write(0);           // Set the servo position to 0 degrees (locked
position)
  digitalWrite(RED_LED_PIN, HIGH); // Turn on the red LED to indicate the door
is locked
  digitalWrite(GREEN_LED_PIN, LOW); // Turn off the green LED
  digitalWrite(FAN_PIN, LOW);    // Turn off the fan
  Serial.println("Door Locked (Red LED ON, Green LED OFF, Fan OFF)"); // Print
to Serial Monitor to confirm the door state
}

void unlockDoor() {              // Define the unlockDoor() function to unlock
the door and update indicators
   lockServo.write(90);             // Set the servo position to 90 degrees
(unlocked position)
  digitalWrite(RED_LED_PIN, LOW); // Turn off the red LED
  digitalWrite(GREEN_LED_PIN, HIGH); // Turn on the green LED to indicate the
door is unlocked
   Serial.println("Door Unlocked (Green LED ON, Red LED OFF)"); // Print to
Serial Monitor to confirm the door state
}

// ------------------ DHT11 + Fan Logic ------------------
void monitorDHTandFan() {         // Define the monitorDHTandFan() function to
read DHT11 data and control the fan
  float temp = dht.readTemperature(); // Read the temperature (in Celsius) from
the DHT11 sensor
  float hum = dht.readHumidity();   // Read the humidity (in %) from the DHT11
sensor

  if (isnan(temp) || isnan(hum)) {  // Check if the temperature or humidity
readings are invalid (NaN = Not a Number)
    Serial.println("DHT Sensor Error"); // Print to Serial Monitor to indicate
a sensor error
    lcd.setCursor(0, 0);            // Set the LCD cursor to column 0, row 0
```

```
    lcd.print("Sensor Error   ");   // Display "Sensor Error" on the LCD
    return;                          // Exit the function to avoid further
processing with invalid data
  }

  if (temp > 28 || hum > 40) {       // Check if temperature exceeds 28°C or
humidity exceeds 40%
    digitalWrite(FAN_PIN, HIGH);    // Turn on the fan by setting the fan pin
to HIGH
    Serial.println("Fan ON (Temp > 28°C or Hum > 40%)"); // Print to Serial
Monitor to confirm fan state
  } else {                           // If temperature is ≤ 28°C and humidity
is ≤ 40%
    digitalWrite(FAN_PIN, LOW);     // Turn off the fan by setting the fan pin
to LOW
    Serial.println("Fan OFF (Temp <= 28°C and Hum <= 40%)"); // Print to Serial
Monitor to confirm fan state
  }

  Serial.print("Temp: ");            // Print "Temp: " to Serial Monitor to
display temperature
  Serial.print(temp);               // Print the temperature value to Serial
Monitor
  Serial.print(" °C | Hum: ");      // Print " °C | Hum: " to Serial Monitor
to separate temperature and humidity
  Serial.print(hum);                // Print the humidity value to Serial
Monitor
  Serial.println(" %");            // Print " %" to Serial Monitor to complete
the humidity display

  lcd.setCursor(0, 0);             // Set the LCD cursor to column 0, row 0
  lcd.print("T:");                 // Display "T:" on the LCD to indicate
temperature
  lcd.print(temp, 1);               // Display the temperature with 1 decimal
place (e.g., 25.0)
  lcd.print("C H:");                // Display "C H:" on the LCD to indicate
humidity (C for Celsius)
  lcd.print(hum, 0);                // Display the humidity with 0 decimal
places (e.g., 50)
  lcd.print("%  ");                // Display "%  " on the LCD to complete the
humidity display and add padding
```

```
  delay(1000);                      // Delay for 1 second (1000 milliseconds)
to refresh the readings every second
}
```