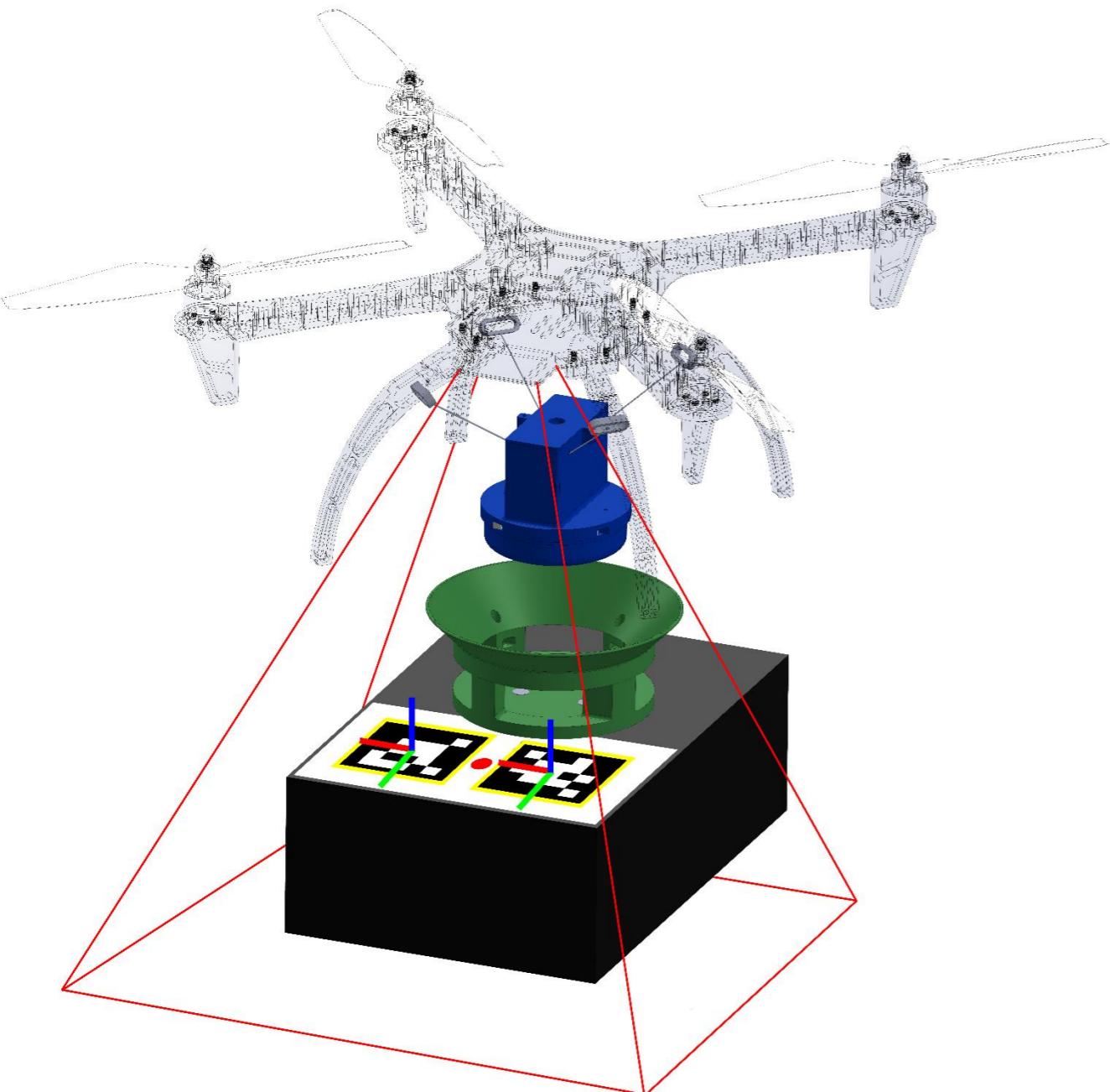


Design Report
FEEG6013 Group Design Project



05

Auto Drone Package System Design

Design of a modular, remotely operated electric package handling system for uncrewed aerial vehicle systems

Project Summary:

Advancements in drone technology and an increasing need for interconnectivity between medical centres have led to the rise of this niche logistic area in which drones could fulfil the stringent requirements of rapid and low-carbon deliveries to rural locations unserviceable by regular motor vehicles. Designed to allow a multirotor to fully handle a package autonomously, this system tackles one of the issues proposed by the NHS e-drone logistic trials in which the elimination of direct human interaction with the drone would improve efficiency while reducing the risk of personnel injury.

To enable a multirotor to remotely collect and deliver a package on its own, a sequential control algorithm utilising computer vision and fiducial markers for a reliable visual reference was developed to detect, track, and predict the location of a package and precisely control the approach of the drone to it via velocity vectors. This was paired with a mechanical interface which allowed the package to be picked up reliably despite deviation in the drone's angle of approach and centre-offset.

Both subsystems are also modular and are capable of easily being modified and fitted onto any size multirotors and/or packages with end-users only needing to learn how to correctly attach the markers and connect the hardware onto the package. Tested with computational finite element analysis, a virtual robotics environment platform, and a prop-less multirotor before being flown on several test flights at Boldrewood campus, the system achieved several of its initial software, avionics, and hardware design requirements in flight, with the rest only proven theoretically due to time constraints.

Group Members:

ID Number	Name	ID Number	Name
31311784	Jiun Tay	30102219	Sam Henry Jia Chuan Teo
31426077	Chen Yong Fong	31487041	Samantha Chung Shie Xuan
31426069	Prabooganesh Suria Kumaran	31426085	Muhammad Imran bin Mustaffa Kamal

Primary Supervisor: Professor James Scanlan

Co-Supervisor: Professor Tom Cherrett

Submitted on: 10/05/2023

Final Design Proposal Summary

Legend
Software
Avionics
Hardware

Flight Control Algorithm
Sequential flight control algorithm software uses computer vision pose estimation techniques of markers to calculate required velocity vectors to control drone's approach
Onboard Computer
Raspberry Pi 4B runs control algorithm script and output flight commands to flight controller unit
Gimballed Camera
Raspberry Pi Camera Module v2 with a 2-axis gimbal transmits stable footage of drone's nadir to computer for processing
Remote Viewing
VNC Viewer allows ground station to remotely program, launch, and view what the drone sees in flight
Fiducial Markers
Two ArUco markers allows computer to detect, track, and estimate package orientation with respect to the drone

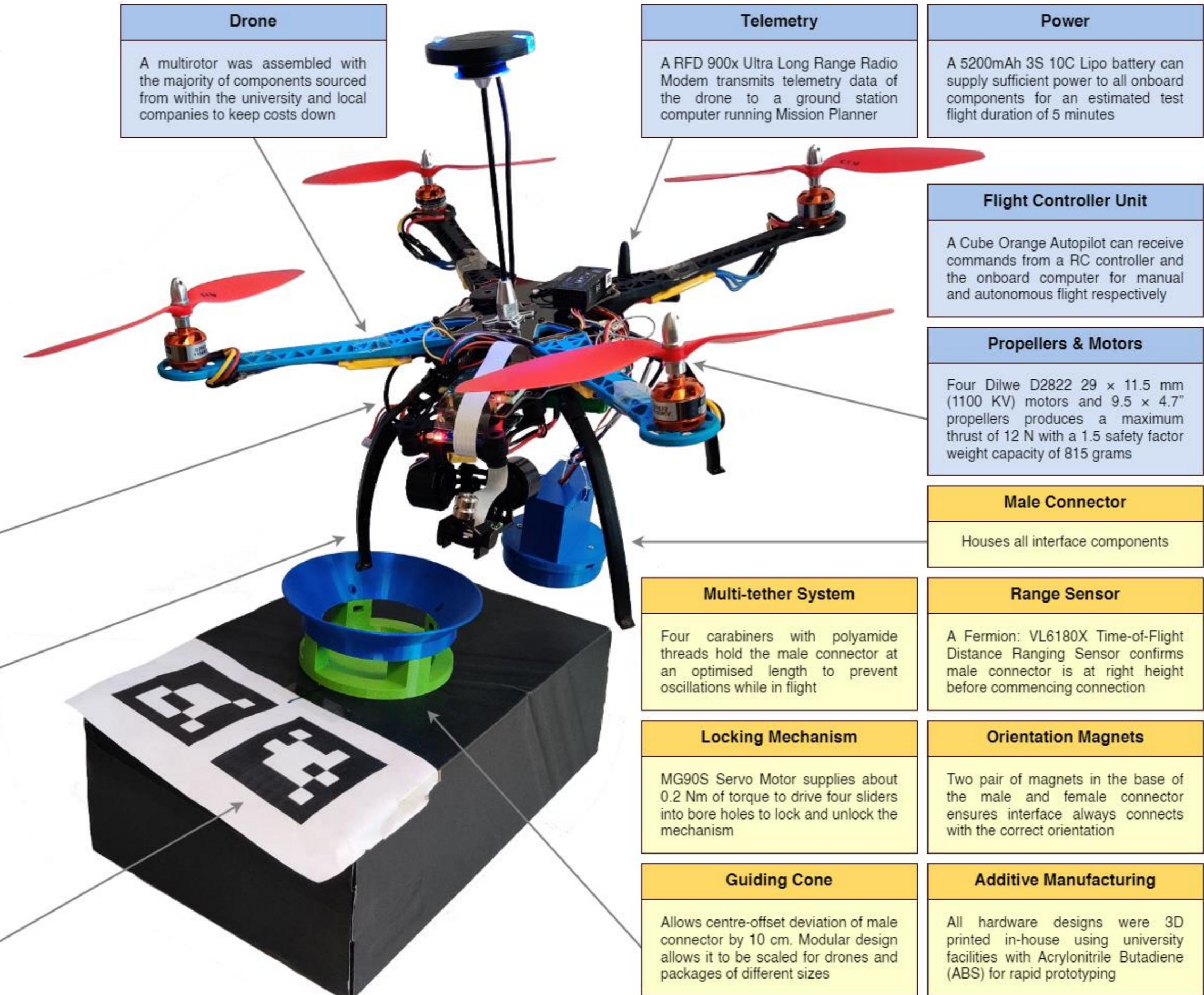


Table of Contents

Final Design Proposal Summary.....	1
Academic Integrity.....	2
Acknowledgements.....	2
1 Design Brief	2
2 Design Specification	4
3 Software.....	5
4 Avionics	13
5 Hardware	16
6 Research & Analysis.....	19
7 Final Design Proposal.....	26
8 Project Review	28
9 Conclusion.....	29
10 References	29
Appendix.....	30

Academic Integrity

We, members of the Modular Operational Remote Package Handling Electric UAV System (MORPHEUS) Group Design Project (GDP), confirm that we have read and understood the University of Southampton's guidelines for academic integrity, and have worked within their expectations in this report. We are aware that failure to act in accordance with these regulations may lead to penalties, and we consent to the University copying and distributing any or all work.

Acknowledgements

The team would like to thank a number of people which were vital for the completion of the project:

- Professor James Scanlan, for allowing us to take on this interesting self-proposed project and supporting us at every step of the way.
- Professor Tom Cherrett, for his expert advice on the current developments of drone logistics and support for the project.
- Dr David Toal, for providing valuable feedback and suggestions for the project.
- The team at SotonUAV, for supplying the multirotor frame as well as a substantial amount of spare avionic components that we used to build the drone with.
- Max Miles, for his aid in operating the propeller thrust measurement rig.
- Gina Celestine, Andrew Giles, and Dave Hills, for your assistance in ensuring that the necessary Health & Safety approvals for the flight tests were obtained.

1 Design Brief

1.1 Introduction

The advancement of Uncrewed Aerial Vehicle (UAV or Drone) technology in recent years has enabled an ever-increasing number of stakeholders to invest in the novel idea of transporting and delivering goods with them. It was found that the global drone market is now expected to grow at a Compound Annual Growth Rate (CAGR) of 49.0% with partially autonomous, hybrid drone platforms capable of short-duration (<30 mins), short-range (<25 km), and lightweight (<2 kg) deliveries projected to lead the market by 2030 [1].

It can also be seen that the slow implementation of rules and regulations, safety issues, a lack of landing zones within densely populated cities, along with the fragile nature of UAVs in poor weather conditions means that it is unlikely that wide-scale drone logistics will become commercially viable in the immediate future [2]. A reasonable assumption could therefore be made that the integration of drones into current logistic networks in the near future would mainly comprise of niche, specialised areas of application such as medical, military, or industrial deliveries.

Regardless of which sector should come to dominate the drone delivery market in the future, it is clear that the benefits brought from these UAVs such as rapid access to remote regions [3] and their lower environmental impact as compared to traditional gasoline-based vehicles [4] would see further advancements in the commercialisation of drone platforms and services. Furthermore, a review into existing multirotor systems which are available for purchase indicates a lack of development in the drone's capability to autonomously pick up and drop off a cargo.

Most multirotor drone platforms feature underslung payload configurations (several of which could be seen in Section 5.2) that though could automatically release its package(s) either via clips [5], nettings [6], or internal cargo holds with bay doors [7], they still require manual loading by a human operator. There were also published patents [8] related to autonomous article loading, but these have been found to be in the early stages of development and are unlikely to be adopted in scale due to its requirement of designing a completely new platform.

As such, this GDP focuses on the development of a mechanical interface named MORPHEUS which is capable of being attached to a drone and allow it to autonomously pick up packages. In particular, it seeks to address one of the issues currently faced by the e-drone NHS logistic trials [9]: developing automated methods by which drones can deliver and collect payloads without the need for direct human interaction. This would improve the efficiency of transporting patient samples from rural clinics to major hospitals or the delivery of crucial medical supplies vice versa while reducing the risk of injuries by removing the need of human operators.

The team designed various aspects of MORPHEUS' subsystem, which includes: a mechanically robust connector hardware, computer-vision-based flight control algorithm software, and avionics of the drone itself. These systems were then manufactured, tested, and verified using simulations and real-world flights. These are outlined in this report.

1.2 Project Aims & Objectives

As this project is a technological demonstrator of a novel concept, it only has a single aim of displaying that a multirotor could have the capability of autonomously locating a specifically modified package for the task, aligning itself with the package before picking it up, fly to a target destination with it in tow, and safely detaching it at the end to complete the mission. The team set several objectives to ensure that this aim would be successfully achieved:

- Design and build a modular attachment system that would serve as the interface between the drone and the package.
- Design and build a corresponding package counterpart to be picked up.
- Design and build a lightweight chassis to house all equipment required by MORPHEUS.
- Acquire and assemble a multirotor for MORPHEUS to be installed on.
- Install on-board vision and processing capabilities.
- Develop a control algorithm software based on computer vision.
- Develop ways to test said algorithm without the need of flying the drone.
- Integrate autonomous capabilities with the drone.

1.3 Project Assumptions

Along with the aforementioned project aims and objectives, the team also introduced several assumptions to simplify the project and ensure that the main aim could be accomplished on time before resources were directed to making the system more robust or economical:

- Project is a miniaturised technology demonstrator only. No considerations shall be made in terms of the financial and legal aspect of the end product as this is beyond the scope of the project.
- Prototype shall be a miniature UAV (small enough to be man-portable) in a multirotor configuration, with a maximum combined system and payload weight of 500 grams.
- The designed system shall only operate in a GPS-accessible location, in suitable weather conditions, with no obstructions between the drone and package.

1.4 Resources

An initial GDP budget allocation of £850 was given to the team by the University along with a \$500 drone frame and several avionic components by SotonUAV. The drone was assembled at the design workshop in the University Engineering faculty and flown in the indoor flight test lab. The team had access to the University's Engineering Design and Manufacturing Centre (EDMC), student workshop, electronics workshop, and Testing Structures Research Laboratory (TSRL). Furthermore, each student was given certain manufacturing hours and material usage amounts.

1.5 Team Structure

The team was arranged in a way to ensure that every member was able to work on an area they had the most expertise with while allowing proper synergy between respective departments. The project was broken down into five areas, those being: Management which handles the logistics and timely

completion of project deliverables; Hardware who focuses on the design and manufacturing of physical components; Software which creates and implements autonomous function of the drone; Avionics which handles the electronical aspect of said drone; as well as Research & Analysis which conducts relevant testing of the completed system. All team members work across several areas to ensure that all decisions were considered by multiple people with a breakdown of each member's role available to be seen in Figure 1.

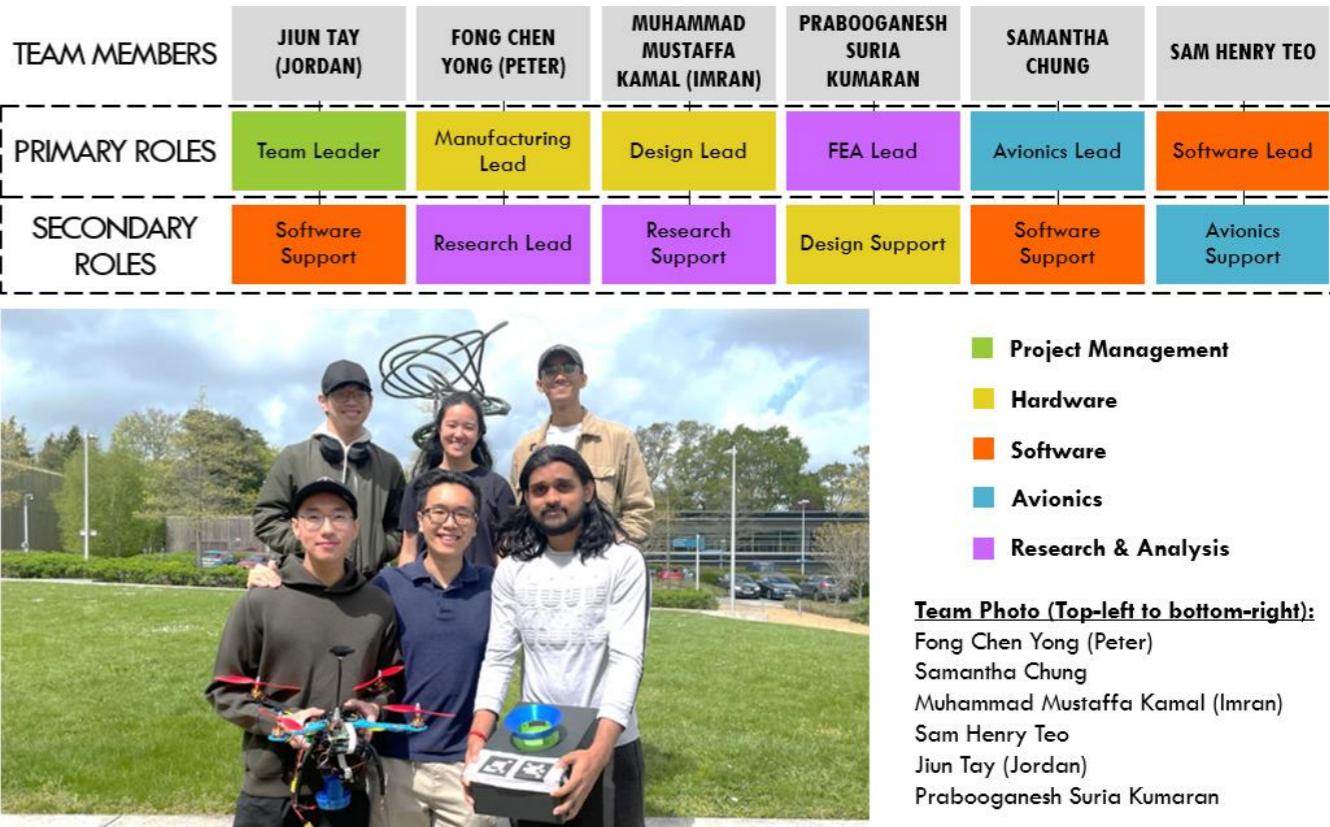


Figure 1: Team structure working on MORPHEUS alongside team photo

1.6 Design Process

The development of multiple sub-systems meant to be integrated together into a complex engineering system would be difficult to achieve without a clear and structured design process to manage it. A flowchart of this process could be seen in Figure 2 which covers all stages of the design, manufacturing, and testing process that the team followed to develop a computer vision control algorithm alongside a mechanical attachment interface which allowed a multirotor to autonomously pick up a package.

Firstly, design requirements were set based on the project's aims and objectives which acted as both the constraints to work within as well as the goals for the team to work towards. As this was a self-proposed GDP intending to serve as a technological demonstrator of a new product, initial research was then conducted to determine a range of potential solutions to tackle the problem. These were then discussed and narrowed down before influencing the initial design concepts. Different designs were analysed and then evaluated by the team before the most appropriate design was chosen to be further refined. If manufacturing or purchasing of a part was required, the budget would first have to be approved by the team before it could continue. If it was found that the time or finance needed to acquire

a part were too great, alternative methods of obtaining it were explored and if no viable option existed, another design concept would have to be used instead. The process of manufacturing the components includes creating the engineering drawings required for the University to use as the small size of all manufactured parts allowed them to be easily and quickly 3D printed. Once a design has been developed or manufactured, individual testing of it would then take place and its performance assessed.

This was done based on the sub-system's ability to achieve the design requirements set out at the start of the GDP. If it was unable to accomplish this, the process repeats with suggestions to improve its performance investigated. The requirements were also reduced if it was determined that the original aims and objectives could still be feasible with these limitations. Finally, the integration of all sub-systems onto the drone and the testing of the final assembly was the last stage of the design process. Compatibility issues were sorted by the re-iteration of individual components and once the drone was able to demonstrate its technical feat, further research and analysis on the system were conducted to identify potential areas of improvement for future work.

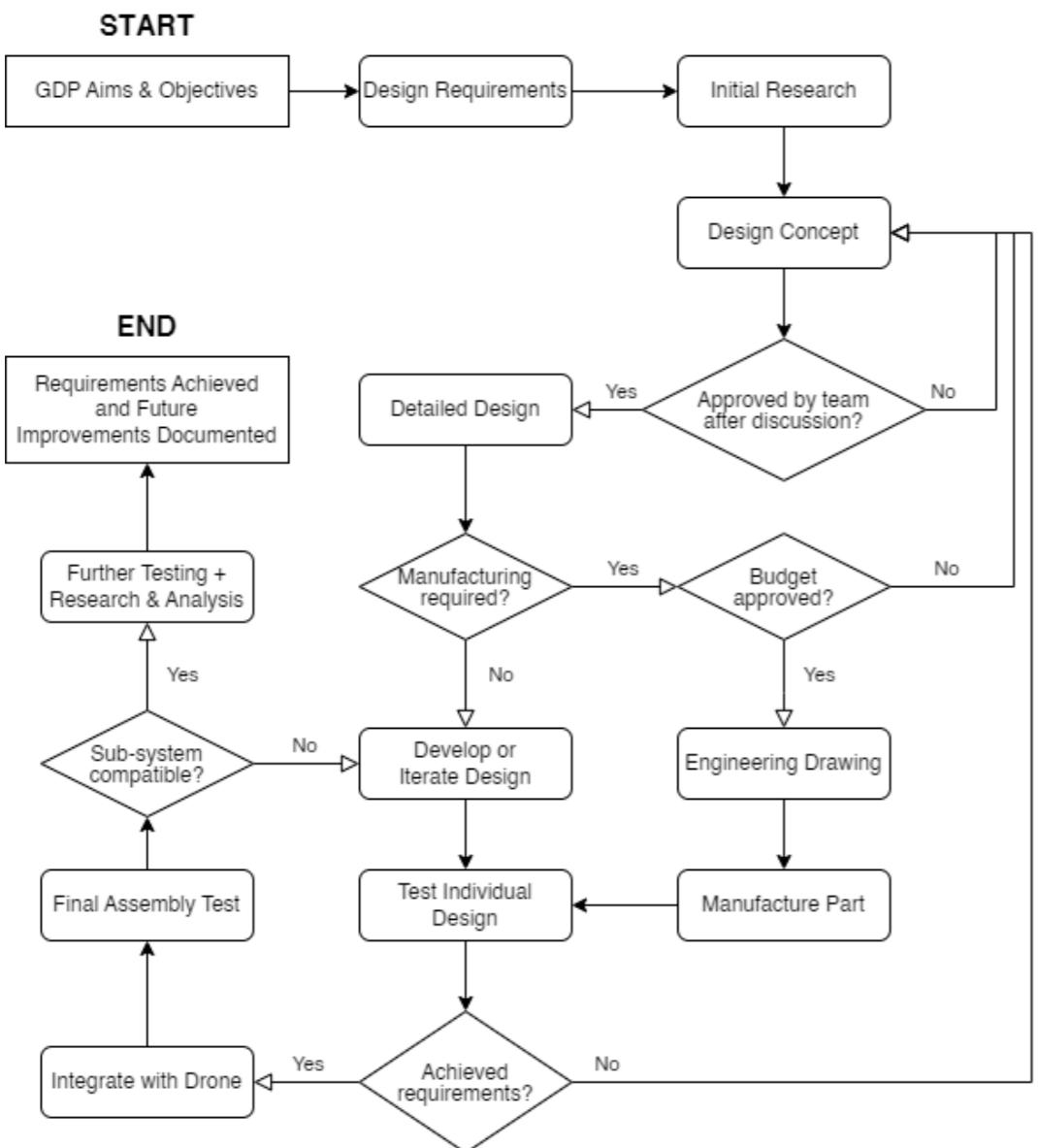


Figure 2: MORPHEUS design process flowchart

2 Design Specification

2.1 System Requirements

To achieve the project's aims and objectives that were set out in the previous section, the following requirements have been set for each subsystem as a target to influence the iterative design process. As the connecting interface is composed of two parts to allow a drone to attach itself onto a package, the component that will be installed on the drone shall henceforth be termed as the 'male connector' whereas its counterpart that is attached onto the package would be called the 'female connector' of the connecting interface.

2.1.1 Software Requirements

- Program shall be able to remotely fly a drone to a set location via GPS coordinates, visually detect the package at a test ceiling altitude of at most 3 metres, precisely control its descent and hover above a package before commanding the interface to collect the package.
- Main program shall have the necessary failsafes to exit the script and safely recall the drone to land at its home location whenever an unexpected error in the code is experienced.
- Control algorithm should have a processing speed of at least 50 Hz for real-time applications.
- Precision of control should allow the drone to hover above a package while having a positional accuracy of 10 cm and a rotational accuracy of 20 degrees. These values were chosen as the female connector will be limited by the size of the package and the orientation of the package is important to prevent instabilities from arising while the drone-package system is in flight.

2.1.2 Avionics Requirements

- Power subsystem shall be able to provide sufficient electric power to all components during the test flights while posing little to no risk of electrical shock or fire.
- Drone shall be able to be flown manually with a remote controller and remotely via a script.
- Flight Controller Unit (FCU) shall be able to wirelessly receive flight instructions from a ground station computer while sending telemetry to it.
- Avionics components shall be easily maintainable such that the replacement of any component can be done so quickly and with basic tools.

2.1.3 Hardware Requirements

- 500 grams mass limit for connecting interface and underbelly chassis (excluding package, drone, and other subsystem components) from off-the-shelf multirotor capability.
- Size of the female connector shall not exceed 15 x 15 cm to allow it to fit on top of a standard small parcel while allowing room for visual fiducial markers to be placed for control purposes.
- Male connector shall be capable of extending below the underbelly of the drone as well as securely attach itself to the female connector while the drone is hovering within the tolerances set out by the software subsystem requirements.
- Chassis shall be mountable on the multirotor and be able to house all components needed for the operation of MORPHEUS such as the connecting interface, camera gimbal, and battery pack.

3 Software

3.1 Computer Vision Literature Review

One major aspect of enabling a drone to autonomously pick up a package on its own is its capability in sensing inputs from the environment, processing said data, and relaying the necessary outputs required to actuators based on the information it received. For the purpose of this project, it was decided that a Computer Vision (CV) control algorithm would be developed to achieve this task. CV methods acquire, process, analyse, and understand digital images to produce numerical or symbolic information which are usually in the form of decisions [10]. The interdisciplinary scientific field of CV is vast, and it has been implemented in many applications such as medical diagnosis and inspection of manufactured goods. These variety in tasks utilises different types of CV like image segmentation, facial recognition, edge detection, pattern detection, image classification, and feature matching [11]. However, only the object detection and tracking techniques are prevalent in terms of drone vision.

This allows a drone to find objects and keep track of their trajectories in a video sequence which is useful for navigation and obstacle avoidance. CV can also be used to automate a drone's Precision Landing (PL) approach which is of particular interest. There have been several literatures that have investigated the usage of CV and fiducial markers to achieve an autonomous drone landing [12, 13, 14], but these were mainly focused on the development of more versatile docking platforms. Currently, there is little to no formal research on implementing CV for the purpose of detecting and picking up an object. Furthermore, there were limited PL experiments that were conducted with real-world drones and no instances of introducing a hybrid CV approach to extend the possible range of control which is novel to this project.

3.2 Framework

Two object detection and tracking approaches were developed for the control algorithm, one based on shape detection and the other on the usage of fiducial markers. The Python Open Source Computer Vision Library (OpenCV) was used for this task as it has programming functions mainly aimed at real-time CV and it has extensive online documentation. A hybrid approach was taken as it was found that the detection range of fiducial markers was shorter than that of a simple shape. As the marker would be placed above a package, this would limit the size of the marker to a maximum length of the package in the best possible scenario. To allow the drone to detect the parcel from a higher altitude, it was decided that the program would first utilise shape detection and centroid estimation to bring the drone down to a low enough height for the fiducial markers to be acquired. The software will then use two fiducial markers and pose estimation to perform a more precise approach and leave it right above the package before passing control over to the mechanical interface. However, it should be noted here that due to time constraints and a lack of available testing locations, the final design proposal of MORPHEUS will only include the fiducial marker technique as the prototype drone will not be flying above a height of 3 metres which is low enough for the markers to be detected without the assistance of a hybrid CV system. A flowchart of the PL control algorithm could be seen in Figure 3 and the two constituents making up the hybrid CV is further discussed in the following subsections.

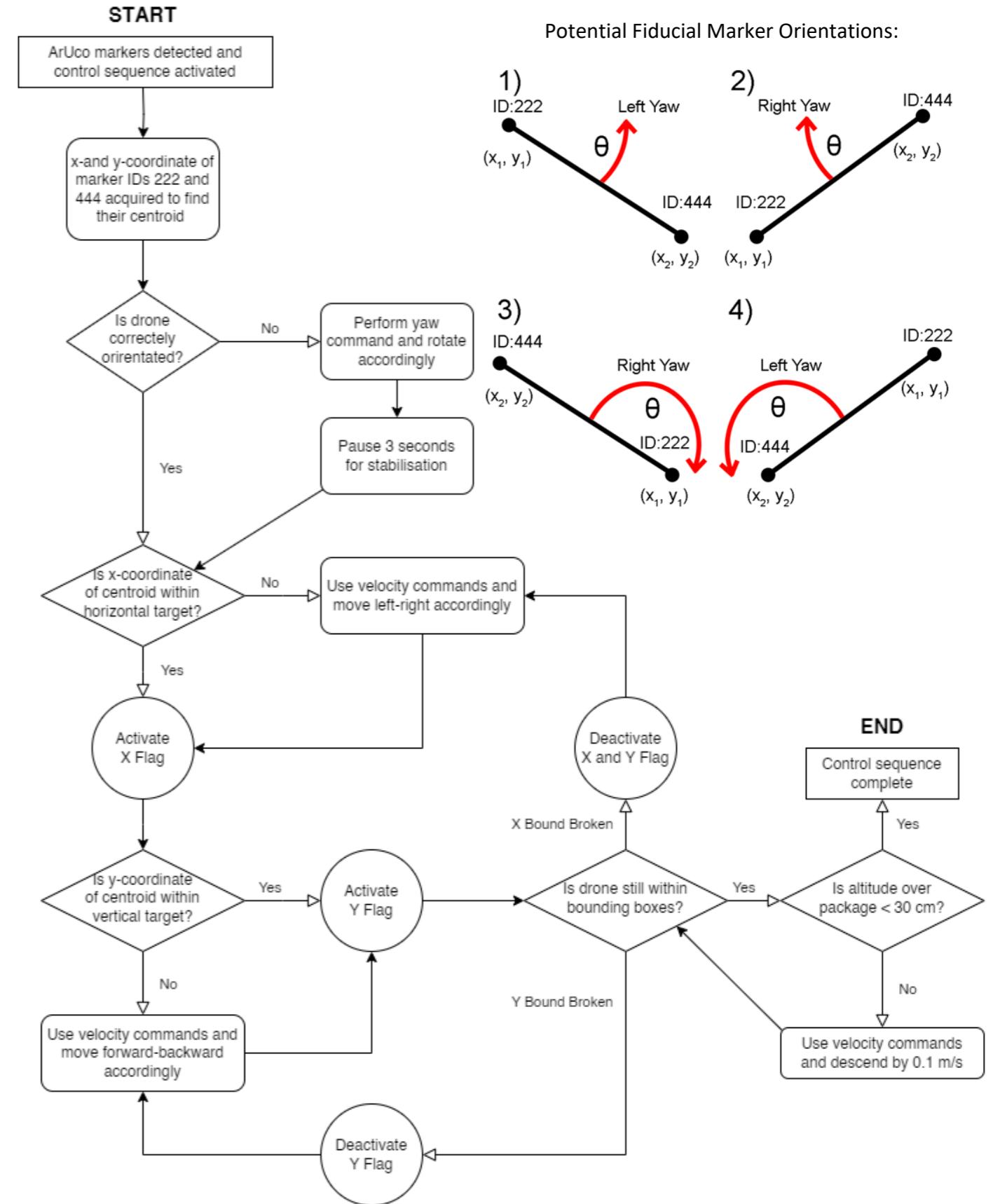


Figure 3: Precision approach control algorithm flowchart with all potential orientations of fiducial markers

3.3 Geometrical Shape Detection

The use of geometrical shapes was first proposed as a mechanism for the drone to locate the package. The goal of this process is to simply locate the package by analysing its shape, which is often rectangular. It is assumed that the package will be placed in an isolated area such as an empty field and as such there wouldn't be any other objects in the vicinity that would interfere with the detection process. Once the package is located, the drone will descend to an altitude where the fiducial markers can be detected by the CV and hence the control algorithm can be initialised. If the detected shape is not a rectangle, the drone will not descend and simply perform another parameter sweep until the package is located or the drone's battery is depleted. While a parcel is typically rectangular or a square, the shape detection algorithm can always be tweaked according to a specific package's shape. Figure 4 shows an example scenario where shape detection can be used to locate a package by its shape.

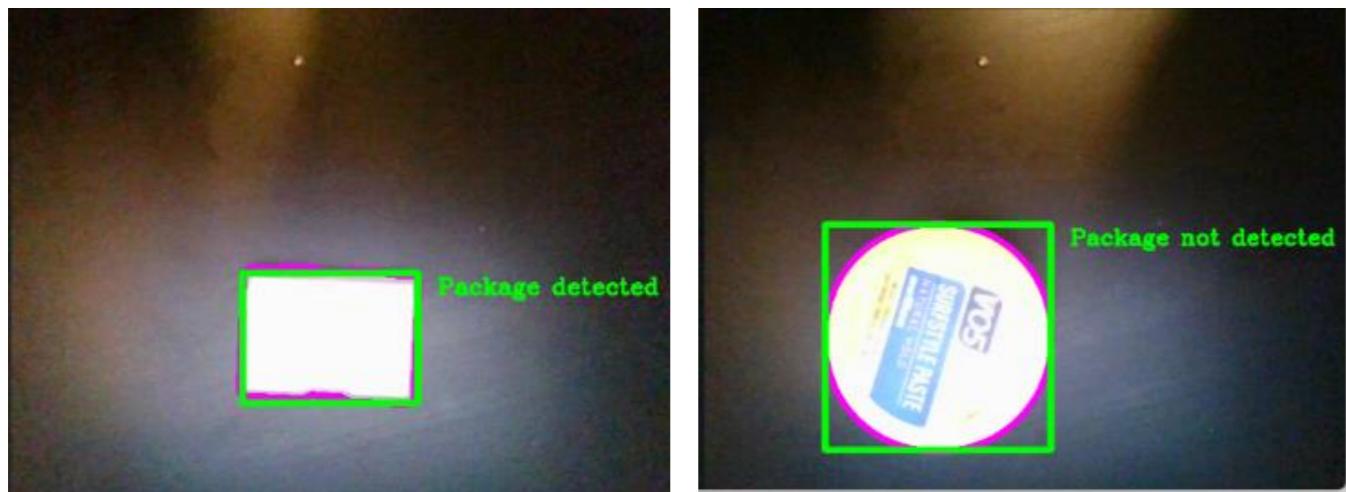


Figure 4: Utilisation of shape detection to identify and locate the targeted package

Due to time constraints and complexities in creating a hybrid system with the utilisation of shape detection and fiducial markers, it was decided that only the fiducial markers would be implemented. This would mean that the drone will have to hover at a lower altitude of less than 3 metres to detect the fiducial markers which now not only act as a mechanism for identifying the package but also as a reference for the control algorithm. However, the integration of shape detection in locating the package along with the fiducial markers for the control algorithm would surely make the software more robust and therefore makes a compelling discussion for future works.

3.4 Fiducial Markers Detection

The specific class of fiducial markers that were used are known as ArUco markers – synthetic square markers composed of a black border and an inner binary matrix which determines its identifier (ID) number [15]. They are an inbuilt feature of OpenCV which derived this method of a visual fiducial system from the AprilTag

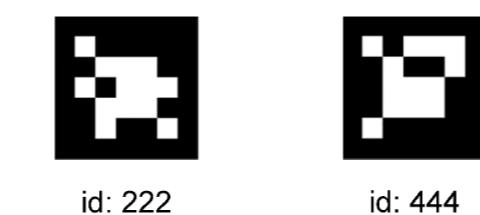


Figure 5: ArUco markers used in MORPHEUS

Project, a conceptually similar two-dimensional bar code to that of a QR code but designed to encode fewer data payloads (between 4 and 12 bits) which allows them to be detected more robustly and from

longer ranges [16]. The 10-bit family of ArUco markers were selected which allowed IDs of 0 to 999. This was chosen as a good middle ground as denser markers with more bits would be harder to track from a distance whereas coarser markers result in a higher number of false positive detections.

A minimum of two markers were placed on the package to provide information for the control algorithm. Arbitrary IDs of 222 and 444 were chosen which could be seen in Figure 5. Apart from acting as a visual fiducial system, these numbers can also be used to identify the package itself with a combination of any two identifiers allowing 498,501 unique IDs. For logistic purposes, a smart system would also be able to look up and match this ID to an existing database to verify necessary details such as whether the drone is picking up the right package and where the address of delivery is. Increasing the bit-size or including additional markers on the package would also allow for more possible combinations if required. The detection rate of these markers is highly dependent on the size of the marker itself, and though not explored in this GPD as the drone has an altitude ceiling of three metres, future work could be done on selecting a different fiducial system such as AprilTag 3 to allow for greater acquisition range and thus a higher approach altitude.

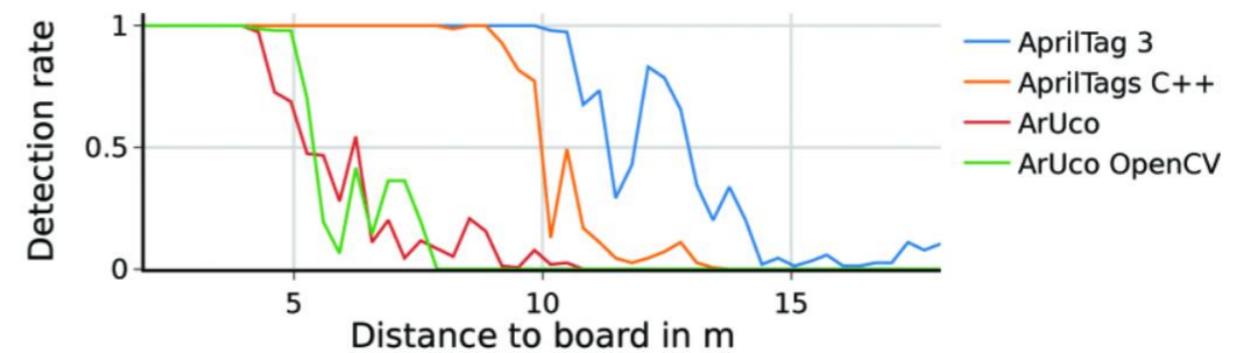


Figure 6: Detection rate of different fiducial markers at different ranges [17]

3.5 Camera Calibration and Pose Estimation Calculation

Apart from allowing the computer to detect and track a package, the usage of ArUco markers also enables the use of pose estimation which is a CV technique that predicts and tracks the location of an object by looking at a combination of the orientation and pose of said object. This is first done by calibrating the camera in use with a chessboard pattern to estimate the camera parameters and allow the determination of a 3D point in the real world to its corresponding 2D projection on the image. Cameras have internal (focal length, fish-eye lens distortion, optical centre) and external (orientation of camera with respect to some world coordinate system) parameters which distorts an image that can be accounted for with the following equations for radial and tangential distortion [18]:

$$x_r = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \quad (1)$$

$$y_r = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \quad (2)$$

$$x_t = x + [2p_1 xy + p_2(r^2 + 2x^2)] \quad (3)$$

$$y_t = y + [p_1(r^2 + 2y^2) + 2p_2xy] \quad (4)$$

where the five unknown parameters are known as the distortion coefficients:

$$\text{Distortion coefficients} = (k_1 \ k_2 \ p_1 \ p_2 \ k_3)$$

and these could be solved for by first finding the focal length (f_x, f_y) and optical centres (c_x, c_y) by providing several sample images of the chessboard taken by the camera and using specific points such as square corners in the chessboard where both their real world and image coordinates are known to create a camera matrix:

$$\text{Camera matrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

which is then used to determine the distortion coefficients with OpenCV's inbuilt function [18]. At least 10 different test patterns of varying orientation should be used to get a better result. Once these two parameters are found, the undistorted x- and y-coordinate of a marker's centroid alongside its distance with respect to the centre of the camera can be computed as seen in Figure 7. These values were then fed into the main algorithm to control the drone.

3.6 Control Algorithm

To accomplish the goal of an autonomous pick-up system, it was crucial to recognise that the drone will have to be positioned in a way that allows the interface to load the package securely. This means that a robust control algorithm would have to be thought of to realise this ambition. Therefore, three main control phases were identified:

- Orientation
- Alignment
- Height

These three control processes are the main building blocks of the control algorithm. The orientation and alignment control processes dictate the position of the interface relative to the centre of the package. If any of the 2 control processes are not in sync, the male connector of the interface would not be able to attach to its female counterpart due to mispositioning. This is especially crucial for the final phase of the control algorithm, which is the height control process. This process simply descends the drone to a specific height which allows the male connector to attach itself physically to the female connector, completing the autonomous pick-up algorithm.

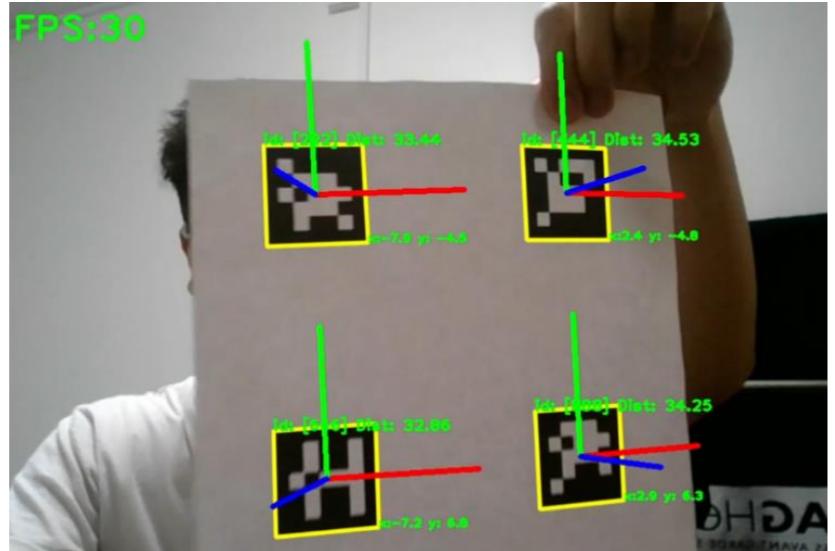


Figure 7: Initial pose estimation of ArUco markers

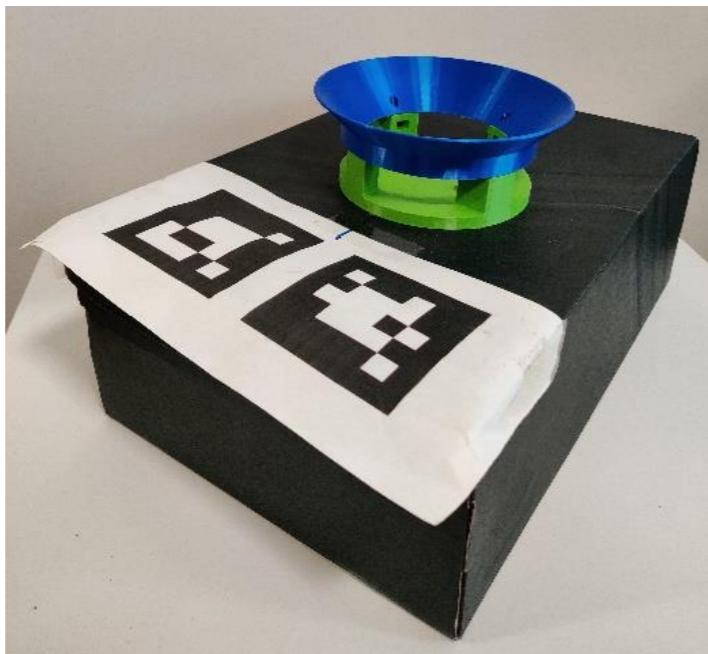


Figure 8: Markers on test package

3.6.1 Code Initialisation

As described in Section 3.4, two ArUco markers, ID 222 and 444 were used in this project. The markers were attached on the top surface of the package. The placement of the IDs allows the CV to detect them easily. Figure 8 shows the marker IDs on the package used during the flight tests conducted in Boldrewood. The control algorithm begins once the markers are detected. The CV first acquires the presence of both IDs before proceeding with the orientation control process.

3.6.2 Pose Estimation Application

The detection of the markers enables the CV to obtain the camera pose. A 3D transformation from the marker coordinate system to the camera coordinate system creates the camera position with respect to the marker, which is defined by the rotation and translation vectors (rvec and tvec respectively) [15]. By calling the parameter tvec, the pose estimation in all three-coordinate system, x, y, and z of any individual marker can be obtained. Figure 9 shows the marker coordinate system with the z-axis pointing outwards while Table 1 briefly describes the output parameter tvec and their respective coordinate system extracted. By taking advantage of tvec, it can be used as reference for the drone to position itself relative to the centre of the parcel. It also should be noted that rvec is not used in the control algorithm.

Table 1: Output parameter "tvec" and its respective coordinate system

Output Parameters of Tvec	Description
tvec[i][0][0]	Extracts x-coordinate of a marker ID
tvec[i][0][1]	Extracts y-coordinate of a marker ID
tvec[i][0][2]	Extracts z-coordinate of a marker ID

The approach taken in this project to solve the orientation, alignment and height control processes revolves around the coordinate system extracted from the pose estimation of each individual marker. Through the pose estimation, a simple and efficient framework for the three main control phases was thought of, which serves as the foundation of the control algorithm.

3.6.3 Orientation

Iteration #1

The control algorithm begins with the orientation process, which is crucial as it also determines the orientation of the male interface. The interface is meant to connect in a certain orientation to ensure that the drone-package system is stable while in flight. If the drone is not orientated correctly, it would mean that the interface would not be able to load the parcel due to the misorientation between the male and female connector. Therefore, the goal of this process is to ensure that the male connector of the interface is orientated in the same orientation as the female connector. The female connector is fixed at the centre of the package as seen in Figure 8. Figure 10 compares an example case of misorientation between the drone and the package with its desired orientation.

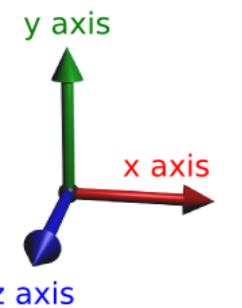


Figure 9: XYZ Marker coordinate system

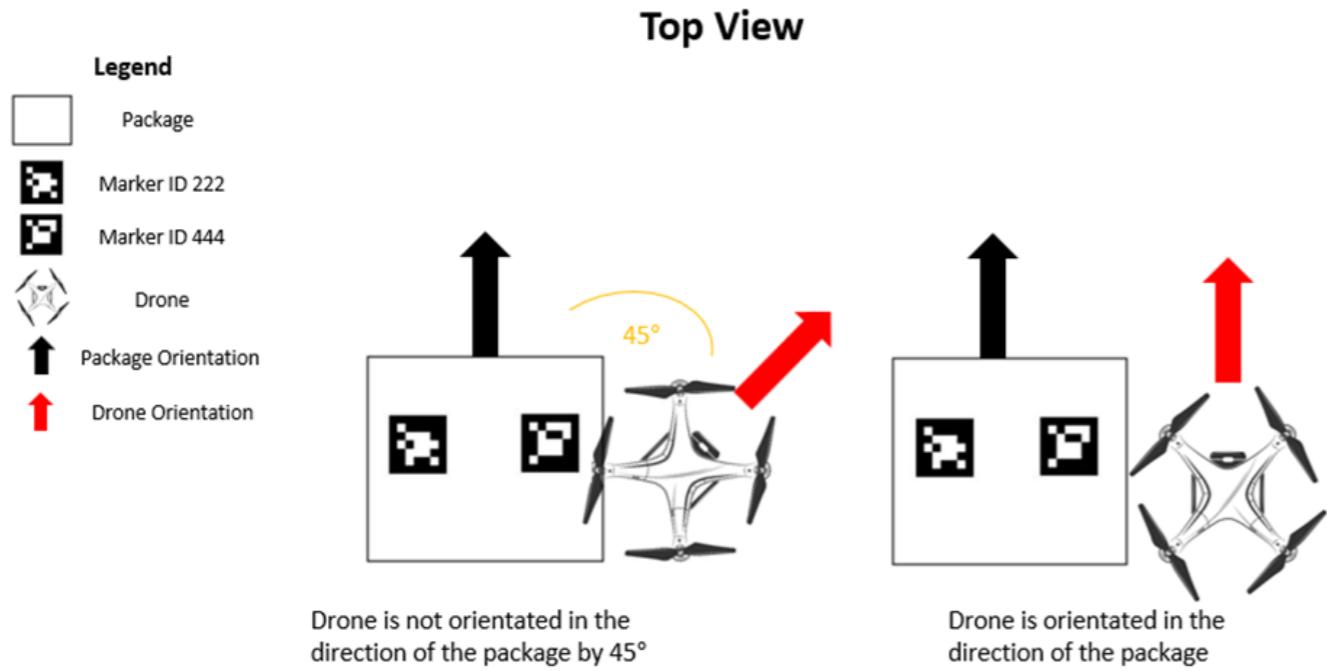


Figure 10: An example case of misorientation between the drone and the package

This control process involves both marker IDs 222 and 444. Using the conditional loop, the y-coordinates of both IDs were extracted and then compared to one another. The code block in Figure 11 illustrates the working principle of this approach, where the y-coordinates of IDs 222 and 444 were compared during this process. Note that `ref_222_y` and `ref_444_y` are variables that extracts the y-coordinate of ID 222 and ID 444 respectively through `tvec[i][0][1]`. It should also be noted here that the phrases “print(“Stop Ori”)” and “print(“Rotate”)” are placeholders for the DroneKit commands.

It can be observed that the magnitudes of the y-coordinates varied with one another whenever they are misoriented, as shown in Figure 12. Therefore, one solution to resolve the orientation issue is to simply equate the values of the y-coordinates of both IDs. The desired orientation can be obtained when the y-coordinates of both IDs are equal. To account for external forces such as wind and drone hovering stability due to pitching and yawing, a threshold of ± 0.2 cm was allocated. This value was chosen as it provided a good middle ground between accuracy and speed to achieve the target during simulations. The flowchart in Figure 13 briefly illustrates the working mechanism of the orientation control process.

```
# Threshold to account for external forces; wind, etc
threshold_ori = 0.2
if ref_444_y <= ref_222_y + threshold_ori and ref_444_y >= ref_222_y - threshold_ori:
    print("Stop Ori")
    ori_done_flag = True
else:
    print("Rotate")
```

Figure 11: Snippet of code showing the approach and conditions set for the orientation control process of iteration #1

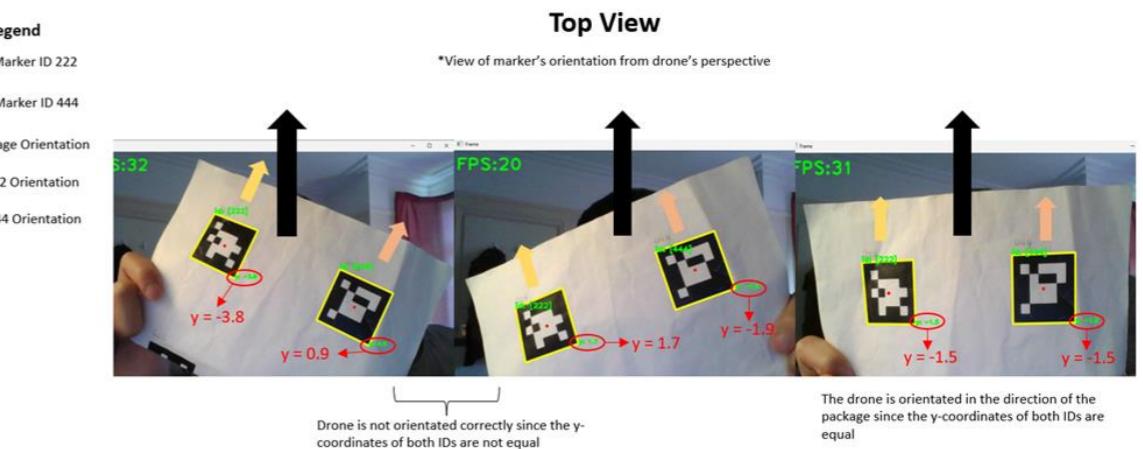


Figure 12: Working principle of the orientation control process

Iteration #2

Though this solution does resolve the orientation issue, executing it during every code loop slows the algorithm down slightly. Therefore, this section introduces a more elegant solution for this. From Figure 3, there are four possible combinations of the two markers' relative position to each other given any rotation angles of the drone. In each case, the initialisation of the code upon detection of both markers captures and stores the initial coordinates of each marker as x_1, y_1 and x_2, y_2 respectively. Using simple logical arguments relating the four parameters and their values in the marker coordinate system to each other, the program can be directed to the correct scenario to calculate the yaw angle, θ required to align the rotational orientation of the drone with the package and send a single command at the start to yaw the drone into this position. As the drone's loiter stability is good at maintaining its yaw in flight, this simplification frees up computation power for more important positional control.

An example calculation of the first scenario can be seen in Figure 14. If the y-coordinate of marker reference 222 is larger than 444 and the x-coordinate of 444 is larger than 222, this particular section of the code will run to compute the value of the rotation variable to yaw the drone with that angle. This is done using trigonometry with the opposite length of the triangle found with $y_1 - y_2$ and the adjacent length with $x_2 - x_1$ before taking their arctangent. As the NumPy library outputs the angle in radians, the last line of the code snippet converts it to degrees for easier debugging and visualisation.

```
if ref_222_y > ref_444_y:
    if ref_444_x > ref_222_x:
        rotation = (np.arctan((ref_222_y - ref_444_y) / (ref_444_x - ref_222_x))
                    * (180 / np.pi))
```

Figure 14: Snippet of code for an example calculation

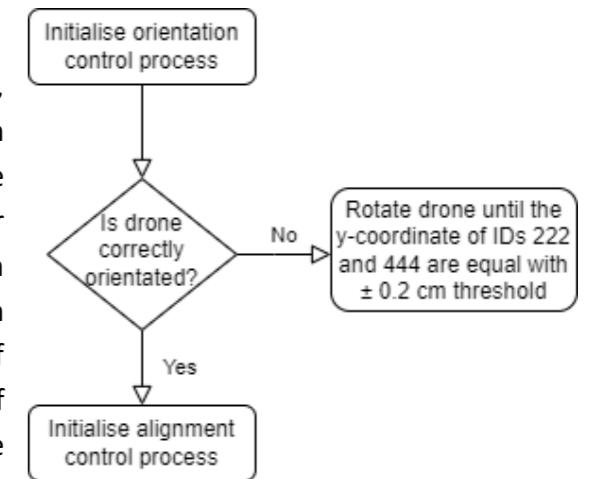


Figure 13: Brief flowchart of the orientation control process for iteration #1

3.6.4 Alignment

One aspect that the orientation control process does not solve is the positioning of the drone directly above the centre of the package. Figure 15 illustrates the various misalignment scenarios between the drone and the package. Note that the centre of the package can also be used to illustrate the position of female connector of the interface. Failure to ensure that both the drone and package is aligned would mean that the male connector of the interface cannot physically attach itself to its female counterpart. In real world scenarios, the drone is often misaligned in both axes. Therefore, the alignment control process aims to fix this issue by aligning the drone to the centre of the package as shown in Figure 16.

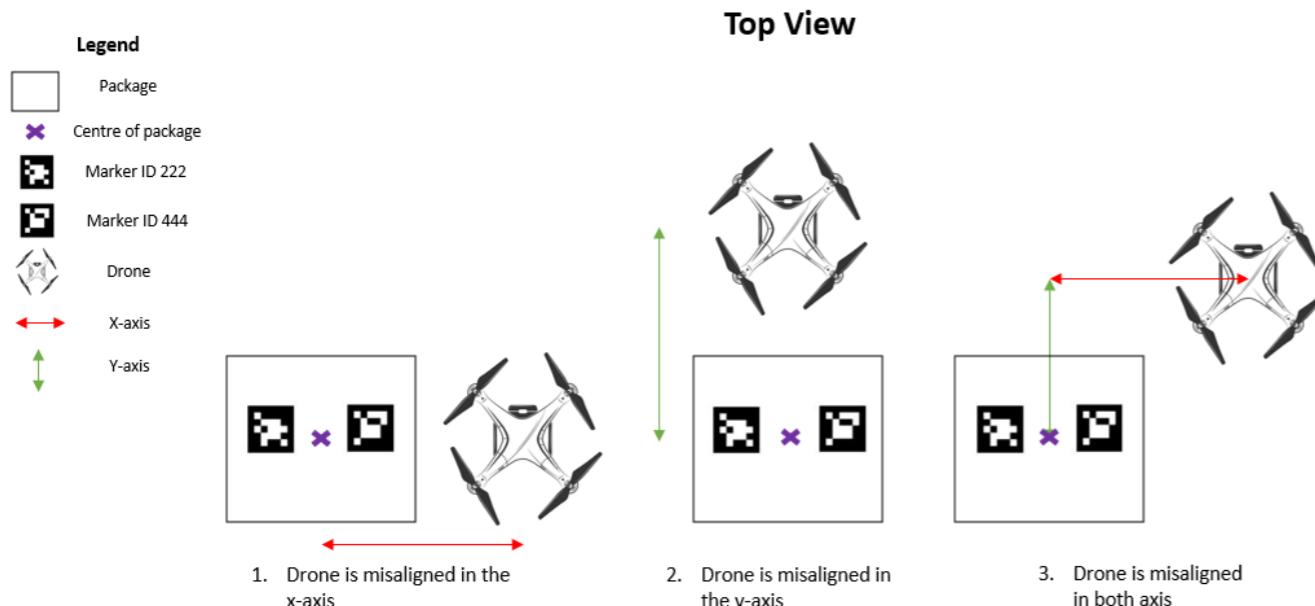


Figure 15: Misalignment scenarios relative to the centre of the package

Both markers were used as a reference to align the drone with the package. The alignment control is split into 2 sub-processes: alignment in the x-axis followed by alignment in the y-axis. By extracting both x- and y-coordinates of the markers through the tvec, the coordinates were used as references to align the drone to its desired position. In this case, the drone should be aligned to the centre of the package to enable the interface to attach itself onto the package. The code block in Figure 17 demonstrates the approach used in the alignment process. Note that avgx and avgy are variables that were computed from the x- and y-coordinates of ID 222 and 444 to generate a virtual centroid that sits in-between both markers.

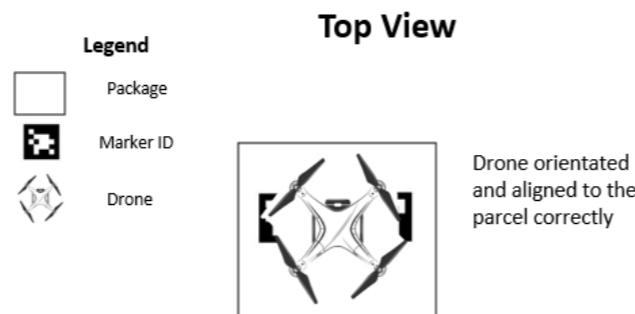


Figure 16: Drone aligned to the centre of the package

```
# Alignment Control Process
if ori_done_flag:
    # Alignment in the x-axis
    if x_done_flag == False and alignment_done == False:
        if avgx > 0.5:
            cv2.putText(frame, f"Drone Moving Right", (7, 70), cv2.FONT_HERSHEY_PLAIN,
                       2, (0, 255, 0), 2, cv2.LINE_AA)
            send_ned_velocity(0, 0.5, 0, 0.02)
        elif avgx < -0.5:
            cv2.putText(frame, f"Drone Moving Left", (7, 70), cv2.FONT_HERSHEY_PLAIN,
                       2, (0, 255, 0), 2, cv2.LINE_AA)
            send_ned_velocity(0, -0.5, 0, 0.02)
        else:
            x_done_flag = True
```

Figure 17: Code showing the approach taken and conditions set for the alignment process in the x-axis

The drone is considered to be aligned with the centre of the package when the values of avgx and avgy are 0. This was possible by considering the relative position of the markers with respect to the female connector on the package as well as the drone camera's distance from the male connector. By ensuring that these two distances were the same, targeting avgx and avgy values of 0 (i.e., the camera is right above the centroid of both markers) will also mean that the male connector is aligned with the female connector while the drone is descending. Similar with the orientation control process, a threshold (± 0.5 cm) was also allocated to account for external forces. If the drone is out of range of these set values, it is implied that the drone is misaligned and therefore a feedback loop will be sent to the drone through the send_ned_velocity function, commanding it to either move right or left at a speed of 0.5 m/s until the correct range of values are met. The same procedures were then repeated for the alignment in the y-axis where avgy was used instead to obtain the desired alignment in the y-axis. Similarly, a threshold of ± 0.5 cm in the y-direction was allocated.

3.6.5 Height

The height control is the final phase of the algorithm, which commands the drone to descend before loading the package via the interface. This process incorporates the autonomous pick-up mechanism, as illustrated in Figure 19. However, this process requires the drone to descend to the right altitude to physically connect the male connector to the female connector. The code snippet shown in Figure 18 shows the approach taken in this process.

```
# Height Control Process
if tvec[i][0][2] >= 35:
    cv2.putText(frame, f"Drone Descending", (7, 100), cv2.FONT_HERSHEY_PLAIN,
               2, (0, 255, 0), 2, cv2.LINE_AA)
    send_ned_velocity(0, 0, 0.5, 0.02)
else:
    everything_done = True
```

Figure 18: Snippet of code showing the approach taken and conditions set for the height control process

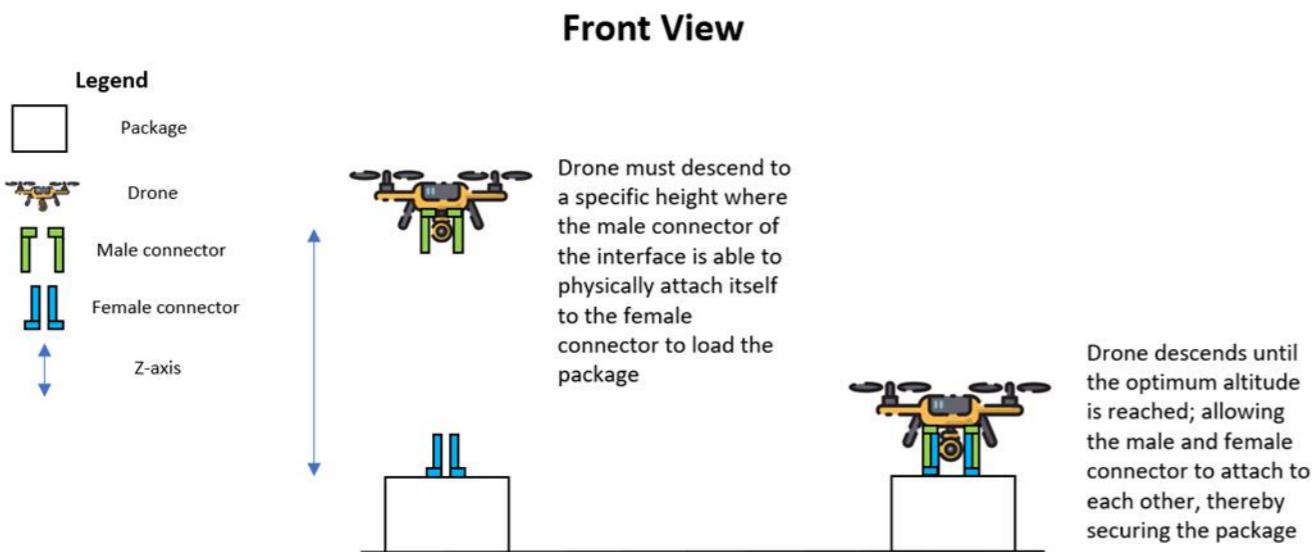


Figure 19: Working principle of the height control process

In this occasion, the z-coordinate of the tvec from ID 444 was utilised. Based on the length of the multi-tether system which was used to support the male connector (introduced in Section 6.2), an optimum height was set at 35 cm above the top surface of the package through a conditional loop, as this is the altitude where the male connector of the interface will be able to connect to its female counterpart on the package. Through the if-else statement, the drone descends to an altitude of 35 cm with the help of the send_ned_velocity function. The drone then stops descending once this altitude is reached and begins hovering to allow the interface to attach itself to the package. The flowchart in Figure 20 briefly illustrates the working principle of the height control process.

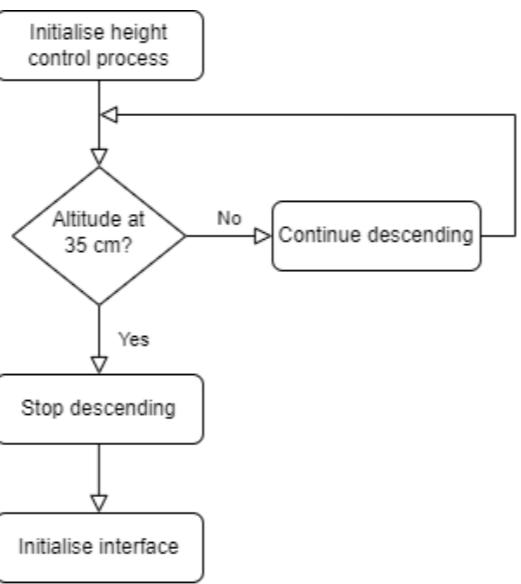


Figure 20: Brief flowchart of the height control process

3.6.6 Flagging

As described initially, the control algorithm is designed in a sequential manner. The drone first orientates in the direction of the package followed by alignment to the centre of the package and finally, the drone descends to pick-up the parcel. At the same time, the drone will have to obey strict conditions that were set in the orientation and alignment process so that the interface is positioned accurately to secure the package. However, an issue that while loops often encounter is that they only have one conditional. Therefore, the use of flags in complex code such as the control algorithm is encouraged to check conditionals throughout the loop [19]. The idea is that a variable logs the occurrence of an event. For instance, it is often set as “True” if the event happened and set “False” if the event did not happen [20]. The use of flags ensure that each control sequence is executed accordingly in a practical and efficient

manner. The code blocks in Figures 17 and 21 illustrate example use cases of flags in the alignment control process while Table 2 describes the variables and the events that they represent. The same concept was applied for the orientation and height control processes.

```

# Alignment in the y-axis
if x_done_flag == True and alignment_done == False:
    if avgx > 0.5 or avgx < -0.5:
        x_done_flag = False
        alignment_done = False
    else:
        pass
if avgy > 0.5:
    cv2.putText(frame, f"Drone Moving Forward", (7, 70), cv2.FONT_HERSHEY_PLAIN,
               2, (0, 255, 0), 2, cv2.LINE_AA)
    send_ned_velocity(0.5, 0, 0, 0.02)
elif avgy < -0.5:
    cv2.putText(frame, f"Drone Moving Backward", (7, 70),
               cv2.FONT_HERSHEY_PLAIN,
               2, (0, 255, 0), 2, cv2.LINE_AA)
    send_ned_velocity(-0.5, 0, 0, 0.02)
else:
    y_done_flag = True
    alignment_done = True
  
```

Figure 21: Snippet of code showing the usage of flags in the alignment control process

Table 2: Flag variables used in the alignment control process

Variable	Event
x_done_flag	Is alignment in the x-axis done?
y_done_flag	Is alignment in the y-axis done?
alignment_done	Is the alignment control process done?

From Figure 17, it can be observed that x_done_flag and alignment_done were initially set at “False”. This indicates that the alignment in the x-axis has yet to happen and as a result, the alignment control process is not done yet. Hence, the drone will execute the alignment process in the x-axis according to the conditions given. Once the conditions are met, x_done_flag is set to “True” as the drone has completed its alignment in the x-direction. However, alignment_done is not set to “True” yet as the drone still needs to complete the alignment in the y-axis which is described by the code in Figure 21. Before the y-axis alignment is executed, the code goes through the alignment in the x-axis once again to verify its status as described in the first 4 lines of code. If the drone happens to be misaligned in the moment, the alignment process in the x-axis is repeated. Otherwise, the code continues with alignment in the y-axis. Finally, y_done_flag and alignment_done are toggled to “True” as the alignment in both axes is completed. This indicates to the code that the alignment control process is done, and the height control process is ready to be executed.

Another advantage of using flags in the code is the ability to reiterate through all the conditions used in the control algorithm, ensuring reliability in its positioning. Due to constant shift in CV frame perspective during the descend phase, the coordinates will also change as the CV slowly approaches the package. Therefore, there was a need to recheck that all conditions set in the alignment controls are still met. Furthermore, external factors such as wind environment may affect the drone's alignment during descend phase. The code in Figure 22 shows how flags were utilised as a reiteration mechanism during the descend process. At every instant when the drone descends, the alignment of the drone in both axes is reiterated, realigning the drone according to the conditions set when necessary.

```

if y_done_flag:
    if avgx > 0.5 or avgx < -0.5:
        x_done_flag = False
        y_done_flag = False
        alignment_done = False
    elif avgy > 0.5 or avgy < -0.5:
        y_done_flag = False
        alignment_done = False
    else:
        pass

    # Height Control Process
    if distance >= 35:
        cv2.putText(frame, f"Drone Descending", (7, 100), cv2.FONT_HERSHEY_PLAIN,
                   2, (0, 255, 0), 2, cv2.LINE_AA)
        send_ned_velocity(0, 0, 0.5, 0.02)

```

Figure 22: Code showing the usage of flags for the alignment reiterative process during drone descend

It was considered good practice to constantly reiterate through the control sequence to ensure that the drone was always positioned as closely to the centre of the package for the interface to secure the parcel successfully.

3.7 Simulation of Control Algorithm

During the development of the control algorithm, the team needed a way to first test the program without flying the drone. This is due to the high-risk nature of operating the drone in which an unwanted crash due to a code exception could result in significant resource losses. As such, a robotic simulator called CoppeliaSim, formerly known as V-REP (Virtual Robot Experimentation Platform), was used to model a drone with simplified physics. The code could then be inputted into this software with debugging and improvements made to it based on visual feedback of the drone's simulated behaviour.

Figure 23 shows an example snippet of the simulation during one of the earlier phases of code testing. ArUco marker #444 was used as the target for control. The sequential algorithm corrects for rotation using trigonometry before using the x- and y-coordinates of #444 to align the drone above the marker. Once all the criteria are met, flags direct the code to descend until a height of 35 cm is achieved. The closed loop feedback system then unflags if any set threshold is exceeded which allows the drone to realign itself again in case of disturbances.

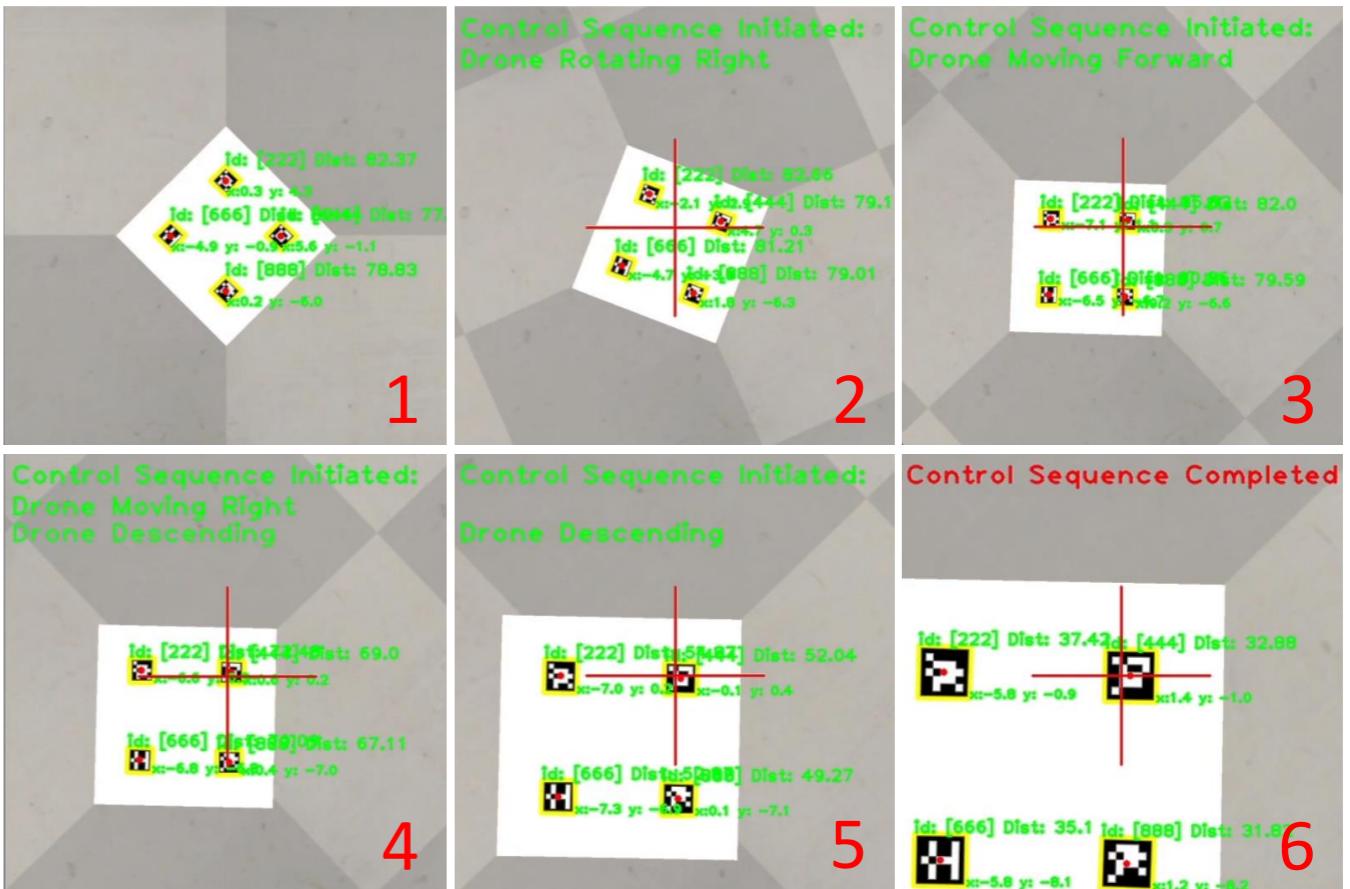
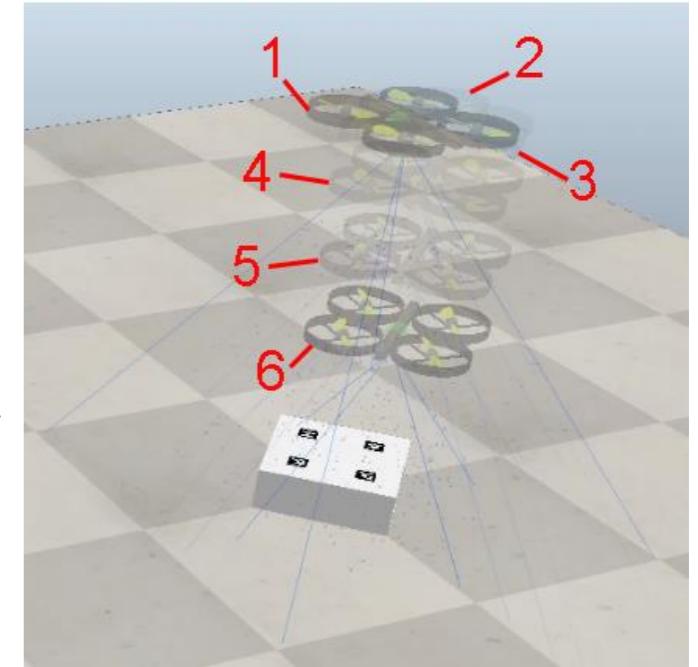


Figure 23: Simulated view of camera frame while control algorithm is active (above) and respective simplified motion of drone (right)

3.8 Interface Code

After the control algorithm is completed, the interface control sequence will follow through. A simple interface control code is written before integrating it to the main control code. The servo motor is initialised at 0°, running at a frequency of 50 Hz as specified in the datasheet of MG90S micro servo motor [21]. A brief flowchart of the parcel pickup process and the main loop for parcel loading and unloading are shown in Figure 24 respectively. After initialisation, the Time-of-Flight (ToF) sensor which is located at the bottom of the male connector will detect the distance between itself and the female connector continuously at a rate of 2 Hz (500 milliseconds) for a precise detection of real-time distance. Once the distance of 2 cm or lesser is detected, the servo motor will set its position to 135° to allow for connection between the connectors and this angle set by servo is obtained through a series of trial-and-error tests during debugging phase. When the parcel is loaded, the drone will be fed with velocity commands to fly to the specified location.



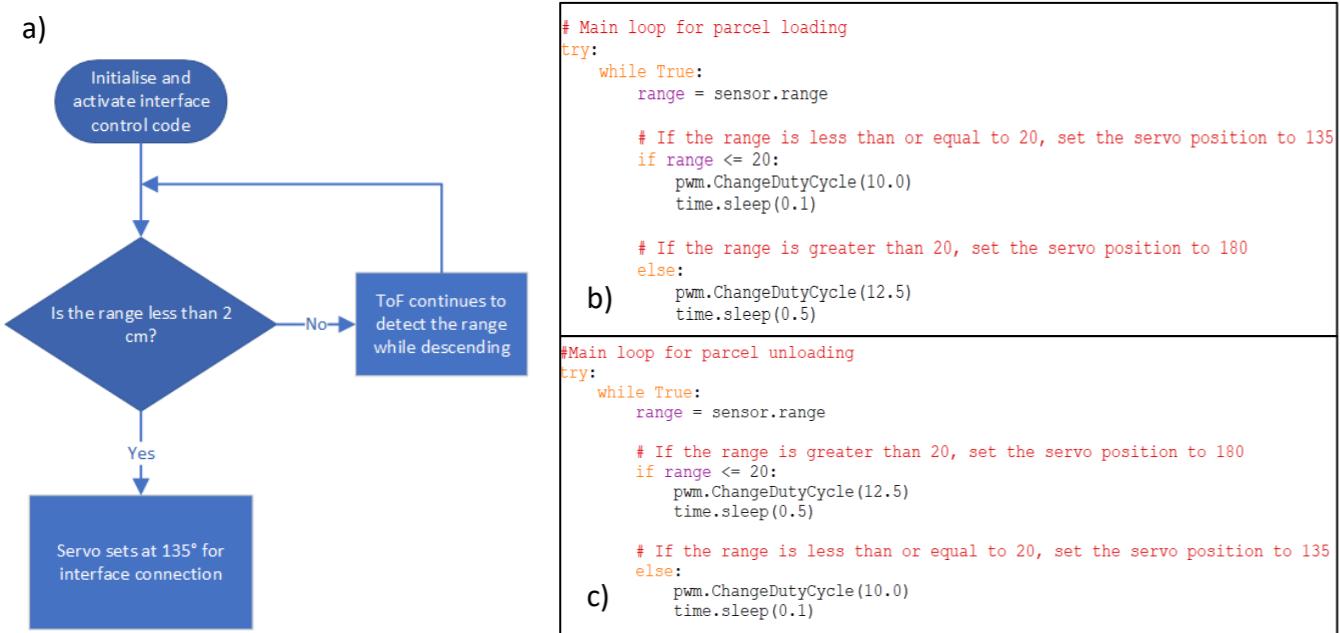


Figure 24: (a) Brief flowchart of interface control for parcel loading and main control code for (b) parcel loading as well as (c) parcel unloading.

3.9 Microcontroller & Camera Module

The micro-controller is arguably the most important component in the system. It is the main unit which communicates the control algorithm to the drone via the flight controller and vice versa. Raspberry Pi (RPi) 4 model B as shown in Figure 25, was selected as the micro-controller for this project due to two reasons:

- It provides sufficient computing power required to communicate the control algorithm to the flight controller at a reasonable frequency.
- RPi also has a built-in Python integrated development environment called “Thonny”. Thus, this provides the ease of porting over the control algorithm code, which is written in Python to RPi.



Figure 25: RPi 4 Model B & RPi camera module 2

The RPi camera module, also shown in Figure 25 was chosen for the CV due to ease of integration where the settings of the camera can be tweaked directly from the RPi itself. Moreover, it provides a high-resolution video capture. As explained in Section 3.5, the camera module was first calibrated to perform the pose estimation used in the control algorithm.

Once the initial setup of RPi such as the installation of OpenCV is completed, there was a need to find the most efficient solution to communicate the control algorithm to the drone, which enables the drone to perform manoeuvres such as rotating and descending. However, this high-level communication can only be achieved once two crucial APIs, MAVProxy and DroneKit are installed in the RPi.

3.9.1 Raspberry Pi Setup

A few essential python packages had to be installed to enable the high-level communication between the RPi and FCU. These packages are described in Table 3 and were installed via the RPi terminal. Once these essential packages have been installed, the control algorithm can be communicated from the RPi to the drone via the FCU.

Table 3: Installed packages on RPi along with their respective functions [22, 23]

Python Package Installation	Description
sudo apt-get update	Keep all packages up to date in Debian-based Linux distribution
sudo apt-get upgrade	Official package installer for Python 3
sudo apt-get install python-pip	Header files required to build Python extensions
sudo apt-get install python-dev	Python interface to the wxWidgets
sudo apt-get install screen python-wxgtk4.0	Facilitates serial communication
python-1xm1	Provide communication with the ArduPilot flight controller
sudo pip install pyserial	Provides ground control stations (GCS) that can be complemented with other ground stations such as Mission Planner
sudo pip install dronekit	
sudo pip install MAVProxy	

3.9.2 Remote Viewing

Remote viewing was necessary as the RPi is attached directly on the drone. VNC (Virtual Network Computing) viewer allows the connection to the remote RPi and watch its screen in real time, eliminating the need for an external monitor to be connected directly to the Pi itself. This also saves additional payload from the drone, allowing more allowance for other accessories such as the camera gimbal. Among other benefits is the flexibility to modify Python scripts on-the-go via remote access. This is especially beneficial for quick bug fixes encountered during flight tests. Furthermore, it provides the ability to monitor the CV in real time as illustrated in Figure 26.

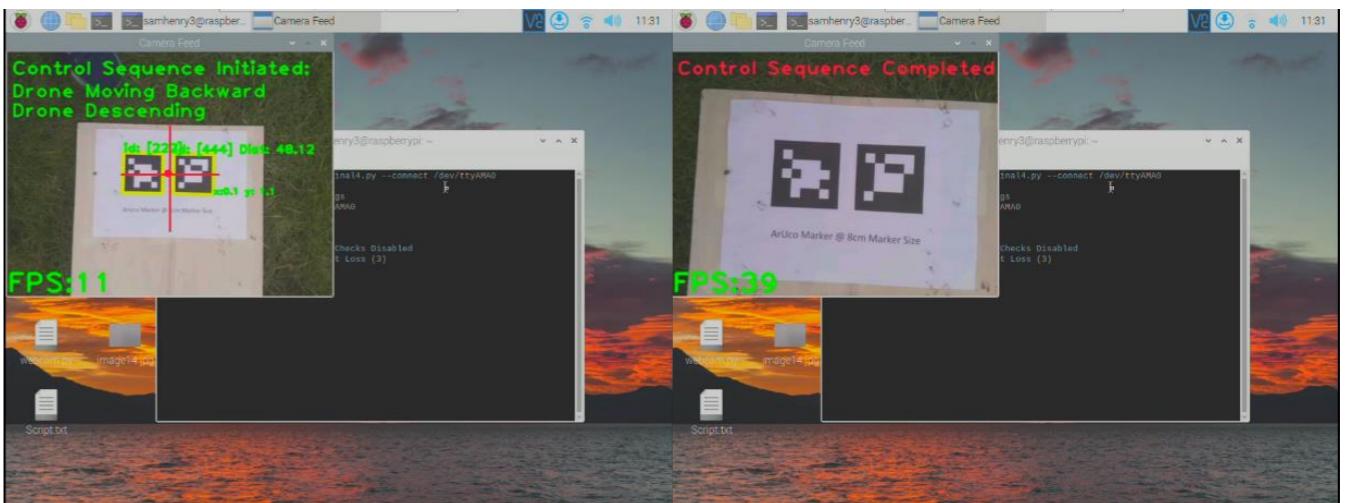


Figure 26: Usage of remote VNC when conducting outdoor tests

4 Avionics

To tie together all of the subsystems of MORPHEUS (the control of the drone and pickup of the package), a suitable drone was required for testing and demonstration. Aspects of the project to test and monitor with the use of the drone include:

- Effects of the package and the pickup interface on the stability of a drone in flight.
- Automated control of the drone through DroneKit directly from the Raspberry Pi.
- Real world accuracy of alignment obtainable by the computer vision software on a drone.
- Accuracy of alignment required by the pickup interface.

To limit the project scope concerning the drone, as it is only a part of the project, it was decided that specific requirements of the drone, e.g., flight time, range, maximum payload etc. is not of great importance. For example, the maximum weight and performance of the mechanical interface are to be tested separately. As the entire system is designed to be modular and scalable to any Vertical Take-Off and Landing (VTOL) drone, the drone used in this project is for testing and demonstration purposes only. At the beginning of the project, the option to purchase an off-the-shelf drone or design and manufacture a drone was discussed. It was decided that we would build our own drone due to cost efficiency and flexibility in its programming and troubleshooting.

4.1 Drone Objectives and Specifications

The goal when designing the drone was to provide a platform to integrate all of the subsystems together. It was decided that the drone would be small for easy and frequent test flights for the automatic flight and also because of the budget constraint of £850. At the beginning of the project, it was very uncertain about how much weight the drone had to be able to lift as the specifications and design of the subsystems (pickup interface and control system components) that it had to carry was uncertain. A S500 drone frame kit was obtained from SotonUAV early on. From there, the components of the drone were decided upon and purchased. These are listed in the section below.

4.1.1 Avionics Components

The table below lists all of the main components of the drone, excluding the battery and propellers, including their respective quantities, weights, and the source in which each component was obtained. The total weight of all of these components is 1260 grams.

Table 4: List of avionic components

Component	Full name	Mass (g)	Quantity	Source
Frame	S500 Multi-Rotor Air Frame	405	1	SotonUAV
Battery	Multistar High Capacity 5200mAh 3S 10C Multi-Rotor Lipo Pack	331	1	Amazon
Flight controller	Cube Orange Autopilot	75	1	GDP kit
Motor	Dilwe D2822 29x11.5mm (1100 KV)	57	4	Amazon
Propellers	9.5 x 4.7" Red Nylon RC Propellers	55	1	Amazon
GPS	Here3 GPS module	49	1	GDP kit
ESC	NUOBESTY 30A ESC	20	4	Amazon
Radio module	RFD 900x Modem	20	1	GDP kit
Receiver	FrSky X8R	17	1	GDP kit

4.1.2 Systems Diagram

Figure 28 shows the connections between all the components whereas Figure 27 illustrates the control system flowchart used to switch flight control between the pilot in command and onboard computer. During flight, there are always two communication links between the drone and the ground: the two-way communication radio telemetry to the ground control station (Mission Planner) and the one-way connection between the RC transmitter and receiver on the drone. The Raspberry Pi on the drone can be powered on and a control script run to establish an additional link to control the drone. To control the Raspberry Pi remotely, it is viewed on an additional laptop using VNC viewer. Control of the drone is switched between the Raspberry Pi and RC transmitter using the different modes available (mainly controlled by the RC transmitter): when set to GUIDED, control is given to the Raspberry Pi, and when the mode was switched to STABILIZE, control is set to be taken over by the RC transmitter, particularly in the case of emergencies, or when the script for automatic flight is not working as intended.

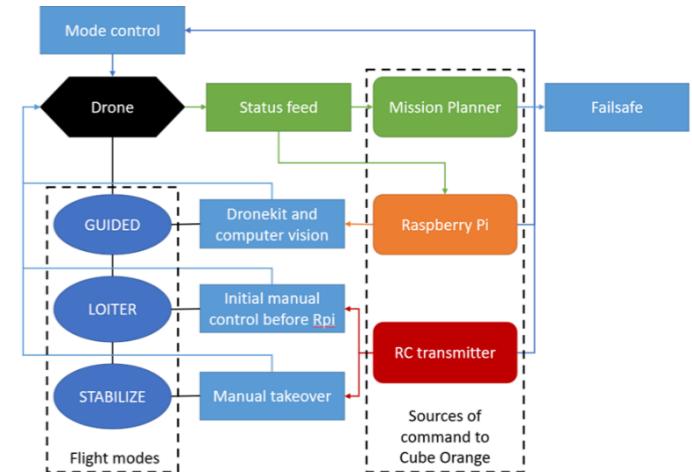


Figure 27: Avionics control system flowchart for different flight modes and command sources

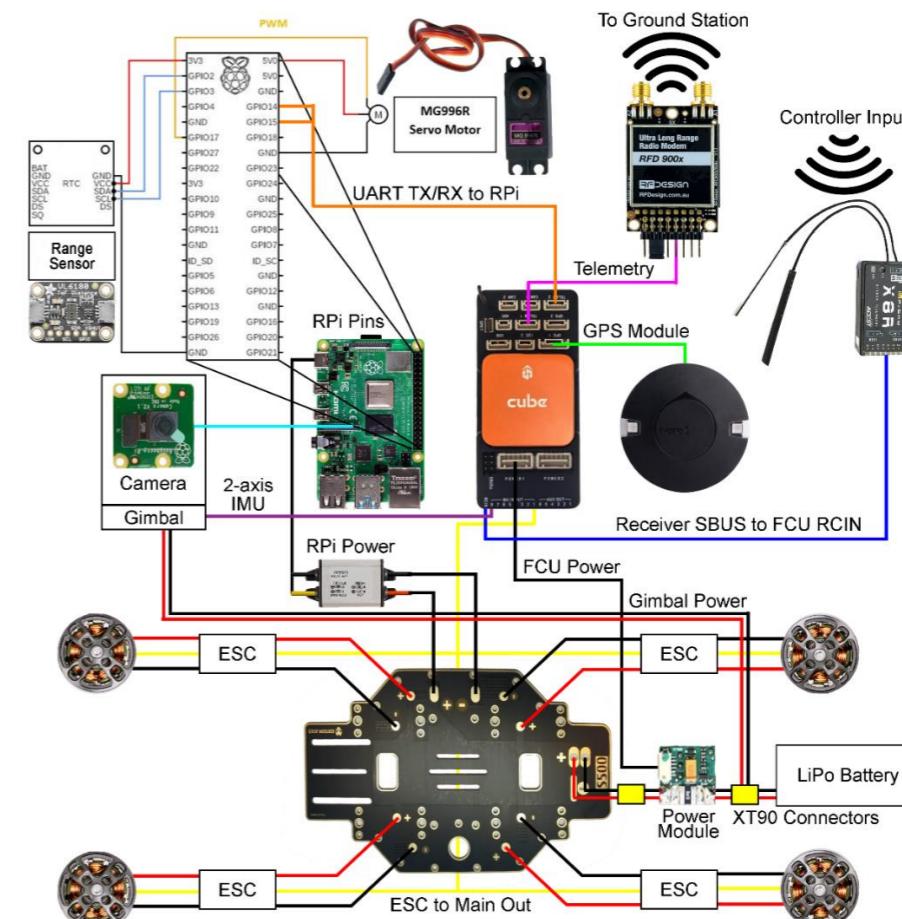


Figure 28: Avionics component diagram

4.2 Drone Performance

Before initial testing was performed, a set of $9.5 \times 4.7"$ red nylon RC (red) propellers were purchased from Amazon. After some flight tests were conducted, particularly when the gimbal system and interface were included, it was speculated that the thrust provided from the propulsion system were inadequate. Due to this, another set of $11 \times 4.5"$ APC (black) propellers for multirotors were later purchased from 3DXR to increase payload capacity. A thrust measurement testing rig produced by Tyto Robotics was used to scientifically determine the lifting performance of the drone. Two of the motors and the two different propellers were tested with the same Electronic Speed Controller (ESC) and battery of the drone. The results obtained are summarized in Figure 29.

During the experiment, it was found that with one of the motors (labelled with “Bad_” in the figure above) stuttered and did not spin when throttle was increased to above approximately $1430 \mu\text{s}$ with either propeller attached. Interestingly, with no propeller and the same motor, this problem did not occur. This result helped to explain some of the results that occurred during prior flight tests, where particularly with heavier payload (the larger battery and mechanical interface attached), the drone sent an error for ‘potential thrust losses’. It was decided that the motor was faulty and a new one was purchased to replace it, after which the same error did not occur again.

The results used to quantify the performance of the drone were taken to be with the non-faulty motor (with the orange and green lines in the figure above). The maximum thrust obtained from each of the propellers are summarised in the table below. Revolutions per minute (RPM) values were unable to be taken because the motors used were too small and so too far away from the sensor.

Table 5: Tested propellers and their associated thrust

Propeller	Maximum thrust (N)	Calculated total maximum thrust (N)	Weight capacity (g) – 1.5 safety factor
$9.5 \times 4.7"$ Red	3.0	12	815
$11 \times 4.5"$ Black	4.3	17.2	1169

From these results, the ability of the drone to carry any extra payload weight is very poor. This was due to initial urgency to quickly perform flight tests, which led to purchasing of unreliable, and potentially underpowered propulsion component. Thus, when conducting flight tests, detailed in later sections, real-life flights with a package attached were not performed.

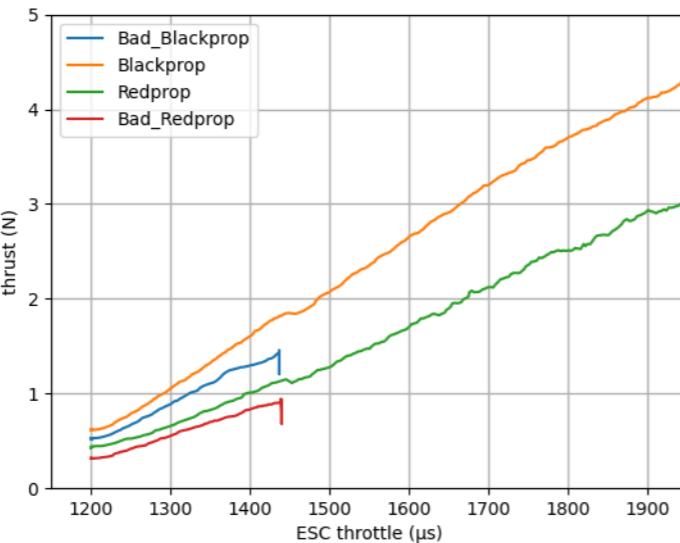


Figure 29: Thrust performance for different propellers and motors

4.2.1 Payload

The drone is required to carry the computer vision control system and the mechanical interface at empty load (without a package), and this is what determined the performance requirements of the drone. The components and their weights are detailed in the table below.

Table 6: Payload components and weights

Subsystem	Component	Mass (g)
Control system	12V to 5V 3A 15W DC power converter	40
Control system	Raspberry Pi 4 model B board	46
Control system	Raspberry Pi Camera v2	4
Control system	Camera gimbal	319
Control system	Gimbal stabilizing weight	64
Mechanical interface	Male Connector	188
Mechanical interface	Female Connector	142

With these components, when the drone is not carrying a package, the total payload weight is 661 g (excluding the female connector of the mechanical interface). After picking up a package, this payload weight increases by 142 g for the female connector and the mass of the package itself.

4.3 Flight Tests

Although the drone was small enough to fly indoors, it required GPS signal to fly in LOITER and GUIDED mode. Thus, permission was obtained from Health & Safety for the period from January to May 2023 to fly a drone tethered to the ground of a grass plot within Boldrewood Campus. Earlier flights tests were focused more on the manual flying of the drone, assessing its stability and performance with and without excess weight such as the mechanical interface and the gimbal system, shown in the figure below. During this time, many errors were found and debugged such as “bad battery” and the “potential thrust loss” as mentioned previously in Section 4.2.



Figure 30: 2-axis gimbal used to stabilise the camera with male connector behind it

Both propellers (red and black) were also tested in real flight at this stage. From the previous Tyto results, it is known that the black propellers produced more thrust. However, during flight testing it was

found that, with the black propellers, the drone was very unstable and wobbled significantly back and forth before crashing before being able to properly take off. This is because the larger propellers have a higher moment of inertia, so quick changes in RPM for stabilization become more difficult, therefore causing instability. Due to this, in all further flights, the red propellers were used.

4.3.1 Iron bird tests

Iron bird tests (shown in Figure 31) were also performed for the control code with the ArUco markers to ensure that the code was producing the correct instructions to the drone before actual flight tests. These tests were very promising, and although slightly slow, the output directional commands were as expected, and automatic flight tests were scheduled soon after that.

4.3.2 Automatic Flight Tests

Once the performance and stability of the drone were satisfactory, additional flight tests were conducted where the drone was fully controlled by the computer control system using the Raspberry Pi. The setup of the outdoor test area could be seen in Figure 32 alongside the initial proposed test sequence. It should be noted here that due to time constraints, integration with the what3words API (which can route the drone to any 3-metre square worldwide based on three words), inclusion of shape detection for the computer vision system, and implementation of parameter sweeping techniques were not achieved. As such, only autonomous testing of the ArUco markers were tested in this project. Commands from the DroneKit-Python library such as takeoff, hover, and land were initially trialed alongside velocity-vector-based directional controls in the North-East-Down (NED) frame. Once these were proven to command the drone in the intended way, the code for drone alignment to the ArUco markers (which were placed on the surface of the package) by computer vision was tested. The control process tested is as follows:

1. The script is run, and camera feed starts.
2. The drone takes off manually by RC transmitter in LOITER mode.
3. Mode is switched to GUIDED and Raspberry Pi automatically takes over and begins control sequence by converting pose information of the markers into velocity vector flight commands.

From these tests it was found that when on the ground, the camera feed runs smoothly. However, once the drone lifts off the ground, the camera feed experiences a severe drop in quality and eventually cuts out. It was already proven earlier that the control code worked during iron bird testing, so the iron bird test was again repeated on the field with the same setup with the only difference being the propellers

to eliminate any other factor that could be contributing to its failure such as communications or GPS signal on the field. The team concluded that the cause of failure of the camera feed during flight is due to a faulty ribbon cable as replacing it with a substitute and ensuring that it was not bent fixed the lack of a camera stream issue. However, this was as far as flight testing of the final design was possible as the team could no longer arrange any other suitable days to conduct further tests. Therefore, the team has recommended continuing the diagnosis of the current design to find and fix other unanticipated issues as a consideration for future teams who might want to continue this project.

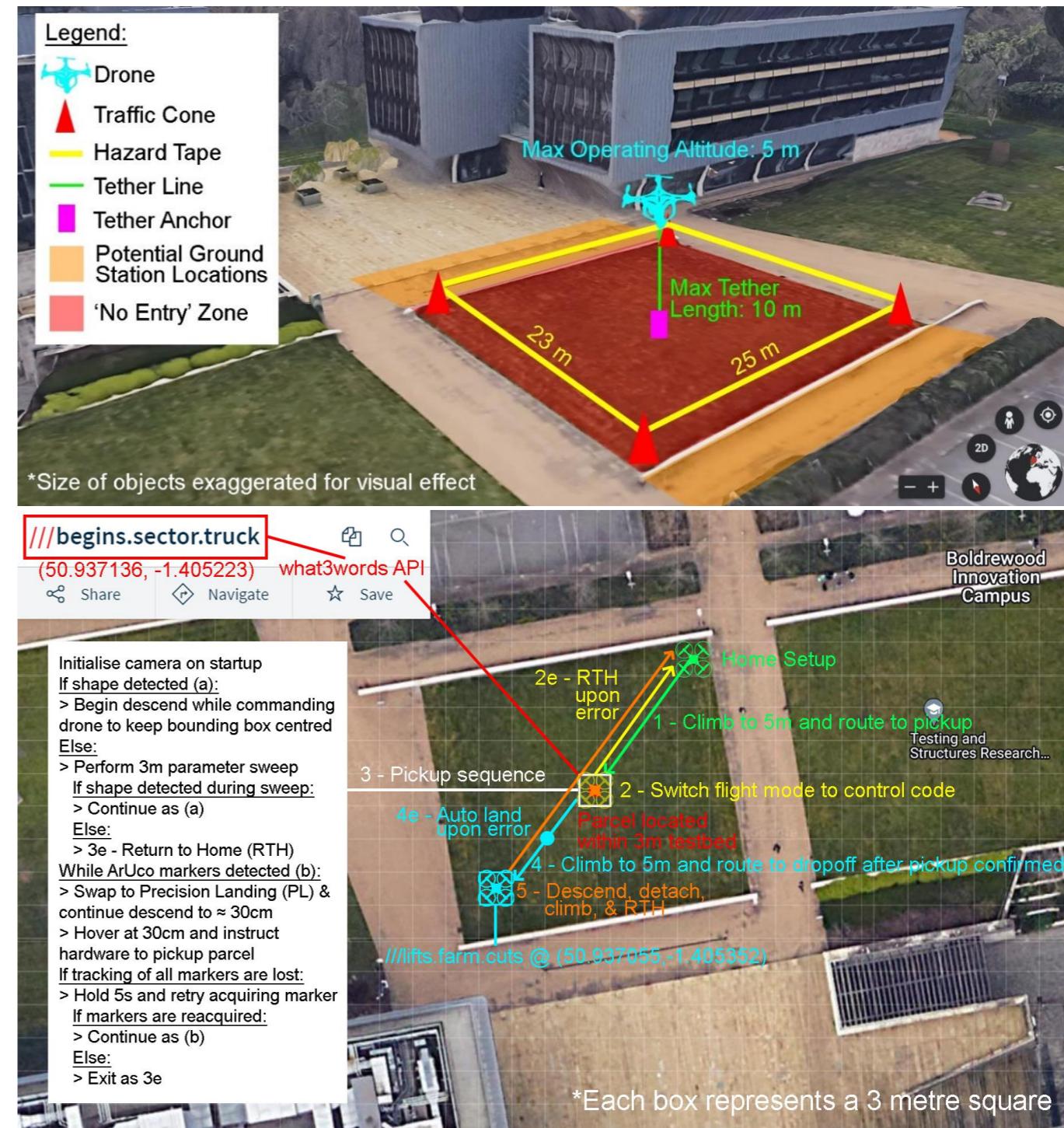


Figure 32: Setup of Boldrewood campus area (top) and initial proposed software test sequence (bottom)

5 Hardware

5.1 Idea Conceptualization

Drone loading mechanisms for parcel deliveries are currently available and being tested by multiple companies. This project intends to focus on the autonomous system based on, but not restricted to, ideas conceptualized by the currently available designs (Appendix C). Current drone package delivery systems being tested for commercial use have components or a combination of manual loading and unloading, and autonomous loading and unloading systems. However, there are currently no designs for a fully autonomous loading and unloading system, without the use of human assistance or an additional system external to the drone. Thus, the design for this project intends to develop a combination of both autonomous loading and unloading system.

5.2 Current Drone Package Delivery Designs

Various methods of loading and unloading designs are used by companies manufacturing drones for logistics purposes. The type of loading and unloading mechanisms used typically depend on the use case of the system as it is difficult to manufacture an optimal system that can be used in various situations. Concepts and ideas are developed from analysing the various loading and unloading mechanisms currently being used by drone manufacturing companies such as Elroy Air, Aura UAV, Aergility, and Drone Delivery Canada. It is to be noted that due to details of the loading and unloading process not fully explained by some companies, how the processes work is assumed based on the images provided.

From some of the available designs used for manual loading and unloading systems, a few design concepts are recognized. Aergility uses the interior body of their drone with a relatively large storage area due to a large drone used (size comparison with an adult from picture). An operator is required to manually load and unload the packages from inside the drone's storage compartment. Draganfly uses a similar storage design to Aergility however, the storage compartment is observed to be a separately manufactured box attached to the barebones drone frame. Aura UAV uses two storage systems with two storage shelves where packages are strapped onto and a netting underneath the drone's body. The packages on the shelves require straps to keep the packages in place thus an operator is required to perform the loading process. Velos Rotors seems to have a detachable parcel underneath the body of the drone which seems to require a manual loading and unloading process.

There are very few autonomous loading and unloading systems used on drones found. One of them is from Avidrone Aerospace where the package is held under the drone's body using multiple clamps that hold onto the package. The loading process is not disclosed however from the demonstration video provided it is likely that a manual loading process is required. The unloading process on the other hand is autonomous as the drone open the clamps that release the package upon arrival at the intended destination. Elroy Air uses a detachable storage bay where packages are filled into where the drone autonomously picks up and attaches the storage bay on the underside of the drone using a lever and pulley system hooking onto the metal bars above the storage bay. This method seems suitable for deliveries of multiple parcels at one time.

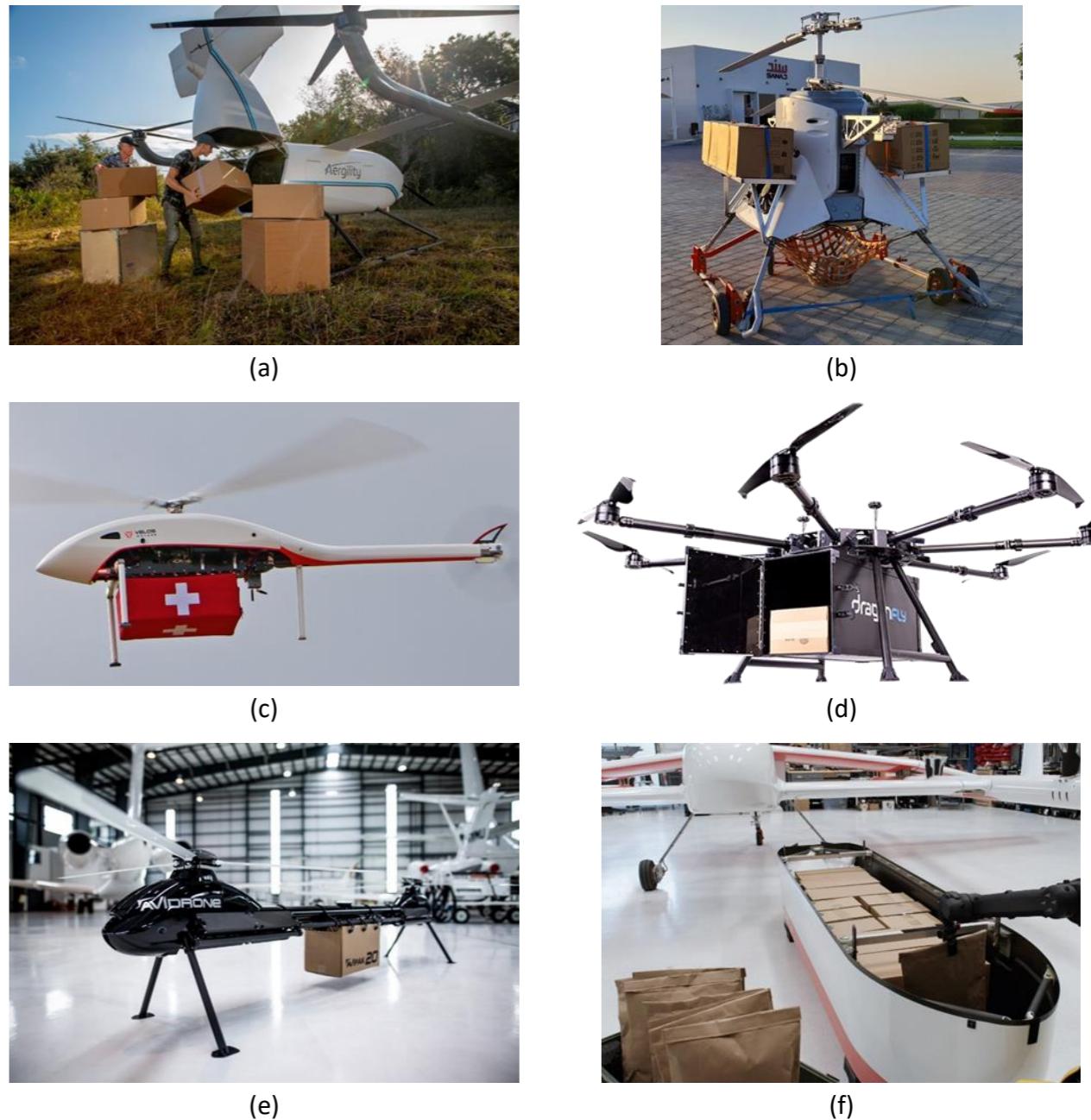


Figure 33: Images of various commercially available drone platforms and their loading mechanisms. In appearance: Aergility, Aura UAV, Velos Rotors, Draganfly, Avidrone Aerospace, Elroy Air

5.3 Design Requirements

No strict restrictions are posed when conceptualizing the design of the fully autonomous interface system. However, the interface is designed to fit a small multirotor drone with a 500-gram payload limit, specified by the drone manufacturer, with an additional focus to aid the software avionics system when loading the package onto the drone. A binary weighting matrix, as shown in Table 7, is used to identify the significant design criteria to provide a more focused design direction. From the criteria that are evaluated, importance is most significant for the design to ease the software system, easily repairable, is reliable and durable.

Table 7: Binary weighting matrix of mechanical interface

Mechanical Interface Design	Reliable	Lightweight	Durability	Minor Effects on Drone Flight Performance	Modular	Easy to Repair/Service	Simple Mechanism	Acts as Aid to Software System	Scores	Biased Scores (scores + 1)	Normalized Scores
Reliability	x	1	1	1	1	0	1	0	5	6	16.7%
Lightweight	0	x	0	0	1	0	0	0	1	2	5.55%
Durability	0	1	x	1	1	0	1	0	4	5	13.9%
Minor Effects on Drone Flight Performance	0	1	0	x	0	0	1	0	2	3	8.33%
Modular	0	0	0	1	x	0	1	0	2	3	8.33%
Easy to Repair	1	1	1	1	1	x	1	0	6	7	19.4%
Simple Mechanism	0	1	0	0	0	0	x	0	1	2	5.55%
Support to Camera Guidance Software	1	1	1	1	1	1	1	x	7	8	22.2%
						Total	36	100%			

Table 8: Initial concepts for the three main components of the hardware subsystem

Solution Sub-functions	A	B	C	D	E
Connection method					/
Locking mechanism	Motor & bar linkages	Motor, bar linkage, and cord	Electro-magnets	Mechanical push-latch/springs	/
Number of linkages between drone and interface	1	2	3	4	6

Although the design process used a binary weighting matrix and a Morph table to assist in the initial design, a ‘fail fast, fail early’ approach was implemented as rapid prototyping allows for multiple design iterations given the available project period. The use of a detachable storage container was chosen for this project as it is necessary to allow for an autonomous delivery system design. Three main

components of the system were identified, the male and female connectors, method of locking and unlocking mechanism, and number of linkages between the drone and interface. One of the key design factors of the interface was to use simple mechanism reducing the likelihood of failure and improving reliability. The interface should also not hinder the drone’s flight performance and aid the camera guided software system in attaching the package.

For the first design concept, the push latch mechanism was initially chosen as it would be fully mechanical and electronic components would not be necessary for the mechanism to function as compared to the other methods. However, through discussions with the team, it was necessary to include a motor to simplify the locking mechanism as a simple locking and unlocking mechanism would require much more time to design. Four linkages between the drone and interface were chosen as it would provide better stability between the drone, interface, and package as compared to fewer number of linkages. One linkage would leave the whole system susceptible to a pendulum effect which could affect the drone’s flight performance. Using three linkages was not suitable as the load between the interface and the drone would be uneven when compared to using an even number of linkages.

5.4 Final Design and Iterations

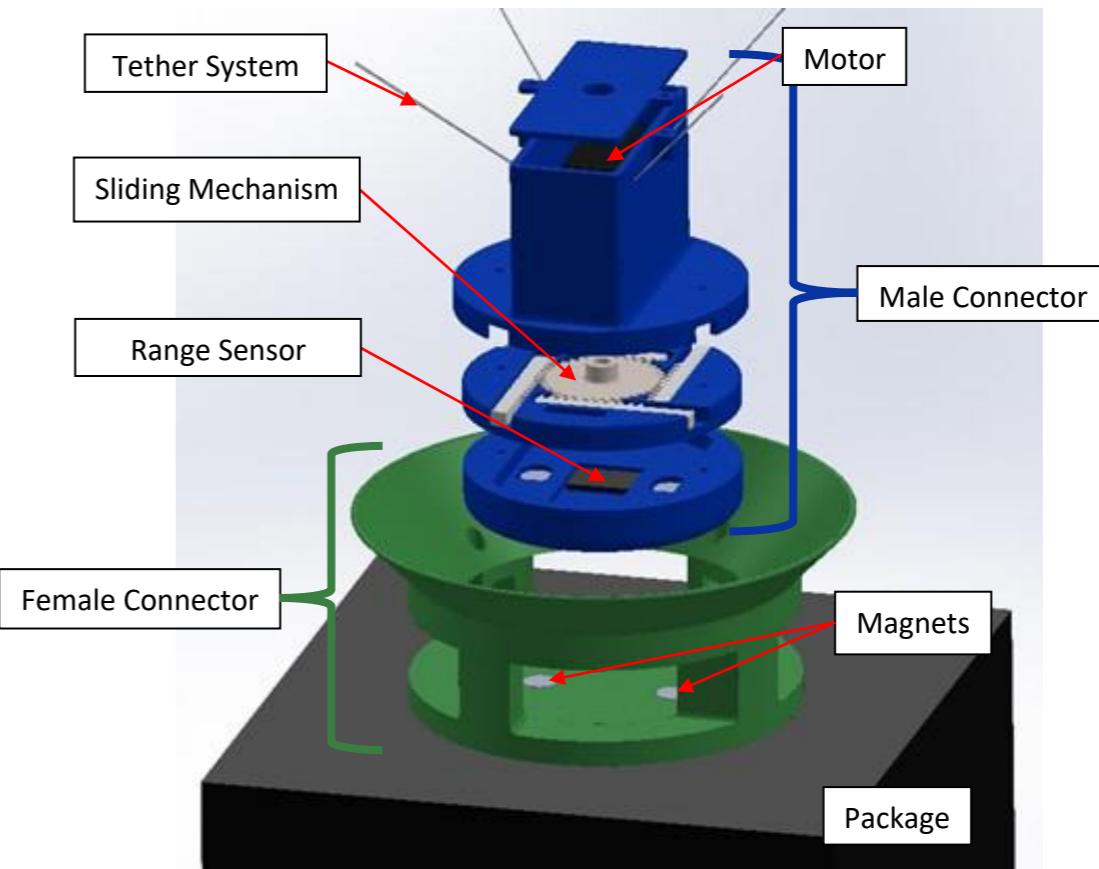


Figure 34: Exploded Computer-Aided Design (CAD) model of final proposed mechanical interface design

The final design as shown in Figure 34 is the 6th iteration prototype. It is made of two main parts, the male and female connectors. The female connector is to be attached on top of the delivery package where the top of the female connector is a cone to assist the software system when connecting the interface together. The cone allows for a 10 cm radius deviation from the centre which helps reduce the

need for a strict accuracy requirement in the image detection software. The male interface consists of five components, the casing, slider mechanism, ToF sensor, alignment magnets, and motor. As the male connector descends into the female connector, four magnets, two horizontally on the bottom of the female connector and similarly inside the bottom part of the male connector housing, aligns the interface together. Once the interface meets, the ToF sensor identifies that the distance between the interface is less than 1 mm, and the motor initiates to slide the mechanism locking the male and female interface together before take-off.

This design is significantly different from the initial prototype design, Figure 35, which was much more complicated and received criticisms on its functionality and feasibility. Among the criticisms received for the design was the strict accuracy required for the software system to connect the pins into the connector, the amount of connector arms between the interface and the drone and whether they are required in the first place, and lastly the potential negative effects on flight performance of the drone with interface and package connected.

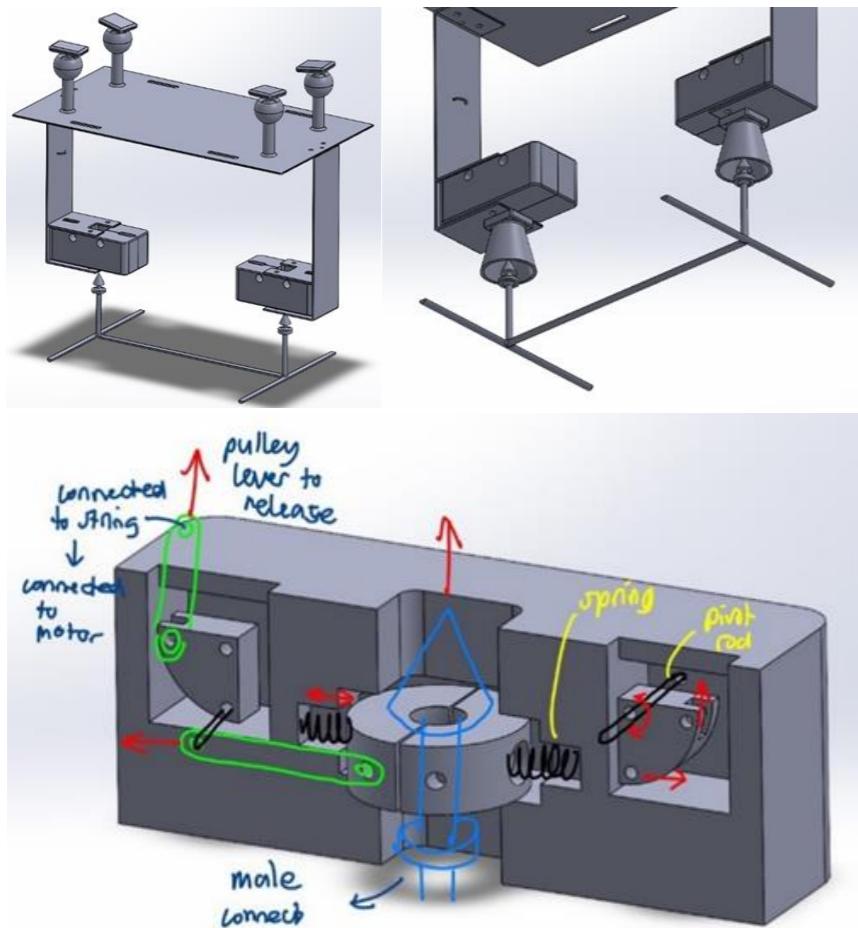


Figure 35: CAD model of hardware subsystem initial prototype design

Following the feedback received on the first iteration design, a complete redesign was necessary with focus on the feasibility and functionality of the design. Firstly, the push-latch mechanism was changed to a slider mechanism with the design of the 2nd prototype for a simpler design and potentially improved reliability. A further improvement of the slider mechanism that was chosen to be part of the male interface to be connected to the drone while the female interface was conceptualized as two hooks

attached to the top of the package as shown in the 3rd prototype in Figure 36. This design was accepted to be manufactured as the first physical prototype. Further refinements and iterations of this design was needed to improve its reliability and functionality after testing. This included the design of the 4th prototype with an improved hook and a larger size male connector size to ease the assembly and disassembly of the internal components. The ultrasonic sensor was also changed to a ToF sensor as it provided greater maximum and minimum range and better sensitivity for more accurate and reliable range reading. The 5th prototype is a similar design to the 4th prototype only changing the ultrasonic sensor housing to a much smaller ToF sensor housing.

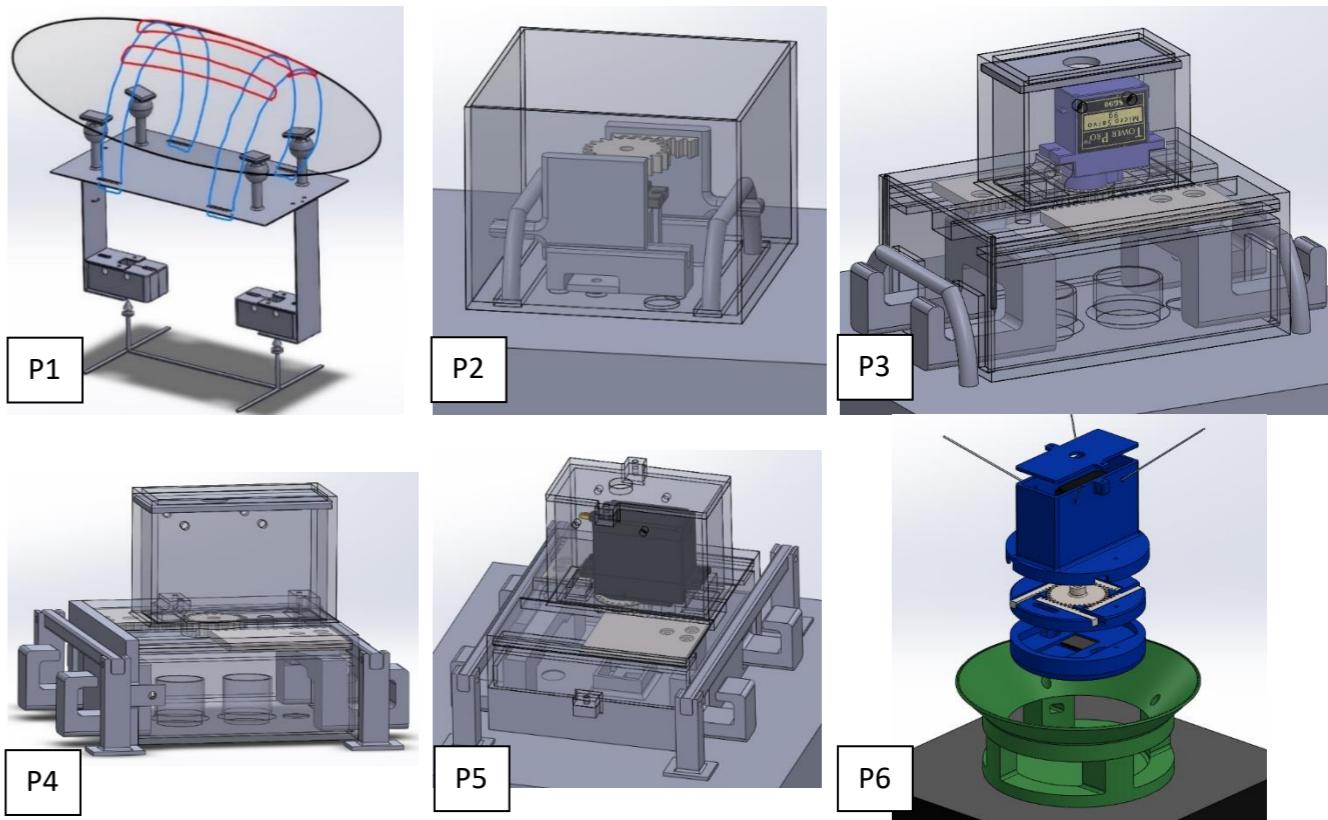


Figure 36: Several design iterations of slider mechanism

Following further developments of the software system, a connection angle deviation was necessary for the mechanical interface which meant that the penultimate design was inadequate to meet this requirement as it required very minimal, close to zero angle deviation from the software system. The 6th iteration and final design was developed from basic concepts of the previous iteration using the slider mechanism, alignment magnets, and introducing a conical and cylindrical interface design which allowed for a larger connection angle deviation other than allowing a larger connection area.

Testing of the final iteration provided satisfactory results in terms of reliability and functionality. Although no further iterations were made due to time constraints, a further improvement that could have been made to the design of the interface was to remove the limitation of the connection angle for the locking mechanism to work. This would have eliminated the potential for unsuccessful locking. However, a design of this sort might allow the package to rotate freely during flight thus the effects must also be considered.

6 Research & Analysis

6.1 Vibration on Drone

When considering multirotor UAVs for payload deliveries, it is important to acknowledge that like other transport means, the drone and its payload will experience vibrations during transport [24]. The vibration and shock to the drone are usually caused by structural vibration and shock resulting from period excitation from drone motors and propeller blades, and the interaction between the drone propellers and the airflow [25]. By understanding the input vibrations, as well as the shock and atmospheric conditions generated during drone operation as well as its potential hazards, we were able to implement measures to mitigate the risks.

6.1.1 Causes of Vibration

Drones experience vibrations due to various reasons, including motor and propeller imbalances, aerodynamic effects, and structural instability [26]. These vibrations can occur due to a product defect, and wear and tear. Propeller imbalances can occur when the propellers are out of balance or damaged. During a flight, the drone creates turbulence in the air around it. As a result, several aerodynamic effects such as propeller wash, vortex shedding, and airfoil flutter can induce vibration in the drone. Researchers have found that the drone's frame and motor mounts were the main sources of vibration [27]. Drone frames are usually made of lightweight materials to improve flight performance. However, this makes them more susceptible to vibrations; the more mechanical a drone is, the more susceptible it is [28]. The drone used for our project has a lot more added parts, and screws and nuts to hold everything together. This negatively affects the stability of the drone frame and induces more vibrations.

6.1.2 Effect of Vibrations

The effect of drone vibrations can be classified into two, short-term and long-term effects. The most noticeable effect of drone vibration is reduced flight performance. The drone's stability, manoeuvrability, and overall flight performance are negatively impacted by these vibrations resulting in increased difficulty for the drone to execute certain tasks, including package handling [29]. Another effect of drone vibration is reduced sensor accuracy. The sensors equipped on the drone such as the camera, ToF sensor, and GPS receiver will be affected, which can result in poor image quality and subsequently errors in data collection and processing. Drone vibration over an extended period may lead to premature failure of the drone components [25]. This can result in costly repairs and delays in delivery operations due to downtime for the drone.

6.1.3 Steps taken to Mitigate Vibration

In order to reduce the vibrations experienced by the drone as much as possible, several measures were executed. The drone propellers were balanced using precision balancing equipment and technique where the propellers were ensured to be evenly weighted. The motors used were tested through observation and using the ESC to ensure they were spinning equally as intended. Motors that were visibly faulty - not spinning at the same speed as the others – were considered faulty and replaced. To optimise the drone's structural design, vibration-damping materials and structures were incorporated. Materials with high stiffness-to-weight ratios such as rubber mounts on the motors were added to

dampen vibrations. All the screws and nuts were routinely inspected to ensure there are no loose components on the drone. The aerodynamic design of the drone was optimised through the arrangement of all the drone components to reduce the effects of turbulence and other aerodynamic forces that can induce vibration. To ensure the quality of the video feed from the camera is as desired, an active gimbal system was added to the system and the connections of the ribbon cable were secured in place.



Figure 37: Propeller balancer used to ensure even weight

While expensive measurements have been taken over the past three decades to characterise the vibration and atmospheric conditions in truck and rail environments, relatively few major research studies have been conducted to study the same factors in the aircraft environment [30]. As drone deliveries become a norm, the effect of vibration on drones and their payload will be further researched and understood, promoting innovative ideas to mitigate any risk involved, and improving the overall safety of the industry.

6.2 Multi-Tether System

When it comes to carrying payloads in UAVs, there are two approaches that are usually considered: a specialized cargo compartment within the UAV itself or attaching the cargo externally to the UAV. When the load is carried externally, the motion of the UAV induces cable swing and oscillations which in turn affects the motion of the aircraft [31]. There have been multiple studies and research conducted to investigate different control strategies to mitigate this effect. This includes an analytical closed-form study done in the University of Bologna [31]. In this article, the effect of additional non-actuated degrees of freedom that comes with the suspended payload on the stability of the multirotor are deeply addressed, and many different control strategies to deal with it are summarised. One particular strategy of interest is where the cable itself is modelled as a series of interconnected links to keep the cable in its vertical position [32].



Figure 38: Multi-tether system

Another comprehensive study was conducted in Qassim University, Saudi Arabia [33], which also included multiple control algorithms for dealing with the problem. In this article, SITL and HITL simulations and physical vehicle testbeds were also addressed to test the movement of the drone and load. Based on the amount of research done, it was decided that the problem induced quadcopter instability from a hanging payload would not be a significant issue. Thus, an external attachment system with a cable would be more suitable for the system as it enhances the drone performance during unloaded flight. It is also more suitable for the modularity of the system and is easier to load and unload autonomously.

All things considered, the male connector attaching to the drone uses the multi-tether system. The multi-tether system has many advantages over the traditional single-tether systems. The primary benefit of the multi-tether system is its increased stability. With four reels attaching the male connector to the drone, it is able to maintain a more stable position even in a windy or turbulent condition. This stability is ideal for the application of connecting the male connector to the female connector during the package loading process. Another reason we utilised multi-tether system over the single-tether system is because it guarantees increased safety. In the event that one of the reels snaps, the other reels are still able to support the interface, preventing it from falling and damage or injure the people beneath the drone. This added level of safety makes the multi-tether system an ideal choice when delivering packages in crowded or urban environments where it is more prone to collisions or accidents. Overall, the multi-tether system shows a significant improvement over the single-tether system due to its increased stability and safety and thus is the most suitable tether system to act as the medium connecting the drone to the interface.

6.2.1 Mathematical Model for The Cable Length

To calculate the optimum length for the cable, L_s (the length of the cable when it is under slack condition but not too long to affect the flight performance), we need to come out with a viable mathematical model. Figure 39 shows the measurement of the drone, the male connector and the cable length from the front and side view. The cable length will be calculated using Pythagoras theorem, vector dot product rule and formulae of circles.

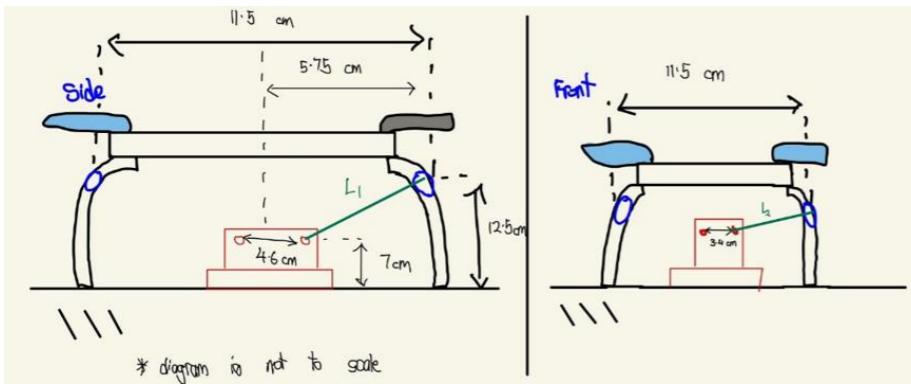


Figure 39: Measurement of the drone, the male connector, and the supposed length of cable (in green) from the front and side view

6.2.2 Methodology

The methodology to calculate L_s is first to calculate the hypotenuses, L_1 and L_2 from the given measurements on the front and side views. Then the measurements will be projected in vector view with the hole on the drone leg as origin as shown in the Figure 40. The L_1 and L_2 calculated previously will then undergo the vector dot product rule to calculate the hypotenuse, L in the direction where the cable will connect from the male connector to the drone. Lastly, formulae of circles below will be applied to calculate L_s :

$$\frac{\theta - \sin\theta}{2\pi} = \text{Ratio of the area of the segment of circle to the area of whole circle} \quad (5)$$

$$L_s = R\theta \quad (6)$$

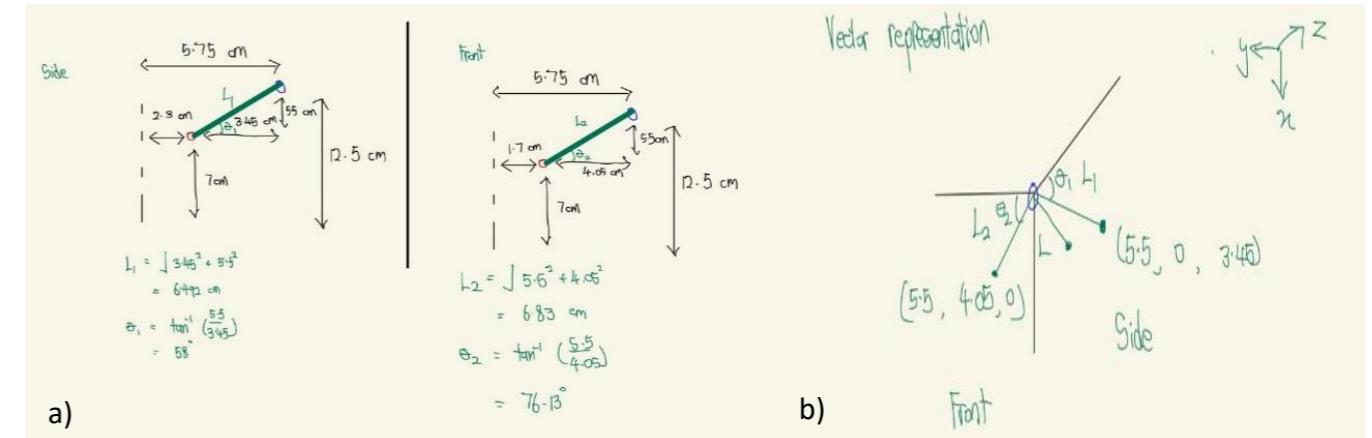


Figure 40: (a) Calculation of hypotenuses of side and front view, (b) Vector representation from the resulted calculation

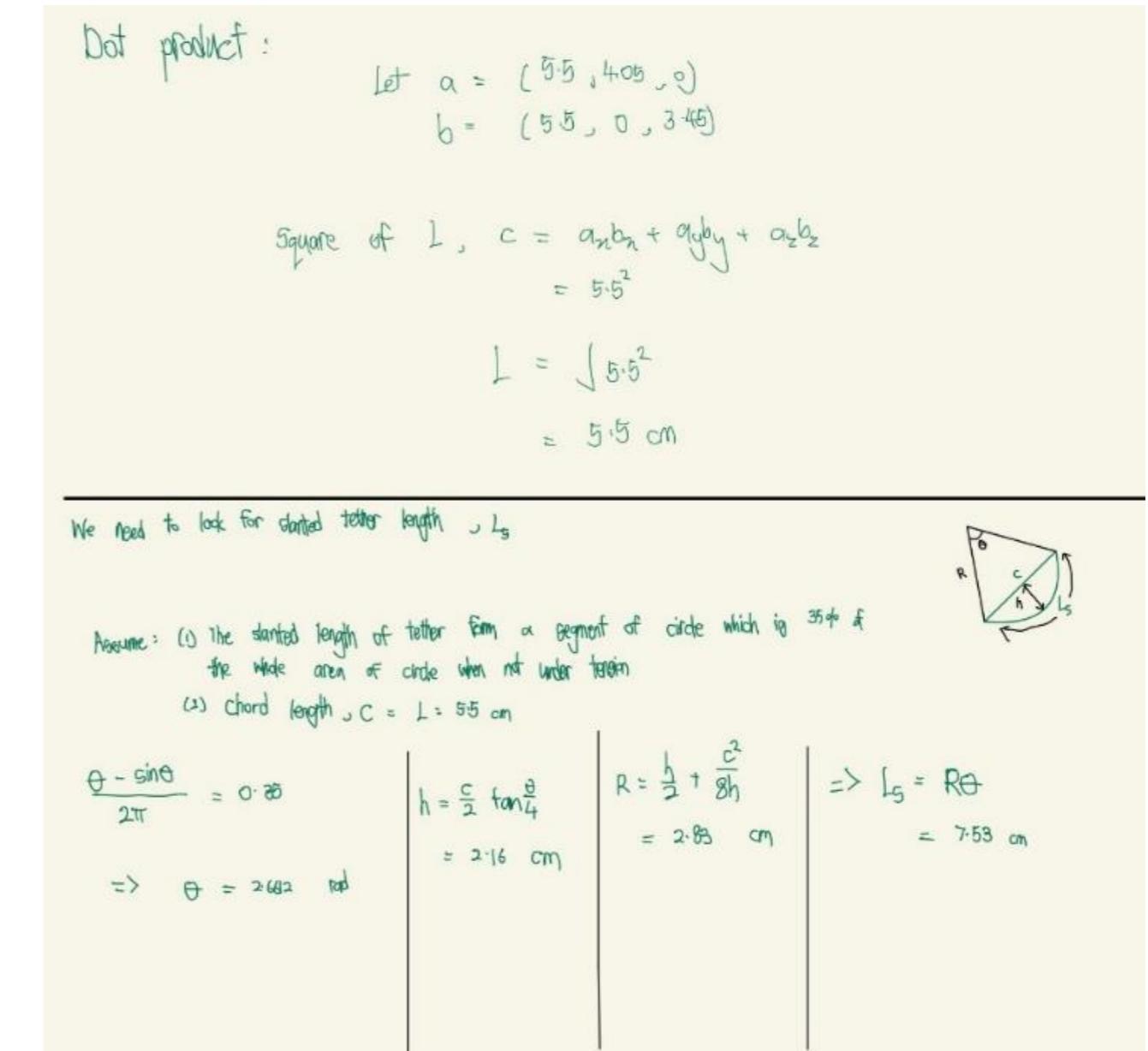


Figure 41: Calculation processes to obtain L_s

The following assumptions were made for this mathematical model and the calculation processes that were shown in Figure 41:

1. All measurement lengths are perpendicular of one another.
2. The holes to secure the cable on both the drone and the interface are small enough to be considered a point.
3. The interface is located beneath the centre point of the drone.
4. The hypotenuse calculated from vector dot product, L is the chord of the segment of circle formed together with L_s .
5. The ratio of the area of the segment of circle formed by L and L_s to the area of whole circle is 0.35.

Based on the calculation, the theoretical optimum length L_s is around 7.5 cm and the actual length used was approximately 9 cm. The discrepancy between the theoretical and actual cable length was due to the incorrect estimation of the assumption (5) above.

6.3 Electronics

One servo motor and a ToF sensor are used to enable the interface control to function properly. ToF sensor will measure the distance between male connector and female connector and the servo motor is responsible for the connection between the two connectors. We will discuss the thought process on the selection of both electronics.

6.3.1 Servo Motor

Servo motors are widely employed in a wide range of applications requiring precise control of angular position and they are designed to rotate a certain number of degrees in response to a control signal, allowing for precise control over the movement of the mechanical systems. The servo motor can be utilised to facilitate the connection between the male connector and female connector thus making it an essential part of the project. The criteria to a suitable servo motor are: (i) high torque and speed, (ii) durability. The micro servo motors in consideration are MG90S and SG90 and their specifications are summarised in the table below.

Table 9: Specifications of MG90S and SG90 micro servo motor [21, 34]

Specifications	MG90S	SG90
Mass (g)	13.4	9.0
Torque (kgF.cm)	1.8 - 2.2	1.8
Voltage (V)	4.8 – 6.0	4.8 (~5)
Speed (s/60°)	0.08	0.1

Based on the table above, we can see that MG90S is a better suit for the project application as it is able to operate at higher torque and speed at a higher voltage range. Besides, MG90S has a metal gear which provides durability and reliability which fulfils the second criterion. Thus, it can be concluded that MG90S micro servo motor is a better choice than SG90 micro servo motor for the drone application.

6.3.2 Distance Sensor

Ultrasonic, infrared (IR), Light Detection and Ranging (LIDAR) and ToF sensor are being considered for the application on our interface. There are a few factors to be considered when deciding on the suitable

distance sensor such as the ability for long range sensing, high frequency reading and the insensitivity to the external conditions. Table 10 below summarised the consideration points for each of the mentioned distance sensors. Based on the table below, we can see that both LIDAR and ToF sensor satisfy the requirements set earlier. However, based on the cost factor and also the fact that LIDAR will emit light pulse which may be harmful to the naked eye [35], ToF distance sensor is the best choice.

Table 10: Summary of the Capabilities of Ultrasonic, IR, LIDAR and ToF distance sensor [35]

Type	Ultrasonic	IR	LIDAR	ToF
Ability for Long Range Sensing	No	No	Yes	Yes
High reading frequency	No	No	Yes	Yes
Cost	Low	Low	High	Moderate
Sensitive to external conditions	Yes	No	No	No

6.3.3 ToF Distance Sensor

The Fermion: VL6180X ToF Distance Ranging Sensor was chosen for the interface control application. This ToF sensor is selected because of a few reasons such as its ability to measure 5 – 100 mm and is able to provide up to 200 mm in good environment, its robust anti-reflectance properties, and wide operating temperature range [36]. The ToF sensor consists of three main components: a transmitter, a receiver, and an instrument which is the timing circuit. The transmitter emits a pulse of light and the receiver detects the reflected light. The timing circuit measures the time between the emission of the light pulse and the detection of the reflected light, and the time delay is used to calculate the distance between the sensor and the object [37].

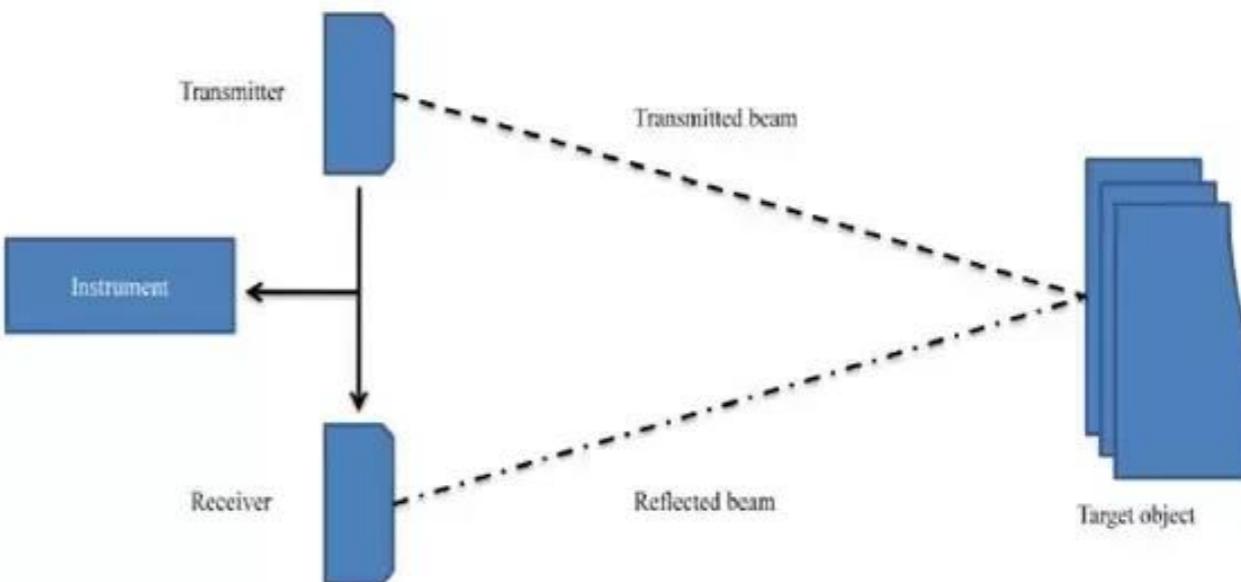


Figure 42: Operating Principles of ToF distance ranging sensor

6.4 Sustainability

As the focus of the project is on the software and the interface and the design of the interface is only at the prototype stage, it is hard to evaluate the sustainability potential of the interface itself. Besides, the

lack of widespread drone delivery operations also hinders the accurate measurement of energy consumption itself [38]. However, we can still evaluate the sustainability of the project (the drone delivery and lithium-ion battery) and how it aligns with the Sustainable Development Goals (SDGs) proposed by the United Nations to achieve a long-term sustainable development. The sustainability of our project will emphasise on Goal 6 (clean water and sanitation), Goal 8 (decent work and economic growth), Goal 12 (responsible consumption and production) and Goal 13 (climate action).

With the rapid development of e-commerce and economic globalisation, the frequency of commercial delivery has increased substantially, resulting in a significant increase in carbon emissions [39, 40]. These carbon emissions are mainly caused by the unrestrained consumption of fossil fuels by trucks and requires hundreds of years to offset the impact it caused to the environment. Due to the high pressure suffered by the logistics delivering companies, they are seeking emerging commercial delivery tool and the drone has the advantages of low carbon emission and low cost. Based on [41], drone assisted delivery reduced carbon emissions by 24.9%, reduced the total cost by 22.13% and shortened the delivery time by 20.65%, making positive impacts on social, economic and environment. As our project was originally intended to deliver medical supply, research has shown that medical drone application in healthcare supply chain greatly contributes to the country's healthcare supply chain by delivering medical supplies which have short shelf life on time, reducing wastage from expired medical items and improving delivery efficiency. Besides, the medical drone application also helps achieving SDGs (Goal 3 and 13), alongside improving the society's socioeconomic situations by lowering mortality rates and providing the beneficiaries with better social and economic lifestyle [42].

Due to its high energy density, low-self discharging rate, and light weight characteristics [43], the drone, avionics and the interface are all powered by a lithium-ion batteries (LIB). However, LIB contains harmful heavy metals (nickel and cobalt) organics which will adversely affect the environment and human health if mishandled. A life cycle assessment is performed to evaluate the environmental impact of LIB and lithium-ion polymer batteries (LIPB) and it was discovered that LIPB is more environmentally friendly than LIB [44]. LIPB is an improved version of LIB, having higher level of safety, better electrochemical stability, and a lower life cycle degradation rate [45, 46], giving delivering companies more incentives to using LIPB in the future to fulfil the vision of sustainability.

6.5 Finite Element Analysis of Final Design

A static structural analysis was conducted on the final prototype design to determine its structural soundness. Since all interface components were 3D printed using ABS plastic, this material was used for the analysis. A maximum load of 0.5 kg was set on the interface components as it was the payload limit of the multirotor and a full stress analysis on the assembly is analysed and displayed by individual components. For the purpose of this analysis, the effect of gravity on the components is ignored to provide a clearer view on the effect of the sole load applied.

Table 11: Table of maximum and average values for total deformation and equivalent stress

Total Deformation (m)	Equivalent Stress (Pa)
7.098×10^{-7}	2.674×10^5
3.499×10^{-7}	5.909×10^3

6.5.1 Interface Assembly

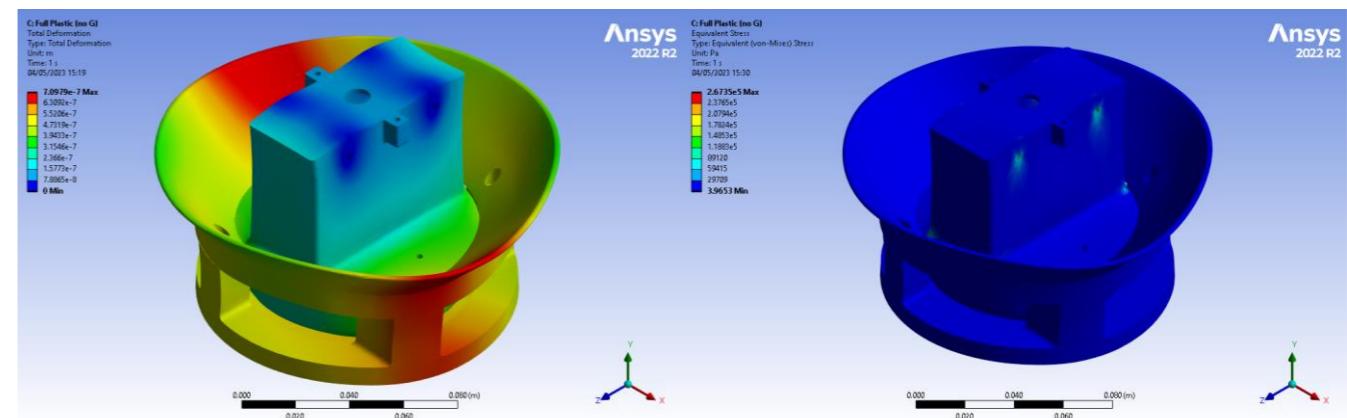


Figure 43: Total deformation (left) and equivalent stress distribution (right) on the interface assembly

Table 11 shows the maximum and average values of the deformation and the stress experienced by the interface assembly. Figure 43 displays the distribution of the stress and the change in deformation by area. As the assembly comprises of two components connected by a slider mechanism, a comprehensive analysis of the individual components follows.

6.5.2 Male Component

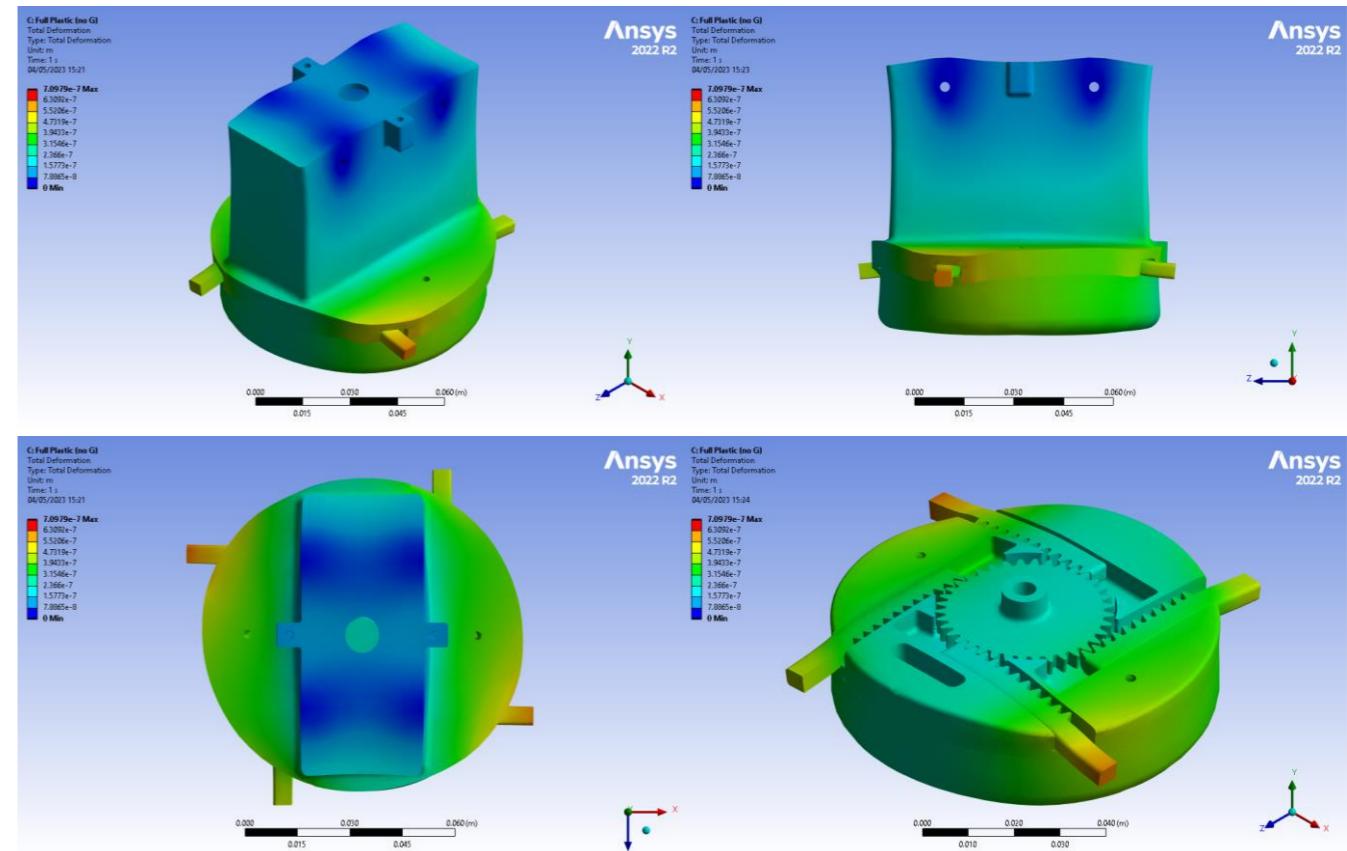


Figure 44: Total deformation on the male component

The images in the figure above show the different deformations on the male interface. The tip of sliders experiences the greatest deformation as expected as serves as the connection between the male and

female components, hence, bearing the total load that is attached to the female interface. The four sliders display unequal deformation but in a symmetrical manner. This could be caused by the shape and design of the male component where two of the sliders are parallel to the fixed part while the other two are perpendicular. The distance from the tip to the fixed part also differs between the pairs. The image on the bottom left of Figure 44 clearly shows this difference.

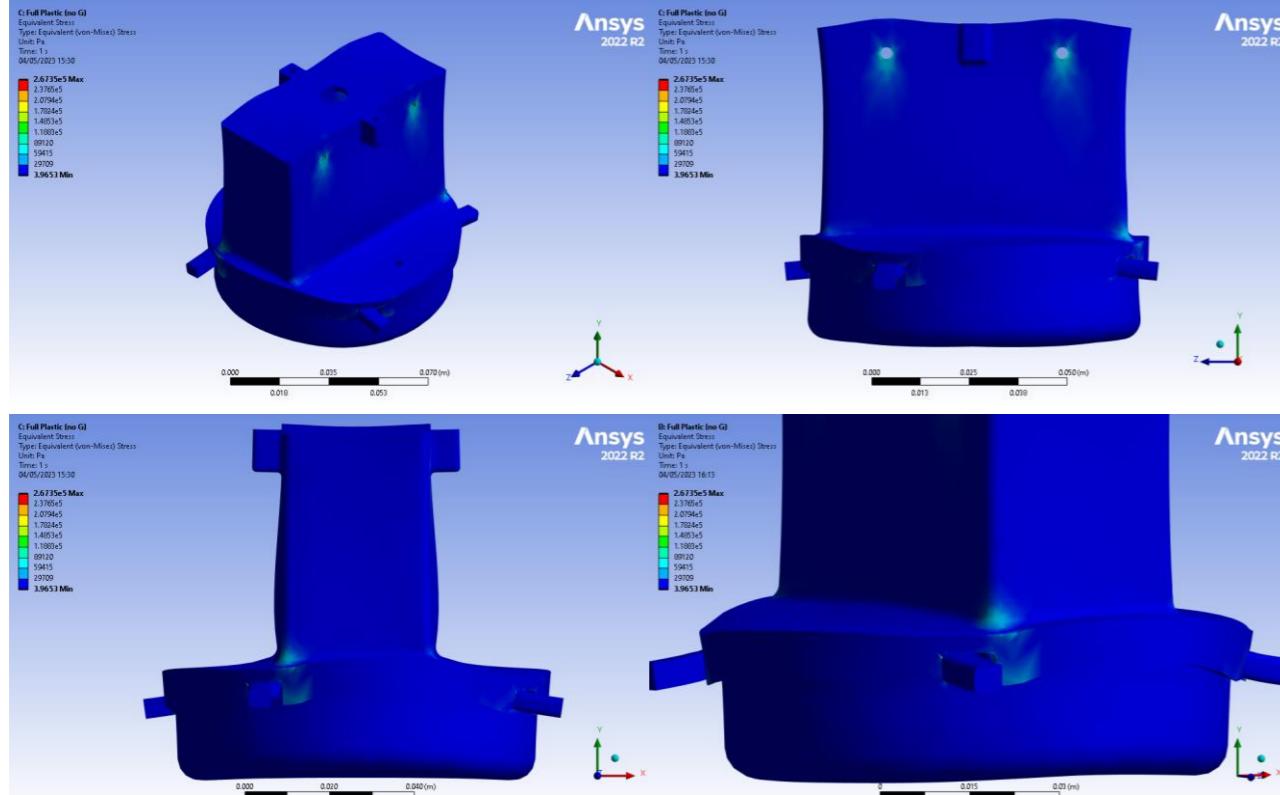


Figure 45: Equivalent stress distribution on the male component

The stress distribution of the male interface above shows two main things. Firstly, a significant amount of stress is experienced at the top of this component as shown in the top right image in Figure 45. This is simply because this part was fixed, inhibiting movement, and causing stress. Secondly, the vertices on the lower part of the component also experience higher stress. This region that covers the sliders is susceptible to stresses due to its proximity to the load bearer and its shape and design.

6.5.3 Female Component

The maximum deformation on the assembly is experienced by the female component as shown in Figure 46 on the following page. The regions with maximum deformation begin from the loading area at the bottom and extend right up to the cone. They show a pattern similar to the male component in that the deformation experienced is unequal but symmetrical. The effect of the fixed part on the male component is transferred through the sliders to result in this pattern. Two regions on the female component experience greater stress than other regions. The first part is the edges of the slots to which the slider from the male component connects (bottom left in Figure 47) and the second area is the connection to the load (bottom right in Figure 47). Both these areas have higher stress as anticipated but only show values no higher than $8.912 \times 10^4 \text{ Pa}$.

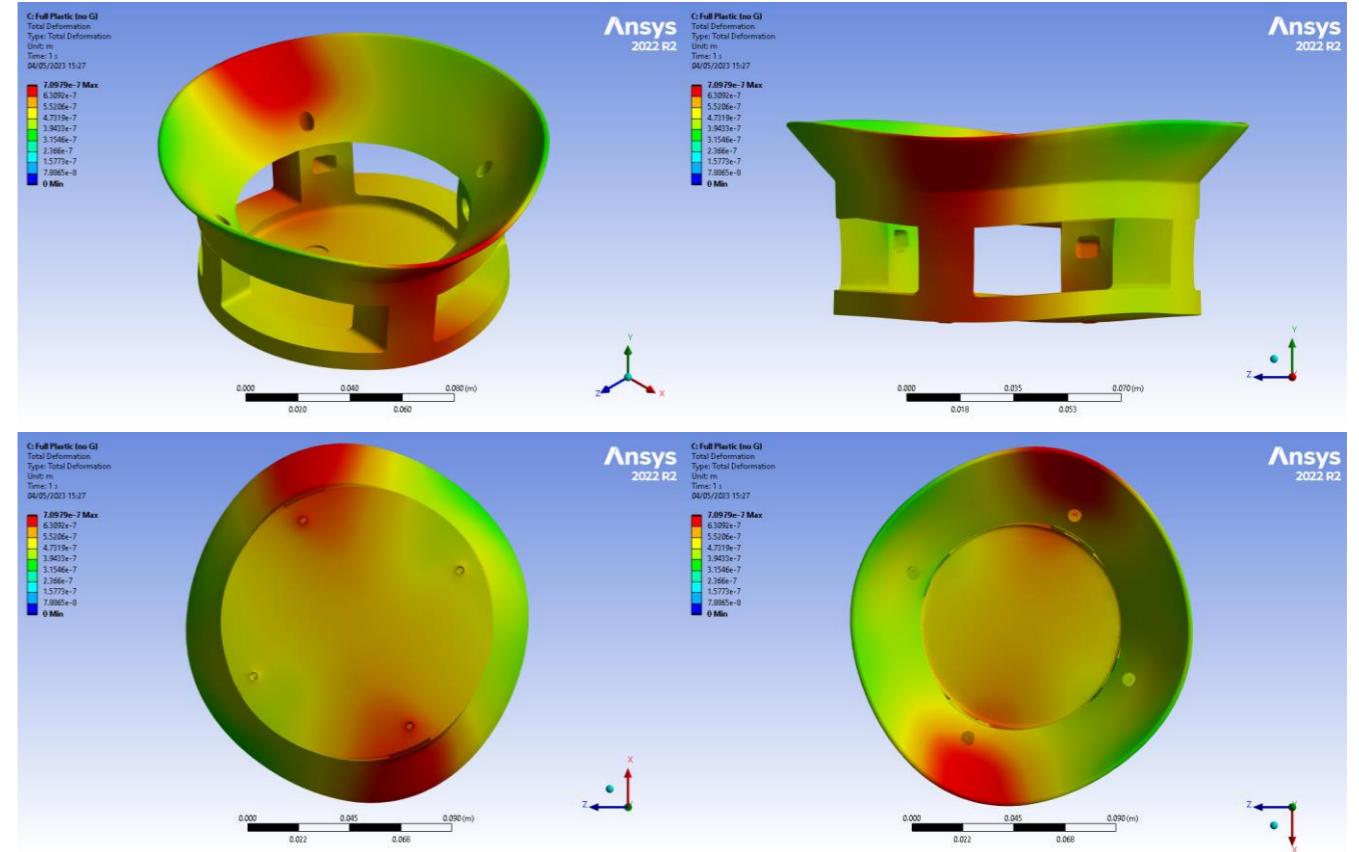


Figure 46: Total deformation on the female component

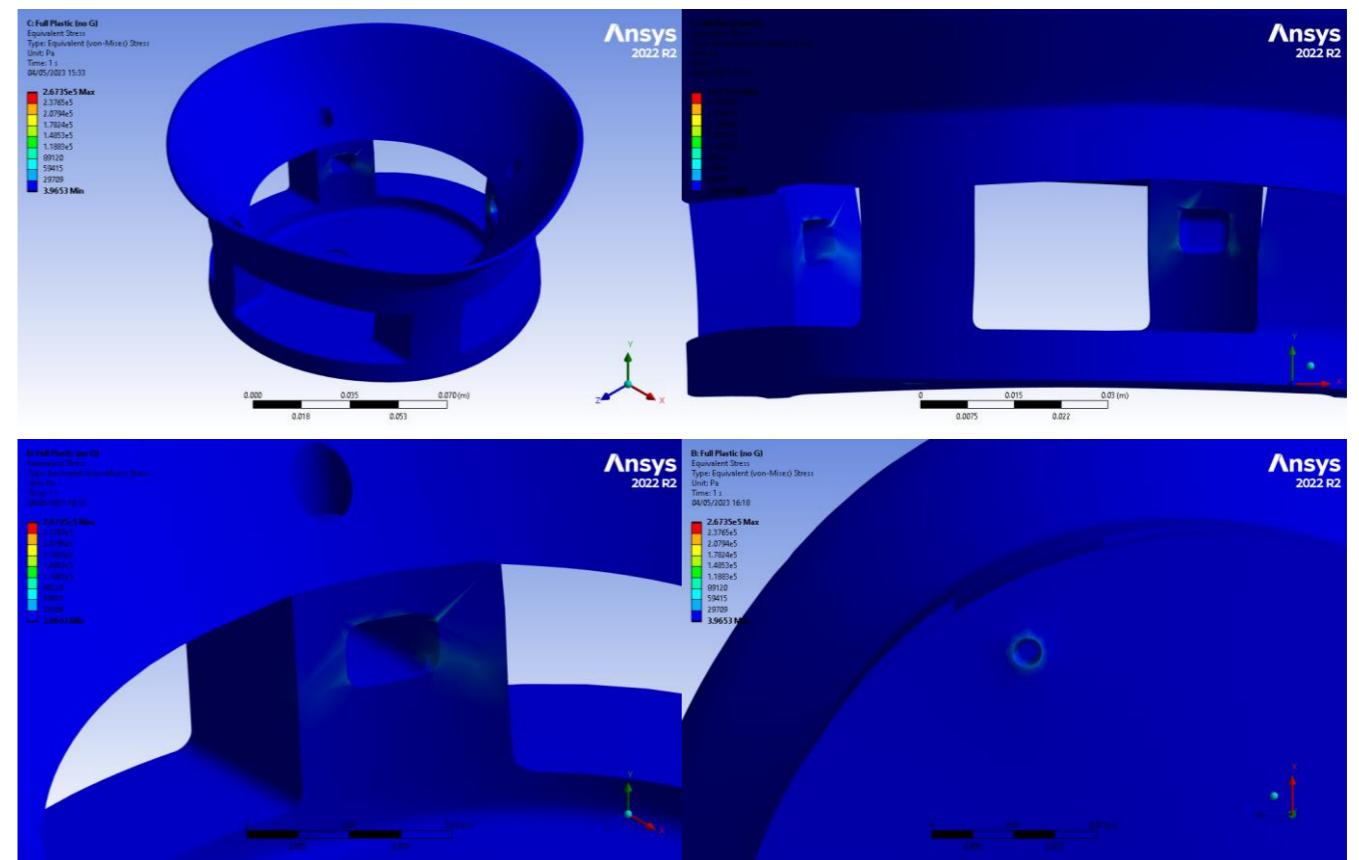


Figure 47: Equivalent stress distribution on the female component

6.6 Downwash Effect

A high velocity airflow that is directed downward is created when a drone's propellers rotate. This downward airflow accelerates the surrounding air, resulting in a drop in pressure above the blades and an increase in pressure below the propellers. In accordance with the Bernoulli's principle, the net imbalance of the pressure distribution generates lift which helps the drone to stay airborne. The flow near the wing tips tend to curl around the tips due to the pressure imbalance and travels from the high-pressure region to the low-pressure region. This circulatory motion of the flow rails downstream of the wing, forming a vortex. The vortices around the wingtip will drag the surrounding air and induce a small downward component of air velocity in the vicinity of the wing and is known as the downwash. When the drone is flying close to the ground, the downward flow of air can diminish the net effective lift created by the rotors, resulting in poor performance during take-off, landing, and low-altitude manoeuvres as observed during our flight test. Furthermore, the downwash will combine with the freestream velocity to create a horizontal air flow around the drone, resulting in a decrease in the effective angle of attack and, as a result, the propellers will generate less lift than they would if there was no downwash [47]. This can have an impact of the drone's control and stability, making it more difficult to maintain a stable flight at low altitude. This can be seen from the sudden drop in height when flying the drone at low altitude even though a constant thrust was applied by the pilot.

6.7 Experimental Testing

All iterations of the prototype were subjected to experimental tests to determine if it meets certain criteria that are required for this project. The criteria include:

- Functionality
- Limitations
- Point(s) of Failure

The functionality test is the most important criterion to be met as it proves the working concept of the design. The limitations encountered on any of the prototype designs are resolved on the following iteration until the final design with minimal to no major limitations was chosen. The point of failure for a particular design was determined by subjecting the prototype to extreme conditions. The behaviour of the interface mechanism right up to its breaking point is observed and used in designing the newer iterations. The experimental tests conducted on the final prototype design are as follows.

6.7.1 Preliminary Test

The prototype was initially subjected to a manual test where the functionality of the design was tested. The main functional criterion for this test is the connection between the interface components. The motor and the slider mechanism were tested separately to ensure they work as intended before the ToF sensor was incorporated into the system. The image on the left of Figure 48 shows the test conducted on an older iteration of the interface design. After conducting several tests, the prototype design showed several limitations which were used to model the final design. Table 12 shows some of the limitations encountered and the measures taken to overcome them. The final prototype design was then subjected to the same tests to ensure the solutions work as intended and all the limitations have been overcome.

Table 12: Limitations encountered during preliminary tests and their solutions

Limitation	Solution
1. Imprecise proximity sensor	Replace with a ToF sensor.
2. The sliding mechanism is not robust	The system was redesigned to improve robustness and to allow for a wider range of angles of approach for connection.
3. The pair magnets used for aligning do not have a large enough range to assist the male interface	A cone was added to the female interface to serve as the first form of assistance for the connection before the magnets align themselves.
4. The male connector does not align with the female connector as envisioned	

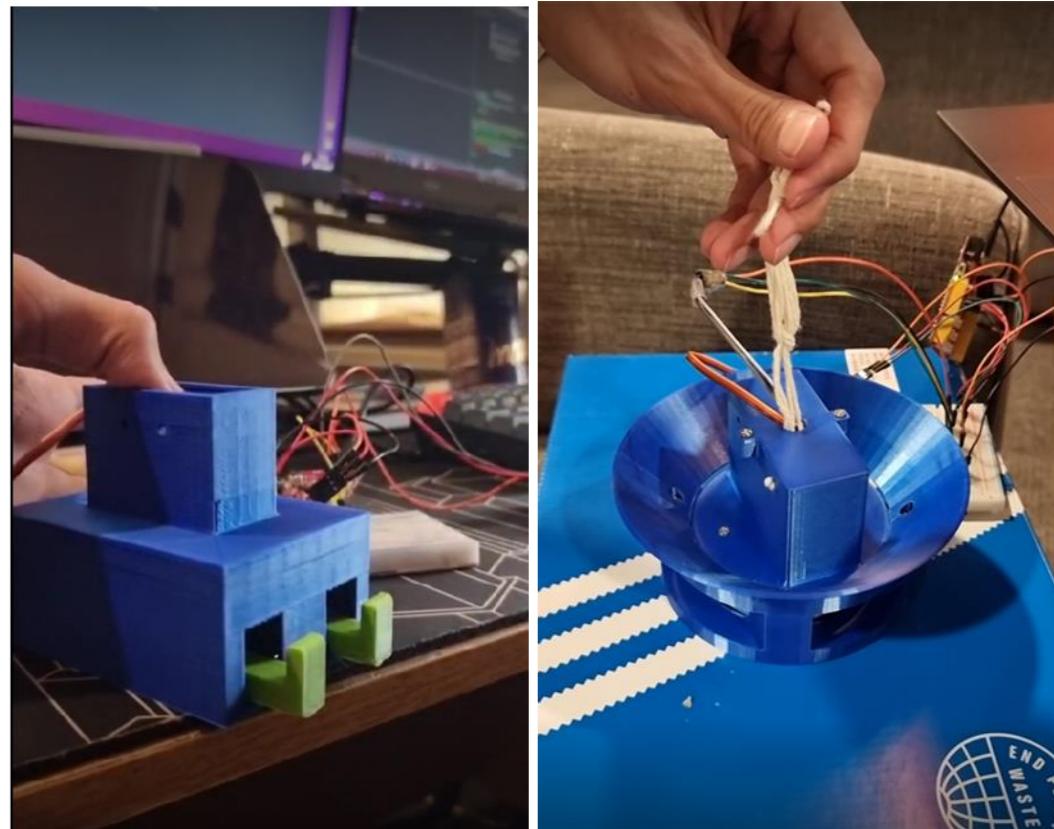


Figure 48: Preliminary tests conducted on old (left) and new (right) prototype designs

6.7.2 Iron Bird Test

With the final prototype design chosen, a full iron bird test was conducted. This allowed the simulation of a real-life scenario without flying the drone. The benefits of the iron bird test include reduced risks, as the propellers of the drone are not attached, rapid and repeated testing since there is no need to constantly recharge the drone batteries, and most importantly, it allows for a closer observation of the working mechanism of the whole system. Figure 49 shows the full system iron bird test that was conducted to observe the functionality of the prototype system when coupled with the avionic components. The camera module on the drone is able to pick up the markers attached to the package. The drone is then commanded to descend with the male component of the interface directly positioned above the female counterpart. The cone on the female component ensures the male component falls

into place and the magnets align the two components. The ToF sensor detects when both components are in contact and activates the slider mechanism to connect the interfaces. The package is now attached to the drone and is ready for delivery. This test proves the functionality of the final prototype design when incorporated with the drone.

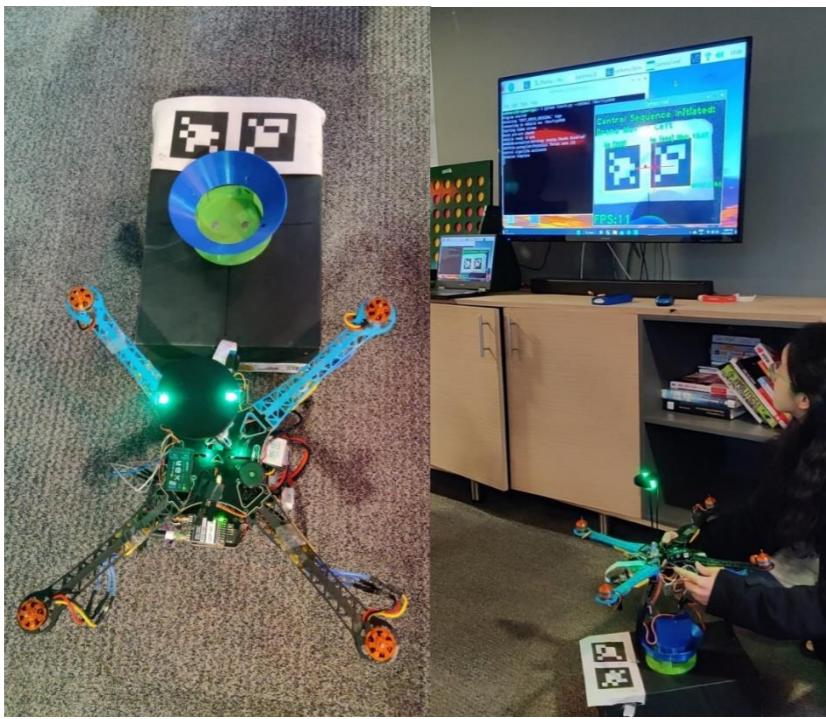


Figure 49: Iron bird test of the complete system

6.7.3 Magnet Test

When designing the final iteration of the interface mechanism, magnets were incorporated to assist in the interlocking of the male and female connections. One set of magnets is permanently attached to the top of the female interface, while another set is attached to the bottom of the male counterpart. During the attachment phase of the whole system, the magnets play the vital role of aligning the separate interfaces and forming the first connection between them. The alignment ensures the slider mechanism of the male interface can fit perfectly into the female interface for a secure connection.

Due to the nature of the magnet alignment mechanism, there is a limit imposed on the male interface for a successful connection. The limit in question is the horizontal angle difference on the XZ plane at which the male interface approaches the female interface in reference to the latter, hence an experiment was devised to determine this limit. The male and female interfaces were set up with a piece of paper with accurately marked angles inserted between the female interface and the package. The male interface was attached to the drone and hovered above the female interface. An ‘iron bird’ test set-up was used to reduce the risk and obtain a more accurate result as there was more control in determining the angle of approach.

Methodology

1. The drone with the male interface was suspended over the female interface at about 20cm.
2. The male interface was aligned at an angle of approach of 0°

3. The drone was slowly brought down at a steady pace all the while ensuring the angle of approach remained constant.
4. Once the male interface is rested on the female interface, the success of the connection was determined by primarily observing the position of the male interface relative to the female interface on the XY plane, followed by lifting the drone to observe if the female interface and the attached package was successfully connected.
5. Steps 1 to 4 were repeated 9 more times.
6. Steps 1 to 5 were repeated at increasing angles of approach until failure and all the results are recorded and tabled.

Examples of a successful connection and an unsuccessful attempt are displayed below.

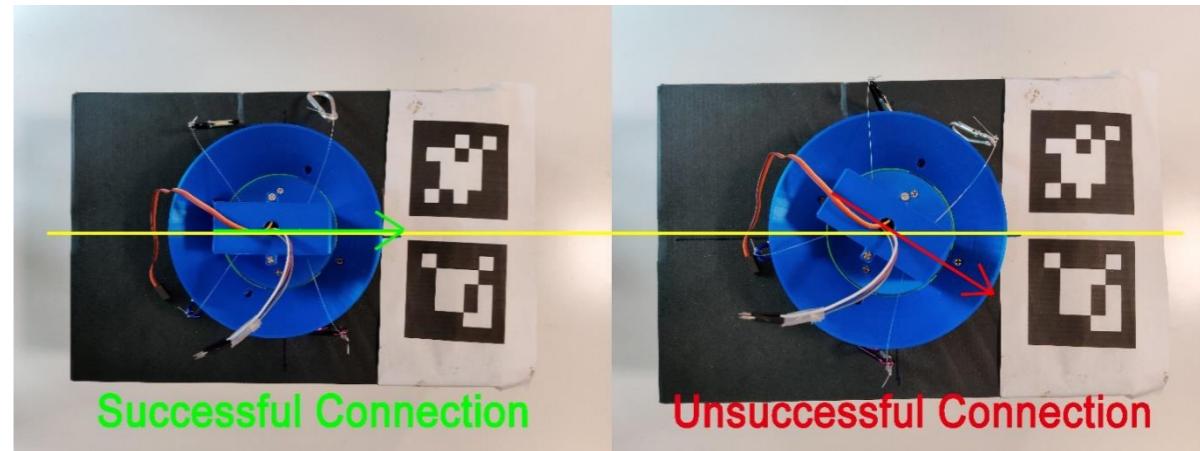


Figure 50: Example of successful and unsuccessful interface connection for angle of approach test

Results & Discussion

Table 13: Table of connection test for various horizontal angles of approach.

Connection Test	Horizontal angle (°)					
	0	20	30	35	40	45
1	/	/	/	/	/	X
2	/	/	/	/	/	X
3	/	/	/	/	X	X
4	/	/	/	/	/	X
5	/	/	/	/	X	X
6	/	/	/	/	X	X
7	/	/	/	X	X	X
8	/	/	/	/	/	X
9	/	/	/	/	X	X
10	/	/	/	/	X	X

Based on the table of results above, it can be observed that at angles of approach of 35° and below the connection is almost always successful. Whereas in 40° and above angles, the connection almost always fails. Hence, it can be reasonably concluded that to ensure a perfect connection success rate during autonomous loading, with the inclusion of a safety factor, the drone must approach the package at a horizontal angle difference of no more than 30°.

7 Final Design Proposal

7.1 Hardware



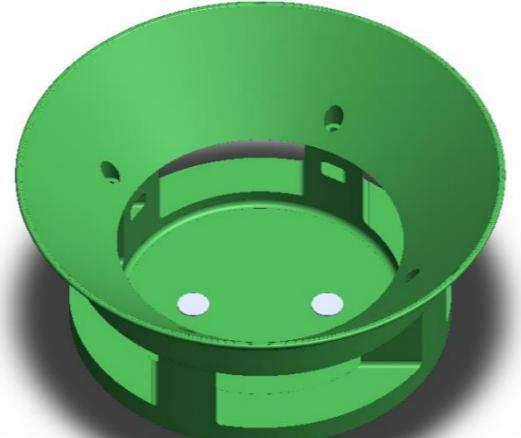
Two-piece Female Connector

The female connector is manufactured similarly to the male connector housing and consists of two pieces, the upper guiding cone and bottom slider locking piece. The upper guiding cone allows for a larger area where the male connector can drop into. The pockets in the lower part of the female connector keeps the rack slider in place and prevents the male and female connectors from disconnecting during flight.

Orientation Magnets

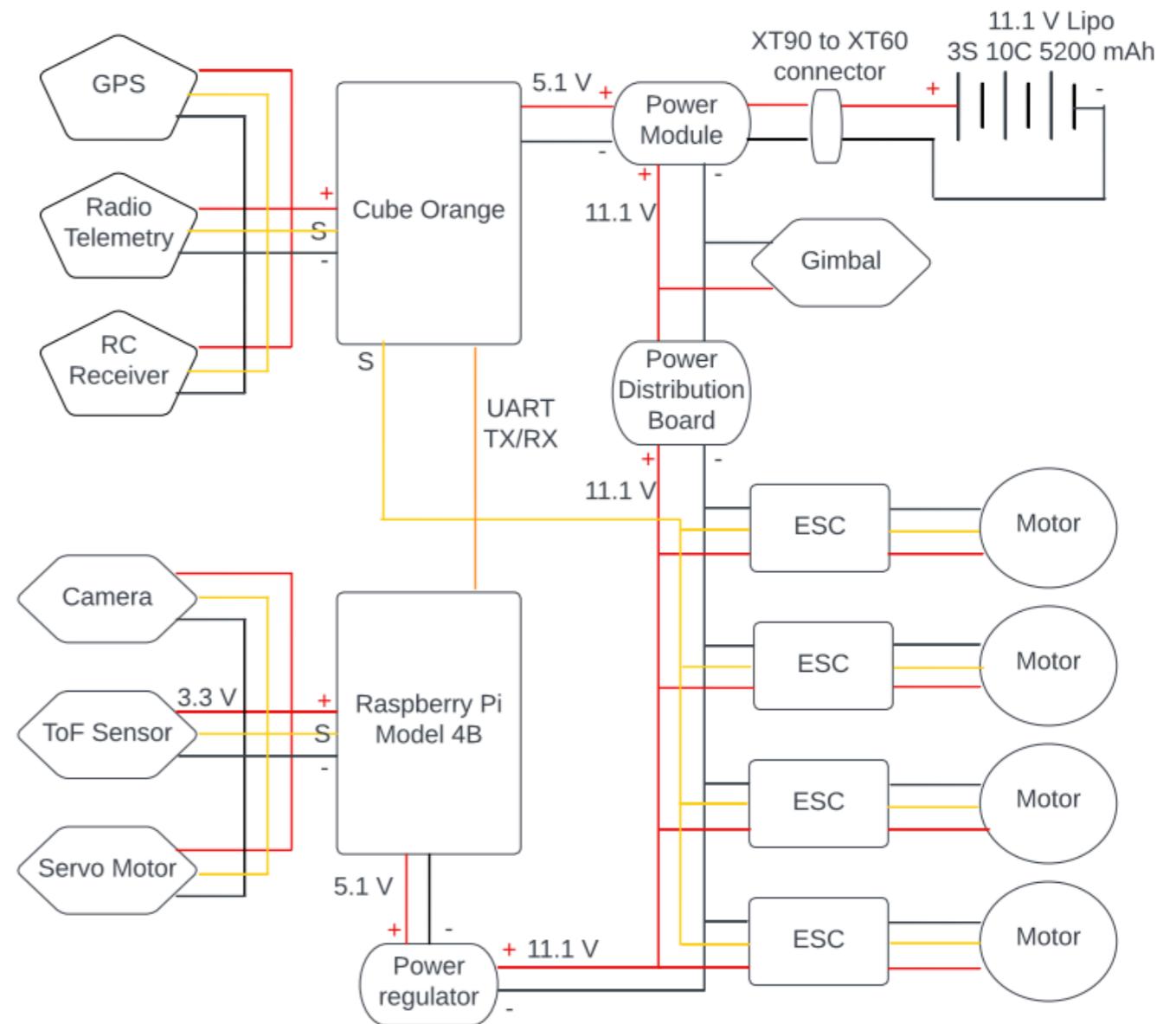
Two magnets are also fitted into the base of the female connector housing to align the male and female connectors allowing successful locking between the parts upon contact.

Multi Tether System	
Four-piece Male Connector Housing	Four polyamide threads are used to hold the male connector in place underneath the body of the drone. An optimized length is measured to prevent large oscillations during flight
Servo Motor	The male connector housing, which holds all the internal components, consists of four parts that are printed using additive manufacturing and are made of Acrylonitrile Butadiene (ABS) plastic
Slider Mechanism	A metal gear servo MG90S motor is used to power the slider mechanism upon successful contact between the male and female connectors
Range Sensor	A spur gear turns and slides out four rack gears from the male connector housing into the individual gear pockets in the lower piece of the female connector
Orientation Magnets	A Time-of-Flight sensor is used to measure distance & act as a verification of contact between the male and female connectors before the motor is instructed to run



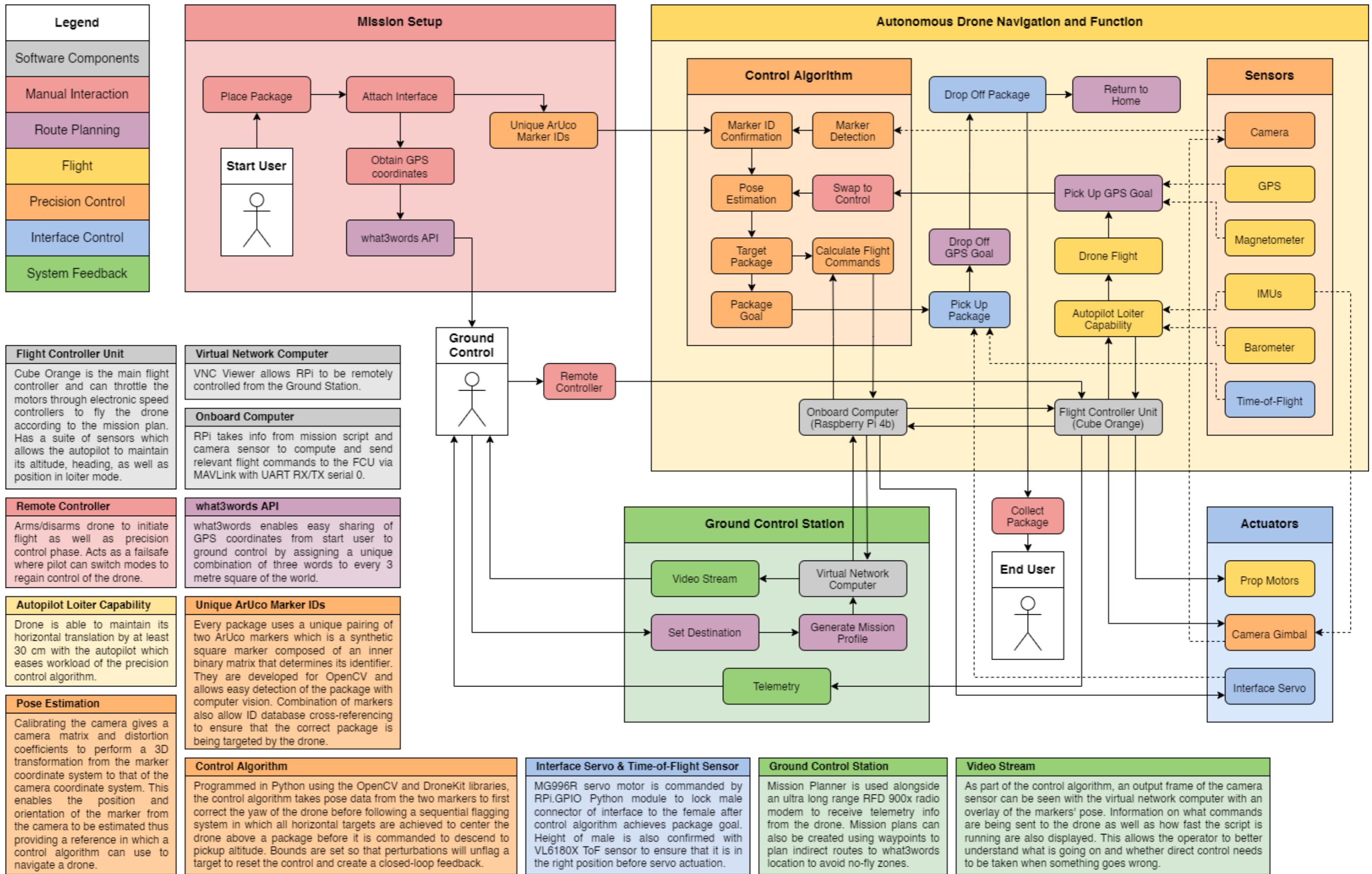
7.2 Avionics

GPS Module	Radio Telemetry & RC Receiver	Electronic Speed Controllers
Transmits GPS location to flight controller with a minimum of 10 connected satellites to ensure good loiter stability (i.e., drone does not translate horizontally in stick-free flight)	A RFD 900x Ultra Long Range Radio Modem transmits relevant flight data such as heading, altitude, etc. to ground station while FrSky X8R RC receiver allows flight controller to receive input from a pilot	Four NUOBESTY 30A ESCs allow the Cube Orange flight controller to control the speed of the motors by raising or lowering their voltages



UART TX/RX Signal	Time-of-Flight Sensor	Servo Motor
A Universal Asynchronous Receiver/Transmitter protocol is used for exchanging serial data between the flight controller and onboard computer. Drone health is sent from the FCU to the OBC and velocity vector commands derived from the CV control algorithm is sent vice versa	A Fermin: VL6180X ToF Distance Ranging Sensor is used by the onboard computer after completion of the control sequence to determine whether the male connector is at the right height for connection	A MG90S servo motor supplies 0.2 Nm of torque to drive the sliders of the locking mechanism into their bore holes to attach and detach the interface

7.3 Software Architecture



8 Project Review

8.1 Achievements

The main achievements of the final design were that a novel computer vision flight control algorithm and modular attachment system was developed for a multirotor to be capable of autonomously detecting, tracking, collecting, and delivering a companion package without human interaction. In its finished state, the system was able to theoretically demonstrate visual acquisition and approach to the package up to 2 metres, with the mechanical interface able to reliably pick up a package with a maximum centre- and angular-offset of 10 cm and 35° respectively in an iron bird configuration. Due to time constraints, it was not possible for the team to test the final design in flight. However, the team has managed to simultaneously demonstrate several key components in the limited number of flight tests they could conduct that serves as fundamental building blocks for the feasibility of this technology to be fully proven in flight. These includes:

- Horizontal stability of the drone in loiter mode is stable in good weather (no precipitation, < 5m/s windspeeds) and is capable of maintaining its translational position without a human pilot.
- The drone's battery is able to supply enough power to all existing flight components as well as additional components which provide the drone with onboard vision and processing capabilities.
- Remote access to onboard computer with VNC viewer allows drone to be programmed for missions and monitored during flight from a separate ground station.
- Autonomous flying of the drone with a Python script is possible with the DroneKit library in which velocity vector flight commands were issued to the FCU from the onboard computer.

8.2 Project Aims and Objectives

In terms of the project aims and objectives that were set out at the start of the design process, the team believes that the majority of these were successfully achieved as follow:

- The attachment system that serves as an interface between the drone and the package is fully modular, allowing it to be integrated on multirotors and packages of any size with minimal work.
- Components for the multirotor was internally sourced as much as possible with on-board vision and processing capabilities added onto it separately. This reduced the total cost and environmental impact as existing components could be reused for other drone-based projects.
- The computer vision control algorithm is capable of detecting fiducial markers from a distance and use their estimated poses to calculate and send the necessary velocity vector flight commands to the FCU to precisely control the drone's approach over a package.
- The control software was theoretically proven with simulations using a virtual robotics platform.

8.3 GDP Requirements

8.3.1 Innovation

Several innovative solutions were devised in order to solve the challenge that was posed to the team. This includes a novel computer vision flight control algorithm and the corresponding software architecture required to support it which allowed the drone to precisely approach a package on its own

with the usage of high fiducial markers. The design for the mechanical attachment system was also developed for modularity as the female and male connector parts can be easily rescaled for multirotors and packages of any size. Not only did this allow for rapid prototyping via additive manufacturing and ease of testing on a small system, but future teams could then integrate these existing systems onto larger drones that would have more commercial viability as compared to miniature UAVs.

8.3.2 Process

As a self-proposed project, this serves as the 1st iteration of the MORPHEUS auto drone package system in which feasibility of the task was to be determined by a technological demonstration. The aims and objectives for this GDP were therefore created based on this challenge, where the required creation of new systems to meet these goals were realised. For each design, a literature review was conducted to determine the solutions currently available before new ideas were generated and discussed within the team. Promising designs were then extensively tested either with a virtual robotics environment, FEA, iron bird setup, or actual flight before the optimal solution was implemented in the final design.

8.3.3 Communication and Project Sustainability

The team has been able to communicate the project with the use of diagrams and animations to allow a wide range of technical and non-technical audience to appreciate the complexity in designing a system that would allow a drone to collect a package on its own. These were included in this report as well as the video and presentation deliverables of the GDP so that information regarding the project could be communicated via different mediums. In terms of sustainability, the team sourced as many avionic components as they could within the University and purchased the rest of them from local companies where possible, to reduce the environmental impact of shipping. Components were off-the-shelf which meant that they could be reused for other drone-based projects. All hardware was additively manufactured in house using university facilities to reduce costs and material wastage.

8.4 Future Work

To continue the work done on the project and to realise the capability of the system in actual flight, any future team should take into account the following considerations to improve the functionality as well as reliability of both the drone and MORPHEUS package handling system:

- Further flight testing of current final design to diagnose and fix issues.
- Complete flight demonstration to assess capabilities and pinpoint areas for improvement.
- Profiling flight control algorithm to search for removable bottlenecks and streamline the code.
- Install onboard Wi-Fi module for long range flight as current system relies on an external hotspot shared with the ground station which limits the range in which remote viewing of RPi is possible.
- Develop a lightweight chassis to house all non-flight components for better presentation.
- Improve mechanical interface. Further iteration of both the male and female connector designs could be done to reduce their mass and cost, or to improve the reliability of connection.
- Implement hybrid computer vision technique to increase detection range and develop searching algorithms for cases in which package is not directly within the drone's vision.
- Integrate package handling system onto a larger drone which can support the additional weight of a fully loaded package to prove market viability.

9 Conclusion

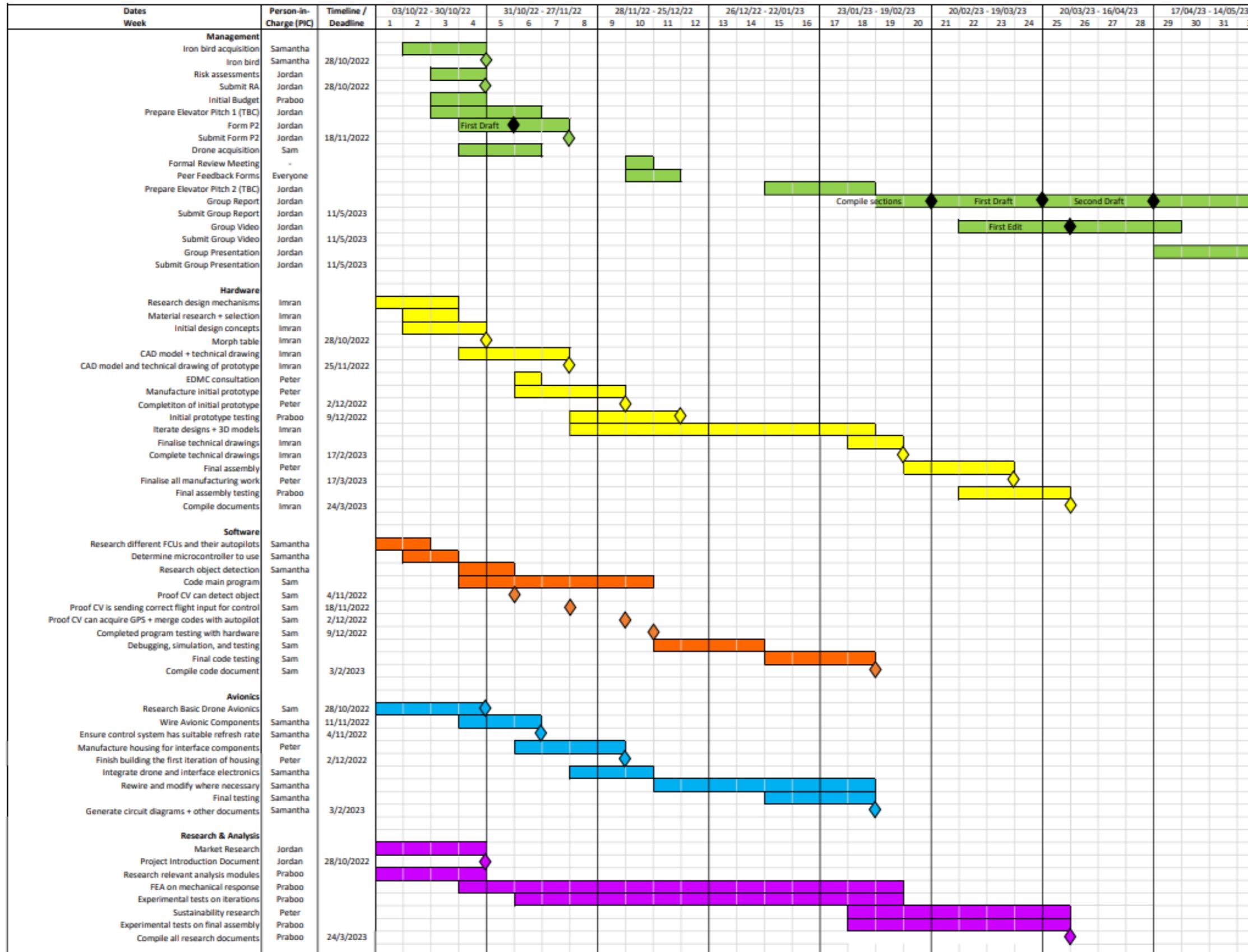
Throughout the duration of this GDP, the team has continually worked to research, design, manufacture, and test new systems that would ultimately allow a multirotor to autonomously pick up a package on its own. This includes the development of a sequential flight control algorithm utilising computer vision to enable a drone to detect, track, and precisely approach a companion parcel as well as a modular mechanical attachment system which acts as the interface between them. This has been achieved thanks to the iterative design process, additive manufacturing techniques, and robotics simulation platforms which allowed the team to rapidly test prototypes in a safe environment before modifying and refining them to produce the final proposed design. The current software architecture allows the drone to be remotely programmed, launched, and monitored from a ground station. The onboard computer is able to communicate with the flight controller unit to send specific commands to autonomously control the drone while in flight and the computer vision control algorithm has been theoretically proven to work using simulations and iron bird testing. A limited number of flight tests verified the airworthiness of the multirotor and the avionics subsystem's capability in supplying the required power to all components, but substantial future work is still required before a complete technological demonstration of the final system could be realised.

10 References

- [1] MarketsandMarkets Research, "Drone Package Delivery Market Forecast To 2030," 2022.
- [2] W. De Jager, "Why Drone Delivery Has Failed To Deliever On Its Promises," DroneXL, 2020. [Online]. Available: <https://dronexl.co/2022/02/21/drone-delivery-failed/>.
- [3] K. Laksham, "Unmanned aerial vehicle (drones) in public health: A SWOT analysis," *Journal of Family Medicine and Primary Care*, vol. 8, no. 2, pp. 342-346, 2019.
- [4] P. Jiyoong, K. Solhee and S. Kyo, "A Comparative Analysis of the Environmental Benefits of Drone-Based Delivery Services in Urban and Rural Areas," *Sustainability*, vol. 10, no. 3, p. 888, 2018.
- [5] Heven Drones, "Urban," [Online]. Available: <https://www.hevendrones.com/product/urban/>.
- [6] AURA-UAV, "AURA100," [Online].
- [7] Top Flight Technologies, "Airborg H8 10K," [Online]. Available: <https://www.topflighttech.com/products/airborg-h8-10k-with-top-flight-hybrid-power-system.html>.
- [8] A. C. LTD, "A distribution drones and a delivery system for loading a delivery article on an article loading section formed as an empty space in a base portion". Patent KR102100437B1, 2018.
- [9] E-drone, "NHS Logistics: Challenges and Opportunities," [Online]. Available: <https://www.e-drone.org/nhs-logistics>.
- [10] L. G. Shapiro and G. C. Stockman, Computer Vision, Prentice Halls, 2001.
- [11] Arm Limited, "What is computer vision - Arm," [Online]. Available: <https://www.arm.com/glossary/computer-vision>.
- [12] C. G. Grlj, N. Krzna and M. Pranji, "A Decade of UAV Docking Stations: A Brief Overview of Mobile and Fixed Landing Platforms," *Drones*, vol. 6, pp. 17-38, 2022.
- [13] A. Khazetdinov, A. Zakiev, E. Magid, M. Svinin and T. Tsoy, "Embedded ArUco: a novel approach for high precision UAV landing," in *International Siberian Conference on Control and Communications (SIBCON)*, 2021.
- [14] M. Bhargavapuri, A. K. Shastry, H. Sinha, S. R. Sahoo and M. Kothari, "Vision-based autonomous tracking and landing of a fully-actuated rotorcraft," *Control Engineering Practice*, vol. 89, pp. 113-129, 2019.
- [15] OpenCV, "Detection of ArUco Markers," 2022. [Online]. Available: https://docs.opencv.org/4.x/d5/dae/tutorial_aruco_detection.html.
- [16] APRIL Robotics Laboratory, "AprilTag," University of Michigan, 2010. [Online]. Available: <https://april.eecs.umich.edu/software/apriltag>.
- [17] J. Kallwies, B. Forkel and H.-J. Wuensche, "Determining and Improving the Localization Accuracy of AprilTag Detection," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, Paris, France, 2020.
- [18] OpenCV, "Camera Calibration," 2022. [Online]. Available: https://docs.opencv.org/4.x/dc/dbb/tutorial_py_calibration.html.
- [19] The University of Maine, "Loops and Flags," 2017. [Online]. Available: <http://umaine.edu/computingcoursematerials/wp-content/uploads/sites/511/2017/10/Loops-and-Flags.pdf>.
- [20] University of Kentucky College of Engineering, "flags-py.html," [Online]. Available: <https://www.cs.uky.edu/~keen/115/reading/flags-py.html>.
- [21] "MG90S Micro Servo Motor Datasheet," [Online]. Available: https://www.electronicoscaldas.com/datasheet/MG90S_Micro_Servo_Motor_Datasheet.pdf.
- [22] 3D Robotics, "About DroneKit," 2016. [Online]. Available: <https://dronekit-python.readthedocs.io/en/latest/about/overview.html>.
- [23] ArduPilot Dev Team, "MAVProxy," 2023. [Online]. Available: <https://ardupilot.org/mavproxy/index.html>.
- [24] A. Oakey, T. Waters, W. Zhu, P. G. Royall, T. Cherrett, P. Courtney, D. Majoe and N. Jelev, "Quantifying the Effects of Vibration on Medicines in Transit Caused by Fixed-Wing and Multi-Copter Drones," *Drones*, vol. 5, no. 1, p. 22, 2021.
- [25] J. Verbeke and S. Debruyne, "Vibration analysis of a UAV multirotor frame," in *Proceedings of the of ISMA 2016 International Conference on Noise and Vibration Engineering*, Leuven, Belgium, 2016.
- [26] H. T. Nguyen, V. T. Nguyen, T. Nguyen and D. Nguyen, "Vibration characteristics of quadrotor UAV: Computational and experimental study," *Aerospace Science and Technology*, vol. 120, p. 105861, 2020.
- [27] S. Wang, X. Sun and X. Zhu, "Experimental and numerical investigation of vibration characteristics of hexacopter UAV," *Journal of Intelligent & Robotic Systems*, vol. 92, no. 1, pp. 55-67, 2018.
- [28] E. Ciobanu, "How to Fix Drone Vibration Problems?," droneblog, [Online]. Available: <https://www.droneblog.com/drone-vibration/>.
- [29] Y. Zhang, Y. Liu, Z. Huang and H. Sun, "Effects of rotor imbalances on the vibration characteristics of quadrotor UAVs," *Journal of Aerospace Engineering*, vol. 34, no. 1, 2021.
- [30] C. Ge, K. Dunno, M. A. Singh, L. Yuan and L.-X. Lu, "Development of a Drone's Vibration, Shock, and Atmospheric Profiles," *Appl. Sci.*, vol. 11, no. 11, p. 5176, 2021.
- [31] E. L. de Angelis and F. Giulietti, "Stability and control issues of multirotor suspended load transportation: An analytical closed-form approach," *Aerospace Science and Technology*, vol. 135, p. 108201, 2023.
- [32] F. A. Goodarzi, D. Lee and T. Lee, "Geometric control of a quadrotor UAV transporting a payload connected via flexible cable," *International Journal of Control, Automation and Systems*, vol. 13, no. 6, pp. 1486-1498, 2015.
- [33] H. M. Omar, R. Akram, S. M. S. Mukras and A. A. Mahvouz, "Recent advances and challenges in controlling quadrotors with suspended loads," *Alexandria Engineering Journal*, vol. 63, pp. 253-270, 2023.
- [34] "SG90 Micro Servo Motor Datasheet," [Online]. Available: SG90 Servo Datasheet pdf - Micro Servo. Equivalent, Catalog (datasheetspdf.com).
- [35] Shawn, "Types of Distance Sensors and How to Select One?," 2020. [Online]. Available: <https://www.seeedstudio.com/blog/2019/12/23/distance-sensors-types-and-selection-guide/>.
- [36] The Pi Hut, "Fermion: VL6180X ToF Distance Ranging Sensor (Breakout)," 2023. [Online]. Available: <https://thepihut.com/collections/maker-store/products/fermion-vl6180x-tof-distance-ranging-sensor-breakout>.
- [37] D. Kim and e. al., "A Comprehensive Review of Time-of-Flight (ToF) Sensors," *Sensors (Basel)*, vol. 21, no. 2, p. 518, 2021.
- [38] J. Zhang, J. F. Campbell, D. C. S. Li and A. C. Hupman, "Energy consumption models for delivery drones: A comparison and assessment," *Transportation Research Part D: Transport and Environment*, vol. 90, p. 102668, 2021.
- [39] F. Neves-Moreira, B. Almada-Lobo, J.-F. Cordeau, L. Guimarães and R. Jans, "Solving a large multi-product production-routing problem with delivery time windows," *Omega*, vol. 86, pp. 154-172, 2019.
- [40] M. A. Figliozzi, "Carbon emissions reductions in last mile and grocery deliveries utilizing air and ground autonomous vehicles," *Transportation Research Part D: Transport and Environment*, vol. 85, p. 102443, 2020.
- [41] D. Shindell and C. J. Smith, "Climate and air-quality benefits of a realistic phase-out of fossil fuels," *Nature*, vol. 573, no. 7774, pp. 408-411, 2019.
- [42] Z. Meng, Y. Zhou, E. Y. Li, X. Peng and R. Qiu, "Environmental and economic impacts of drone-assisted truck delivery under the carbon market price," *Journal of Cleaner Production*, vol. 401, p. 136758, 2023.
- [43] J. Peters, M. Baumann, B. Zimmermann, J. Braun and M. Weil, "The environmental impact of Li-Ion batteries and the role of key parameters—A review," *Renew. Sustain. Energy Rev.*, vol. 67, pp. 491-506, 2017.
- [44] J. Yang, F. Gu, J. Guo and B. Chen, "Comparative Life Cycle Assessment of Mobile Power Banks with Lithium-Ion Battery and Lithium-Ion Polymer Battery," *Sustainability*, vol. 11, no. 19, p. 5148, 2019.
- [45] N. Espinosa, R. Garcíavalverde and F. Krebs, "Life-cycle analysis of product integrated polymer solar cells," *Energy Environ. Sci.*, vol. 4, pp. 1547-1557, 2011.
- [46] S. Liang, W. Yan, X. Wu, Y. Zhang, Y. Zhu and H. Wang, "Gel polymer electrolytes for lithium-ion batteries: Fabrication, characterization and performance," *Solid State Ion.*, vol. 318, pp. 2-18, 2017.
- [47] A. Paziresh, A. H. Nikseresht and H. Moradi, "Wing-Body and Vertical Tail Interference Effects on Downwash Rate of the Horizontal Tail in Subsonic Flow," *Journal of Aerospace Engineering*, vol. 30, no. 4, p. 04017001, 2017.

Appendix

Appendix A – Gantt Chart



Appendix B – Budget

Item	Cost (£)	Subtotal (£)
Direct Cost		
Servo motor	12.59	
ToF Sensor	6.60	
Heat Shrink Tube	6.99	
Electronic Speed Controller	27.96	
Ball Bearing	3.99	
Counterweight	11.06	
Brushless Motors	161.79	
Raspberry Pi	70.80	
Memory card	8.29	
Ribbon cable	2.49	
Gimbal	50.75	
Drone Batteries	40.99	
GPS mount	12.99	
Battery adapter	6.07	
Carabiner	2.49	
Velcro	2.99	
Propellers	25.57	459.00
Indirect Cost		
Fasteners	7.94	
Voltage checker	10.99	
Retractable cord	18.50	
Ground hooks	2.70	
Hazard tape	13.50	
Laser	6.64	
Anemometer	28.49	
Hazard tape	11.99	
Tether	4.00	
Ground spike	16.99	121.74
Variable Cost		
3D Printing	150.00	150.00
Total Cost		730.74
Sunk Cost		
(Inherited Drone Components)		
S500 Quadcopter Frame	28.95	
The Cube Orange	350.00	
Here3 GPS Module	175.00	
RFD 900x Modem	280.00	
RC Controller	58.00	
FrSky X8R	21.25	913.20

Appendix C – List of Commercial Drone Platforms Mentioned in Section 5

1. Aergility - <https://www.aergility.com>
2. Aura UAV - <https://aura-uav.com> (No longer available)
3. Velos Rotors - <https://velos-rotors.com>
4. Draganfly - <https://draganfly.com/products/heavy-lift>
5. Avidrone Aerospace - <https://www.avidroneaerospace.com/>
6. Elroy Air - <https://elroyair.com/>