Design Report
FEEG6013 Group Design Project

49
Integration of Drone and Rover
Integration of a Drone with a Ground Rover for Navigation and Docking

Project Summary:

Our project focuses on the development of Hub for Aerial Navigation and Ground-based Autonomous Reconnaissance (H.A.N.G.A.R), a docking platform serving as the link between unmanned aerial vehicles (UAVs) and unmanned ground vehicles (UGVs). Over the next decade, forecasts suggest a growing adoption of UAVs, including drones, notably in search and rescue, and drone-based aid applications emphasising on minimal human interaction post-pandemic.

Designed to allow the docking platform to autonomously detect, track and secure a drone within its range while maintaining stability, this system addresses one of the most common issues of limited flight duration due to battery constraints. The platform potentially enables secured UAV battery charging to take place while connected to the UGV. The mechanical design of the docking platform prioritises strength and stability, featuring a full 360° rotation capability and dynamic horizontal and vertical extensions to reach drones. Combined with a concurrent control algorithm utilizing computer-vision-based tracking and platform levelling capabilities, the system seamlessly performs tasks to ensure the drone lands smoothly on the platform.

All subsystems, from mechanical design to software development integrates seamlessly, with validation in simulated environments preceding real-world manual flight tests at Boldrewood campus. The system achieved several design requirements of accurately detecting, tracking and securing a handheld drone, replicating real-life landing approach, with future efforts focused on achieving autonomous flight, marking a potential breakthrough in drone operations.

Group Members:

ID Number	Name	ID Number	Name
31426042	Ken Ooi Mo Herng	31145817	Toh Zheng Aun
30640458	Ung Shern Khai	32248873	Gai Zhe Wong
32492634	Jalen Tan Khai Zhen		

Primary Supervisor: Dr Mohamed Torbati

Co-Supervisors: Dr Mohammad Soorati
Dr Ayodeji Abioye

Submitted on: 9/5/2024

Table of Contents

Academic Integrity

We, members of the 'Integration of Drone and Rover' 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 work.

Acknowledgements

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

- Dr Mohamed Torbati, for taking on this self-proposed project and supporting us at every step of the way throughout the project.
- Dr Mohammad Soorati, for his expert advice and support of the project from the perspective of the ECS department.
- Dr Ayodeji Abioye, for his unwavering support and guidance for the project, notably for his instruction of UAV piloting and facilitating flight tests.
- Dr Peter Glynn-Jones, for providing valuable feedback during the project review meeting.
- Soton UAV, for providing advice and components at the initial phase of the project.
- Health and Safety Flight Test team, for test flights approval at Boldrewood Campus Design Studio
- EDMC technicians, for his help and advice on the manufacturing of the docking mechanism.
- To our families and friends, for their extended support throughout this journey.

1. Introduction

This GDP focuses on developing a docking platform called H.A.N.G.A.R, serving as the mechanical interface between an unmanned aerial vehicle (UAV) and unmanned ground vehicle (UGV). UAVs, including drones, are small aircraft flown without a human pilot on board, offering flexibility in take-off and landing, even in challenging terrain. In recent years, UAVs have garnered attention for their versatility in undertaking various tasks, especially in hazardous or inaccessible locations [1]. Statistics from [2] indicate significant growth in the global drone services market, projected to reach a Compound Annual Growth Rate (CAGR) of around 28.5% during the forecast period of 2024-2030.

Specific applications such as search and rescue are expected to witness a CAGR progress of 15.2% over the next decade [1, 2]. Furthermore, with post-pandemic scenarios, particularly within the Healthcare and Social Assistance sectors, are forecasted to increase drone adoption, primarily due to the upsurge of drone-based aid for minimal human interaction [3]. These trends underscore the widespread adoption of drones. However, a common challenge faced by drones is their limited flight duration due to onboard battery capacity, constraining operational time and efficiency [4].

To address this challenge, the concept of a mobile docking system attached to a moving UGV, or rover, for secured landing is explored – the first step towards mid-operation recharging of UAV batteries. Consequently, in support of the heterogenous swarm robot research conducted by the Electronics and Computer Science (ECS) department of University of Southampton (UoS), our project emphasises the overall development of the newly proposed docking platform to ensure the safe and secure landing, mitigating potential damages or accidents.

As such, our docking platform's primary function is to accurately detect, track, and secure the UAV onto the platform autonomously. The team addressed various aspects of H.A.N.G.A.R's subsystems, including: (1) Designing and manufacturing a mechanically stable and robust docking platform capable of movement in three core degrees of freedom (360° rotation, extension, and tilting) covering a significant area despite its compact size, (2) Developing a computer vision-based UAV detection and control algorithm involving platform levelling and tracking functionalities to ensure seamless landing and securing of the UAV.

These subsystems were then constructed, tested, and validated through FEA and Simulink simulations, coupled with real-world flights. The final integration of all subsystems, powered by a single battery resulted in the development of a fully operational docking platform which enabled UAV detection, tracking, and securing while maintaining platform stability, which will be outlined in this report.

2. Design Brief

2.1. Project Aims & Objectives

In support of the ECS department of University of Southampton, our project aims to explore the integration of a drone with a ground rover to enhance its navigation capabilities. This will also be used to build the initial phase of a heterogenous swarm robotic research, which will include multiple, different robots for various applications, such as search and rescue, surveillance etc.

The primary aim or focus of our project is to develop a reliable docking platform, that will be secured onto a ground rover, to accurately detect, track and secure a flying drone under predetermined conditions. The team has set several objectives to ensure that the aim would be successfully achieved:

1. **Avionics Objectives**
    a. Acquire and assemble the drone, with excellent knowledge over the functionalities of manually controlling the drone for testing purposes.
    b. Integrate autonomous flight capabilities with the drone.

2. **Hardware Objectives**
    a. Design a versatile docking platform capable of accommodating imprecise drone landings and effectively securing the drone in position post-landing.
    b. Develop and build the docking platform, considering the feasibility, manufacturing requirements and suitable material associated for each sub-sections and individual part.

3. **Software Objectives:**
    a. Install computer vision packages on given onboard computers.
    b. Develop a computer vision algorithm for the vision-based tracking system.
    c. Develop a control algorithm for the docking platform to execute platform levelling and, marker tracking and drone capturing functionalities.

4. **Electronics Objectives:**
    a. Research potential electronic components to be purchased based on the suitability regarding its specifications and functionalities.
    b. Construct a concise budget plan over the purchasing of electronic components.

2.2. Project Assumptions

1. Along with the project aims and objectives, several key assumptions were drawn out to smoothen the project, ensuring timely completion before resources were allocated to enhance system robustness or cost-effectiveness. The detection range is capped at a maximum limit of two meters. Any detections beyond this range may necessitate additional improvements to the detection system.
2. The docking platform is designed to integrate the ground rover and drone provided to the group, adhering to specific size specifications and dimensions.
3. The docking platform will exclusively function under a controlled and favorable environmental condition, without any obstacles obstructing the drone and rover.

2.3. Resources

An initial GDP budget allocation of £850 was given to the team, along with various components from the ECS department as stated Table 1 below.

Table 1. Components list provided by ECS department of University of Southampton.

Jetson Nano Waveshare Developer Kit Bundle with IMX219-77 Camera			
x1 Jetson Nano	x1 IMX219-77 Camera	x1 64GB Micro Card	x1 Card Reader
x1 Jumper Cap	x1 15-pin FFC	x1 Power Adapter	
UAV Platform (Drone)			
x1 Frame Kit (Frame, 4 Motors, ESC, 4 Propellers)		x1 Pixhawk Cube Orange Flight Controller	
x1 Cube Pilot GNSS Module		x1 4S 5000mAH LiPo battery	
UGV (Rover)			
x1 UGV Kit (JSUMO ATLAS All Terrain Explorer Robot 4x4, including Arduino Board)			

The assembly of the docking platform took place at the design workshop within the University Engineering faculty, and flight tests were conducted in the Boldrewood Design Studio. The team had access to the University's Engineering Design and Manufacturing Centre (EDMC), student workshop, and electronics workshop. Additionally, each student was provided with specific manufacturing hours and material usage allowances.

2.4. Team Roles

The team roles were allocated based on the specialized skills of each group member. The team was divided into four principal departments: (1) Avionics, (2) Hardware, (3) Software, and (4) Electronics, overseen by the management department, which involves setting up communication channels, scheduling meetings, setting deadlines, and ensuring the timely delivery of project milestones. The hardware department is responsible for developing design concepts and manufacturing processes. The software department focuses on constructing the algorithms behind the prototype, including the development of logical frameworks and system architectures for the project.

Each department has its own lead but crossover between departments has helped our group to achieve greater teamwork, synergy, and better understanding between departments. Each member's role can be seen in Figure 1 below.

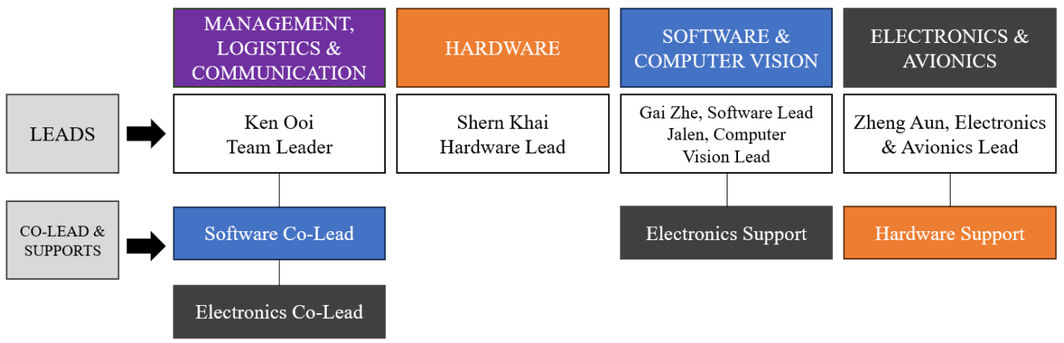


Figure 1. Team structures and roles.

2.5. Design Process

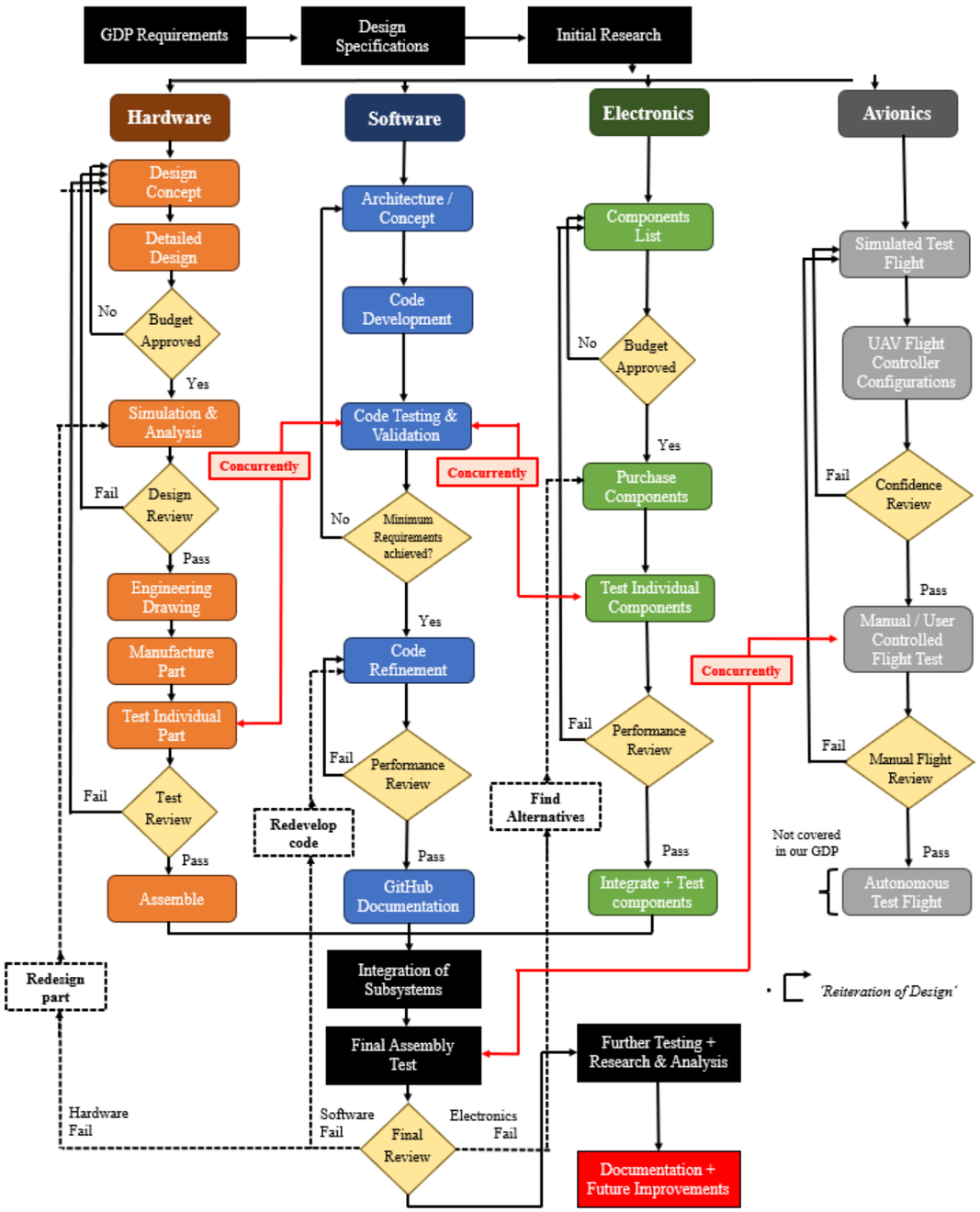


Figure 2. Design process flow chart.

A structured and clear design process was outlined to identify key milestones and critical junctures in the timeline. The design process to conceptualise and develop a docking platform has been presented in Figure 2, which was divided into four subsystems, each adhering to sequential stages to achieve a reliable docking platform. The double diamond design process, as seen in Figure 3 has also been adopted to ensure a smooth and systematic design process. In the discover stage, the team came across various unique platform design and software architectures based off literature as well as other internet resources, where the design concepts were generated and defined based on our proposed requirements, assessed, prototyped, tested and reiterated before moving to the final delivery of the project. Anticipated delays and potential issues concerning the overall timeline were duly considered during the project planning phase, as documented in Appendix A.

Figure 3. Double diamond design process chart.

(a) GDP Requirements & Design Specifications

As this project is self-proposed, the initial stage of the design process involves understanding our GDP requirements and objectives. This step guides the team in setting design specifications for all subsystems, including hardware, software, electronics, and avionics.

(b) Initial Research

This stage includes conducting a comprehensive literature review to identify similar work within the integration of UAV and UGV, with a specific focus on developing a docking mechanism. Emphasis is placed on examining potential methodologies for detection, prioritising reliability, and accuracy, as well as designing the mechanism and control algorithms crucial for project success.

(c) Implementation

Each department worked concurrently throughout the project, with careful synchronisation to ensure timely delivery and allow for design refinements.

a. Avionics

The initial step involves configuring the drone and mastering user-controlled flight through simulated practice. While not explicitly outlined in Figure 2, there are correlations with hardware and software subsystems, such as designing and manufacturing a battery holder. Note that autonomous flight study is omitted from the GDP due to reasons detailed in the section 4.3.3.

b. Hardware

The initial phase involves brainstorming various mechanism concepts and analysing their advantages and limitations for effective comparison. This is followed by SolidWorks CAD modelling and FEA simulations to refine designs before manufacturing individual parts. Testing and assembly were conducted to evaluate the design and explore potential enhancements or alternatives.

c. Software

Following the construction of a high-level architecture overview, code development spans across various areas such as autonomous flight via ArduPilot, computer vision detection, electronic actuators, and platform leveling control algorithms. Concurrent development by the software team ensured sufficient time for code refinement to enhance performance. Additionally, all code is documented and uploaded to GitHub.

d. Electronics

Constructed a component list, tailored to specific purposes, such as rotation, extension etc is essential for budget management. Purchased and tested individual components occur simultaneously to facilitate timely integration and testing of all components.

(d) Integration of Subsystems

The next critical stage of the design process involves integrating findings from each subsystem to function as a unified docking mechanism. Although thorough testing of individual components can smoothen this integration process, time is allocated to address unforeseen issues.

(e) Final Review – Design Iteration

The final stage involves the final assembly testing of the docking mechanism and analysis of its performance. This allowed the team to carry out iterations and modifications to the design, code, components to achieve all the aim and objectives established at the beginning of the project.

3. Design Specifications

3.1. System Requirements

To achieve the project's predefined aims and objectives outlined earlier, the following system requirements have been set for each subsystem to guide the iterative design process. These requirements have been set in accordance with the standard UGV and UAV requirements and specifications. Consequently, any references to "drone" in subsequent sections of the report will denote UAV, while "rover will refer to UGV, for a broader context.

1. Avionics Requirements

- The total weight of the drone, including all components and battery must be below 2.5 kg for safe and stable flight.
- All the components are securely attached and balanced on the drone so that no additional torque is applied for maximum stability and minimal disturbance.
- The power source must be sufficient to power all electronics and sensors on the drone safely with no risk of electrical shock or fire.
- The flight controller should be able to receive flight commands from a ground computer station or a remote control via telemetry. In emergencies, the drone must have a fail-safe feature to ensure it lands safely with no damage.

2. Hardware Requirements

- Able to safely secure a landing drone in position, on top of a ground rover.
- Flexible to accommodate different landing conditions, including:
  - Moving platform
  - Inclined or uneven ground
  - Wind disturbance
- Able to tolerate ±10 cm misalignment and offset from the centre during landing.
- The setup must be stable during and after the landing of the drone.
- Strength to withstand a load (drone) of 2.5 kg plus any additional weight introduced by other components on the platform.
- Sufficient force/torque output from motors/actuators involved according to expected load (about 6 Nm with safety factor).
- Flexible to accommodate a wide range of drones.

3. Software Requirements

- Autonomous detection, tracking and securing a drone on a moving rover.
- Have failsafe scripts to gracefully abort the program, turning off all the sensors and actuators.
- The complete algorithm responsible for autonomous detection and tracking must have a response time of 100 ms or below to be practical in real life applications.
- Have controllers to generate appropriate control signals to actuators with the aid of sensor inputs as feedback.

4. Electronics Requirements

- All electronic components for the docking platform must fit within or on the rover and not impact its movement.
- On-board power source can reliably power all components during operation for 20 minutes.
- Preventive measures are incorporated to mitigate risk of damage to electronic components during accidents.
- Sensors with relatively accurate output for current application:
  - Angle measurement
  - Distance measurement between 2 cm to 300 cm.
  - Compatible with onboard computer(s) for computer vision.

4. Avionics

The primary objective of the avionics subsystem is to enable autonomous flight and marker tracking for autonomous landing. This was achieved by utilising a Jetson Nano as the on-board computer and a camera to provide visual input. The team was provided with a pre-assembled drone and electronic components from Dr. Soorati and Dr. Ayodeji as depicted in Figure 4 and Figure 5. The enclosed areas in Figure 4 are identified for team-driven improvements and elaborated. Additional components such as brackets were added to ensure the component's security and stability during flight. Prior to any autonomous test, manual flight tests were conducted with docking platform prototypes to generate additional ideas. Simulations of autonomous flight utilising the Jetson Nano was conducted to replicate real-life flying scenarios. However, due to the restrictions of the pre-installed software in the flight controller, the project was unable to proceed with the autonomous flight which is further discussed in Section 4.3.3.

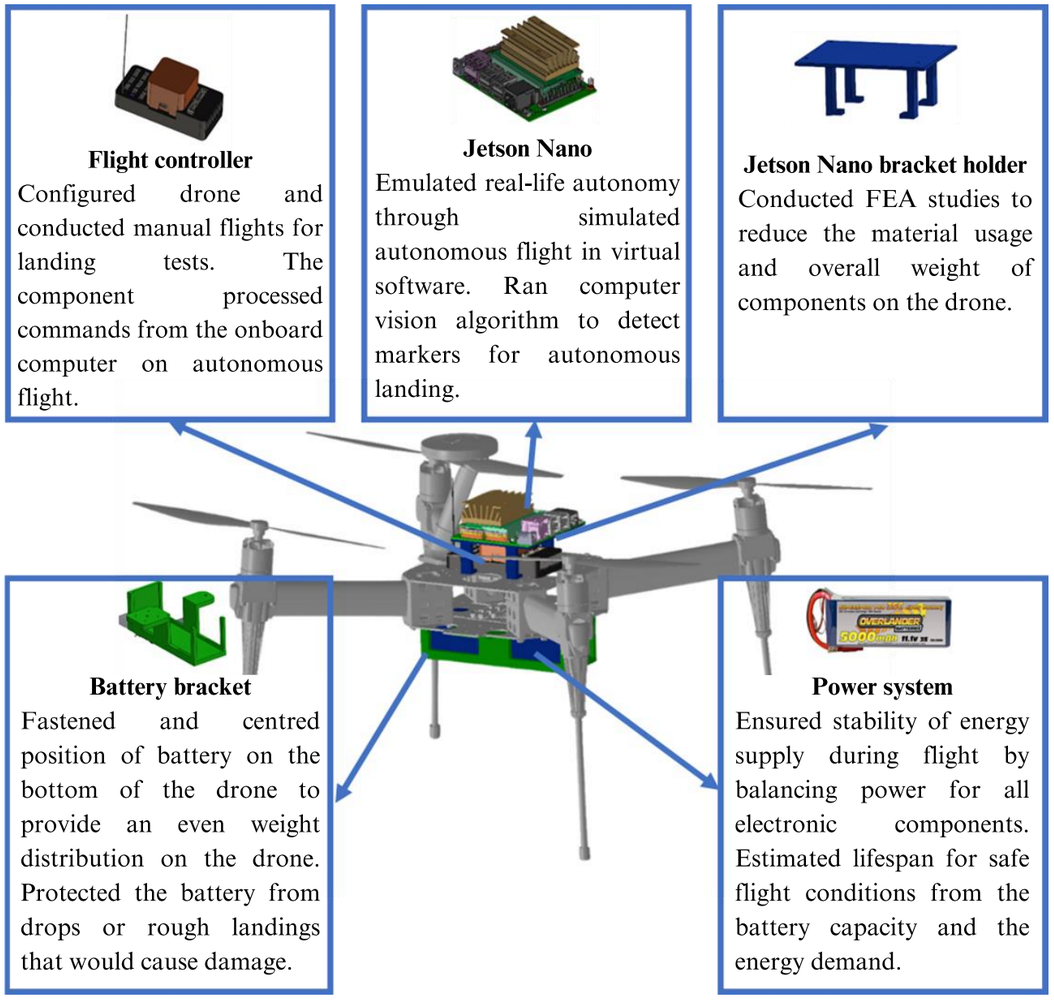


Figure 4. Overview of drone assembly

4.1 Drone Design & Electronics

The first section of focus would be the power balance and mechanical design of the drone. The final assembly of all components on the drone is shown in Figure 5. Since the Jetson Nano bracket would not bear stress, they were manufactured with 3D printing to reduce the overall weight of the payload. The weight is also reduced by conducting FEA on the bracket as shown in **Error! Reference source not found.**, to reduce the amount of material. The maximum stress is 0.044 MPa in Figure 6 (a) and the maximum displacement is 0.026 mm in Figure 6(b) while under load from the weight of the Jetson Nano. The weight of the bracket went down from 60 g to 28.6 g with 52% reduction in total weight.

Figure 5. Physical full drone assembly

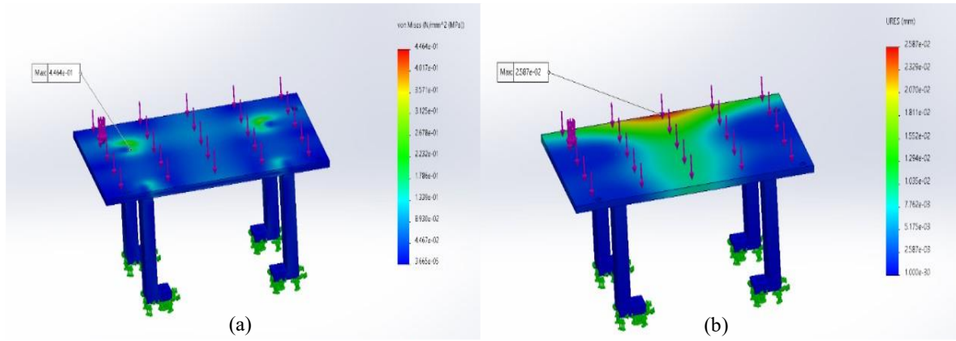


Figure 6. FEA simulation of Jetson Nano holder to observe (a) stress and (b) displacement.

Additionally, the power requirements of each component were used to ensure that the battery on the drone could power the Jetson Nano, flight controllers, and four motors at the same time without affecting its stability. The specifications of all electronic components on the drone are summarised in Table 2 and the battery lifetime is estimated as 15 minutes of safe flight operations.

Table 2. List of main components on the drone.

Component and illustration	Specification	Comments
NVIDIA Jetson Nano [5] 	• Voltage Input: 5.0 V • Current Input: 1 A – 2 A • GPU: 128-core NVIDIA Maxwell™ • CPU: Quad-core ARM® Cortex® A57 • Memory: 4 GB 64-Bit LPDDR4	Readily available from supervisors and could be lent with no additional cost. The specifications were high enough to run computer vision and autonomous flight simultaneously.
Cube Orange Flight Controller [6] 	• Voltage Input: 4.1V - 5.7V • Current Input: 2.5A • Processor: 32bit ARM® STM32H753 Cortex®-M7	Readily available, setup with drone to allow internal processing of flight mechanics and autonomous flight through ArduPilot.
Hexsoon 20A ESC [7] 	• Voltage Input: 9 V – 26 V DC • Voltage Output: 5 V & 12 V • Current Output: 6x 25 A – 40 A	Already setup on drone and distributed power to all electronic components on the drone, including Jetson Nano.
HS2216 KV:920 Motor [8] 	• Max Voltage: 15 V • Max Current: 18 A • Kv: 920 rpm/v	Already setup on drone and provided sufficient thrust for lift off and stable flight.
LiPo Battery [9] 	• Capacity: 5000 mAh • Voltage: 14.8 V • Constant Discharge: 35 C • Burst Discharge: 65 C	Readily available, could power all components and had sufficient capacity to operate for 15 minutes safely.

4.2 User-Controlled Flight

Multiple tests were conducted with the drone while being piloted manually. This included testing initial prototypes of docking platform as shown in Figure 7. The physical tests resulted in identifying multiple fundamental issues with the design that resulted in a revamp of the concept. This ultimately also led to the realisation that a passive docking platform would not satisfy the project requirements and an active system would be required instead, which will be elaborated in the hardware section 5.3.1.

Figure 7. Docking platform test

Furthermore, testing of the camera vision was also conducted with all components attached to the drone and manually piloted as shown in

Figure 8. Results successfully show that the detection algorithm works up to 2.5 m in height and can detect as close to 0.1 m. The tests also proves that the battery can power all components on the drone without affecting its stability. The ability to conduct tests utilising the drone in manual flight provided insight to generate new concepts that was adapted to our designs.

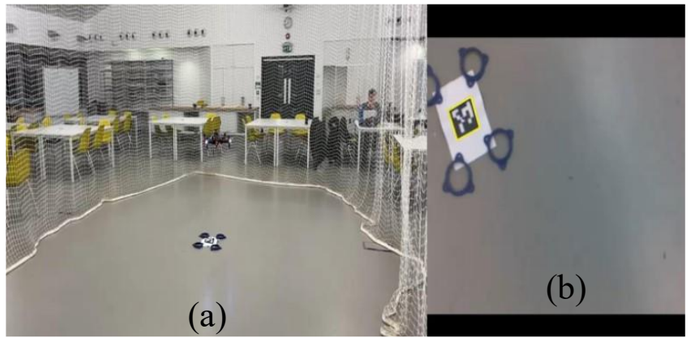


Figure 8. Experiment of manual flight tests for (a) computer vision and (b) computer vision test results.

4.3 Autonomous Flight

4.3.1 Background of Drone Programming

The first step of autonomous flight was to understand the world of drone programming, which was a relatively novel concept for this team of mechanical engineers. This section documents the pertinent hardware and software of modern drones.

4.3.1.1 Flight Controller (FC) Firmware

The FC and its firmware on a drone are analogous to its brain, dictating how the drone behaves. Notable open-source firmware includes ArduPilot, PX4 and BetaFlight. The provided drone uses the ArduPilot firmware deployed on a Cube Orange FC as shown in Table 2. The primary functionalities of a FC are:

- **Drone stability**: By default, drones are inherently unstable and require active stabilization. FC firmware abstracts the inner control loop responsible for stability using available sensor data such as accelerometers and gyroscopes. As a result, users can focus on higher-level functionalities, such as specifying waypoints to be followed by the drone.
- **Interpreting human inputs**: A pilot can operate the drone with external hardware, such as a remote controller with joysticks. These signals are translated into commands by the FC.

- **Communicate with other devices via MAVLink**: Communication with and/or between drones are conventionally encoded following the MAVLink protocol. The types of messages range from acknowledging the current state of the drone (e.g., arm, disarm, altitude) to the ability to send custom commands to the drones, such as commanding it to arm and ascend to a specified altitude.

The last functionality is the most important aspect of realising autonomous flight. A companion computer can be added to the drone and communicates with the FC. The state of the drone, such as its altitude and pose, can then be extracted from the FC by the companion computer to generate an appropriate MAVLink command to steer the drone autonomously.

4.3.1.2 Ground Control Station (GCS)

GCSs are interfaces to the drone that serves as a mean for users to communicate with the drone. This can refer to the software that presents data with an interface, either in the form of a graphical user interface (GUI) or a command line interface (CLI). It can also refer to the physical hardware setup of the GCS, which can be as simple as a laptop to a sophisticated dedicated console.

With a GCS, a bi-directional communication pathway is established between the user and the drone. Real-time data from the drone can be monitored whilst commands can be given to operate the drone. An example includes configuring a mission where a drone moves from point A to point B, which is communicated to the drone via radio telemetry or the internet.

4.3.1.3 Companion Computer & MAVLink Libraries

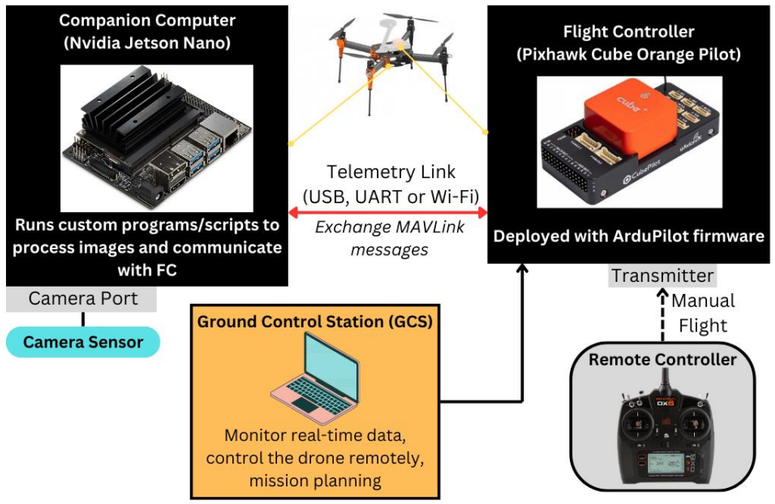


Figure 9. Hardware architecture of drone programming.

Although automated flight can be performed by GCSs, available features will be limited to simpler messages and commands such as arm, take off and hover. To integrate more complex tasks with autonomous flight, like commanding the drone to move towards a marker with computer vision, a companion computer is usually used in addition to the FC to off-load this burden. In the context of the project, a camera sensor needs to be connected to the companion computer to process images to calculate its current offset from the FC. Then, the companion computer can send a MAVLink command to the FC to manoeuvre the drone as desired. The Nvidia Jetson Nano computer was the perfect choice for the

companion computer, since it was made readily available by Dr. Soorati and Dr. Ayodeji, whilst simultaneously satisfying both compact and performant criteria due to its dedicated graphics processing unit (GPU). For the companion computer to send MAVLink messages to the FC, MAVLink libraries are used. There are several options depending on the programming language and framework, such as DroneKit, Pymavlink, MAVSDK and MAVROS. The holistic hardware architecture of drone programming is depicted in **Error! Reference source not found.**.

4.3.2 Drone Simulation

Before deploying programs or scripts onto the companion computer to enable autonomous flight, it is imperative to simulate its behaviour in software due to safety and monetary concerns associated with drone malfunctions. Python scripts for marker detection were developed, which is further discussed in Section 6.2. In addition, python scripts that integrated MAVLink libraries were also written, capable of sending MAVLink commands to the drone based on its state. Such a simulation is done via ArduPilot's Software-in-the-loop (SITL), which runs ArduPilot completely on a drone frame. By initializing a SITL drone, other programs can connect to the simulated drone, such as a GCS software or custom programs that are able to exchange MAVLink messages after establishing a telemetry link (e.g., Wi-Fi).

By default, the simulated drone exists in the form of a CLI. Therefore, an external simulator was pursued to gain access to high-fidelity physics and a GUI for visual feedback of drone performance. Amongst many simulators, choices were narrowed down to between CopelliaSim and Gazebo, where the latter was chosen due to the supervisors' familiarity with the software that may prove crucial when aid is needed. A drone was simulated in personal computer of team members by running the ArduPilot SITL in Gazebo and successfully received commands from custom python scripts, demonstrating the possibility of autonomous flight.

4.3.3 Shifts in Project Effort

Simulating the drone led to the discovery that the initial design requirements (indoor vision-based tracking) could not be natively supported by the ArduPilot. While ArduPilot is a popular drone software for autonomous flight, it typically utilises GPS signals as inputs to its Extended Kalman Filter (EKF) to correct its own pose. Without GPS signals for corrective action, sensor reading errors will accumulate with time. The option of applying lower-level control commands, such as manual thrust, was also explored in Gazebo. However, the simulated drone experienced unacceptable sways in the simulation due to a lack of awareness of its own position. This has become an impediment in the project as autonomous flight in GPS denied environments was one of the design requirements initially set forth. Several methods were devised to resolve this issue:

1. Reprogram the flight controller with PX4 firmware and test the performance of the drone, which may yield improvements over the current ArduPilot firmware since PX4 has more popular usage in research and development applications.
2. With the aid of IMUs stationed in the flight controller, self-tune a controller to output low-level commands to always level the drone.
3. Incorporate additional sensors to make up for the lack of GPS data indoors. This includes the use of optical-flow sensors, which are essentially cameras that take frames of images and allow the pose of the drone to be inferred from the distance travelled between consecutive frames.

During that point in time, option 3 was the most pragmatic, but was not acted upon because of a lack of documentation of how to integrate optical-flow sensors with the existing drone hardware. Option 1 and 2 were disregarded because that would involve reprogramming or overwriting some of the existing functionalities of flight controller firmware like ArduPilot and PX4, which is outside the expertise of our team of mechanical engineers. After careful considerations and meetings with our supervisors, it was decided to pivot the focus of the project fully towards the docking platform to achieve a more complex active mechanism. It was also to concentrate the effort to achieve one goal with excellence, instead of attempting to fulfil multiple goals with moderate results. To conclude the avionics section, project outputs include detailed documentation of the conceptual framework behind autonomous flight that would have otherwise been realised if not for limitations in firmware and instrumentation documentation.

5. Hardware

5.1 Overview of Drone Docking Platform & Background Study

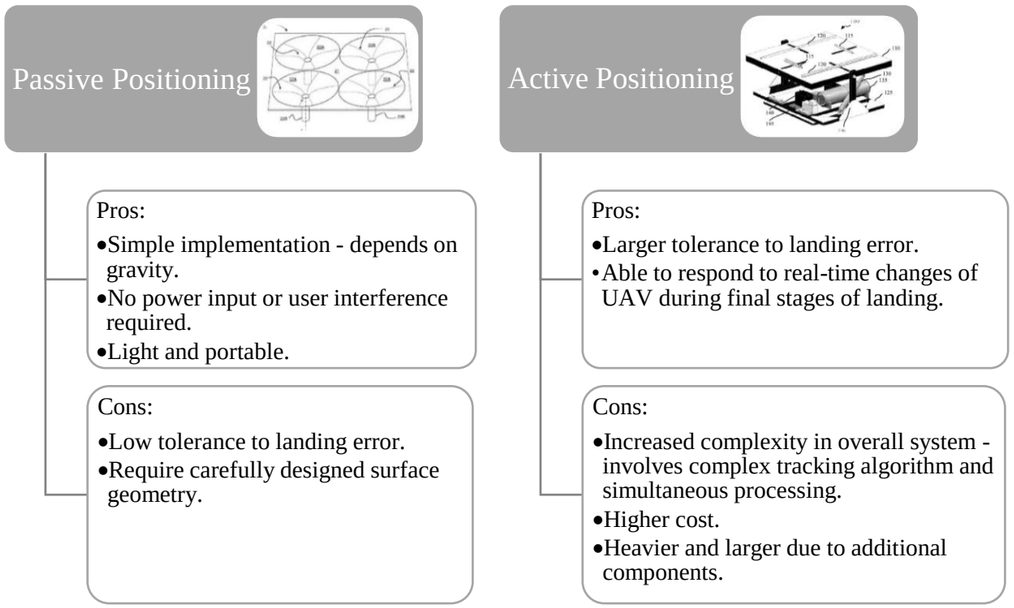


Figure 10. Pros and cons of passive and active docking approaches [4, 10].

Over the years, numerous drone platform designs were built and implemented to retrieve and position a drone on a platform. Generally, it is categorised into passive and active positioning approaches. Passive methods depend on the gravity and specific surface geometry to slide the drone into position. No external power source or inputs are required. Active methods respond to uncertainty through electronic control system and actuators. Sensors and cameras can be incorporated for real-time detection to generate responses accordingly. A robust and sophisticated system can be very complex as it requires different algorithms while processing multiple inputs and outputs simultaneously. The pros and cons of both approaches are summarised in Figure 10. In short, the selection of approach depends on specific application and cost.

Thus, the goal of this project is to design a docking platform that can:

- Counteract uneven terrain and always maintain a levelled docking platform.
- Portable and modular (i.e., can be mounted on a vehicle)
- Accommodate landing errors.

- Secure drone in position after landing.

5.2 Binary Weighting Matrix

From the background study done, there are several key parameters that define different design elements of a docking platform. These parameters are documented and ranked according to its importance with a binary weighting matrix as illustrated in Figure 11 and Figure 12.

Binary Weighting Matrix

Docking Mechanism	Design Requirements	Structural Integrity & Materials	Stability	Omnidirectional Landing	Portability	Compact Design	Low Weight	Modularity	Ease of Assembly	Cost	Responsiveness	Large Landing Tolerance	Ability to Hold UAV	Counteract Inclination	Score (Sum of Rows)	Biased Scores (Scores + 1)	Normalised Scores (Biased Scores/Total Scores)
Physical Design	Structural Integrity & Materials		0	1	1	1	1	1	1	0	1	1	1	1	10	11	**12.09%**
	Stability	1		1	1	1	1	1	1	1	1	1	1	1	12	13	**14.29%**
	Omnidirectional Landing	0	0		0	1	1	1	1	0	0	0	0	0	4	5	5.49%
	Portability	0	0	1		1	1	1	1	0	0	0	0	0	5	6	6.59%
	Compact Design	0	0	0	0		1	0	0	0	0	0	0	0	1	2	2.20%
	Low Weight	0	0	0	0	0		0	0	0	0	0	0	0	0	1	1.10%
	Modularity	0	0	0	0	1	1		0	0	0	0	0	0	2	3	3.30%
	Ease of Assembly	0	0	0	0	1	1	1		1	0	0	0	0	4	5	5.49%
Financial	Cost	1	0	1	1	1	1	1	0		1	0	0	0	7	8	8.79%
Performance	Responsiveness	0	0	1	1	1	1	1	1	0		0	0	0	6	7	7.69%
	Large Landing Tolerance	0	0	1	1	1	1	1	1	1	1		1	1	10	11	**12.09%**
	Ability to Hold UAV	0	0	1	1	1	1	1	1	1	1	0		0	8	9	9.89%
	Counteract Inclination	0	0	1	1	1	1	1	1	1	1	0	1		9	10	10.99%
															Total	91	**100.00%**

Figure 11. Binary weighting matrix.

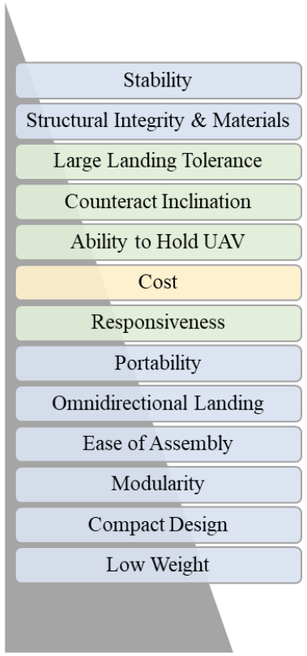


Figure 12. Ranking of relative importance.

Stability took the top spot as an unstable setup would risk damaging both the drone and ground rover especially during landing. The ***structure and material choice*** of the docking platform will affect its strength, which is crucial in withstanding the drone's weight and potential impact during landing. Ability to provide ***large landing tolerance, counteract inclination*** and ***secure drone in position*** corresponds to the main goal of the project. While achieving these goals, it is essential to keep the **cost** down due to limited project budget and make it feasible for future works. The ***responsiveness*** of the docking platform towards constantly changing surroundings is crucial for the successful drone retrieval. Meanwhile, ***portability*** enables quick and easy deployment of the docking platform at anytime and anywhere. Offering ***omnidirectional landing*** allows the drone land in any orientation, eliminating the need for high precision landing. ***Ease of assembly*** is important, especially when mass production is to be expected. Placement of components should also be given attention to not interfere with each other. Design ***modularity*** is considered as it could potentially increase the value and versatility of docking platform. Outside of this project, retrofitting the platform onto other drone-rover systems can be made possible with minor modifications. A ***compact design*** would help reduce material waste, overall weight, and design footprint. The overall ***weight*** of the docking platform is of low concern as it will be mounted on the rover.

5.3 Docking Platform Design Process

5.3.1 Design Concept

Initial study and literature review suggested that autonomous drone landing can be relatively accurate with tolerances of ± 10 cm [11]. The passive positioning approach is a simple system but its tolerance to landing error is not exactly clear. Its performance may vary depending on various factors including the specific drone model as well as weather disturbance during landing. Nevertheless, several design concepts based on passive positioning approach is generated to understand the limits of this approach before identifying the actual range of landing accuracy via manual flight test. The design concepts in Figure 13 exploits the surface geometry to slide the drone into position as it descends.

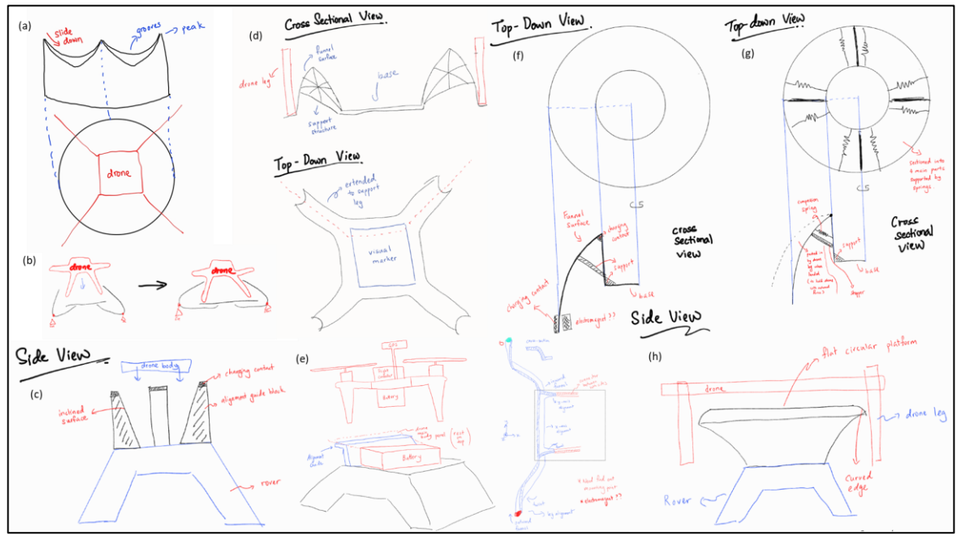


Figure 13. Design concepts based on passive positioning approach. Approach of each concept: (a) petal/crown with sliding surface, (b) clamping mechanism, (c) prism alignment guide, (d) UAV-complement surface with leg support, (e) alignment surface with leg extension, (f) circular dome, (g) spring-loaded circular dome and (h) flat surface with curved edges.

To counter surface inclination, a gimbal system was considered as a potential candidate. However, investigation into its implementation was not conducted as the indoor test flight reveal several issues with the passive positioning approach. Note that the petal/crown design Figure 13(a) was utilised for the preliminary test due to its simple geometry that enables quick fabrication with cardboard. Large variation was observed in each landing attempt even in controlled indoor environment and the design only allowed a limited (±5 cm) tolerance to landing error (Figure 14). Besides that, the platform sizing depends on the specific size of the UAV, requires specific landing direction as well as risk of damaging UAV blade due to elevated geometry.

Thus, design concepts are reiterated with consideration of incorporating active system to achieve the project goal and satisfy the key parameters identified in the previous section (Figure 15). The idea of complementary attachment on drone to aid landing is also explored before focusing on the direction of actuator-driven platforms with active tracking capabilities due to concerns on strings getting tangled.

Figure 14. Preliminary test with the petal/crown design simple prototype (a) drone sliding into position and (b) failed attempt with the drone stuck under wrong landing direction.

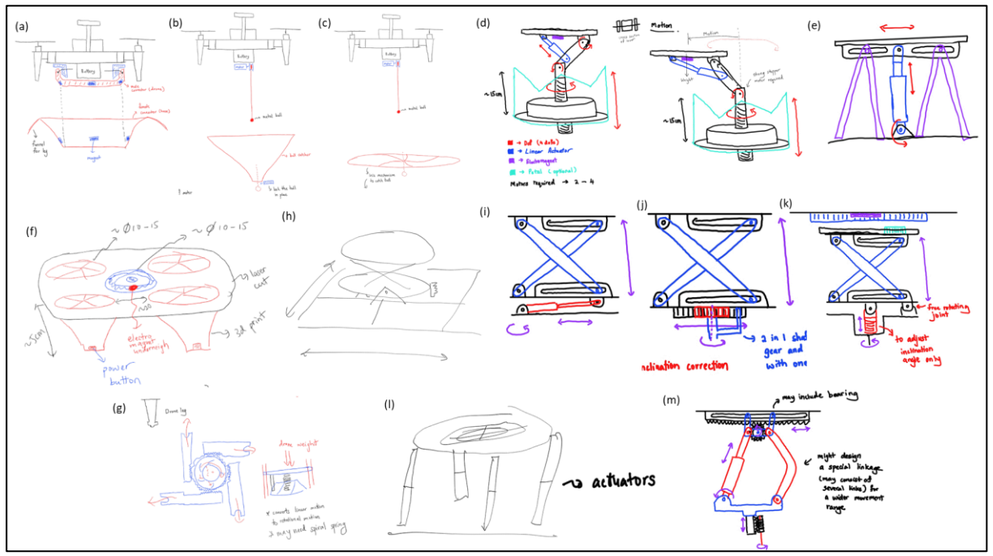
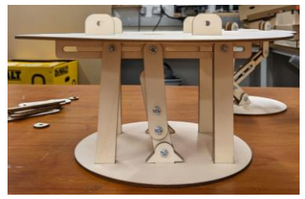


Figure 15. Reiterated design concepts: (a) funnel with electromagnet, (b) funnel with guiding ball, (c) iris mechanism with guiding ball, (d) single actuator arm, (e) dual actuator linkage system, (f) funnel with iris mechanism, (g) UAV leg locking mechanism, (h) scissor mechanism with plane motion, (i-k) scissor mechanism with title control, (l) 6 DOF platform with linear actuators and (m) linkage system.

Z-axis motion introduced in certain design concepts in Figure 15 could be useful in catching the drone during the last stages of landing. However, the implementation is complex due to the coupled relationship between the degree of freedoms and may not be worth the effort for limited benefit for the overall performance. The scissor mechanism could be the potential solution for z-axis motion, but it consumes more space and complicates tilt control system. The six DOF platform in Figure (l) boast great flexibility and reach but is too costly and requires precise control system. The single actuator arm as well as actuated linkage system showed the highest potential due to their relatively simple construction while offering the DOFs required. Simplified version of potential design concepts was fabricated to visualise and to further understand the characteristics of each concept (Table 3). The single actuator arm design was adopted as it has the potential to achieve greater reach with a smaller footprint and enable straightforward control on the extension and tilt angle with less actuators involved. Concerns regarding torque load, reach and tilt performance can be resolved through careful selection of a suitable actuator based on force calculations (refer section 5.3.2.1 and 5.3.2.2).

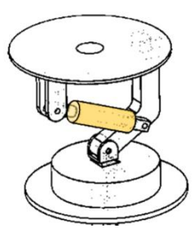
Table 3. Strengths and weaknesses of potential design concepts.

Potential Design Concepts	*Strengths*	*Weaknesses*
 *Single actuator arm approach*	• Only 3 actuators are required to extend, tilt, and rotate. • Straightforward tilt control via adjusting actuator length.	• Large torque acting on the single pivot. • Maximum reach and tilt limited by the extension range of actuator.
 *Dual actuator linkage system*	• Better stability with the additional supports.	• Many actuators involved (5 in total). • Limited reach (constrained by length of groove and actuator extension range). • No straightforward approach to control tilt.
 *Linkage system*	• Easy tilt control of platform.	• Complex linkage system. • Stability concern due to additional DOF. • Slightly higher centre of gravity due to actuator installed at a higher position.

5.3.2 Single Actuator Arm Mechanical Design

Inspired by automated robotic arm, the concept incorporates a linkage system controlled by three actuators. In overall, the docking platform would be able to move in three core DOFs which are crucial in achieving the project goal: extension, tilt, and rotation. It consists of a platform surface pivoted onto an actuator- driven arm and a variable length actuator on the other end. This provides tilt control which is an essential part of the self-levelling surface mechanism. The other end of the variable length actuator will be attached to the main arm which is pivoted onto the base. The rotational motion of the arm around its pivot will be translated into linear motion of the platform surface and thus producing an extension and retraction motion. The base will be attached to a mechanism that introduce rotational motion to increase the coverage area of the platform. The design process of each key element: the surface levelling mechanism, arm and actuator, base and top platform will be presented in the following sections.

5.3.2.1 Surface Levelling Mechanism



Self-levelling of platform surface is achieved via a variable length actuator responding to inputs from a sensor. By pivoting the platform surface at one end, the extension and retraction of actuator will change the surface's tilt angle to counteract uneven terrain. There are a few potential candidates of variable length actuators including linear actuator, leadscrew and hydraulics that can be implemented to perform the task. The actuator selection depends on size, weight, extension range as well as control approach. An analysis is done to understand the pros and cons of each type of actuator,

which is documented in Figure 16. Hydraulic system is eliminated from further testing as it consumes too much space to house the hydraulic power unit to remain portable and there is no straightforward method to acquire its instantaneous position. There is also risk of fluid leakage from the system which increases the complexity. At this stage, there is no sufficient information to select between the linear actuator and leadscrew as the former can generate higher force, while the latter is light and has self-locking capability. Thus, it is decided to perform a test on the actual capability of both actuators and the details are presented in Section 5.3.2.1.

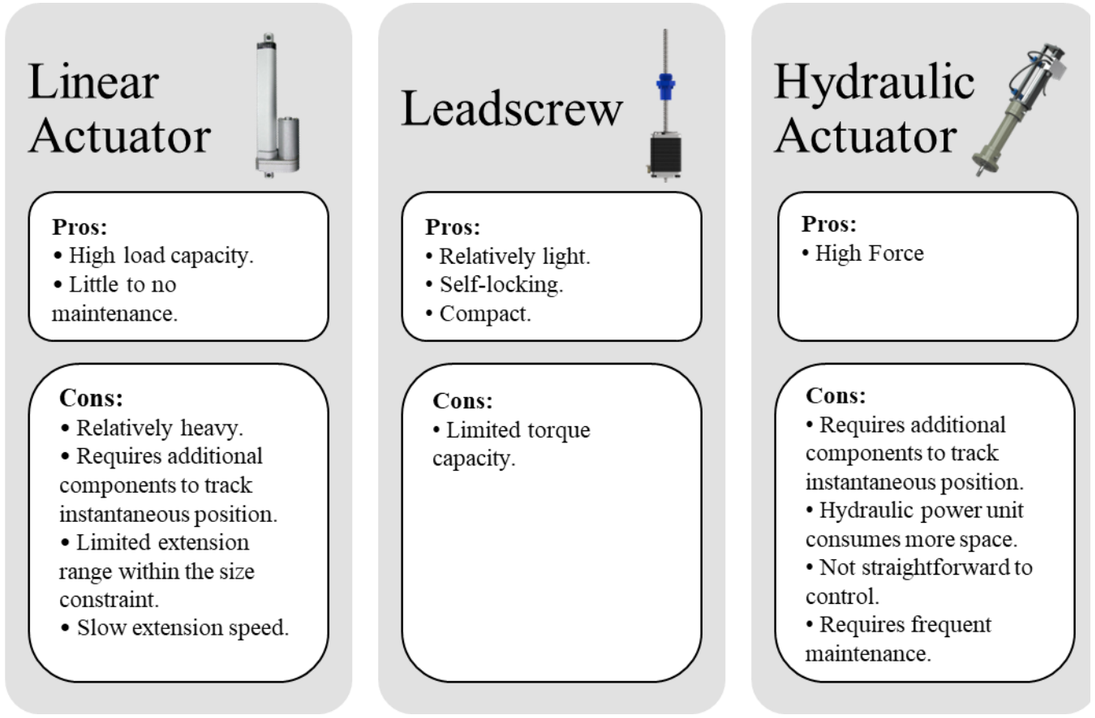
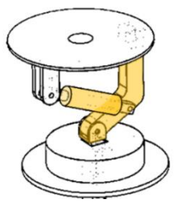


Figure 16. Pros and cons of different types of actuators.

5.3.2.2 Arm and Actuation



The arm geometry and the attachment point placement are a crucial part of the arm design as it needs to provide support and allow movement in the desired path without interfering with other elements. There are three main attachment points on the arm including the pivot to the base, actuator and top platform as illustrated in Figure 17. The distance between each other can produce different results in terms of reach, tilt control, force, and torque loads on the actuators. The axial force acting on the actuator is done via resolution of force with an assumed platform mass of 3.5 kg (i.e., 1 kg platform and 2.5 kg drone) acting on the centre of the top platform. The associated torque requirement for leadscrew can be determined as below:

$$T_{leadscrew} = \frac{Fd}{2}\left(\frac{L + \mu d\pi}{\pi d - \mu L}\right) \quad (1)$$

where the parameters are tabulated in Table 4 according to the specifications from a leadscrew that fit within the size constraints and is readily available in the market. The required actuation torque for the entire platform around the base pivot is:

$$T_{base\ pivot} = mg \times extension_{max} \quad (2)$$

The extension is defined as the horizontal distance between the base pivot and centre of the top platform and is measured with the dimensionally accurate assembly in SolidWorks as illustrated in Figure 18. An initial arm geometry is generated for each actuator with consideration on providing sufficient clearance during operation (refer Figure 17), while an arbitrary distance between attachment points is assigned as reference. Parametric study of the attachment point distance is done to maximise extension range while reducing maximum force and torque load on the actuators and the normalised results are presented in Figure 19.

Table 4. Actuator Specifications.

Parameter	Specifications
Leadscrew	
Lead, $\boldsymbol{L}$	8 mm
Diameter, $\boldsymbol{d}$	8 mm
Coefficient of Friction, μ	0.2
Max Torque, $\boldsymbol{T_{leadscrew}}$	0.4 Nm
Mass, $\boldsymbol{m_{leadscrew}}$	0.45 kg
Linear Actuator	
Max Axial Force, $\boldsymbol{F_{linear\ actuator}}$	900 N
Max Displacement, $\boldsymbol{l_{displacement}}$	50 mm
Mass, $\boldsymbol{m_{linear\ actuator}}$	0.96 kg

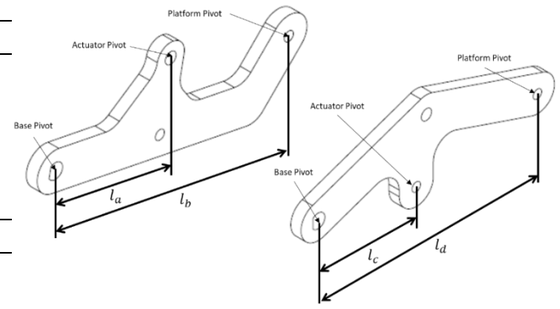


Figure 17. Arm geometry and attachment point

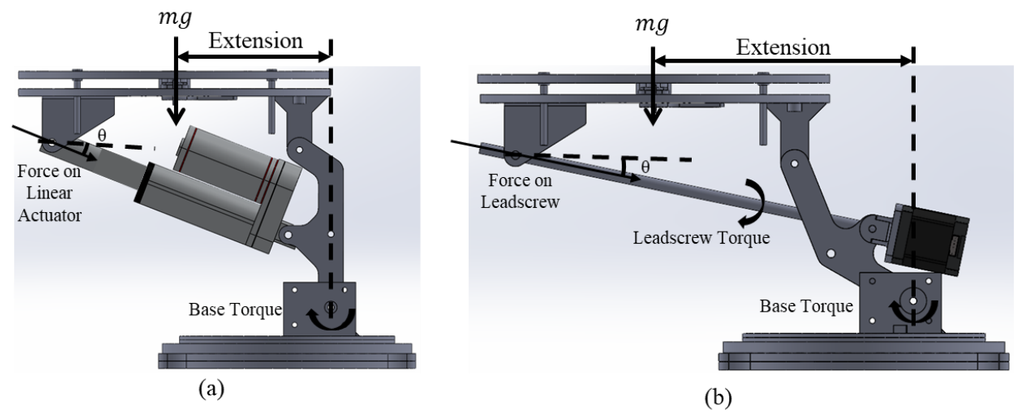
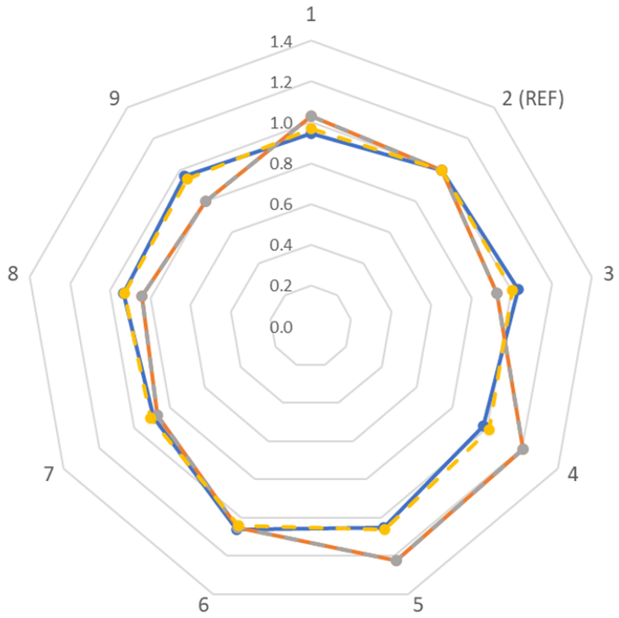


Figure 18. (a) Linear actuator and (b) leadscrew under maximum extension with sufficient clearance.

From the calculations, the force and torque loads involved are still within the limits of both actuators. It is clear that leadscrew is able to provide greater extension range (205.65 mm vs 96.81 mm), and thus having more spare room to counter platform tilt across all configurations than linear actuator. Note that the tilt angle is controlled by actuator length, and it is found that there is a linear relationship between the tilt angle and actuator extension. The linear actuator is eliminated due to its limited extension and tilt

angle range, which are of high importance in this project. It is also slow to extend (14 mm/s), and its heavy weight might also introduce instability when fully extended. From Figure 19, Configuration 9 is considered as the optimised design as forces and torque load on the leadscrew are significantly reduced while keeping a similar extension range. The larger displacement achieved by leadscrew is accompanied by a larger torque on the base pivot. Thus, a 5.85 Nm DC motor is sufficient to power the platform. However, the actual extension of the final prototype is to be software limited to a lower value to leave room to control platform tilt.



Normalised Extension Range*
Normalised Max Axial Force on Leadscrew**
Normalised Leadscrew Torque**
Normalised Base Pivot Torque**

Reference Performance:
Max Extension Range = 205.65 mm
Max Axial Force on Leadscrew = 138.66 N
Max Leadscrew Torque = 0.31 Nm
Max Base Pivot Torque = 6.26 Nm

CONFIGURATION	l_c (mm)	l_d (mm)
1	57.52	123.50
2 (REF)	57.52	128.50
3	57.52	133.50
4	62.52	123.50
5	62.52	128.50
6	62.52	133.50
7	52.52	123.50
8	52.52	128.50
9	52.52	133.50

* Higher is better; ** Lower is better

Figure 19. Normalised performance of leadscrew.

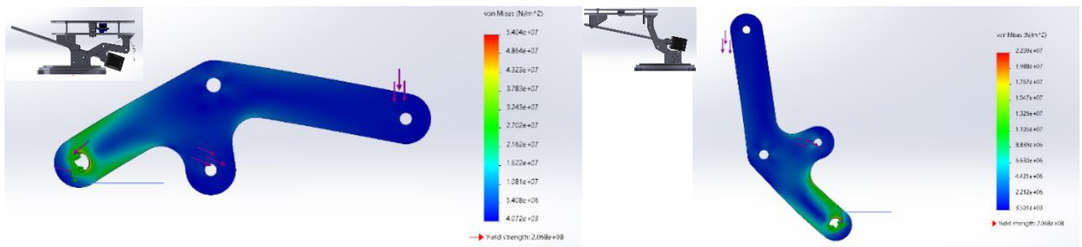


Figure 20. Stress distribution on arm geometry under expected load at extreme positions.

An FEA is conducted on the optimised arm at extreme position (where highest loading is expected to occur at minimum and maximum extension) with an expected loading condition of 3.5 kg platform mass and 5.9 Nm of torque at the base pivot. Prior to the FEA, a quick test with 6 mm plywood showed that the keyhole connected to the DC motor that generates torque is damaged with only 0.2 Nm torque load. Stainless steel is the material of choice due to its resistance to rust and great strength, which is crucial for this weight-bearing component. The results in Figure 20 show that stress is concentrated at the keyhole, agreeing with the quick test results. Due to stainless steel's enhanced strength, the reduction of thickness from 6 mm to 3 mm saw negligible difference in deflection. To further reduce weight, materials in areas experiencing low stress is removed as shown in Figure 21. Physical testing performed on the stainless-steel arm fabricated through waterjet cutting showed no visible deflection under expected load, further verifying the simulated results.

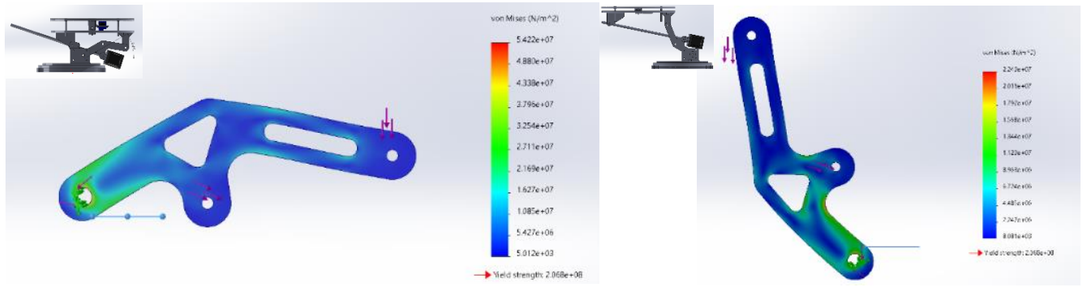
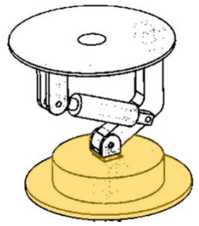


Figure 21. Stress distribution on optimised arm geometry under expected load at extreme positions.

5.3.2.3 Platform Base



The platform base is where the arm is mounted and is a natural choice to control the rotational motion of the platform on top. A rotating bearing is installed between the moving and non-moving base layers to ensure a smooth rotating motion. Introducing an actuator to rotate around the centre axis is a simple and straightforward solution. However, the rotational motion introduces issues with wires and cables running down from the components on top. To reduce the risk of wires getting twisted and tangled as the base rotates, wires are designed to route through the centre. The base pivot is offset to one side instead of at the centre to make space for wires to pass through it. This change also helps to adjust the centre of gravity closer to the geometric centre of the base (due to the nature of uneven distribution of weight).

A simple internal meshing gear system driven by a stepper motor is proposed to control the base rotation. This approach clears up the centre area while offering extra control over the rotational speed and torque through gear ratio selection. The gear ratio is selected at 5.5:1 to produce an output rotational speed of about 0.5 revolution per second and leverage the torque output. The dimensions of the spur and internal gear is calculated with the following relation:

$$m = \frac{d}{N_{teeth}} \tag{3}$$

where the module, m is set at 1.75 to ensure the internal gear is sized correctly and fit underneath the moving base layer. The spur gear attached to a stepper motor is mounted to the non-moving base layer. Besides that, the DC motor that operates the arm is mounted on the moving base layer with its designed housing. Several iterations are done on the material and fabrication approach (i.e., plywood with laser cutting to PLA plastic with 3D printing) to improve strength, stability, wire management as well as the aesthetics as illustrated in Figure 23.

Physical testing was done in between each iteration to identify and discover any underlying issues before progressing further. Incompatibility of gear system, fitment issues due to manufacturing tolerances, material strength and challenge in assembly process are some of the problems revealed during testing.

Initial Design

- A simple circular base.

Iteration 1

- Base is split into 2 main layers: moving and non-moving layers.
- Added ring bearing between 2 base layers.
- Added mounting points on the fixed base to the rover.
- Added internal gear.
- Added mounting points for arm.
    - Loose tolerance causing the arm to be flimsy.

* Made with consideration for laser cutting using plywood

Iteration 2

- Added a flat DC motor mount and corresponding attachment point on the base.
    - Encountered difficulty in assembly process.
    - Loose fitment led to DC motor moving out of position during operation.
- Added spur gear for the internal gear with gear ratio of 3:1.
    - Rotation speed too high.
- Added mounting points on the fixed base layer for stepper motor to operate the gear system.
- Added hole at the centre for routing wires.
    - Too small to fit all wires and accommodate the fragile flat camera cable.

Iteration 3

- Focusing on solving issues found in iterations 1 & 2 and replacing laser cut plywood with 3D printing parts.
- Reiterate base with integrated arm pivot structure.
    - Improved arm stability.
- Reiterate DC motor mount for better support and stability.
- Improved assembly process.
    - DC motor mount can slide into place, solving assembling issue in Iteration 2.
- Reiterate spur gear to achieve gear ratio of 5.5:1 to reduce rotation speed.
- Increase hole size to accommodate more wires than initially expected.
- A special slot is made to hold camera cable separate from other wires.

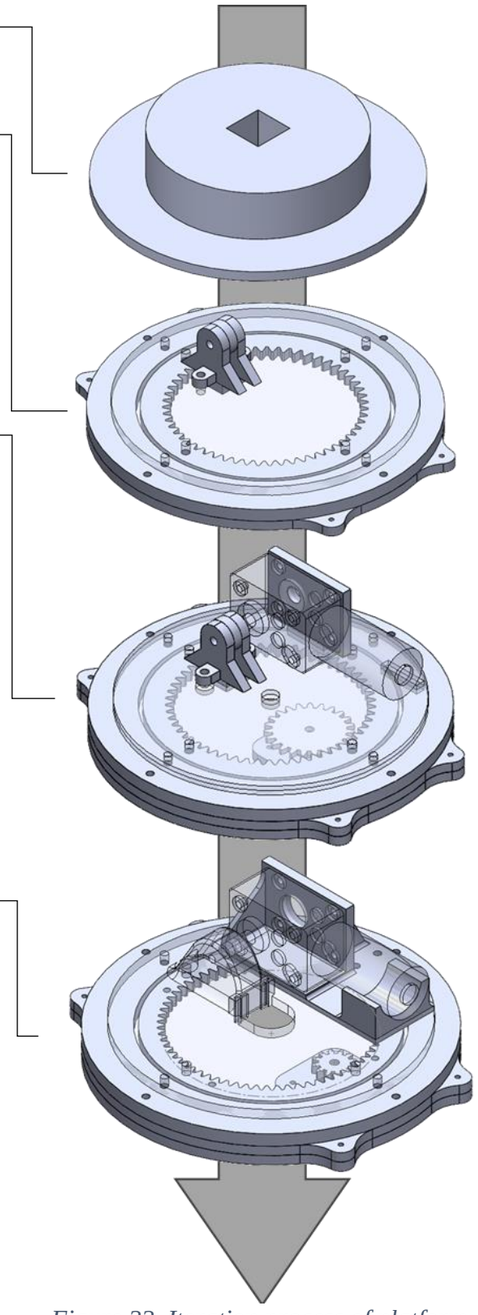
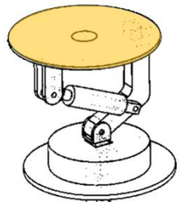


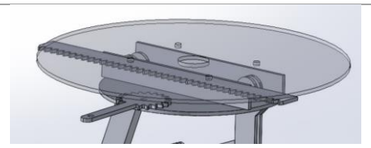
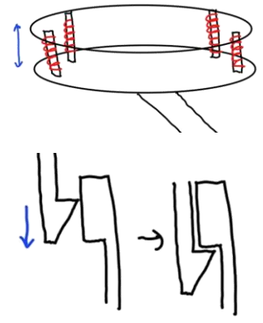
Figure 22. Iteration process of platform base design.

5.3.2.4 Top Platform



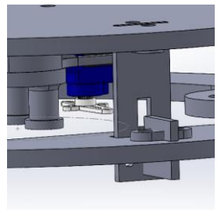
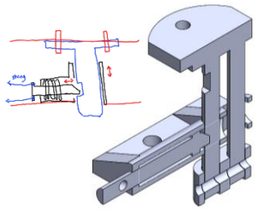
The top platform surface is circular and sized to accommodate slight error or response delay from the active tracking system in all directions during landing. It will also serve as a mounting point for several essential sensors to detect drone which will be discussed in detail in **Section 7**. However, populating the area with electronic components can get messy with all the wiring. Besides that, the space consumed by these components may also reduce clearance to the actuator and interfere with the platform motion. Introducing vertical offset could potentially solve the issue, but design ideas revolving around double-stacked top platform is explored (Table 5) to not only solve the aforementioned problem, but to further improve the platform's functionality.

Table 5. Double-stacked top platform ideas.

	Strengths	*Weaknesses*
 *Adjustable top layer*	• Increase extension in the x-axis. • Potentially reduce response time to reach the opposite end.	• Requires an additional actuator on the top. • Increase weight of the top platform. • Increase control system complexity.
 *Damping system with integrated hold-and-release mechanism*	• Introduce z-axis motion. • Springs introduce damping to absorb impact during landing. • Potentially speeding up the last stages of landing. • A lightweight servo motor is sufficient to release mechanism. • No precise control system is required.	• Requires an additional mechanism to control z-axis movement.

Owing to the large extension achievable with the leadscrew, it is not necessary for further x-axis extension offered by the adjustable top layer. The additional weight introduced might also cause instability due to weight imbalance under extreme extension. The second design idea incorporates spring-loaded top platform with integrated hold-and-release mechanism. Damping from the springs could help to protect the drone from excessive shock during landing. In combination with an electromagnet, the pre-compressed mechanism can be released via a string attached to a servo motor to immediately retrieve the drone at the right timing when it is in proximity from the top layer surface. This eliminates the need for constantly aiming on the platform centre during landing, which can be difficult when wind is present. This idea has undergone further development considering the benefits it brings with just a simple mechanism. There are two iterations of the mechanism to control the top layer in either compressed or released state in Table 6.

Table 6. Hold-and-release mechanism iterations.

	Pros	*Cons*
Iteration 1 	• Less components. • Simple geometry.	• High wear rate. • Inconsistent. • High force required to release mechanism.
Iteration 2 	• Smooth and consistent operation. • Easy to click in place and release. • Negligible wear. • Integrated with springs to hold top layer.	• Requires a specifically designed geometry.

The first iteration despite its simple construction, retired due to immense wear and inconsistent performance shown during testing. Replacement part is required after a few cycles due to broken surface. As the mechanism is separated from the springs pushing against the top layer, imbalance force was introduced and caused a significant deformation on the intermediate layer. The second iteration that is controlled with a single servo motor offers a smoother operation and better consistency. The new mechanism also provides a sturdier connection between both layers of the top platform than the previous iteration. The spring-loaded piston can click into position to hold the top layer in the compressed state, and it can be released by pulling the piston attached to the servo motor via a string. The mechanism provides a 2.5 cm height buffer that can be compressed under the UAV's weight to provide damping. The springs are specifically selected to have a spring constant of approximately 250 N/m to be stiff enough to support the weight of the top layer, while being sufficiently soft to be compressed under drone's weight for damping.

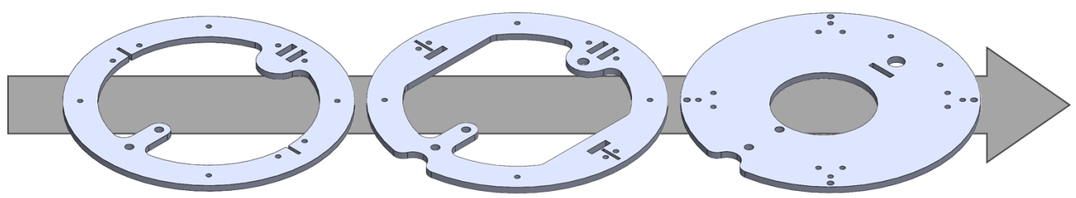


Figure 23. Iterations of the intermediate layer.

The deflection observed in the first design iteration is also contributed by the geometry of the intermediate layer. The big cut out made to provide clearance for the servo motor and electromagnet weakens its structural strength. This is proven by the FEA conducted where a large displacement error occurs under predetermined load with a 6 mm plywood. The layout is optimised by minimising cut out size while providing sufficient clearance for the electronic components (Figure 23). Additional cut out are made for the camera cable separate from other wires to protect it from damage. Beyond the top platform, the wires and cables will pass through the arm through a specifically designed wire shield before reaching the base (Figure 24). The wire shield is a two-piece part with designated path for the fragile camera cable and offers an easy way to manage wires. Throughout the iterations, cut outs designated for first iteration of hold-and-release mechanism is replaced for the updated iteration. The top platform material is also swapped to acrylic for its strength and aesthetic. The FEA results in Figure 25 revealed that the deflection is significantly smaller at less than 0.5 mm. Aluminium is also a potential candidate for its strength, but its electrical conductivity poses concerns to other electronic components attached to the platform surface.

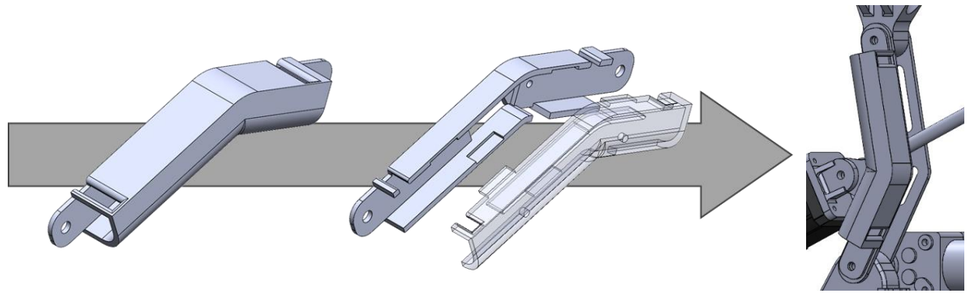


Figure 24. Iterations of wire shield (left) and attachment on arm (right).

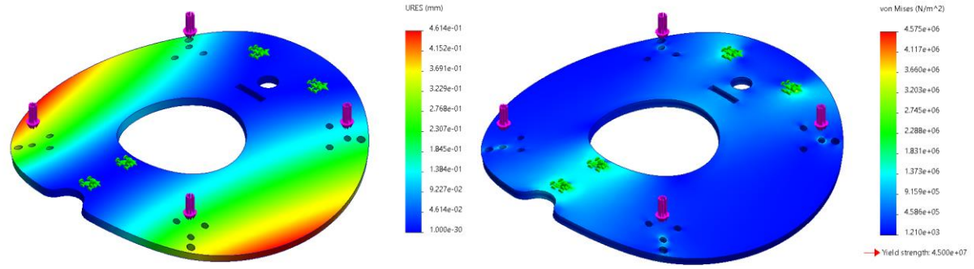


Figure 25. FEA results of 6 mm acrylic of the intermediate layer: deflection (left) and stress (right).

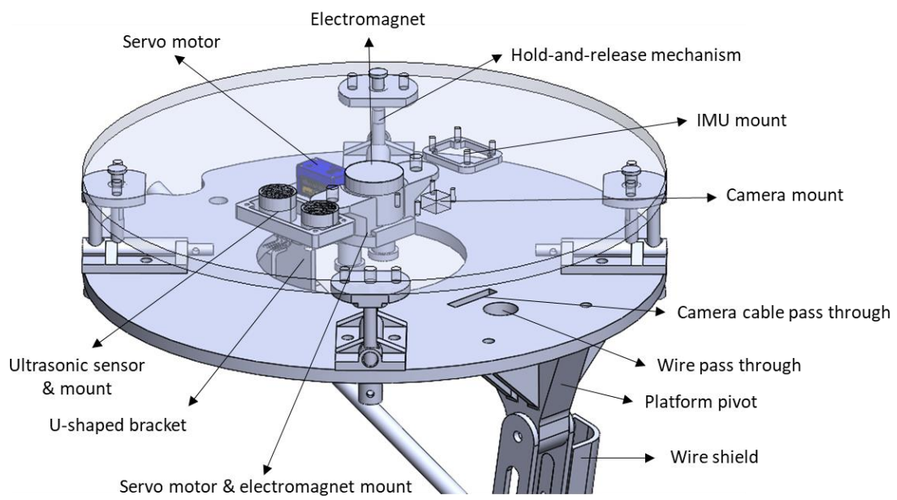


Figure 26. Layout of components on the top layer.

The top layer houses all the electronic components and are arranged in a way to maximise functionality, ease of assembly as well as cable management as in Figure 26. To ensure a perfect fit and alignment, mounts for each electronic component, including the electromagnet, servo motor, IMU, ultrasonic sensor, stepper motor and camera are designed and fabricated. The pivot connecting the top platform to the arm is redesigned for improved strength and integrated with a bypass hole for wire management as illustrated in Figure 27. A standard U-shaped bracket (Figure 28) is installed on the other end to provide an attachment point for the opposite end of the leadscrew. The mount for electromagnet and servo motor is integrated with guide pillars to guide the strings controlling the hold-and-release mechanism (Figure 29).

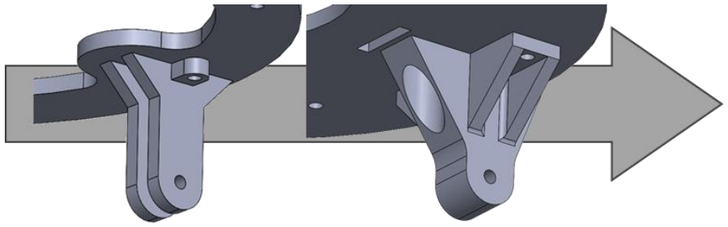


Figure 27. Iterations of pivot connecting the top platform and arm.

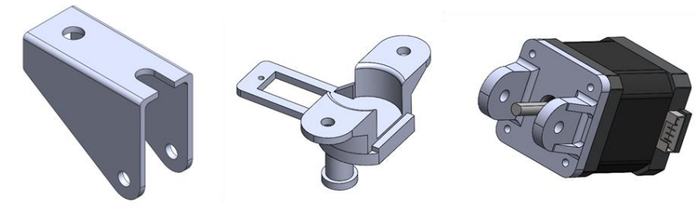


Figure 28. (a) U-shaped bracket, (b) integrated electromagnet and servo motor mount, and (c) leadscrew stepper motor mount.

Figure 29. Guide pillars on the electromagnet and servo motor mount guiding the strings attached to the hold-and-release mechanism's piston.

6. Software

6.1 Overview of Software Architecture

During the initial stages of the project, team effort was equally split into autonomous landing and the docking platform. A flowchart of the logic behind autonomous landing is shown in Figure 30, highlighted by the blue background. The transition to an active docking platform calls for a new software logic to be drafted and is shown in the rest of Figure 30. As autonomous flight was discontinued, explained is section 4.3.3, the new project assumption was that the drone would be able to fly autonomously and detect the rover using marker detection (assume blue background has been achieved). To simulate this project assumption during testing, our group manually held the drone at an elevated position to simulate the drone hovering over the rover. The idea of tracking a marker still applies but is reversed such that tracking is performed by the platform, where the actuators will strive to reduce the displacement error between the platform and the marker via closed-loop feedback systems. Once a desired translational setpoint has been reached and the drone is sufficiently close to the platform in altitude, the final drone catching sequence will commence - activating an electromagnet and a servo motor to extend the platform vertically. Throughout the entire process, the platform is actively levelled to provide a flat surface ideal for landing. The subsequent section (Section 6.2) delves into the detection phase, including the comparison of marker types, pose and sizes. Section 6.3 discusses the software framework to implement the algorithm in the figure below. Finally, a Simulink simulation carried out for controller tuning is minutely discussed in Section 6.4.

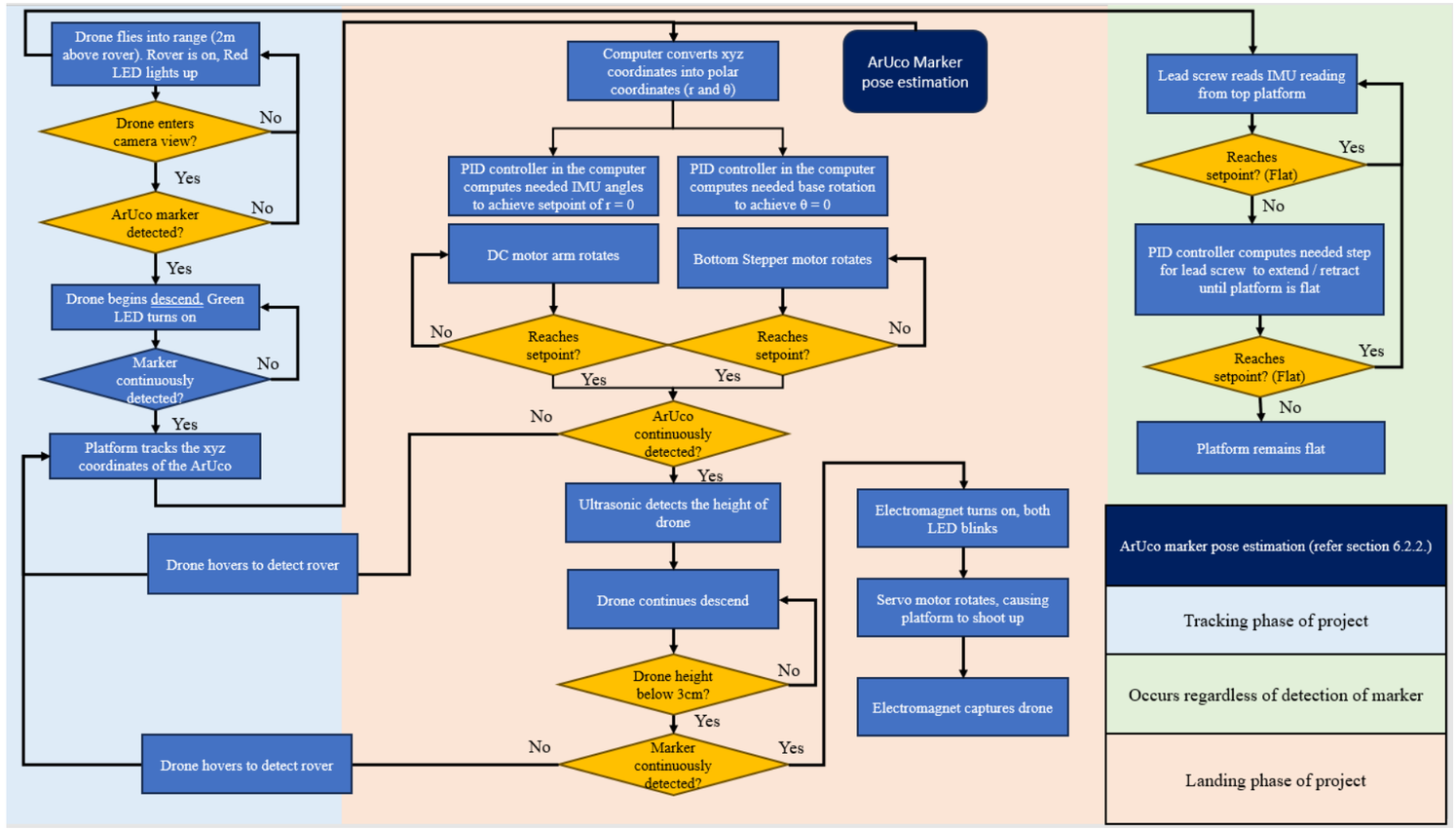


Figure 30. Overview Flow Chart of software.

6.2 Marker Detection

6.2.1 **Detection Design Process**

Given the significance of the detection phase in the functionality of the docking platform, a comprehensive literature review of previous relevant studies has been undertaken, with a primary focus on the detection of drones and rovers.

Attention has been directed towards several critical aspects, which led to the implementation of ArUco marker detection via OpenCV. The full design process flow, outlining each stage has been illustrated in Figure 31.

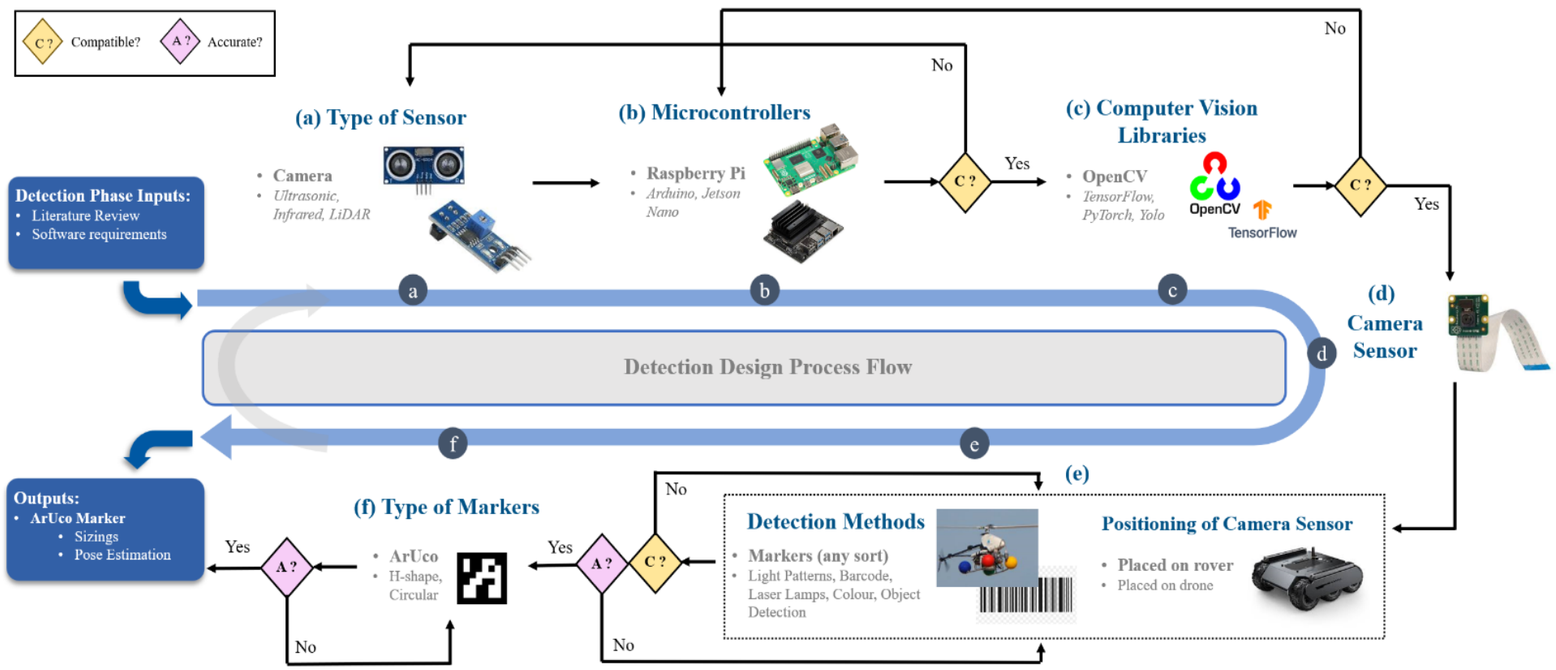


Figure 31. Detection Design Process Flow.

(a)(b) Sensor & Microcontroller Compatibility

The first stage involves selecting the most suitable sensor type and compatible microcontroller platform for communicating computer vision and control algorithms to the docking platform. Initially, three ultrasonic sensors were considered to be placed at certain degree of angle with respect to each other for object detection due to their affordability and compatibility with various microcontrollers, including Arduino. However, inconsistent and fluctuating performance of ultrasonic led us to explore LiDAR technology, which uses 3D mapping but proved cost prohibitive. Ultimately, we determined that **camera modules (**such as the IMX219-77 for Jetson Nano and RPI Camera Module V2 for Raspberry Pi) were the suitable option given that they offered the most effective detection capabilities despite their shorter range, supported by previous work conducted by [12].

While Jetson Nano initially seemed preferable for its higher processing power, **Raspberry Pi 4 (4GB RAM)** emerged as the superior choice due to its native support for PWM pins, which is crucial for control over electronic actuators, mentioned in Section 6.4. Moreover, Raspberry Pi's robust processing capabilities and versatility make it ideal for developing complex computer vision algorithms. Additionally, its compact size and low power consumption make it well-suited for integration into space-constrained drone and rover without compromising functionality.

(c) Computer Vision (CV) Libraries

Given our decision to move forward with cameras, CV became a central focus of the project. The process of CV methods is to acquire, process, analyse, and understand digital images, producing numerical or symbolic information to make prompt decisions [13]. Initially, we considered employing CV libraries that has robust algorithms, which includes PyTorch and TensorFlow. However, as illustrated in the comparison chart in Figure 32 below, **Python's Open-Source CV library (OpenCV)** was selected for several reasons.

OpenCV is renowned for its efficient implementation of computer vision algorithms including object tracking, camera calibration etc, resulting in high speed and real-time processing capabilities due to its extensive use of C/C++, which are crucial for real-time applications like drone and rover detection. Furthermore, OpenCV has a large and active community with extensive forums, documentation, and contributions from developers worldwide, including libraries such as a predefined arucoDict for ArUco markers, which will be covered further in Section 6.2.2. While the performance of OpenCV may be relatively lower, it demands less computational resources, aligning well with our project objectives and the RPI microcontroller, which does not entail any deep learning tasks.

	OpenCV	TensorFlow	PyTorch
Suitable for Computer Vision Applications	✓	✗	✗
Suitable for Deep Learning Applications	✗	✓	✓
Speed & Efficiency	✓	✓	✓
Performance & Accuracy	-	✓	✓
Ease of Integration & Development	✓	✗	✓
Community Support & Documentation	✓	✓	✗

Figure 32. Comparison chart of computer vision libraries - OpenCV, TensorFlow, PyTorch.

(d) Camera Module

After previous rounds of iteration, we opted for the **RPI Camera Module V4**, depicted in Figure 33, for the CV needs. Its seamless integration with the RPI allows for direct tweaking of camera settings from the RPI itself and offers access to numerous built-in libraries, including Pi Camera. Additionally, it boasts high-resolution video capture capabilities, enhancing accuracy for our detection purposes.

Figure 33. RPI 4 & RPI camera module.

(e) Detection Methods & Positioning of Camera Sensor

The compatibility between detection target and positioning of camera sensor serves as key contributing factors to the overall performance of the CV system.

A deep analysis comparing the advantages and limitations of placing the camera sensor on the rover and drone has been conducted, considering factors stated in Table 7. These four factors have been evaluated according to their significance and impact on the integration of the docking platform. They have also been assigned scores reflecting their performance. The scale for both weightage and score ranges from 1 to 5, with 5 representing the greatest importance. It is important to note that these scores are informed by findings from past literature reviews, as tabulated in Table 7, serving as the basis for generating a radar graph depicted in Figure 34 for better visualisation.

Table 7. Data of weightage and scoring for the factors of positioning of camera sensor.

Factors	Weightage	Placed on Rover		Placed on Drone	
		Score	Total Score	Score	Total Score
Field of View	2	2	4	5	10
Landing Flexibility	2	1	2	4	8
Stability & Accuracy	4	4	16	2	8
Alignment with Project Goal	5	5	25	1	5

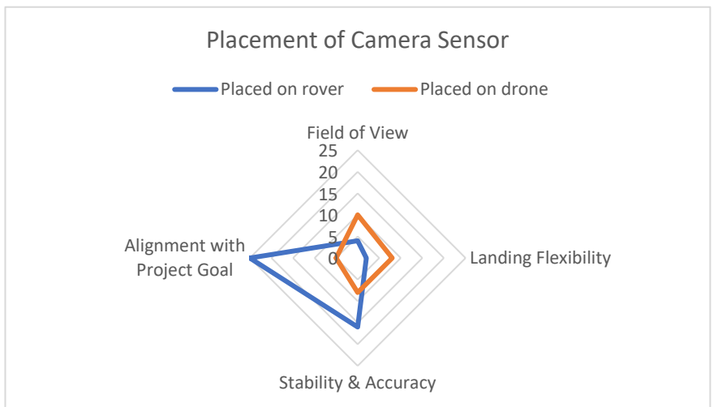


Figure 34. Radar graph comparing the factors affecting positioning of camera sensor.

As illustrated in Figure 34, **positioning the camera on the rover** presents advantages in terms of stability and accuracy, attributed to the rover's stable reference point. This placement minimizes potential camera issues, such as blurring caused by vibrations, which could occur if the camera were exposed in the air when mounted on the drone. Despite offering a narrower camera viewing angle, this configuration aligns with our project goal of detecting a single drone at a maximum altitude of 2 m. Additionally, opting to place the camera on the rover reduces payload weight on the drone, further reinforcing our decision to pursue this option.

As for the **detection method**, or in layman terms, ‘what we are detecting’, several options have been explored, taking into account factors such as accuracy, processing complexity, and resilience to environmental conditions:

- Barcode

This method is commonly employed in commercial settings as referenced by [14], which offers simplicity in implementation. However, it is prone to damage and may lack accuracy when viewed at certain angles due to overlapping lines.

- Laser Lamp

This option explored by [15] leverages light for detection, where an infrared laser lamp will be placed on the drone, hence boasts long-range capabilities and resilience to environmental factors. Nevertheless, it necessitates additional hardware setup which may result in higher complexity and cost for enhanced accuracy, including two infrared cameras being place at a certain distance away from each other along the landing runway tasked to detect the laser lamp.

- Color Landmarks

A trinocular system with coloured landmarks installed on the drone explored by [16] allows estimation of the drone's position for docking by the rover as seen in Figure 35. While this method utilizes existing drone features for detection, it may require extensive calibration to optimize accuracy and restrict flexibility.

Figure 35. Coloured marks on drone.

- Markers

Shapes like H, T, circular, and ArUco markers studied in [17-19] offer easily detectable visual indicator and requires minimal computational demands. One of the limitations of detection of markers was due to proximity, given that the camera could not sense the entire marker if it was too close. However, integrating cameras with ultrasonic sensors can mitigate this issue, which was covered in the preceding hardware and subsequent electronics sections.

Markers emerge as the preferred detection method due to reliable performance across different detecting orientations and distances, and minimal computational requirements. Their widespread adoption and support within the CV community further underscore their suitability for implementation.

(f) Type of Markers

After opting for the markers option, the subsequent stage in the design process involves determining the most suitable type of marker based on detectability and reliability. Initially, concentric circles were considered due to their widespread adoptability for object detection. However, a reiterative process involving the development of separate codes for each type of shape, considering that each type utilizes different libraries (e.g., ArUco markers utilize the arucoDict containing markers of different IDs was undertaken) was conducted.

These codes were programmed to print whether the marker was detected every three seconds. During this interval, the marker was manipulated to different extreme positions, distances, and angles to assess detectability under various conditions. A total of 20 captures were conducted, and the number of successful detections, indicated by the printing of 'Marker Detected', was recorded.

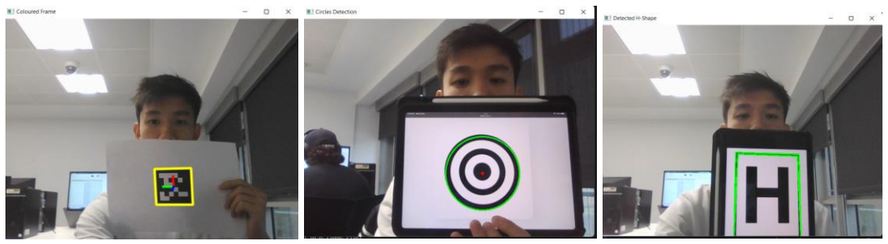


Figure 36. Marker detection process for various markers, ArUco, circular, H-shape (left to right).

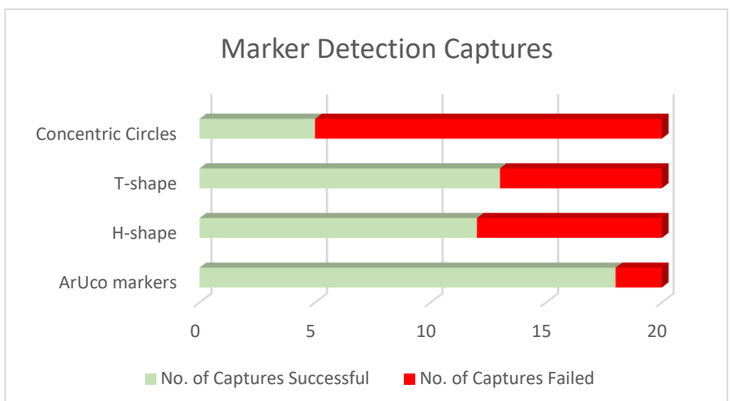


Figure 37. Marker detection captures data for ArUco, concentric circles, T-shape and H-shape.

As illustrated in Figure 37, ArUco markers exhibited the highest success rate, with approximately 90% (18 out of 20 captures) yielding positive detections. Consequently, ArUco markers are deemed the most reliable option. It is worth noting that the reliability of detection may also be influenced by the code itself, which was self-written. Therefore, the resulting data may not entirely reflect real-world outcomes but serves as justification for the selection of ArUco markers for our project. As shown in Figure 37, ArUco markers exhibited the highest success rate, with approximately 90% (18 out of 20 captures) yielding positive detections. Consequently, ArUco markers are deemed the most reliable option. It is worth noting that the reliability of detection may also be influenced by the performance of the code itself, which was self-written. Therefore, the resulting data may not entirely reflect real-world outcomes but serves as justification for the selection of ArUco markers for our project.

6.2.2 ArUco Marker

Figure 38. ArUco marker ID 25.

ArUco marker is a synthetic square marker composed by a wide black border and an inner binary matrix which determines its identifier (id). The black border facilitates its fast detection in the image and the binary codification allows its identification and the application of error detection and correction techniques [20]. A random arbitrary ID of 25 was chosen for our project, which could be seen in Figure 38. In addition to serving as a visual detection system, these numbers can also be utilized for broader and future applications, such as identifying specific types of drones. Each ID can represent a particular drone, offering flexibility possibilities and standardization benefits. The next milestone would be to determine the optimum marker sizing and calibrate the camera. This calibration significantly influences the pose estimated coordinates of the marker relative to the centre of the camera frame, which are subsequently utilized as input coordinates for the control algorithm during the tracking phase. The complete workflow, including design iterations, is depicted in Figure 39 below.

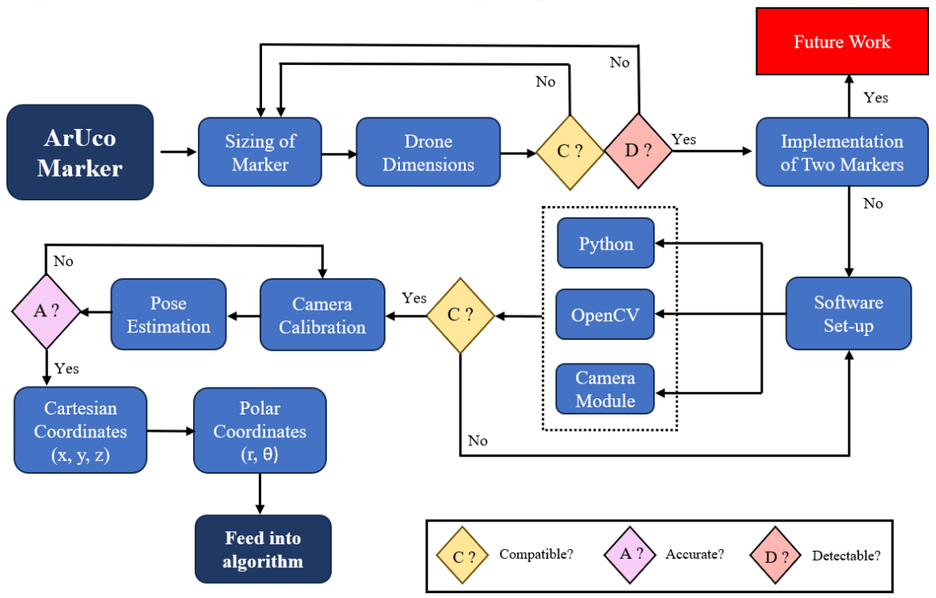


Figure 39. ArUco marker pose estimation design process flow.

6.2.2.1 Sizing of Marker

The marker size determines the size of the internal matrix. For instance, a marker size of 4x4 is composed by 16 bits. A variety of ArUco marker sizes were tested to determine the optimal size based on the suitability and size specifications of the drone, where the marker would be positioned below the battery holder. A straightforward yet effective test was conducted by adjusting the vertical distance (height) of the camera relative to the marker by manually moving it with our hands, observing the marker's detection status via remote viewing. The collected data are depicted in Figure 40.

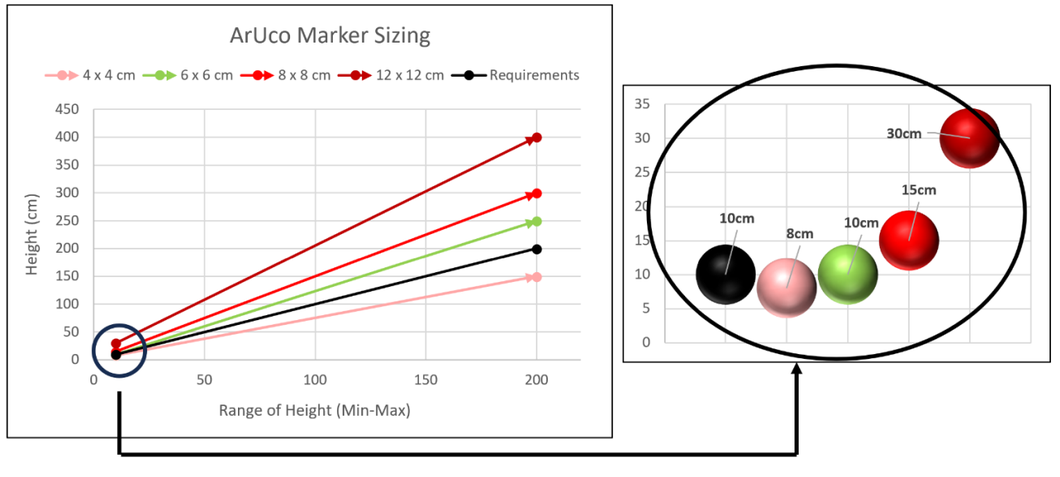


Figure 40. Range of detected height (min-max) comparing ArUco marker sizings.

A total of 4 sizes were tested, ranging from 4x4cm to 12x12cm, with the black line representing the ideal requirements based on several key factors:

- Measurement of available space to accommodate the marker on the drone.
- Minimum height required for the marker to be detected before the drone is safe to land.
- Maximum height initially set in our software requirements (approximately 2 m)

Upon visual inspection, the green line representing a **6x6 cm marker size** appears to strike a balance among all three aforementioned requirements, thus emerging as the most suitable option, by possessing a detection range of 0.1 m to 2.5 m.

6.2.2.2 Implementation of Two Markers

The implementation of two ArUco markers, with one significantly smaller than the other, was considered to aid in close proximity detection. The concept involved the camera detecting the larger marker and accurately estimating its pose when there is a considerable distance between the camera and marker. However, as the distance decreases, the accuracy of pose estimation would diminish slightly until it becomes unable to detect due to the close proximity. Consequently, when the drone is very close to the platform, CV algorithm would switch to utilizing the smaller marker for precise positioning to ensure a smooth landing. Incorporating additional markers on the drone would provide more possible combinations if needed.

Nevertheless, we opted against this approach due to the minimum detectable distance of 0.1 m for the chosen 6x6 cm marker. This limitation can be addressed through the design of the hold and release mechanism as detailed in Section 5.3.2.4, which will vertically extend the docking platform by 2.5 cm to capture the drone, supplemented by the usage of ultrasonic sensors that will be further discussed in the Electronics section.

6.2.2.3 Software Set-up

Improper installation of Python and OpenCV versions on the Raspberry Pi 4 can result in compatibility issues, leading to code execution problems. To address this, compatibility reiterations were conducted based on thorough researc to install specific versions of OpenCV (version 4.5.3.56) and Python (Bullseye 3.9.4).

Furthermore, a remote viewing setup has been established for the RPI embedded within the drone. Utilizing VNC (Virtual Network Computing) viewer enables real-time connection to the remote RPI, allowing for screen monitoring without the necessity of directly connecting an external monitor to the RPI. This setup offers extra flexibility, allowing the modification of Python scripts on-the-fly through remote access.

6.2.2.4 Camera Calibration & Pose Estimation

ArUco markers enables the use of pose estimation which is a CV technique that involves a 3D transformation from the marker coordinate system to the camera coordinate system, yielding the camera position relative to the marker [20]. This position is defined by rotation (rvec) and translation (tvec) vectors, where they are transformed from cartesian (x, y, z) into polar (r and θ) coordinate system, as summarized in

Table 8. This conversion is undertaken to facilitate the integration of ArUco detection with the control algorithm, Where the actuators rely on polar coordinates as input to govern their rotation and extension.

Table 8. Output parameters tvec and its coordinate system.

IOutput Parameters tvec	Cartesian Coordinates Description	Polar Coordinates Description
Tvec[i][0][0] Tvec[i][0][1] Tvec[i][0][2]	Extracts x-coordinate Extracts y-coordinate Extracts z-coordinate	Converts x and y-coordinates into radius, r and angle, θ. z-coordinate is not applied.

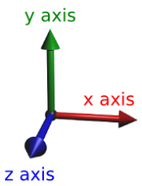


Figure 41. Cartesian coordinate system.

Prior to pose estimation, it is crucial to calibrate the camera to obtain values of the camera matrix and distortion coefficients, which directly impact accuracy. Cameras possess both internal (focal length, lens distortion, optical center) and external (orientation relative to a world coordinate system) parameters that distort images, necessitating correction through equations of radial and tangential distortion. For detailed formulas or equations of these two parameters, please refer to the link provided:[21].

Validation of these values is essential, with cartesian coordinates chosen for their visual clarity. The initial validation approach involved visually inspecting the printed coordinates while moving the marker within the camera frame. A reiterative concept of grid-like mapping approach, detecting multiple ArUco markers of varying IDs simultaneously was implemented to increase the reliability. By measuring the distance between markers and comparing it to the pose-estimated distance, accuracy could be evaluated.

Vertical	Actual Distance	Pose Estimated
Marker 1 and 2	~ 50 mm	~ 53 mm
Marker 1 and 3	~ 50 mm	~ 40 mm
Marker 2 and 3	~ 100 mm	~ 93 mm

Horizontal	Actual Distance	Pose Estimated
Marker 1 and 2	~ 100 mm	~ 120 mm
Marker 1 and 3	~ 100 mm	~ 130 mm
Marker 2 and 3	~ 0 mm	~ 10 mm

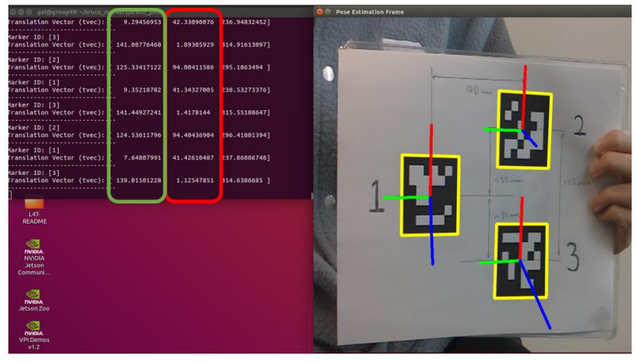


Figure 42. Pose estimation validation using x2 (left) and x3 (right) ArUco markers. Note that up to x6 markers were used for validation.

Initially, the PI camera module V4 calibrated using the simplest calibration approach – chessboard pattern. However, the results showed significant inaccuracies based on the validation approach described earlier, with large error differences. Subsequently, we iterated through several other calibration methods, including using (1) a single ArUco marker, (2) multiple ArUco markers (CharUco and ArUco board), and finally compared the performance results regarding the accuracy of pose estimation under various conditions, as presented in Table 9 below. The calibration process remained consistent across all approaches, involving capturing at least 40 images from various angles and orientations of the calibration object under similar environmental conditions. These images were then used to generate the camera matrix and distortion coefficients using a self-developed camera calibration code.

Table 9. Validation approach of multiple ArUco markers and its respective error differences. The x2 ArUco markers approach was chosen as a simplified representation for validating one axis only, assuming it has a similar effect on all other cases, involving x and y axes.

Method of Validation	Validation Axis	Error Difference (|Pose Estimated – Actual Distance|)		
		Chessboard Pattern	Single ArUco Marker	ArUco Board
Visual Inspection	z-axis	~22 cm	~25 cm	~15 cm
x2 ArUco markers	x-axis	~12 cm	~8 cm	~3 cm

There is notable improvement in accuracy when using the ArUco board, resulting in relatively lower error differences along the x and y axes. However, there remains a significant amount of error along the z-axis, although this is not considered a critical issue. The error tends to increase with distance along the z-axisi, but since the detection height is limited to a preset value of 2 m and ultrasonic sensors are implemented, this discrepancy is deemed acceptable.

6.3 Framework Selection

A software framework is foundational structure that dictates the architecture and offers a set of tools or libraries for developing applications, usually via predefined classes, functions, and conventions to streamline the development process. They are designed to allow developers to focus on implementing the high-level functionalities of the applications, relying on the framework to abstract low-level tasks to promote reliability, scalability, and performance of applications.

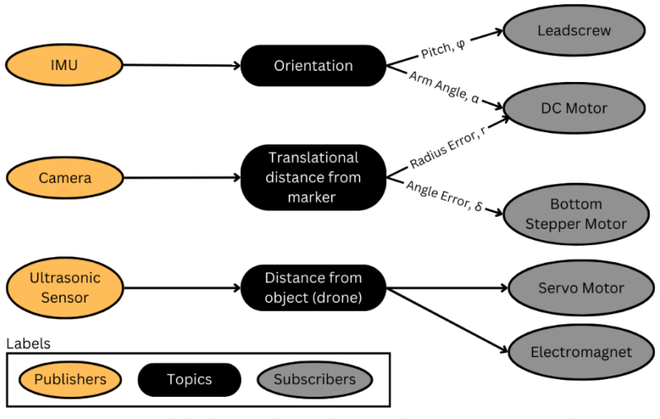


Figure 43. Publisher-subscriber architecture in ROS.

The Robot Operating System (ROS) has risen as the de-facto framework for robotics programming. It provides a collection of tools, most notably for low-level device control, package management and hardware abstraction. The proliferation of ROS also facilitates sharing and distribution of code among robotic programmers globally, encouraging collaborative robotics software development. The most prominent feature is its publisher-subscriber architecture which establishes decoupled and scalable communication between different parts of a system organised as "nodes". Publisher nodes are components of the program that are responsible for generating and sending messages through a channel or queue formally known as ROS topics, while subscribers are nodes that receive these messages. A simple ROS graph, as depicted in Figure 43, is drafted for the purpose of achieving the algorithm proposed in Figure 30. As an overview, publisher nodes sample sensor data and publishes them to relevant topics, whereas actuators subscribe to these topics and perform appropriate actions.

Despite the myriad of advantages of choosing ROS as the overarching framework, it was ultimately not chosen for this project due to several reasons. First, ROS is notoriously difficult to install. ROS receives an update every year [22], bringing new changes that may conflict with older versions. Moreover, an updated version of ROS - ROS 2 – was incepted in 2015 with substantial upgrades, but its novelty suffers from a lack of third-party plugins and documentation. The deprecating concerns of ROS 1 and the immaturity of ROS 2 were the final straw. Notwithstanding, the ROS 2 framework should be used in the future once it has passed its embryonic stage.

To maximize code portability and reduce development time, Python was chosen as the programming language to create the main algorithm. Although the lack of a framework like ROS jeopardizes scalability of the program, by writing the code in a portable and object-oriented approach, most of the code can be easily migrated to fit within a framework like ROS 2 if demanded in the future. To emulate the publisher-subscriber architecture drafted in Figure 43, the *multiprocessing* library was used to run multiple processes simultaneously. Each process is a python function that corresponds to a node, which includes:

- **IMU Reading**: Constantly reads the quaternions sampled from the IMU, processes them and publish the readings as Euler angles by updating its value on a shared block of computer memory. Equivalent to a ROS publisher.
- **Camera Image Processing**: Constantly captures a frame using the camera and perform pose estimation of an ArUco marker if it is detected. This updates the pose values on a block of computer memory. Equivalent to a ROS publisher.
- **Ultrasonic Sensor Reading:** Constantly computes the time of flight of an upward-fired ultrasound wave and compute the distance from an object atop the platform, then publishes this floating-point value on a block of memory. Equivalent to a ROS publisher.
- **DC Motor** and **Bottom Stepper Motor**: Reads the translation offset between the platform and the marker as published by the camera and actuate the DC motor and Bottom Stepper Motor to alter the extension of the platform. Equivalent to a ROS subscriber.
- **Leadscrew:** Reads the pitch angle sensed by the IMU and actuate the leadscrew to level the platform horizontally or to any desired pitch angle. Equivalent to a ROS subscriber.
- **Servo Motor** and **Electromagnet:** Reads the distance published by the ultrasonic sensor. If this distance is below a 3cm threshold, the servo motor and electromagnet will be activated as part of the hold-and-release mechanism.

The processing time for each process were compared between traditional linear programming, where one function executes after another, with concurrent programming using the *multiprocessing* module. Processing an image and pose estimating an ArUco marker comprises the largest percentage of execution time, nearing 70 ms, whereas sampling the IMU for quaternion readings and outputting corresponding Euler angles take 20 ms. Time taken for actuation for both approaches are almost negligible. It should be noted that each function ran slower individually in the case of concurrent programming due to the high computational costs of managing multiple processes. For example, image processing time is 65 ms for the linear program but took the concurrent program 67 ms.

Nevertheless, it was proved that parallelism reduces the response time of the algorithm compared to linear execution. The response times for dominant processes were measured and plotted as a bar chart in Figure 44. For linear programming, time taken to execute multiple tasks are the sum of their individual execution times, as shown by the stacking of bars that denotes the accumulation of time. With concurrent programming, the response time takes as long as the most time-consuming process, hence the bars

overlap, as highlighted in Figure 44. The nature of concurrency also implies that actuators operate on more recent sensor (IMU and camera) readings.

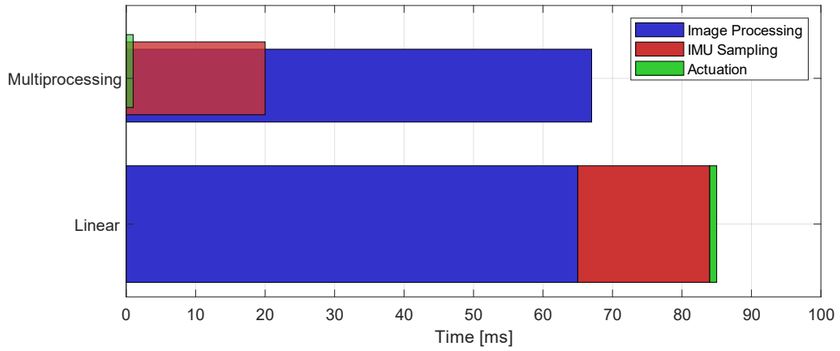


Figure 44. Process execution times of linear and concurrent programming.

6.4 Control Algorithm

During landing, the platform must be able to track the erratic behaviour of the drone that is contributed by its rapid inner-loop stabilization and external factors such as wind. Feedback control is essential in this context to reduce steady state error whilst fortifying the system against perturbations. There are two main control architecture candidates – Proportional-Integral-Derivative (PID) control and model predictive control (MPC). The former was chosen due to low computational complexity and ease of implementation. MPC has the potential to yield greater accuracy but requires extensive list of parameters and testing to obtain an adequate model. To realize the parameters for the PID controllers, a Simulink model was setup. The complete model is shown in Figure 45, with the methodology to model each subsystem discussed in subsequent subsections.

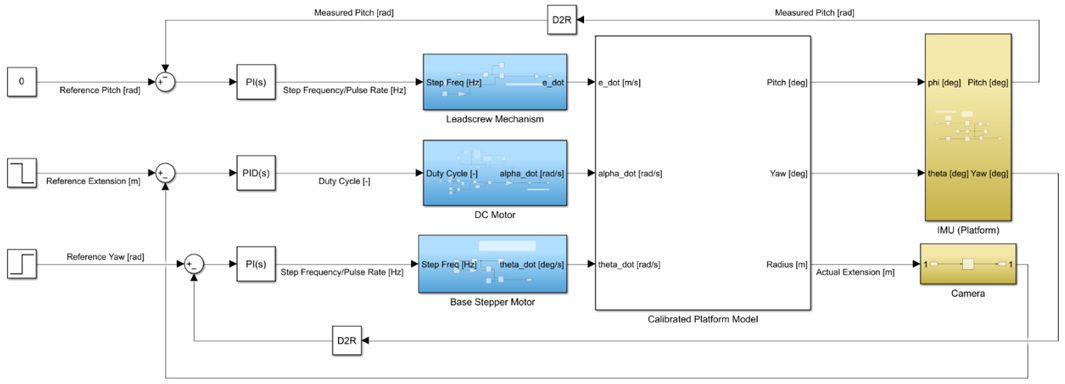


Figure 45. Full Simulink model with actuators, platform model, sensors and PID closed-loop feedback.

6.4.1 Plant Modelling

A kinematic model of the platform was imported as CAD files into Simulink via the Simscape Multibody plugin, which allowed the joint angles and distances to be altered. Three parameters were selected as inputs: the yaw rate $\dot{\theta}$, the angular velocity of linkage arm $\dot{\alpha}$ and the extension rate $\dot{e}$, as annotated in Figure 46. These three inputs correspond to the actuations of the bottom stepper motor, arm DC motor and leadscrew mechanism respectively. The Simscape model outputs are the pitch angle ϕ, platform offset D, and yaw θ.

The kinematic model provided insights into the relationship of each actuator. It was found that the leadscrew's extension holds a linear relationship with the rotation of the DC motor, which can be expressed as Eq. (4) below. However, this assumption only holds when the platform is flat and would break if the rover is positioned on an inclined surface.

$$y = 0.0624\alpha - 0.1917 \tag{4}$$

where y is the extension in metres and α is the degree of rotation in radians.

Furthermore, it was observed from the model that the usage of polar over cartesian coordinates greatly simplifies the algorithm. Using a reference frame where the Z axis is the vertical distance to the drone; the camera system will return the distance between the platform and the drone in cartesian coordinates X and Y. Through conversion to polar coordinates, each coordinate can be isolated to a single actuator –the DC motor to change the radius R by changing the arm angle α, and the bottom stepper motor to change the platform yaw θ as shown in Figure 46.

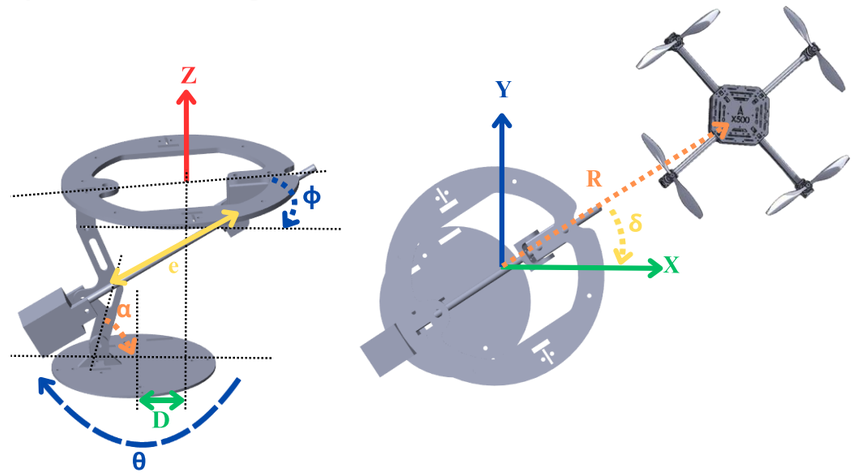


Figure 46. Overview of Simscape model parameters and coordinates.

This relationship is simulated in Simulink and the results are plotted to validate the assumption and shown in Figure 46 below.

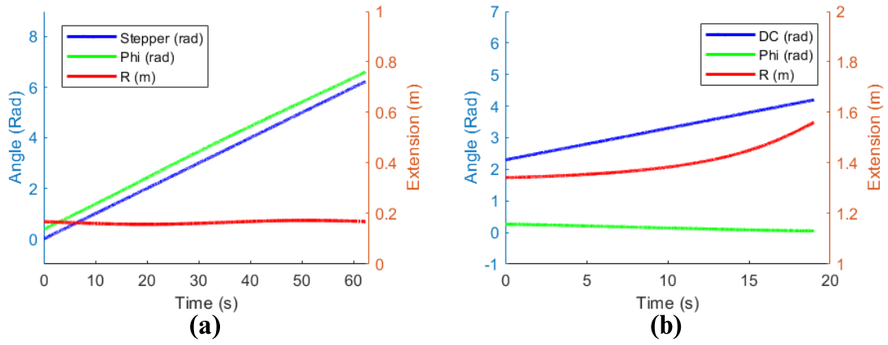


Figure 47. Relationship between stepper motor's angle (a) and DC motor's angle (b) against the polar coordinates.

As evident from the graph on Figure 46, (a), a single rotation of the stepper motor would result in a linear relationship with the θ; while having minimal effect on the R with only 0.01 m difference. Meanwhile, the effects of the DC motor shown in Figure 46, (b) on the R coordinates would be 0.1 m, while θ would only be 0.1 rad. Hence these results indicate that the model can be separated into two individual closed-loop PIDs enveloping each motor as actuators which would work simultaneously to reduce the error in the R and θ coordinates.

6.4.2 Sensor Modelling

The plant outputs are measured by the IMU (pitch ϕ and yaw θ) and the camera (platform offset θ). The IMU outputs are each modelled using zero-order hold and transport delay to simulate delays introduced by its sampling frequency (100Hz) and I2C communication to transmit 256 bits of quaternion data. A noise of $0.1°/s$ was also added as white noise, as per the datasheet of the IMU component. These are fed back and compared to reference inputs to the leadscrew and base stepper motor to close the loop.

The second type of sensor present in the model is the camera, which outputs translational offset between an ArUco marker with the platform. Due to the difficulty of quantifying camera noise, which will be a complex endeavour due to an amalgamation of various factors such as pixel density and shutter speed, visual distortion was neglected with the assumption that it was completely removed during calibration discussed in section 6.2.2.4. Nevertheless, a transport delay of 60 ms was added in accordance with the image processing time measured in section 6.3.

6.4.3 Actuator Modelling

Three actuators are modelled, which moves the docking platform by changing its yaw, arm angle and extension respectively. As discussed in the electronics section (section 7), the yaw is controlled by a stepper motor, the extension is changed via a leadscrew mechanism powered by a stepper motor, whereas the arm angle is controlled by a brushed DC motor. The angular velocities of the stepper motors are modelled kinematically following Eq. (5), with the shaft angular velocity ultimately obtained accounting for any micro stepping and gear ratios. Saturation blocks were later added to the model to limit their step frequencies to 3kHz and 5kHz stepper motors controlling the extension and yaw respectively, since higher frequencies were found to induce undesirable vibrations due to resonance in the actual system.

$$Angular\ Velocity\ \left[\frac{rad}{s}\right] = Step\ Frequency\ \left[\frac{step}{s}\right] \times Step\ Angle\ \left[\frac{rad}{step}\right] \tag{5}$$

The DC motor responsible for rotating the linkage arm is modelled as an equivalent RL electric circuit. With a known input voltage V_{in}, current I and estimated values of its armature resistance R, inductance L, back-emf constant K_E from its datasheet [23] – its angular velocity ω can be solved by through the differential equation shown in Eq. (6).

$$V_{in} = IR + L\frac{dI}{dt} + K_E\omega \tag{6}$$

6.4.4 Controller Tuning

Three controllers exist corresponding to the three actuators. For the stepper motor and leadscrew, the controller outputs control signals that vary their step frequencies, while the controller for the DC motor changes its duty cycle. Although the model is not exhaustive, it can yield useful preliminary control parameters to be used in its physical counterpart. Tuning involves a careful compromise between response time and stability. In the context of drone tracking characterized by erratic movements, response time is prioritized over accuracy since the latter can be rectified by the electromagnet located at the platform to catch the drone. Therefore, an underdamped system is preferred, and the requirements are set to a maximum rise time of 60 ms and a 10% overshoot.

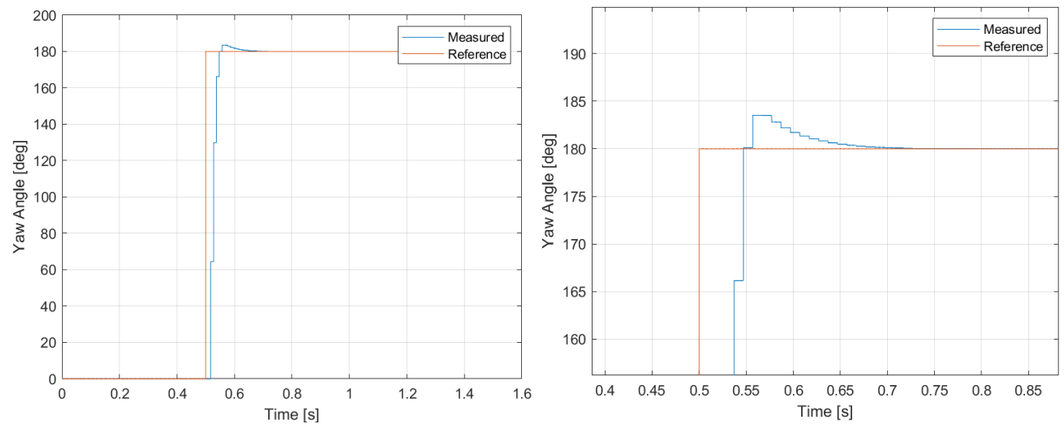
The dynamics of the bottom stepper motor are relatively unaffected by the upper subsystems; hence it is tuned separately with an ideal step response of 180° change in yaw. In contrast, a change in arm angle α directly impacts the platform pitch ϕ provided the extension e remains the same, so these controllers are tuned together. After inputting a step reference of a 10 cm change in extension e, corresponding to the typical landing accuracy of drones, the response of the DC motor and leadscrew is analysed based on its ability to achieve this extension while keeping the platform levelled.

The Ziegler-Nichols method was utilized as a heuristic method for tuning, followed by manual changes to customize its behaviour to the requirements. The IMU white noise exerted a noticeable high frequency response, hence the base stepper motor and leadscrew mechanism which operates on IMU readings was changed to proportional-integral (PI) controllers to avoid accentuating high frequency noise.

The tuned controller gains are listed in
Table 10 and satisfied the requirements in rise time and overshoot. These will serve as preliminary values for the physical system. The tuned response of the base stepper motor is shown in Figure 48 whereas the collective extension and levelling capabilities of the DC motor and leadscrew are modelled in Figure 49.

Table 10. The simulation-tuned proportional, integral, and differential terms for each controller

Controllers	Type of Control Output	Controller Gains			Performance Parameters	
		K_p	K_i	K_i	Rise Time [ms]	Overshoot [%]
Base Stepper Motor	Step Frequency [Hz]	1,500	0.005	-	20.0	2.0
DC Motor	Duty Cycle [-]	90	400	2.7	53.2	15.2
Leadscrew	Step Frequency [Hz]	40,000	0.01	-	40.0	4.1



(a) (b)

Figure 48. (a) Yaw response of bottom stepper motor to step reference of 180° and (b) its zoomed version.

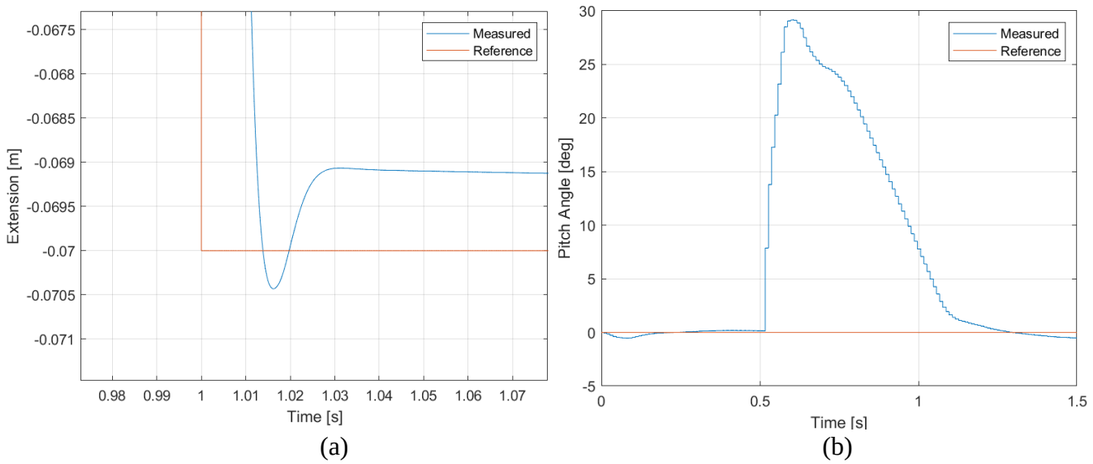


Figure 50. (a) Extension and (b) pitch response of DC motor and leadscrew to a step reference of 10 cm change in extension.

7. Electronics

The electronics section of the project was established to support the docking platform mechanisms and designs. The feedback from this section informs the capabilities of each prototype and how they could improve based on different restrictions and factors. Hence, the design process of selecting electronics begins with the design criteria of the mechanical design as emphasised in the Hardware Section of the report. The flow of the design process is clearly illustrated in Figure 52 below, with multiple iterations until the final electronic components and circuit is produced.

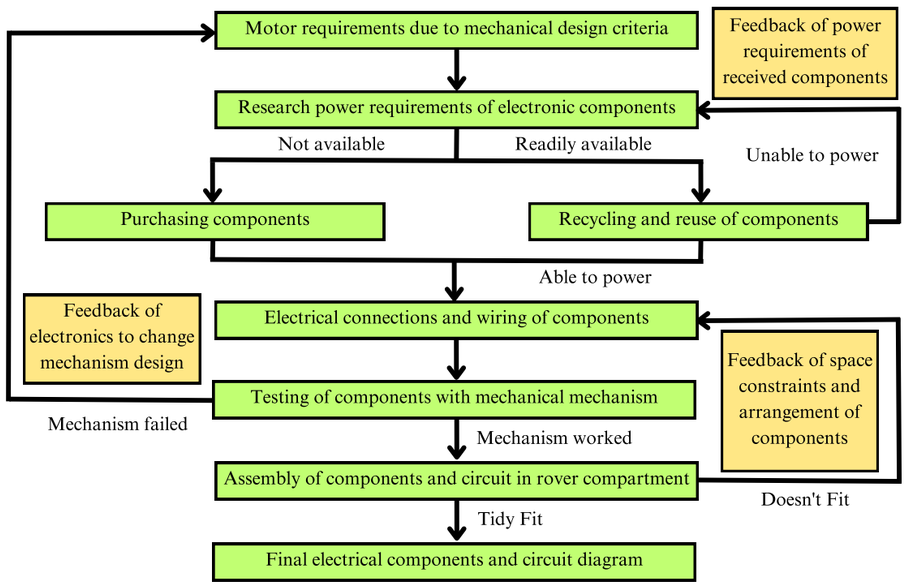


Figure 52. Design process of electronics

Based on Figure 52, the calculated requirements of the mechanisms are used to research on the suitable electronic components and their corresponding power requirements. An example of this would be the selection of each motor for the mechanisms using the calculated torque required as described in Hardware Section 5.3.2.1. Furthermore, multiple digital buck-boost convertors as shown in Figure 49, are chosen to provide the required voltage to each motor as it allowed rapid change of voltages for all components during testing. In addition, the motor drivers for each motor are researched to provide the necessary power to the motor. The project received some repurposed electronic components from previous projects which is utilised to reduce the cost of the project and reduce wastage. Initially, a Nema 17 stepper motor, DC motor, and motor driver A4988 was reused for the project and their required power requirements are researched and the corresponding motor drivers and convertors are purchased.

Figure 49. Digital buck-boost convertor

A linear actuator, as illustrated in Figure 53, was initially chosen as well to actuate the balancing of the top platform. However, during testing, it was found that the actuator was not responsive enough to help maintain a stable platform, due to its speed of extension (14mm/s). Furthermore, other factors such as the weight distribution and aesthetic of the actuator also contributed to the change to lead screw mechanism. This is discussed in depth in Section **Error! Reference source not found.**, but it is noted that the electronics provided additional feedback for the final decision to change the choice of actuators.

Figure 53. Linear Actuator

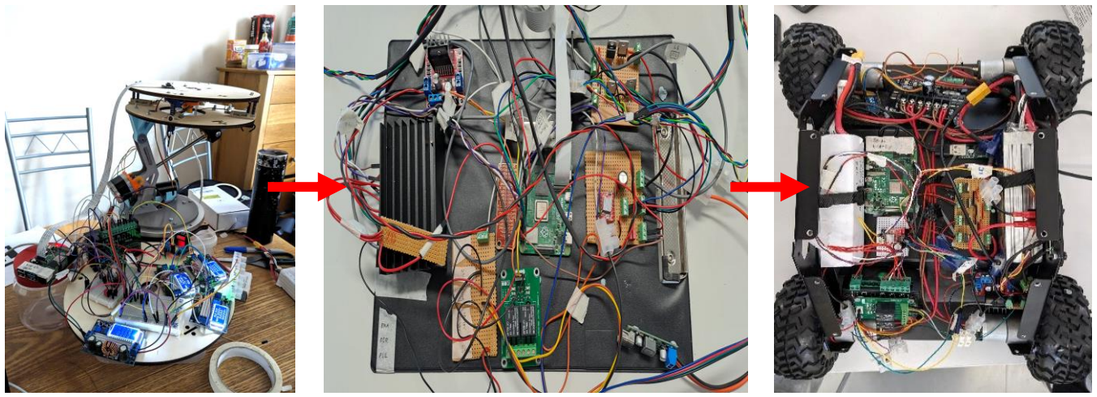


Figure 54. Development of electronic connections and wiring

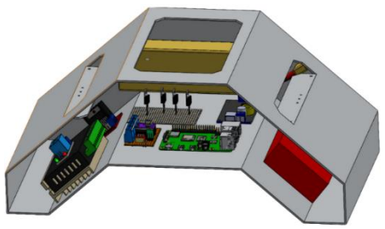


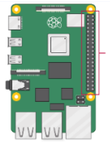
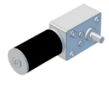
Figure 51. CAD model of rover compartment and components

Once all the components are finalised, they were connected via electrical circuit to test the docking platform. The flow of the progression of development of the circuit is shown in Figure 54. The initial circuit as shown on the left included 4 digital buck-boost convertors as it allowed for accurate voltage and current monitoring during testing. However, this would not fit in the rover's compartment and voltage regulators as shown in Figure 55 are chosen for the final circuit instead. The final components included three 12 V voltage regulators and one 5 V voltage regulator, soldered onto a strip board to ensure it is secured and reduce the space usage. To produce a clean and tidy connection, all the components are laid out on a board, shown in the centre of Figure 54,

Figure 55. Voltage Regulator

to arrange all the components. Furthermore, a CAD model of the rover compartment and all the electronic components that needed to fit inside is created as illustrated in Figure 51 to visualise the allocation of each component. The final connection and wirings fitted into the rover is shown on the right of Figure 54. The finalised list of electronic components is compiled in Table 11 with all their specifications and comments to justify their selections.

Table 11. List of Finalised Electronic Components

Component & Illustration	Specification	Comments
Raspberry Pi 4 Model B [24] 	• Processor: Quad core 64-bit ARM-Cortex A72 running at 1.5GHz • Voltage Input: 5 V	Accurately processes computer vision algorithm while providing flexibility with multiple GPIO pins. Fit within rover with minimal space occupied. Lower cost than Jetson Nano; faster processing than Arduino.
Nema 17 Lead Screw Motor [25] 	• Voltage Input: 12 V • Holding Torque: 40 N.cm • Step Angle: 1.8°	Provided the torque required to support the top platform while maintaining a flat landing surface for the drone. Lighter than a linear actuator and higher accuracy due to radial positioning.
Stepper Motor [26] 	• Voltage Input: 12 V • Holding Torque: 48 N·cm • Step angle: 1.8°	Provided the torque required to accurately rotate the base of the entire docking platform with radial positioning. Reduced overall cost and improved sustainability of the project.
DC Motor GMW40 [27] 	• Voltage Input: 12 V • Torque: 5.85 Nm • Stall Torque: 12.6 Nm	Provided the high torque required to extend and retract the arm mechanism with the weight of the drone included. Stepper motors with similar cost have much lower torque and radial accuracy is less important.
Motor Driver A4988 [28] 	• Voltage Input: 35 V • Current Output: 1 A - 2 A	Provided stable voltage and current to power leadscrew motor and control it. Its compact size allows for soldering onto a strip board, effectively minimizing space usage.
Motor Driver TB6600 [29] 	• Voltage Input: 9 V - 42 V • Current Output: 0.5 A - 4 A • Micro steps: 7	Provided stable voltage and current to power stepper motor and control it. The motor driver allowed micro stepping, facilitating reduced rotation speed, and enhancing precision in movement.
Motor Driver L298N [30] 	• Voltage Input: < 50 V • Current Output: 3 A	Provided stable voltage and current to power DC motor and control it. Small, cheap, and easy to use.
Electronic Relay 	• Voltage Input: 5 V • Current Output: 10 A • Logic Signal: 3.3 V & 5 V at 5 mA	Enabled the control of electromagnet and LED during the capture and release of drone.
Camera Module [31] 	• Resolution: 11.9 megapixels • Video: 1080p50	Camera module compatible with RPI with clear image capture that provided clear and real time camera vision processing.
9 DOF IMU [32] 	• Voltage Input: 3.3 V • Current input: 12.3 mA • DOF: 9	The 9 DOF are directly translated into cartesian coordinates, unlike other commercially available IMU with similar cost that require additional calculations.
Servo Motor 	• Voltage Input: 5 V • Current Input: 100 mA • Torque: 23 N·cm	Small and cheap motor that could provide sufficient torque to trigger the lock and release mechanism.
Ultrasonic Sensor 	• Voltage Input: 5 V • Current Input: 20 mA • Range: 2cm - 400cm	Cheap alternative to easily obtain distance readings to validate camera vision tracking results and trigger the release of top platform.
Electromagnet 	• Voltage Input: 5 V	Sufficient strength to attract and hold drone once it lands on the top platform. Compact size and could fit easily on the top platform.

Once all the electronic components required to power the docking platform are finalised, an electrical circuit was built to provide clear and concise wiring planning and placement of all components within the rover. It also provided a clear guidance on the wiring connections to the Raspberry Pi, which reduced the time required to set up the project after each modification. The addition of a switch to the circuit also allowed for convenient toggling of power to the circuit, improving seamless activation and deactivation of the docking platform. Moreover, the addition of an extra Inertial Measurement Units (IMUs) has been instrumental in acquiring accurate tilt angles for both the rover and its top platform. The first IMU is used to obtain the tilt angle of the top platform to ensure it is constantly flat to capture the drone. Meanwhile, the second IMU is responsible for obtaining the angle of the reaching arm from the DC motor to prevent overextension and damaging the docking platform. This enhancement significantly improved the overall functionality and performance of the system. Furthermore, the operational time of the docking platform is calculated below to estimate the amount of time that it can be deployed on missions. The battery used for the docking platform in the rover is identical to the battery on the drone with specifications included in Table 2.

$$Battery\ Life\ (h) = \frac{Battery\ Capacity\ (Ah)\ *\ Safety\ Factor}{\sum Rated\ Current\ (A) * Load\ Factor} \quad (7)$$

From Eq (7), a safety factor of 0.7 is assumed to ensure the battery is not damaged from deep discharge and the depth of discharge does exceed 70%. This is to ensure the battery does not degrade over time and the lifecycle is not reduced. The operational data and values obtained from the list of components are compiled in Table 12 below.

Table 12. Power requirements of electronic components for battery life calculations

Components	Rated Current (A)	Idle Current (A)	Load Factor
Raspberry Pi [24]	3.00	0.40	1.00
Nema 17 Motor [25]	1.68	0.50	0.80
Stepper Motor [26]	2.50	0.50	0.50
DC Motor [27]	0.92	0.26	0.50
Electromagnet	0.30	0.01	0.01
Battery (source) [9]	Capacity: 5 Ah		
Total Current Drawn (A)	8.40	1.67	-

Battery life (During operations)	Battery life (Idle condition)
0.57 hours	2 hours

As shown in Eq (7) and values from Table 3, the calculated battery life of the docking platform during full operation of tracking and movement would be 35 minutes. Meanwhile, the idle life of the battery calculated using the idle or leakage current of all the components when not in use could go up to 2 hours. This indicates that the project could be deployed on long range missions while the platform is only situationally used to track and allow the landing of drones.

The peak current drawn from the circuit is calculated to be 8.4 A. An electric fuse is added to the circuit to ensure the electrical components are safe from short circuits caused by stray conductors or water leakages. This would cut-off the current flowing to the components when there is a surge of current exceeding the cut-off value that could potentially damage the other components. A safety factor of 1.2 is considered when applying the fuse as it should not allow any current that could damage the electronic components through [33]. The calculated fuse is rated at 10 A and integrated into the system, any current exceeding this value would cause the fuse to burn and cut off the circuit from the power supply. The fuse will need to be replaced after it has been burnt to protect the circuit.

8. System Integration Testing

During system integration, holistic operation was commenced after verifying the operation of individual sensors and actuators. The response time of the system was found to satisfy the initial requirements of 100 ms with *multiprocessing*. Amongst the many processes spawned, image processing comprises the bulk of the execution time about 71 ms, which is interpreted as the response time since all processes run concurrently. The subsequent sections outline the procedures and results obtained when testing the desired functionalities of the docking platform.

8.1 IMU Values Test

First, the readings of the IMU were verified as its accuracy significantly influences the system behaviour. Through various manual and automatic tests, which involved altering the arm angle and leadscrew extension by hand or by pre-coded programs, the sensor accuracy was determined to be within an acceptable range of $\pm 3°$ through crude measurements with a protractor. Next, the yaw θ, pitch ϕ and arm angle α are measured together to ensure they are independent. Figure 56 shows the changes in angles measured by the IMU during an experiment where the platform is yawed manually. As expected, only the measured yaw is varied whereas the pitch and arm angle remained largely unaffected. However, repeated testing reveals occasional presence of artefacts, taking the form of immediate spikes in the IMU readings, which can be seen near the 10.5 s timestamp in Figure 56. Such occurrence is more common during sudden movements; hence its root cause was postulated to be cable movement. To resolve this problem, a simple conditional statement was added to the IMU reading process which ignores an IMU angle reading if consecutive readings differ by more than 50 degrees, which removed some but not all spikes in sensor readings. This issue may be resolved using averaging or more sophisticated algorithms such as Kalman filters but was not implemented due to time constraints.

In addition, a potential risk of actuators exceeding its intended range of motion was observed - predominantly the linkage arm sweeping over the intended range. Risks include damaging the leadscrew stepper motor and/or the base entirely. It was decided that a second IMU was to be added to the linkage arm to resolve this issue by measuring the arm angle. Additional merits of an additional IMU include redundancy in case of a sensor failure, as well as the possibility of integrating both sensor readings in advanced algorithms such as an EKF, though this was not explored in this project and will be left as future work.

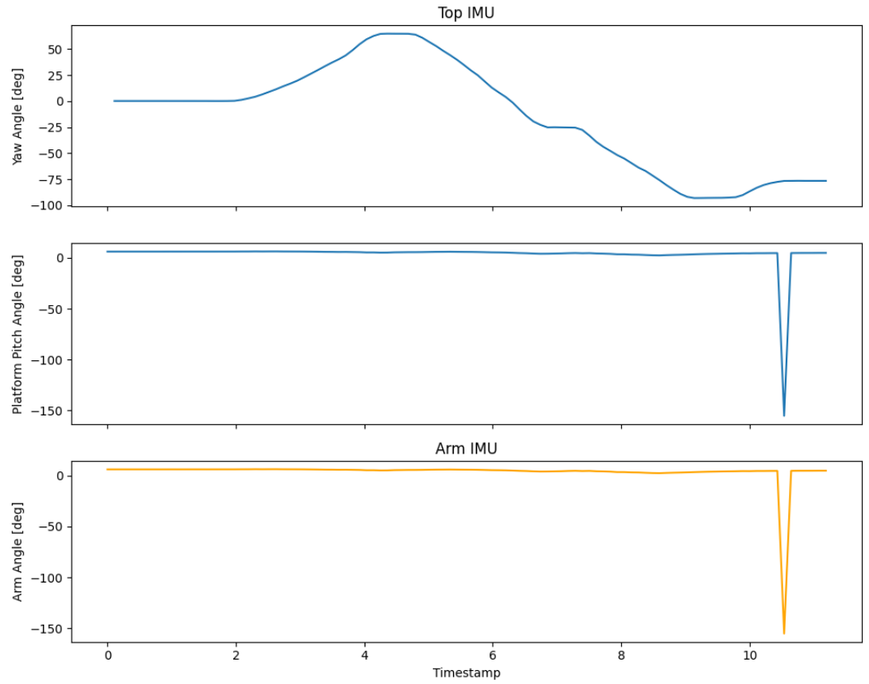


Figure 56. Changes in IMU readings as the yaw of the platform is manually changed.

8.2 Platform Levelling Test

The platform levelling capability of the platform was tested with two methods. The first test involved handheld manoeuvres that manipulate the inclination of the rover, followed by visual inspection of its levelling capability, which was deemed acceptable. This is followed by a more systematic approach of pitch measurement, where the arm angle is swept back and forth to its limits using a pre-coded program, while the leadscrew mechanism actively balances the platform via closed-loop feedback. The results are shown in Figure 58. It is observed that the platform pitch remained relatively level initially, but large oscillations are found at extreme angles of operation, indicating an underdamped. This is likely due to system non-linearities and difference in weight distribution associated with a change in arm angle, both of which are unmodelled in the Simulink simulation. This led to further PID tuning which improved the response but was ultimately unresolved.

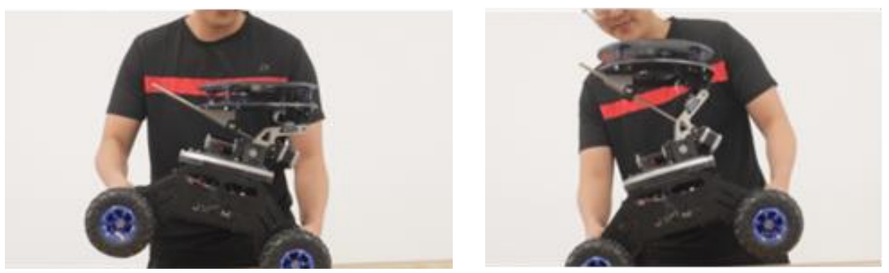


Figure 57. Platform self-levelling test.

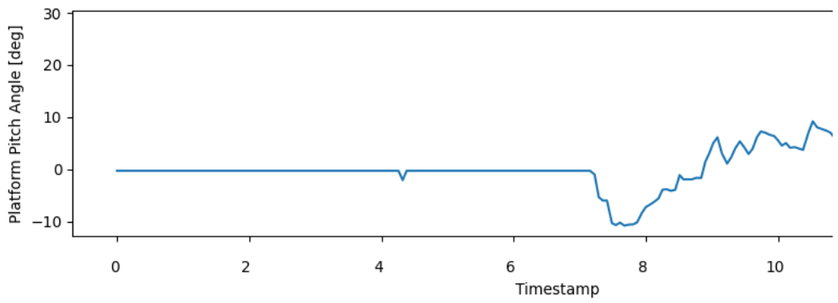


Figure 58. Pitch angle reading sampled from the IMU during self-levelling test where the arm angle is swept from 50° to 130° starting from t=4s.

8.3 Tracking Test

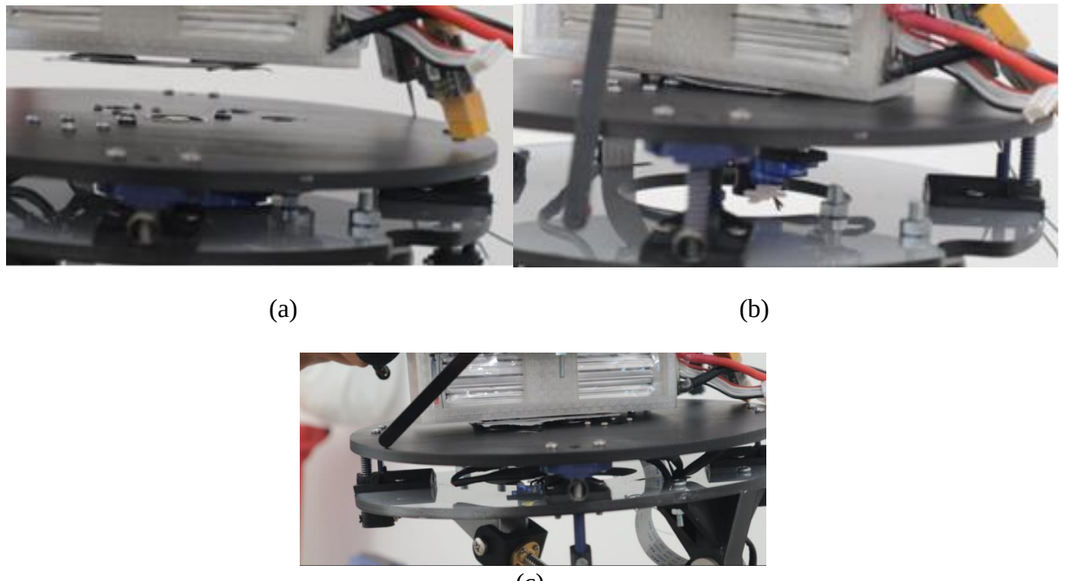
Tracking test first entailed visual inspection of a handheld marker, as illustrated in Figure 59 (a) and (b). Marker detection was verified using print statements embedded into the program and through inspecting the captured frames via VNC. However, several factors affected the success rate of marker detection, including light conditions and marker size, despite them being considered during marker selection (section 6.2). The former was especially prevalent as shown in Figure 59 (c), where the presence of concentrated light source severely compromised the camera's ability to detect a marker. Due to the limited field of view of the camera, smaller markers were found to perform better during the final phase of the docking sequence, with markers of size 3 cm x 3 cm exhibiting much higher detection consistency at the range of 2 m height compared to the original size of 6 cm x 6 cm.

(a) (b) (c)

Figure 59. Experiment of marker tracking using (a) handheld markers, (b) marked handheld drone, and (c) a captured image with concentrated light conditions during the tracking test.

8.4 Hold and Release Mechanism Test

Finally, several tests were conducted to determine the reliability of the hold-and-release mechanism. Since the activation of this mechanism relies on the ultrasonic sensor, the sensor readings were logged and analysed after each test, which were verified with approximate measurements done with a ruler. Upon integration with the servo motor, where the servo motor rotates 90° to release the top layer. It was found that the release mechanism has a 70% success rate. Failures are all attributed to an uneven release, as seen in Figure 60.



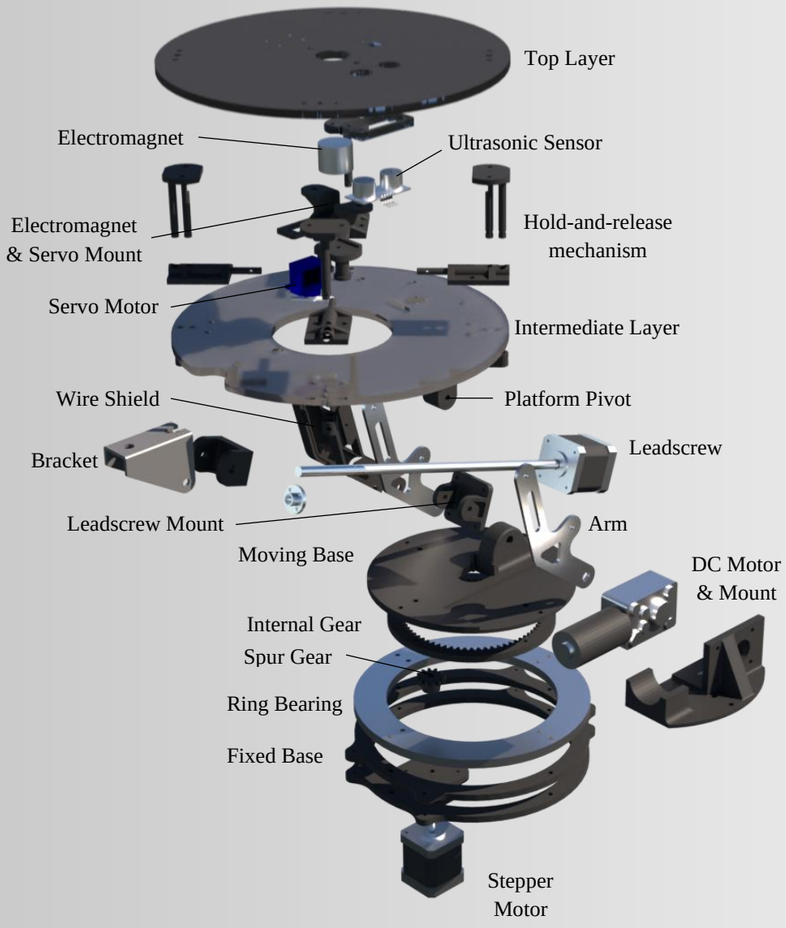
(a) (b)

(c)

Figure 60. Tests of hold and release mechanism in its (a) initial, (b) released position and (c) a failed uneven release.

9. Final Design Proposal

9.1 Hardware



Top Platform

- Platform surface made of acrylic.
- Houses electronic components including IMU, camera, servo motor and electromagnet.
- Z-axis motion with spring-loaded hold-and-release mechanism.
- Mounts and mechanisms are 3D printed.

Arm and Actuator

- Optimised arm geometry and attachment point position.
- Maximum extension range of 198 mm.
- Wire shield with dedicated cable path for wire management.
- Platform tilt controlled via leadscrew extension/retraction.

Base

- Rotational motion driven by stepper motor paired with an internal meshing gear system.
- A ring bearing is sandwiched between the moving and fixed base layers.
- DC motor mount with easy assembly procedure.
- Integrated arm pivot for enhanced stability.

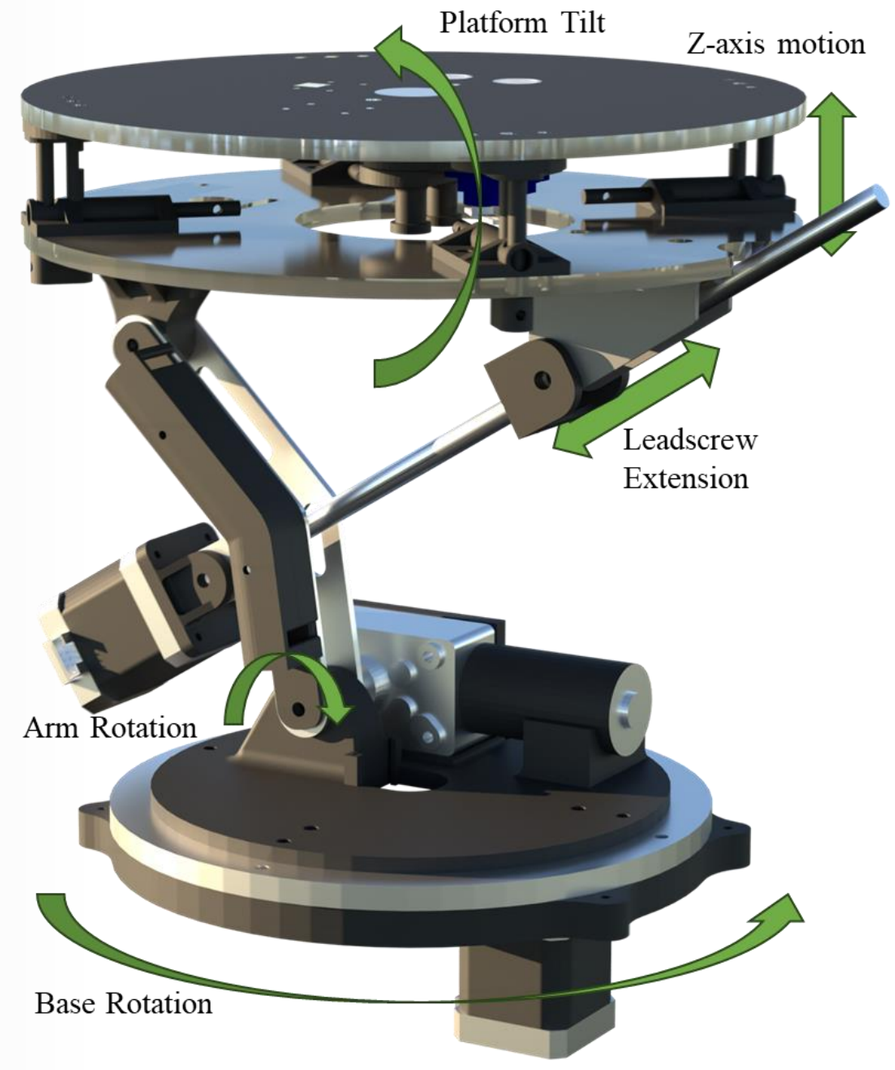


Figure 61. Hardware Final Design

9.2 Electronics

Overall electrical circuit is connected as shown in **Error! Reference source not found.**, including all electronic components and the Raspberry Pi pin connections as well. The fuse and switch are also added in the beginning of the circuit to cut off power when necessary. The electronics of the project can be divided into 5 main subsystems as evident in **Error! Reference source not found.**, which includes the power system, detection system, inertia measurement unit system, actuation system, and capture system.

The inertial measurement unit system only includes 2 IMU sensors which are essential to provide inputs for the control algorithm. The first IMU ensures the top platform is flat while the second IMU ensures the platform does not exceed its rotational limit which could cause damage.

The detection system mainly includes the camera which tracks the Aruco markers and provides input as coordinates to guide the landing platform. Additionally, an ultrasonic sensor is utilised to validate the vertical distance of the platform to the drone and is the trigger for the catch mechanism. The servo motor rotates when triggered and releases the spring-loaded top platform and turns on the electromagnet.

The power system provides sufficient power for all the electronic components in the project. A designated buck convertor is used to provide 5 V to the RPI with USB port adaptors for connection. A strip board consisting of three 12 V and one 5 V voltage regulator powers the three actuators and additional components such as the electromagnet. The battery life is calculated to be 35 minutes during operation and 2 hours of idle condition.

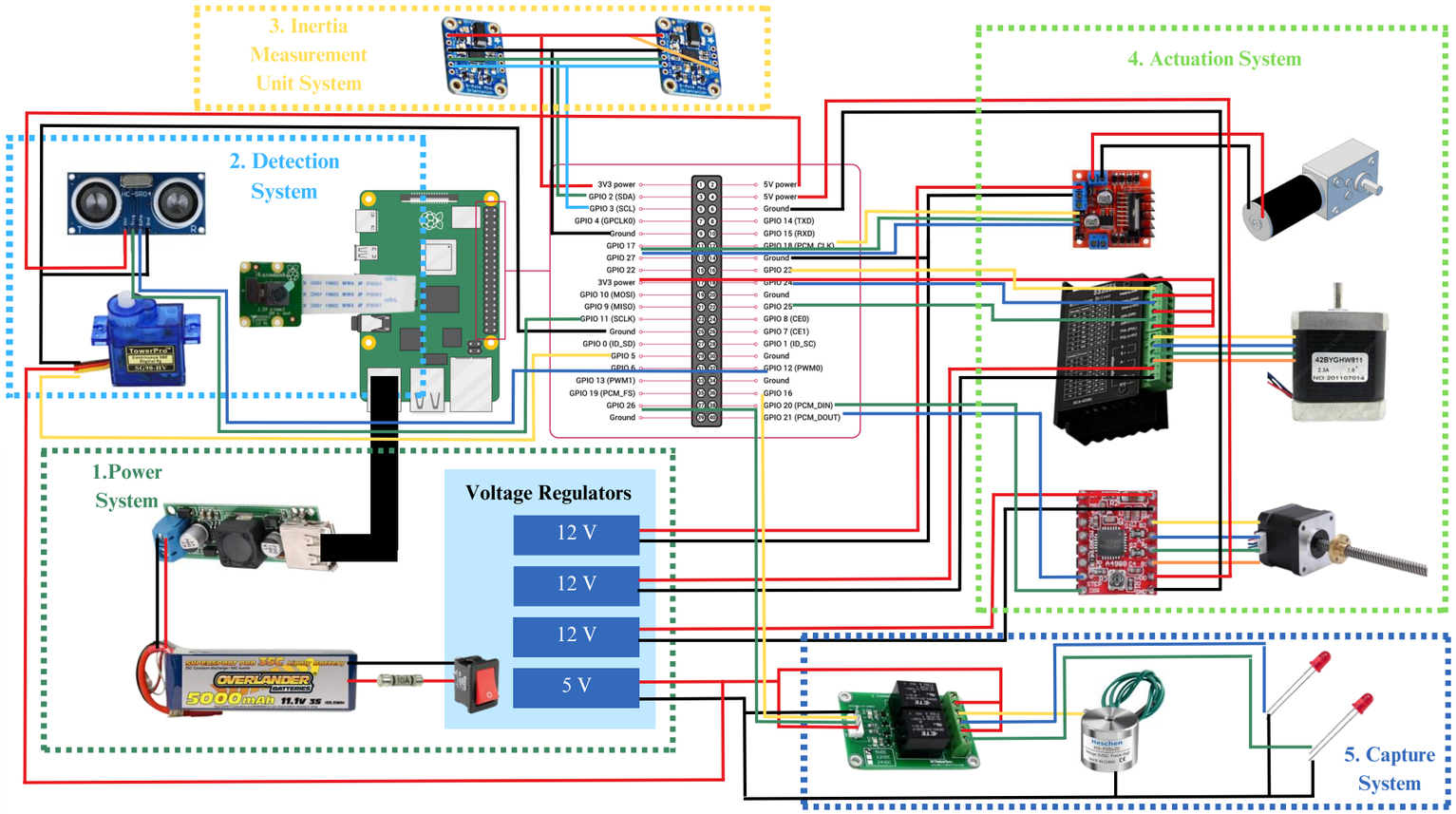


Figure 62. Full circuit diagram with all electronic components and connections

The actuation system includes 3 actuators and their corresponding motor drivers. Each motor is specifically chosen to meet the mechanical requirements of the landing platform. The power requirements and outputs of each motor is controlled by the motor drivers which receive signals from the Raspberry Pi.

The capture system consists of the electromagnet which is responsible for attracting and keeping the drone attached to the top platform. It is controlled by a relay that receives signals from the Raspberry Pi. Additionally, 2 LEDs are used to indicate the status of the landing platform during operations.

9.3 Software Architecture

The final designed software architecture is detailed in **Error! Reference source not found.**. As mentioned in section 6.3, *multiprocessing* was used to implement a collection of concurrent programs. Each program takes the form of a python function, collectively gathered in a single script named `main.py. These processes are run in parallel to leverage the multiple cores of the processor powering the Raspberry Pi. Fixed blocks of computer memories are allocated to store sensor readings, which are constantly written and read by publisher and subscriber processes.

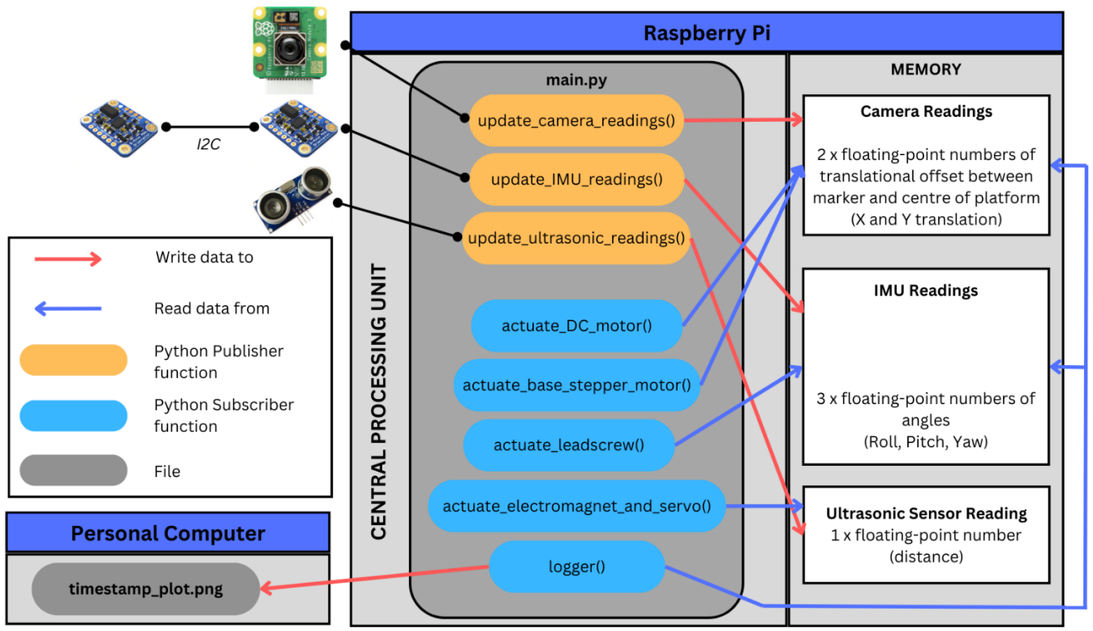


Figure 63. Software Architecture.

Project Review

9.4 Achievements

The main achievements of the final design were the development of a fully operational docking platform capable of autonomously detecting, tracking, and securing a drone. In its completed form, the system showcased visual acquisition of the drone via ArUco marker detection, approached and secured the drone in place with the aid of an electromagnet, coupled with the hold and release mechanism, representing a seamless landing process. The detection height was tested successfully up to 2 m.

Furthermore, the system achieved a 360-degree detection range that enables drone capture while maintaining platform stability. Although time constraints prevented autonomous flight implementation, key components were successfully demonstrated through tracking improvisations, supplemented by manual flight tests, laying the groundwork for system feasibility and future development. These key components include:

- **Mechanical Design**: The docking platform can extend horizontally and vertically by up to approximately 20 cm and 2.5 cm respectively, sweeping across an angle of 93° despite its compact size. The design is sturdy and has great stability, capable of supporting payloads up to 2.5 kg.
- **Software Architecture**: The system features a marker tracking system with an accuracy of ±5 cm, complemented by resilient control algorithms enabling platform levelling and tracking functionalities. Concurrent programming using the multiprocessing module substantially enhances the algorithm's response time, supported by PID-controlled actuators.
- **Electronics**: A single battery efficiently powers all electronic components, enabling control over various mechanism motions. This includes IMU-based positional input reception for base rotation, linkage arm angle adjustment, and extension. Additionally, it allows the activation of the hold and release mechanism and electromagnet to secure the drone.
- **Avionics**: Basic flight controller configurations were successfully implemented. Safe and reliable manual test flights were conducted, demonstrating the system's readiness for further development on autonomous capabilities.

9.5 Project Management

The team successfully delivered a fully operational docking platform due to a comprehensive set of milestones throughout the year, which can be seen in the Gantt chart in the Appendix. Furthermore, the team effectively managed resources by adhering to a budget plan, ensuring that the manufacturing and assembly of the docking platform was kept within budget allowance, which can be seen in the Appendix.

9.6 Project Aim & Objectives

The team believes that majority of the project aims and objectives defined at the beginning of the design process were successfully achieved, as stated below:

- Designed a reliable docking platform that serves as a mechanical interface between the drone and rover provided, allowing it to be integrated with modular systems of similar purposes in detecting, tracking, and capturing something.
- Electronic components were sourced based on the suitability in regard to its specifications and functionalities along with compatibility with the mechanical design to ensure that the whole system can run smoothly for a certain amount of time under an allocated budget.
- The computer vision algorithm is capable of detecting ArUco markers from a certain distance range. Pose estimated coordinates will then be utilised as inputs for the control algorithm.
- The control algorithm, including platform levelling and tracking functionalities was demonstrated by manually maneuvering the drone within the designated detection area, serving as a simplified representation of autonomous landing.

9.7 GDP Requirements

9.7.1 Innovation

Innovative solutions across various aspects were implemented to address the challenge presented. This includes the development of a novel docking platform design with an emphasis being placed on serving as a stable mechanical link between a drone and rover. The robust software architecture supports concurrent programming which improves the overall performance of the computer-vision-based detection and control algorithms. A combination of platform levelling and tracking functionalities of the docking platform will act as the basis of future work revolving around autonomous flight implementation.

9.7.2 Process

As a self-proposed endeavour, this project serves as the first iteration of H.A.N.G.A.R docking platform. To effectively navigate the design process from ideation to prototype to final product, it was key for our group to start by thoroughly understanding the design brief and the requirements of the project. The generated new conceptual ideas were extensively compared and reiterated. Rigorous testing of assembled components and electronic components, coupled with iterative code development for computer vision and control algorithms, ensured the implementation of an optimal solution for our system. Throughout this process, all subsystems worked concurrently, where this parallel development ensures that each aspect of the design aligns seamlessly with the overall objectives, leading to a cohesive and successful product.

9.7.3 Communication

The team maintained effective communication channels, which proved crucial for our project, given the interconnected nature of our work across all four subsystems. We prioritized clear and transparent communication between subsystems, holding weekly internal and supervisor meetings to ensure everyone stayed informed and aligned throughout the entirety of the project. The team also communicated our project by incorporating figures and tables into the report, enabling both technical and non-technical audiences to view the full process of designing a docking mechanism. These visual aids illustrated the complexity involved in developing a system capable of detecting, tracking, and capturing a drone, particularly within the context of integrating drone and rover functionalities. Heavy emphasis was placed on the video deliverables, so that information regarding the project could be communicated via different mediums.

9.7.4 Project Sustainability

During the project, the team emphasised on eco-conscious practices such as recycling and repurposing electronic components acquired from previous projects. This method significantly reduced the waste generated throughout the project's development phase as prototypes could utilise these components first before the concept is finalised. Moreover, the commitment to sustainability extended to the manufacturing phase as recycled cardboard boxes and scrap plywood was leveraged to produce the initial prototypes. Additionally, all the manufacturing and assembly for the project is conducted within the university's design studio and EDMC, hence reducing transportation emissions. The final product primarily composes of stainless steel and acrylic which are chosen for their durability and longevity and they are recyclable after its lifecycle. The material selection is aimed at extending the product's lifecycle and reducing waste generation during replacements. Overall, the project's focus on sustainable practices reflects the team's commitment on minimising environmental impact.

9.8 Future Work

To continue the work done on the project and to realise the capability of the system under autonomous conditions, any future team should take into account the following considerations to improve the functionality and performance of the docking platform, as well as implementation of autonomous flight.

From the avionics standpoint, the next phase involves integrating autonomous flight without GPS into our docking mechanism, which we were not able to achieve due to time constraints and reasons stated in Avionics section above.

In terms of hardware and electronics, conducting more comprehensive tests will reveal areas for potential redesign to enhance system longevity. A key challenge encountered during testing was wiring management, suggesting future teams could address this by modifying the rotating base section and wiring system to prevent tangling and potential malfunctions. Moreover, exploring manufacturing methods like injection moulding could reduce the number of separate parts used. Additionally, efforts to reduce noise and vibration, waterproofing for outdoor use, and implementing integrated circuits to reduce electronic component footprint could be beneficial.

On the software side, further exploration into the feasibility and accuracy of using two ArUco markers of different sizes for CV detection, especially at varying heights, is warranted. ROS 2 should be evaluated as the overarching software framework once it matures, as it enhances code scalability. Additionally, considering faster programming languages like C and C++ instead of Python can further reduce response time. Integration of more sophisticated algorithms, such as Extended Kalman Filter (EKF), can address common issues like sensor readings mentioned in Section 6 (Testing & Integration).

10. Conclusion

Throughout the duration of this GDP, the team has consistently engaged in research, design, implementation, and testing of new systems aimed at developing the optimal docking platform design. The completed docking platform can precisely detect, track, and secure a drone onto it with the aid of an electromagnet within a detection height of up to 2 m, all while maintaining platform stability.

The mechanical design of the docking platform prioritises stability and strength through the utilisation of mechanisms, including the arm actuator, hold and release and surface levelling, along with careful material selection. Integration of these mechanisms allows the docking platform to operate smoothly, covering a significant area through movement in three core DOFs. The platform boasts a full 360° rotation capability, along with horizontal extension of up to 20 cm and vertical extension of 2.5 cm. These features not only enhance landing tolerance but also effectively counteract inclination, ensuring secure attachment of the drone to the platform.

The current software architecture enables remote programming, launch, and monitoring of the docking platform from a ground station. This architecture incorporates a successful development of a concurrent programming framework utilising multiprocessing modules to emulate the software architecture in ROS. As a result, multiple algorithms, including computer-vision-based tracking based on ArUco markers and control algorithms with tracking and platform levelling capabilities, can run simultaneously, reducing overall response time compared to sequential/linear programming.

These achievements are attributed to the iterative design process, manufacturing techniques, and simulations performed, including FEA and Simscape Proportional-Integral-Derivative (PID) control platforms. These tools allowed the team to efficiently test prototypes within a secure environment for refining, before proceeding to manufacture and assemble the fully operational final design.

Achieving a full demonstration of drone hovering, whether manually controlled or autonomously in a GPS-denied environment, and subsequent landing onto the docking platform was not realised due to time constraints. However, a scenario of a hand-held drone replicating real-landing approach was demonstrated, showing the functionalities of vision-based tracking, platform levelling and tracking capabilities of the docking platform. Future work will be required to achieve a complete technological demonstration of the final system.

11. References

[1] P. R. Palafox, M. Garzón, J. Valente, J. J. Roldán, and A. Barrientos, "Robust Visual-Aided Autonomous Takeoff, Tracking, and Landing of a Small UAV on a Moving Landing Platform for Life-Long Operation," *Applied Sciences*, vol. 9, no. 13, p. 2661, 2019. [Online]. Available: https://www.mdpi.com/2076-3417/9/13/2661.

[2] *Market Insights & Analysis: Global Drone Services Market (2024-30):*. [Online]. Available: https://www.marknteladvisors.com/research-library/global-drone-services-market.html

[3] "Global Drone Services Market to be Valued USD 16 Billion in 2023 & Set to Thrive at a CAGR of 28.5% During 2024-30." https://www.marknteladvisors.com/press-release/drone-services-market-growth (accessed.

[4] M. Galimov, R. Fedorenko, and A. Klimchik, "UAV Positioning Mechanisms in Landing Stations: Classification and Engineering Design Review," *Sensors*, vol. 20, no. 13, p. 3648, 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/13/3648.

[5] "NVIDIA JETSON NANO DEVELOPER KIT," NVIDIA, 2024. Accessed: 30/10/2024. [Online]. Available: https://siliconhighway.com/wp-content/gallery/jetson-nano-devkit-datasheet-936542-US-hr.pdf

[6] "CubePilot Cube Orange Flight Controller," 2024. Accessed: 30/10/2024. [Online]. Available: https://docs.px4.io/main/en/flight_controller/cubepilot_cube_orange.html

[7] "EDU-450," CubePilot, 2024. Accessed: 30/10/2024. [Online]. Available: https://docs.cubepilot.org/user-guides/cubepilot-ecosystem/cubepilot-partners/hexsoon/multirotor-frame/edu-450

[8] "Multistar Elite 2216 920KV Multirotor Motor Set (2xCW 2xCCW)." https://hobbyking.com/en_us/multistar-elite-2216-920kv-multirotor-motor-set-2xcw-2xccw.html?___store=en_us (accessed.

[9] "Overlander 5000mAh 14.8V 4S 35C Supersport Pro LiPo Battery 2578." https://www.modelshopleeds.co.uk/catalog/product_info.php?products_id=17318&gad_source=1 (accessed.

[10] Y. Xu, Z. Chen, S. Wang, and J. Wang, "An Active Landing Recovery Method for Quadrotor UAV: Localization, Tracking and Buffering Landing," *IFAC-PapersOnLine*, vol. 56, no. 2, pp. 3366-3372, 2023/01/01/ 2023, doi: https://doi.org/10.1016/j.ifacol.2023.10.1483.

[11] P. Wang, C. Wang, J. Wang, and M. Q. H. Meng, "Quadrotor Autonomous Landing on Moving Platform," *Procedia Computer Science*, vol. 209, pp. 40-49, 2022/01/01/ 2022, doi: https://doi.org/10.1016/j.procs.2022.10.097.

[12] L. W. Young, "Autonomous docking station for drones," 2016. [Online]. Available: https://patents.google.com/patent/WO2017109780A1/en.

[13] L. Shapiro, "Computer Vision," 2000. [Online]. Available: https://cdn.preterhuman.net/texts/science_and_technology/artificial_intelligence/Computer%20Vision%20-%20Linda%20Shapiro.pdf.

[14] J. Bautista, "UAV Docking." [Online]. Available: https://scholarworks.calstate.edu/downloads/d791sn96p.

[15] T. Yang *et al.*, "A Ground-Based Near Infrared Camera Array System for UAV Auto-Landing in GPS-Denied Environment," *Sensors*, vol. 16, no. 9, p. 1393, 2016. [Online]. Available: https://www.mdpi.com/1424-8220/16/9/1393.

[16] C. Mart´ınez, "Trinocular Ground System to Control UAVs." [Online]. Available: https://oa.upm.es/5720/2/INVE_MEM_2009_70553.pdf.

[17] A. Tsai, P. Gibbens, and R. Stone, *Terminal Phase Vision-Based Target Recognition and 3D Pose Estimation for a Tail-Sitter, Vertical Takeoff and Landing Unmanned Air Vehicle*. 2006, pp. 672-681.

[18] S. M. Chaves, "NEEC Research: Toward GPS-denied Landing of Unmanned Aerial Vehicles on Ships at Sea." [Online]. Available: https://stephenchaves.github.io/pdfs/schaves-2015a.pdf.

[19] Z. Fucen, S. Haiqing, and W. Hong, "The object recognition and adaptive threshold selection in the vision system for landing an Unmanned Aerial Vehicle," *2009 IEEE International Conference on Information and Automation, ICIA 2009*, 06/01 2009, doi: 10.1109/ICINFA.2009.5204904.

[20] "Detection of ArUco Marker." https://docs.opencv.org/4.x/d5/dae/tutorial_aruco_detection.html (accessed.

[21] "OpenCV Camera Calibration." https://docs.opencv.org/4.x/dc/dbb/tutorial_py_calibration.html (accessed.

[22] "ROS.org." https://wiki.ros.org/Distributions/ReleasePolicy (accessed.

[23] "Worm Gearbox Gearmotor." MotionCo. https://www.motionco.co.uk/motors-worm-gearbox-gearmotor-c-54_73.html (accessed 4 Jan, 2024).

[24] "Raspberry Pi 4 Model B Datasheet," 2024, vol. Release 1.1. Accessed: 12/11/2024. [Online]. Available: https://datasheets.raspberrypi.com/rpi4/raspberry-pi-4-datasheet.pdf

[25] "JoyNano Nema 17 Stepper Motor Integrated 300mm T8 Lead Screw Bipolar 1.7A 40N.cm Holding Torque 40mm Body for 3D Printer or CNC Machine." https://www.amazon.co.uk/dp/B07DPGJX39?starsLeft=1&ref_=cm_sw_r_apan_dp_VMGRWKXRM471Q040QPAN&th=1 (accessed 2/2/2024.

[26] "42BYGHW811 Stepper Motor (2.5 A, 4.8 kg·cm)." https://www.openimpulse.com/blog/products-page/product-category/42byghw811-stepper-motor-2-5-4-8-kg%E2%8B%85cm/ (accessed 2/2/2024.

[27] "Worm Gearbox Gearmotor." MotionCo. https://www.motionco.co.uk/motors-worm-gearbox-gearmotor-c-54_73.html (accessed 3/2/2024.

[28] "DMOS Microstepping Driver with Translator And Overcurrent Protection A4988," Allegro, 2024. Accessed: 5/2/2024. [Online]. Available: https://www.pololu.com/file/0J450/a4988_DMOS_microstepping_driver_with_translator.pdf

[29] "TB6600 Stepper Motor Driver." https://uk.robotshop.com/products/tb6600-stepper-motor-driver (accessed.

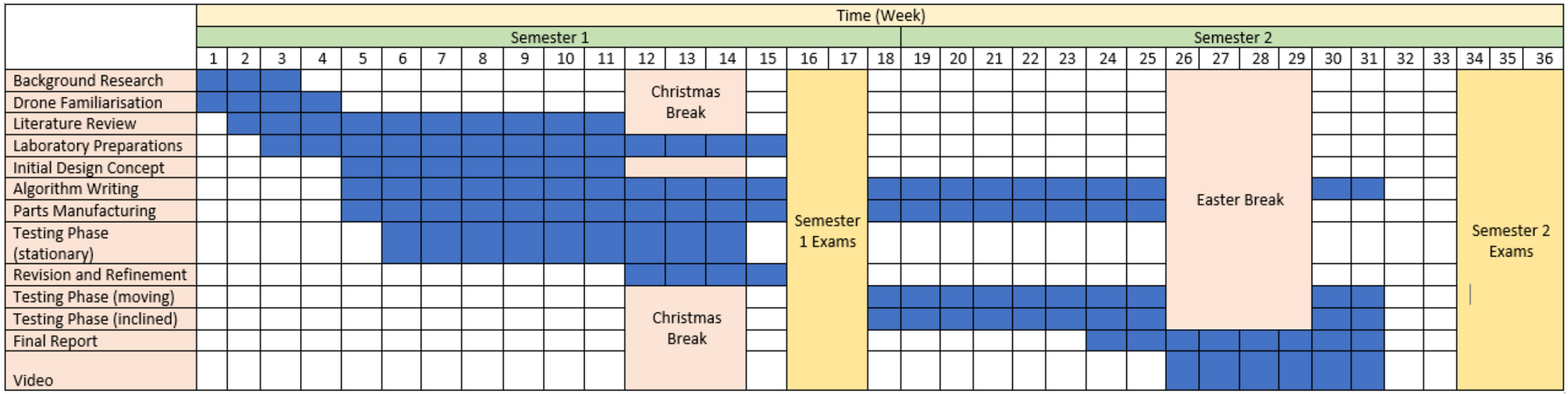
[30] "DUAL FULL-BRIDGE DRIVER L298," 2024. Accessed: 2/2/2024. [Online]. Available: https://www.sparkfun.com/datasheets/Robotics/L298_H_Bridge.pdf

[31] "Raspberry Pi Camera Module 3." https://thepihut.com/products/raspberry-pi-camera-module-3?variant=42305752039619&currency=GBP&utm_medium=product_sync&utm_source=google&utm_content=sag_organic&utm_campaign=sag_organic&gad_source=1 (accessed.

[32] "BNO055 Intelligent 9-axis absolute orientation sensor," 2024. Accessed: 2/2/2024. [Online]. Available: https://cdn-learn.adafruit.com/assets/assets/000/125/776/original/bst-bno055-ds000.pdf?1698865246

Appendix

Gantt Chart



Direct cost

Item	Specifications	Cost (£)	Subtotal (£)
Raspberry Pi	Model 4B, 4GB	90.92	
Micro Sd cards	32GB	25.83	
IMU units	9 DOF	112.47	
Raspberry Pi Camera Module	8MP Raspberry Pi Camera Module 2	12.40	
HDMI adapters for Raspberry Pi	Micro HDMI to HDMI	5.38	
Ultrasonic sensors	Range: 2 -400cm	10.4	
Linear Actuator	100mm stroke length	26.98	
Lead Screw	300mm shaft with Nema 17 stepper motor	30.62	
Buck boost converter	Max 3A	45.79	
Motor driver for DC motor	L298N H bridge	11.99	
DC motor	Maximum Torque: 5.85Nm	46	
Electromagnet	5kg attraction	6.99	
Bearing	200mm	11.99	
Servo motors	SG90-HV	10	
GPIO extender		7.6	
Female to female jumper cables		3.85	
5V relay breakout board	2 Channel Relay	7.5	
Camera extender FFC cable	Compatible with RPI Cameras	8.58	
Motor Driver for Lead Screw	A4988 motor driver	8.99	
Motor Driver for Bottom Stepper Motor (Nema 17)	TB6600 motor driver	18.99	
Voltage Regulators	12V voltage regulators	21.80	
Voltage regulator USB	5V voltage regulator with USB-A output	6.23	
USB-A to USB-C wire	Cable length 50cm	5.75	
			606.43

Variable cost

Item	Cost (£)	Subtotal (£)
3D print	50	
Waterjet parts	20	
Miscellaneous	26	
		96

Sunk cost
(Inherited Drone, Rover and other components)

Item	Cost (£)	Subtotal (£)
Hexsoon Quadcopter frame	575.00	
CUBEPILOT orange	485	
Here 3+ GPS	175	
NVIDIA Jetson Nano	150	
JSumo ATLAS All Terain High Speed Robot	383.61	
Overlander 5000MAH 14.8V 35V LiPo Battery	63.99	
42BYGHW811 Stepper Motor (2.5A)	14.80	
		1847.40

*Note: Given that the Avionics development has been discontinued, **actual sunk cost is £462.40.**