University of Bremen

Faculty 4 Production Engineering

Master Space Engineering I

Master Thesis

Design, Fabrication, and Testing of a 3D-Printed Model Rocket with Integrated Telemetry Systems

Author:	Philippos Georgios Moschidis Mat. Nr. 6254474
Date:	06.06.2025
1. Reviewer:	Dr.-Ing. Benny Rievers
2. Reviewer:	Dr.-Ing. Florian Meyer
Supervisor:	Dr. Petros Bithas

Philippos Georgios Moschidis
Master Thesis
moschidis.filip@gmail.com
Matriculation Nr. 6254474

Design, Fabrication, and Testing of a 3D-Printed Model Rocket with Integrated Telemetry Systems

Entwurf, Herstellung und Test einer 3D-gedruckten Modellrakete mit integrierten Telemetriesystemen

Reviewers: Prof. Dr.-Ing. Benny Rievers & Dr. Florian Meyer
Supervisors: Dr. Petros Bithas
University of Bremen
Space Engineering I (SpE I)
Faculty 4 Production Engineering
ZARM, Am Fallturm 2
28359 Bremen

Abstract

The Hermes Rocket Project focuses on the integration of modern technologies into model rocketry through the design and fabrication of a 3D-printed rocket that includes a telemetry system. The objective of this study is to develop a cost-effective and reusable model rocket capable of achieving an apogee of over 150 meters. The rocket is designed to collect flight data while it also utilises a camera to analyse the flight trajectory post-flight. The rocket has been designed to be lightweight and stable, while also ensuring structural integrity, based on PETG material and modular 3D-printed components. The flight computer, which has been custom-built, includes a Raspberry Pi Zero 2W microcontroller at its core, as well as three sensors: a GPS, an inertial measurement unit (IMU), and a barometric sensor to monitor altitude, pressure, and motion data. With these sensors we can monitor data which can be observed live during the start of the launch and more important they will be analysed post launch together with the onboard video footage. The propulsion of the rocket is provided by Klima D9-5 rocket engines, while the design, simulation and stability of the rocket were ensured through the use of OpenRocket software. The recovery system consists of a ripstop nylon parachute. There were four test flights completed that validated the rocket's performance, telemetry accuracy, and recovery system. The flight data that were collected showed an okay alignment with OpenRocket simulations, with some deviations primarily because of environmental conditions and possibly motor performance. The results confirm the feasibility of using low-cost 3D-printed systems and off-the-shelf components for educational and experimental aerospace applications. This research shows how 3D printing and modern sensors can make model rockets easier to use and perform better, providing valuable information for educational and amateur aerospace projects.

Contents

List of Figures

List of Tables

List of Abbreviations

ABS	Acrylonitrile Butadiene Styrene
BMP	Barometric Pressure Sensor
CAD	Computer-Aided Design
Cal	Caliber (distance between CG and CP in body diameters)
CD	Drag Coefficient
CG	Centre of Gravity
CP	Centre of Pressure
DC	Direct Current
DOF	Degrees of Freedom
DPS310	Digital Pressure Sensor 310
FDM	Fused Deposition Modeling
FPS	Frames Per Second
GND	Ground (electrical)
GPIO	General-Purpose Input/Output
GPS	Global Positioning System
HAT	Hardware Attached on Top
Hz	Hertz (sampling rate)
I2C	Inter-Integrated Circuit
IC	Integrated Circuit
IMU	Inertial Measurement Unit
J-B Weld	Commercial Two-Part Epoxy Adhesive
LED	Light Emitting Diode
mAh	Milliamp Hours
M58x1.5	ISO Metric Thread (58 mm diameter, 1.5 mm pitch)
NAR	National Association of Rocketry
NASA	National Aeronautics and Space Administration
PETG	Polyethylene Terephthalate Glycol
PLA	Polylactic Acid
RAM	Random Access Memory
SBC	Single-Board Computer
SPI	Serial Peripheral Interface
STEM	Science, Technology, Engineering and Mathematics
STEMMA QT	Adafruit Quick Connect I2C System
UDP	User-Datagram Protocol
V	Voltage
$°C$	Degrees Celsius
$°/s$	Degrees per second
μA	Microamperes
μT	Microtesla

Acknowledgement

I would like to thank several people whose support and guidance were crucial to completing my thesis. My academic supervisors, Dr. Benny Rievers and Dr. Florian Meyer, provided continuous guidance and constructive feedback throughout this process, and their support was very valuable. Their expertise was crucial to the completion of this thesis.

Additionally, I am especially thankful to my external supervisor, Dr. Petros Bithas, who without hesitation offered to help me with another thesis and whose contribution led to the successful completion of this project.

In particular, I would like to thank Mr. Theodoros Papafotiou for his support, very helpful ideas and for giving me access to the laboratory facilities of 'VROOM', which were essential for the development of this thesis.

Finally, I wish to thank my family and friends for their support, patience and motivation during this challenging journey.

"Engineering is at the heart of space exploration, turning bold ideas into tangible achievements solving problems that were once considered impossible."
– Aerospace Engineer

1. Introduction

1.1. Background and Purpose of the Study

The field of model rocketry started in the United States in 1957, it resulted from a synthesis of model aviation, the ancient art of pyrotechnics and modern rocket technology. It spawned the development of a novel and innovative approach to the field of model rocketry. Rocketry has for a long time been a platform for scientific exploration, offering students and hobbyists the opportunity to gain hands-on experience in propulsion, physics, aerodynamics, and engineering. In the past, model rockets were constructed by using basic tools and techniques, which restricted their ability to gather and analyse real-time flight data. But as engineering, compact computing, sensor technologies and engine manufacturing developed, it created opportunities for the design of high performance lightweight rockets. Additionally, the integration of telemetry, advanced sensors and 3D-printing enables greater control over stability, more precise data collection and improved recovery mechanisms. The objective of the Hermes Rocket Project is to combine these technological opportunities with practical applications, thereby demonstrating how modern innovations can extend the limits of model rocketry [1].

Figure 1: Shady Side Academy Rocketry Club, Pittsburgh PA in January 1965. Rocketry enthusiasts from over a half century ago [2].

1.2. Objectives of the Hermes Rocket Project

The scope of this thesis is to design, construct and launch a 3D-printed model rocket that is equipped with a custom-build flight computer that collects as well as transmits real-time telemetry data. The aim of this project is to explore the difficulties that come with integration of technologies including the Raspberry Pi, high-precision sensors, wireless communication, camera feed and wireless communication into the field of model rocketry. Additionally, a 3D printer will be used, in order for the Hermes Rocket Project to optimise weight, guarantee flight stability and finally make sure that the rocket is recovered safely for reusability. This study will provide helpful information and framework to the evolving field of model rocketry by integrating this modern technology with applications in education and amateur rocketry advancements.

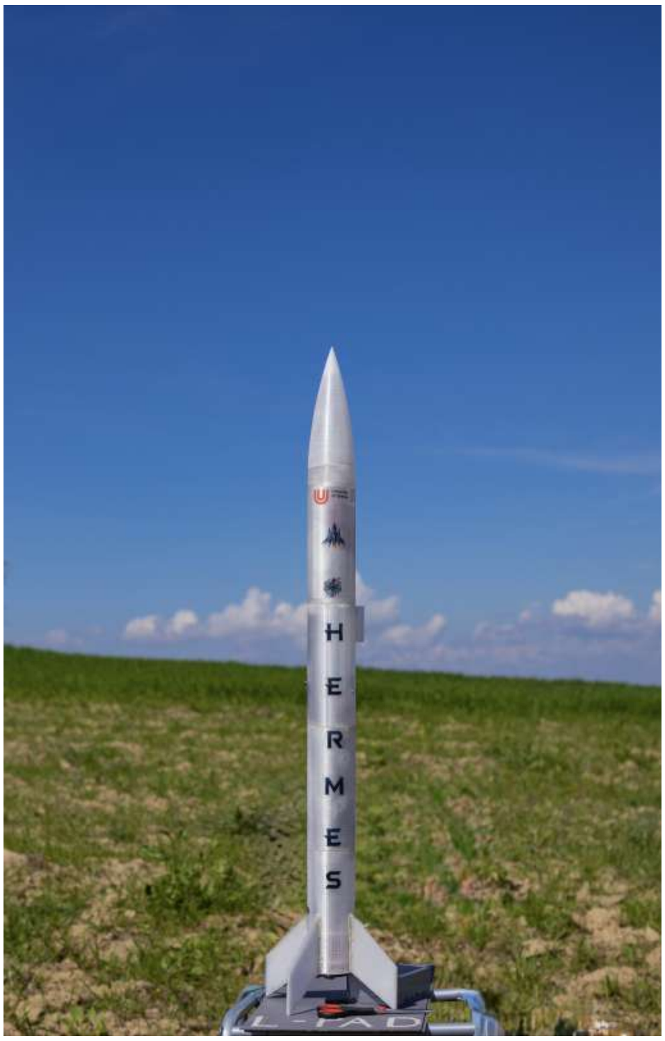

Figure 2: Hermes Rocket before the first Flight

1.3. Scope and Limitations

The main focus of this study is to design, manufacture and test the Hermes Rocket. After the 3D-print of the rocket, the assembly of the flight computer and the setup of the telemetry system were achieved, several tests were conducted with the objective of achieving a minimum apogee of 150 meters. In order for a safe recovery, a number of limitations were encountered during the course of the project. Some of these were the maximum print size of the 3D printer, the availability of the materials used for the rocket and the limited engine power as the project was self funded. Additionally as the testing was held under different conditions than the actual flight, some results may not be applicable to all scenarios. Finally, it should be noted that the project is an experimental design and not aimed for commercial production.

2. Literature Review

2.1. Overview of Model Rocketry

Model rocketry began in the mid-20th century as an educational and entertaining activity. It involves building scaled-down, functional versions of real rockets, often from lightweight, non-metallic materials, for instance paper, plastic or cardboard. This makes them affordable and safe, while factory-made solid-propellant motors guarantee reliability. A recovery system, such as a parachute, enables the rockets to return safely to the ground for reuse. Before 1958, model rocketry faced safety concerns, resulting in restrictions in many regions. However, the establishment of organisations such as the National Association of Rocketry (NAR) introduced standardised safety codes, making model rocketry a safe and popular hobby. Today, it is widely used as an educational tool to promote science, technology, engineering, and mathematics (STEM) concepts in schools, universities, and space camps [1].

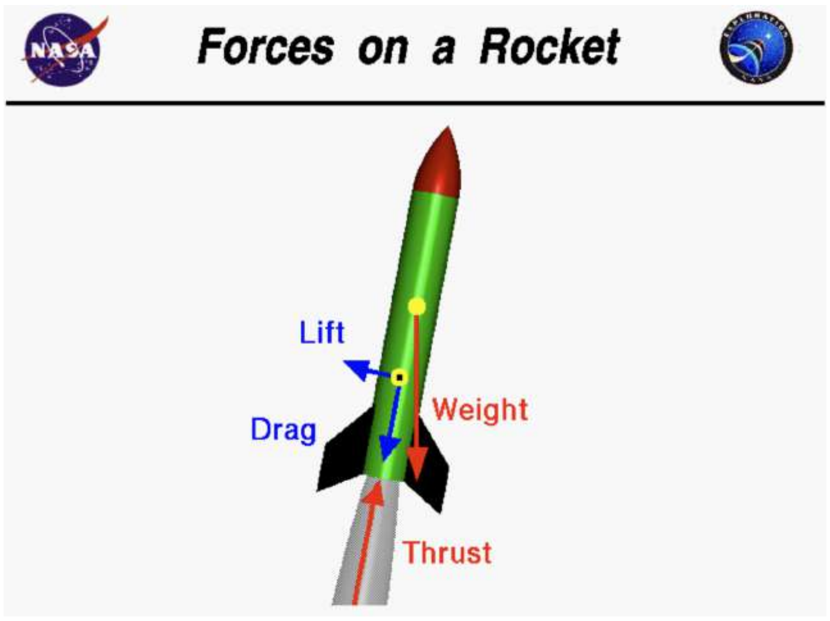

Figure 3: Four Forces in Flight [3]

Similar to aircraft, model rockets experience four primary forces during flight: weight, thrust, lift, and drag as they can also be seen in Fig. 3.

- Weight is the gravitational force acting on the rocket's total mass, pulling it towards the Earth.
- Thrust acts as an opposing force to weight, enabling the rocket to overcome gravity and achieve lift-off.

- Lift stabilises and controls the rocket's flight direction.
- Drag is the air resistance that slows down motion through the atmosphere.

One of the key features of stable flight is the relationship between the centre of gravity (CG) and the centre of pressure (CP). In order for a rocket to remain stable, the CP (which is determined by aerodynamic forces) must be positioned below the CG. This makes sure that, if there is wind or other disturbances, the rocket naturally returns to its intended flight path [3]. Additionally, the motion of a rocket can be described mathematically using the classical equations of motion, which are based on Newton's second law. These formulations are described in detail in *Rocket Propulsion Elements* by Sutton and Biblarz [4],which will provide the basis for the following equations. The net force acting on the rocket is the combination of thrust, aerodynamic drag and gravitational force, as expressed in Eq. (1):

$$\begin{gathered} F_{\text{net}} = m \cdot a \\ F_{\text{thrust}} - F_{\text{drag}} - F_{\text{gravity}} = m \cdot \frac{d^2x}{dt^2} \end{gathered} \tag{1}$$

Continuing, F_{net} is analysed in more detail below. The thrust F_{thrust} is produced by the expulsion of high-speed propellant mass, as followed in Eq. (2):

$$F_{\text{thrust}} = \dot{m} \cdot v_e \tag{2}$$

- Where:
  - $\dot{m}$ is the mass flow rate of the expelled propellant.
  - v_e is the exhaust velocity.

Additionally, the aerodynamic drag, the opposition of the rocket's motion is an important factor that is calculated, as shown in Eq. (3):

$$F_{\text{drag}} = \frac{1}{2} \cdot C_d \cdot \rho \cdot A \cdot v^2 \tag{3}$$

- Where:
  - C_d is the drag coefficient.
  - ρ is the air density.
  - A is the cross-sectional area.
  - v is the velocity relative to the surrounding air.

Finally, the gravitational force is calculated as shown in Eq. (4):

$$F_{\text{gravity}} = m \cdot g \tag{4}$$

These key equations, which are referenced in Eqs. (1) to (4), are important in order to understand rocket dynamics which are widely used in flight simulation tools [1], [5], [6]. The theoretical basis for rocket propulsion is based not only on Newton's Second Law, but also on a number of important principles that are used to describe rocket performance. Sutton and Biblarz [4] provide an important insight into understanding both small-scale as well as full-scale rocket systems. One of the most important equations is the classical rocket equation that is also known as the Tsiolkovsky rocket equation. Which is describing how a vehicle produces acceleration by ejecting part of its mass at high velocity, following the principle of conservation of momentum. This equation, shown in Eq. (5), is critical for estimating the maximum velocity achievable by a rocket:

$$\Delta v = v_e \cdot \ln\left(\frac{m_0}{m_f}\right) \tag{5}$$

- Where:
  - v_e is the exhaust velocity.
  - m_0 is the initial mass.
  - m_f is the final mass after propellant has been burned.

Another significant performance metric is the specific impulse I_{sp}, which measures how efficiently a rocket uses propellant fuel to generate thrust. It is expressed as thrust per unit weight flow of propellant and is given by Eq. (6):

$$I_{sp} = \frac{F}{\dot{m} \cdot g_0} \tag{6}$$

- Where:
  - F is the thrust.
  - $\dot{m}$ is the propellant mass flow rate.
  - g_0 is the standard gravitational acceleration.

The overall thrust produced by a rocket engine is the result of momentum thrust and pressure thrust combined. The calculation is derived from the thrust equation, as illustrated in Eq. (7):

$$F = \dot{m} \cdot v_e + (p_e - p_a) \cdot A_e \tag{7}$$

- Where:
  - p_e is the exit pressure.
  - p_a is the ambient pressure.

  - A_e is the nozzle exit area.

Furthermore, the characteristic velocity is a metric of rocket engine combustion performance that is not reliant upon nozzle performance as shown in Eq. (8). It is therefore utilised for the purpose of comparing different propellants and propulsion systems.

$$c^* = \frac{p_c \cdot A_t}{\dot{m}} \tag{8}$$

- Where:
  - p_c is the chamber pressure.
  - A_t is the nozzle throat area.

Additionally, the thrust coefficient, as demonstrated in equation Eq. (9), demonstrates a relation between the actual thrust produced and the chamber pressure and nozzle area, thereby offering insight into the performance of the nozzle design.

$$C_F = \frac{F}{p_c \cdot A_t} \tag{9}$$

Concluding, the law of conservation of mechanical energy connects the kinetic and potential energy of a system, as shown Eq. (10). In aerospace engineering, this link is used to calculate altitude gain, based on the principle that there are no losses due to drag or other forces [7].

$$E = E_k + E_p \tag{10}$$

- Where:
  - $E_k = \frac{1}{2}mv^2$ is the kinetic energy.
  - $E_p = mgh$ is the gravitational potential energy.
  - m is the mass of the object.
  - v is the velocity.
  - g is gravitational acceleration.
  - h is the altitude.

By cancelling m on both sides of Eq. (11) and solving for h, we obtain:

$$\begin{gathered} \frac{1}{2}mv^2 = mgh \\ h = \frac{v^2}{2g} \end{gathered} \tag{11}$$

This equation can be used in order to estimate the maximum altitude that can be achieved from a known velocity, assuming vertical motion and no aerodynamic losses. Together, these equations form a complete mathematical model for describing and analysing rocket performance, and are often used in research and engineering.

2.2. Advances in 3D Printing for Aerospace Applications

The introduction of 3D printing technology has begun a new era in manufacturing, offering opportunities for innovation and efficiency that were previously unattainable across a number of industries. Specifically, the aerospace industry benefited from the precision, durability, reduced weight, rigid prototyping and customisation that 3D printing provides. Materials used in additive manufacturing include: Polylactic Acid (PLA), Polyethylene Terephthalate Glycol (PETG), Acrylonitrile Butadi- ene Styrene (ABS) and Carbon Fibre Nylon and composites such as carbon-fibre- infused filaments. These materials are known for their strength, light weight, and ability to produce complex shapes that cannot be achieved using traditional manufacturing methods. Aerospace companies that have adopted 3D printing technology include SpaceX, NASA and Relativity Space [8].

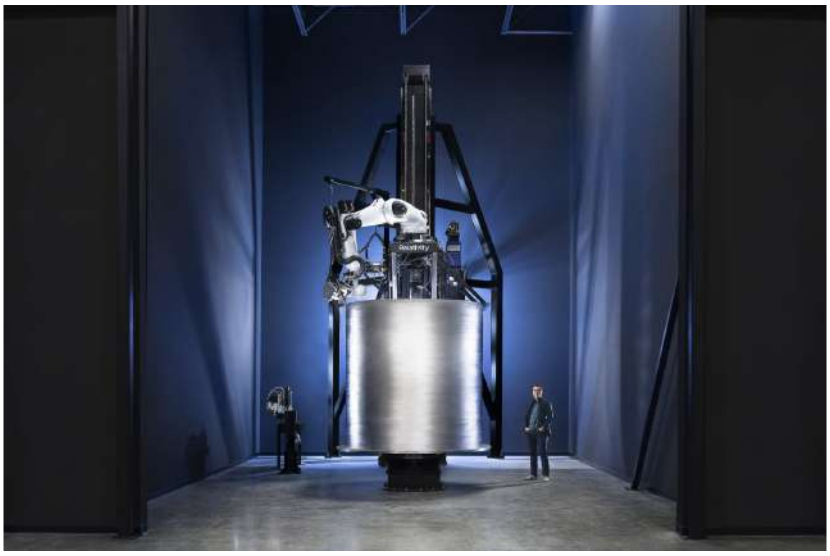

Figure 4: Relativity Space's third generation 3D-printer in its new headquarters, with CEO Tim Ellis standing by for scale [9].

Additive manufacturing can be applied in many different industries, as seen in its use for building model rockets. In particular, costs are significantly reduced compared to traditional machining, making advanced designs accessible to hobbyists. This is achieved by using 3D printing to create lightweight structures where the wall thickness, internal infill and material density can be controlled.

2.3. Role of Flight Computers and Telemetry in Model Rockets

As previously mentioned, flight computers have become important parts in the field of modern model rocketry, offering their users the ability to gather real-time data, monitor telemetry, and utilise autonomous control mechanisms. To be more precise, data collection refers to the gathering of information regarding altitude, velocity, orientation, temperature and pressure. This data can be recorded in real time during the flight for subsequent analysis. With regard to the telemetry system, this enables the transmission of live data to a ground station, thereby facilitating insights into the conditions of the flight. This enables the formulation of solutions to any issues that may arise with the rocket, including its position. Furthermore, flight computers can automate parachute deployment, thereby ensuring the safe return of the rocket. This can be achieved through the use of time-based triggers. In bigger scale rockets, the flight computers that are used, are capable of helping with stability of the rocket through the processing of the inertial measurement unit (IMU) data and then by correcting manoeuvres. Although flight computers can have many advantages, they also could present challenges. In order to function, the flight computer requires a power source. The issue may therefore lie in the management of this power, since the maintenance of a stable supply is imperative for the preservation of data integrity and the prevention of system failure. Additionally, external factors like vibrations, extreme temperatures, and rapid pressure changes can impact the accuracy and reliability of the sensors [1].

With reference to modern rocketry, there has been a notable increase in the accessibility of microcontrollers and sensors to the general public. This development has resulted in a significant role for flight computers in the collection of real-time data, the utilisation of telemetry, and the implementation of recovery mechanisms. The core of the flight computer is the microcontroller, which is primarily responsible for processing the data obtained from various sensors. These sensors may include a barometric sensor for measuring altitude, an IMU for orientation, and a global positioning system (GPS) for tracking. To facilitate live telemetry, wireless communication is essential, and onboard power management is necessary to ensure stable operation during flight.

3. Rocket Design and Fabrication

3.1. Conceptual Design of the Hermes Rocket

In order to achieve the scope of the project the rocket includes a flight computer, a camera and sensors for real-time collection of data. The objective of the mission is to achieve an apogee of around 150 metres, which will enable the flight computer and camera to gather a substantial amount of real-time telemetry and onboard video data for subsequent post-flight analysis.

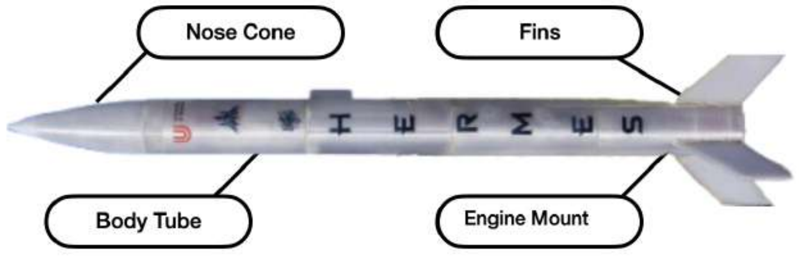

Figure 5: Hermes Rocket with labeled parts

The principal structural elements of the rocket are as follows:

- The body tube represents the primary structural component of the rocket. This constitutes the primary structural element of the rocket, wherein the flight computer, sensors, power supply and parachute for safe descent and durability will be positioned.
- The nose cone is also an important part of the rocket. It is positioned on the upper part of the rocket and is designed with an aerodynamic shape to minimise drag.
- The function of the fins is to maintain stability and control during flight. Their design is based on principles of weight distribution and aerodynamics.
- The engine mount is a device which is used to attach the rocket engine to the rocket structure. This is the location where the Klima D9-5 engines are situated, providing the thrust required for the rocket's ascent.

The design philosophy is based on the principles of lightweight modularity. This approach allows for the construction of a rocket that is both lightweight and easily repairable. In case there would be a failure during any of the flights, the rocket could be re manufactured using the 3D printer, as the material is already bought and there is no need for additional design of the parts. The basic design of the rocket, as well as the flight simulations and stability analysis, were created using OpenRocket which is a free, open-source simulation software used for designing and analyzing model rockets. For the design of the rocket parts, the CAD application Autodesk Fusion 360 was used for 3D modelling.

3.2. Material Selection

Selecting the correct material for the manufacturing of the rocket is of great importance as it will impact its design and performance. In more detail the material will have an affect on the structural integrity, the aerodynamic efficiency, the thermal resistance as well as the overall weight. Factors that are essential for achieving a stable flight. In this section, a comparative analysis of the most commonly used 3D printing materials, namely PLA, PETG, ABS and Carbon Fibre Nylon is made, with respect to their strength, weaknesses, sustainability and price for the Hermes rocket. Every individual material has its distinctive advantages and disadvantages, thus it is important to select the optimal material that meets the most performance objectives for the rocket [8].

3.2.1. PLA, PETG, ABS, and Carbon Fiber Nylon Properties

The table below compares the properties of commonly used 3D printing materials.

Material	**Strengths**	**Weaknesses**
PLA	- Easy to print - Lightweight - Cost efficient	- Low heat resistance softens $\sim 60\ °C$. - Brittle
PETG	- Tough and flexible - Better heat resistance than PLA softens $\sim 80\ °C$ - Slightly more expensive than the PLA	- Slightly harder to print than PLA - Slightly heavier than PLA
ABS	- Strong, durable and impact resistant - Better heat resistance than PLA softens $\sim 100\ °C$ - Good machinability	- Dangerous fumes during printing - Tends to warp and crack
Carbon Fiber Nylon	- High heat resistance and vibration durability softens $>100\ °C$ - Flexible and excellent mechanical properties	- Expensive - More difficult to print

Table 1: Comparison of 3D Printing Materials

3.2.2. Trade-offs in Strength, Weight, and Durability

In order to select an appropriate material for the 3D printer, it is essential to consider the strengths and weaknesses of each material. It is of great importance that the rocket is sufficiently robust to withstand the launch and flight. But, it is also important to keep in mind that using materials that are excessively strong but also heavy may have a big impact on engine efficiency and apogee, as lighter materials improve performance. Additionally, when the engines ignite, they generate heat within the rocket, meaning the use of a material that is both heat-resistant and capable of maintaining its structural integrity under the influence of vibrations is very crucial.

As observed from Table 1, PLA is brittle and therefore unsuitable for applications involving high vibration or stress. In contrast, PETG, ABS and Carbon Fiber Nylon are characterised by high levels of toughness, with the latter exhibiting the greatest resilience. In terms of weight, PLA has the lowest mass, with PETG being a close competitor. ABS and Carbon Fiber Nylon have the highest mass. Ultimately, PLA and ABS have a high possibility of cracking, although the latter shows greater resilience to elevated temperatures. Furthermore, PETG has greater heat resistance than PLA, as well as better durability than the other materials. Finally, Carbon Fiber Nylon has a very good durability and can withstand high temperature [10] [11].

A comparison of the data in Table 1, together with the information on strength, weight and durability, concludes that Carbon Fiber Nylon is the optimal choice, with PETG being a good alternative. However, when considering cost, it is clear that Carbon Fiber Nylon is considerably more expensive than PETG. Consequently, the selected material for the Hermes rocket is be PETG.

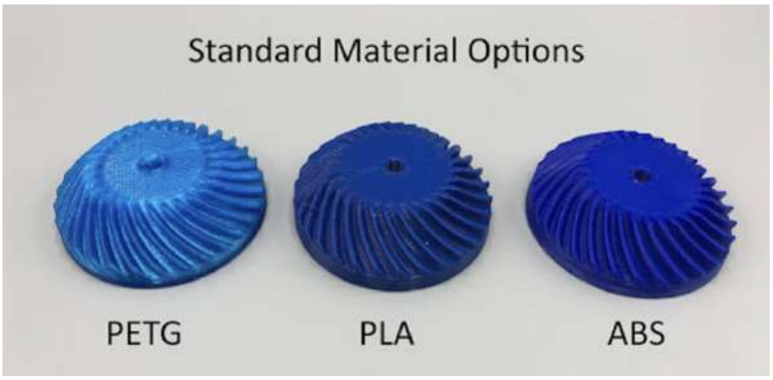

Figure 6: PLA, PETG, and ABS are the most common materials used with filament 3D Printers [12].

3.3. Hermes Rocket Design

This section addresses the design of the rocket and the weight of each component. A number of design guidelines and rules of thumb are employed in the field of rocketry. These rules will optimise performance, stability and aerodynamic efficiency. Prior to the design of the Hermes rocket, it is essential to consider a number of key factors in order to develop an effective and efficient design. It is imperative to consider the internal diameter of the body, as this will determine the location of the flight computer. The largest component of the flight computer has dimensions of $65 \cdot 30$ mm, which equates to a diameter of 67 mm.

3.3.1. Body Tube

With regard to the body tube, a rule of thumb for stability is that the length of the body tube should be approximately 9 to 10 times the maximum diameter of the rocket [1]. In Fig. 7 we can see that the outer diameter is 67 mm, the total length should be 670 mm and a suitable wall thickness for a robust and flexible body tube would be 1.8 mm. These dimensions allow us to calculate the weight of the body tube as shown in Eq. (12), Eq. (13), and Eq. (14). The material selected for this calculation is PETG, which has a density of 1.15 $\frac{g}{cm^3}$.

- The first step is to calculate the inner radius, using the difference between the outer radius and the wall thickness:

$$\begin{aligned}\text{Inner Radius} &= \text{Outer Radius} - \text{Wall Thickness}\\ &= 33.5 - 1.8 = 31.7 \text{ mm}\end{aligned} \tag{12}$$

- Next, the body tube volume is calculated using the cylindrical shell volume formula:

$$\begin{aligned}V_{\text{Body Tube}} &= \pi \cdot h_{\text{Body Tube}} \cdot \left(r_{\text{outer}}^2 - r_{\text{inner}}^2\right)\\ &= \pi \cdot 670 \cdot (33.5^2 - 31.7^2)\\ &= 247020 \text{ mm}^3 = 247.02 \text{ cm}^3\end{aligned} \tag{13}$$

- Finally, the weight is determined by multiplying the volume by the PETG density:

$$\begin{aligned}\text{Weight}_{\text{Body Tube}} &= \text{Volume} \cdot \text{Density}\\ &= 247.02 \cdot 1.15 = 284.08 \text{ g}\end{aligned} \tag{14}$$

Therefore, the final mass of the body tube is determined to be 284.08 grams.

3.3.2. Nose Cone

The shape of the nose cone has an impact on the drag. The most common shapes observed in practice include conical, ogive and parabolic. Ogive and parabolic are typically more aerodynamically efficient than a simple conical shape. Furthermore, the ogive shape is more readily printable than the parabolic, which is

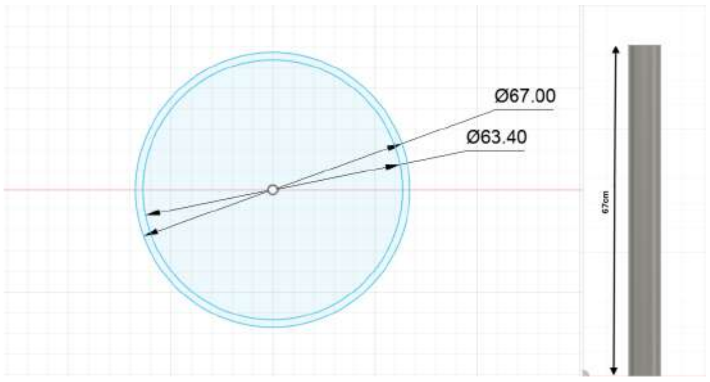

Figure 7: Body Tube of Hermes on Fusion 360

why this shape is typically selected. With regard to length, a widely accepted guideline is to maintain a ratio of between three and five times the diameter of the base to the length of the nose cone. This will ensure that the nose cone effectively reduces drag while also maintaining stability. Given that the diameter of the rocket is 67 mm, the length of the nose cone will be 200 mm. To calculate the weight, the following steps are undertaken, as shown in Eq. (15), Eq. (16), and Eq. (17) [6].

- The first step is to calculate the inner radius:

$$\begin{aligned}\text{Inner Radius} &= \text{Outer Radius} - \text{Wall Thickness}\\ &= 33.5 - 1.8 = 31.7\text{ mm}\end{aligned} \tag{15}$$

- Next, the volume is calculated using the formula for a hollow cone:

$$\begin{aligned}V_{\text{Nose Cone}} &= \frac{1}{3}\cdot\pi\cdot h_{\text{Nose Cone}}\cdot\left(r_{\text{outer}}^2 - r_{\text{inner}}^2\right)\\ &= \frac{1}{3}\cdot\pi\cdot 200\cdot(33.5^2 - 31.7^2)\\ &= \frac{1}{3}\cdot\pi\cdot 200\cdot 117.36\\ &= 23570\text{ mm}^3 = 24.57\text{ cm}^3\end{aligned} \tag{16}$$

- Finally, the weight is determined by multiplying the volume by the material density:

$$\begin{aligned}\text{Weight}_{\text{Nose Cone}} &= \text{Volume}\cdot\text{Density}\\ &= 24.57\cdot 1.15 = 28.26\text{ g}\end{aligned} \tag{17}$$

It is also important to consider the shoulder of the nose cone which can be seen in Fig. 8. It is essential that the shoulder is of sufficient length to ensure stability during flight and to maintain the secure attachment of the nose cone to the body tube until the parachute deployment. The length of the shoulder should be sufficient to ensure that, upon ejection of the engines, the parachute is effectively deployed. The length of the shoulder is typically 1 to 1.5 times the diameter of the body tube.

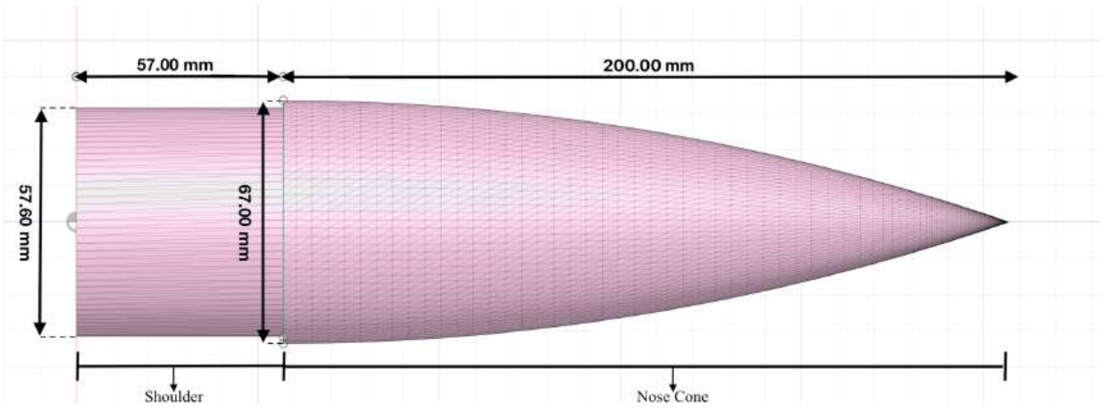

Figure 8: Nose cone and Shoulder of Hermes on Fusion 360

This would equate to a length of 57 mm for the shoulder. In this case, the wall thickness does not have to be the same as that of the body tube or the nose cone, but it can be thinner. In this instance, it will be 0.1 mm. To calculate the weight of the shoulder, the following steps are undertaken, as shown in Eq. (18), Eq. (19), and Eq. (20) [1].

- The first step is to calculate the inner radius:

$$\begin{aligned} \text{Inner Radius} &= \text{Outer Radius} - \text{Wall Thickness} \\ &= 33.5 - 1 = 32.5 \text{ mm} \end{aligned} \tag{18}$$

- Next, the volume is calculated:

$$\begin{aligned} V_{\text{Shoulder}} &= \pi \cdot h_{\text{Shoulder}} \cdot \left(r_{\text{outer}}^2 - r_{\text{inner}}^2\right) \\ &= \pi \cdot 57 \cdot (33.5^2 - 32.5^2) \\ &= \pi \cdot 57 \cdot 66 \\ &= 118100 \text{ mm}^3 = 11.81 \text{ cm}^3 \end{aligned} \tag{19}$$

- Finally, the weight is calculated by multiplying the volume by the density:

$$\begin{aligned} \text{Weight}_{\text{Shoulder}} &= \text{Volume} \cdot \text{Density} \\ &= 11.81 \cdot 1.15 = 13.59 \text{ g} \end{aligned} \tag{20}$$

The final mass is obtained by adding the weight of the nose cone and the shoulder, resulting in a value of $28.26 + 13.59 = 41.85$ grams. But the actual final volume was not the same as calculated, Fusion 360 showed that the volume is 60.18 cm^3 and the mass is calculated in Eq. (21):

- Finally, the combined weight of the nose cone and shoulder is calculated:

$$\begin{aligned}\text{Weight}_{\text{NC\&Shoulder}} &= \text{Volume} \cdot \text{Density} \\ &= 60.18 \cdot 1.15 = 69.2 \text{ g}\end{aligned} \tag{21}$$

3.3.3. Fins

The fins on a model rocket are also very important because they are crucial for maintaining the rocket's stability. They make sure that the rocket maintains a straight trajectory, stopping it from moving sideways. A rocket typically has three or four fins, for Hermes the selection of fins is four as they provide more stability than three fins would. The rules of thumb used for the fins are based on this source [1]. The fin height should be between 0.5 to 1.2 times the rocket diameter, given that the diameter of the Hermes model rocket is 67 mm, the height of the fins should be 70 mm. The root chord should be between one and one-and-a-half times the diameter of the rocket. In this case, the root chord will be 80 mm. The tip chord is typically between 50 and 80 percent of the root chord, which in this case will be 60 mm. With regard to the sweep length, this should be between 60 and 80 percent of the root chord, here it is 67 mm. The optimal sweep angle is between 30 and 45 percent of the root chord in this case it is 43.7 °. The fins will be 5 mm thick to make sure that they have enough strength while maintaining a low weight. These dimensions are illustrated in Fig. 9 below, and they allow us to calculate the weight of the fins, as shown in Eq. (22) and Eq. (23). The material selected for this calculation is PETG, which has a density of 1.15 $\frac{g}{cm^3}$. The fins were designed with a hollow interior in order to optimise weight. While a typical single-piece fin is more straightforward to manufacture, it would result in an excess of mass that could have a detrimental impact on the rocket's stability margin and apogee. In order to address this issue, the two-part design consists of one edge wall layer which has a flat base and a wall attach and another flat base by itself, as it can be seen in Fig. 10. The hollow design results in a notable reduction in the mass of the fins without any compromise to their strength or aerodynamic properties.

- The weight of a single fin is calculated:

$$\begin{aligned}\text{Weight}_{\text{Fin}} &= \text{Volume} \cdot \text{Density} \\ &= 16.2 \cdot 1.15 = 18.6 \text{ g}\end{aligned} \tag{22}$$

- Since the rocket has four fins, the total weight is:

$$\begin{aligned}\text{Weight}_{\text{Fins}} &= \text{Weight}_{\text{Fin}} \cdot \text{Number of Fins} \\ &= 18.6 \cdot 4 = 74.5 \text{ g}\end{aligned} \tag{23}$$

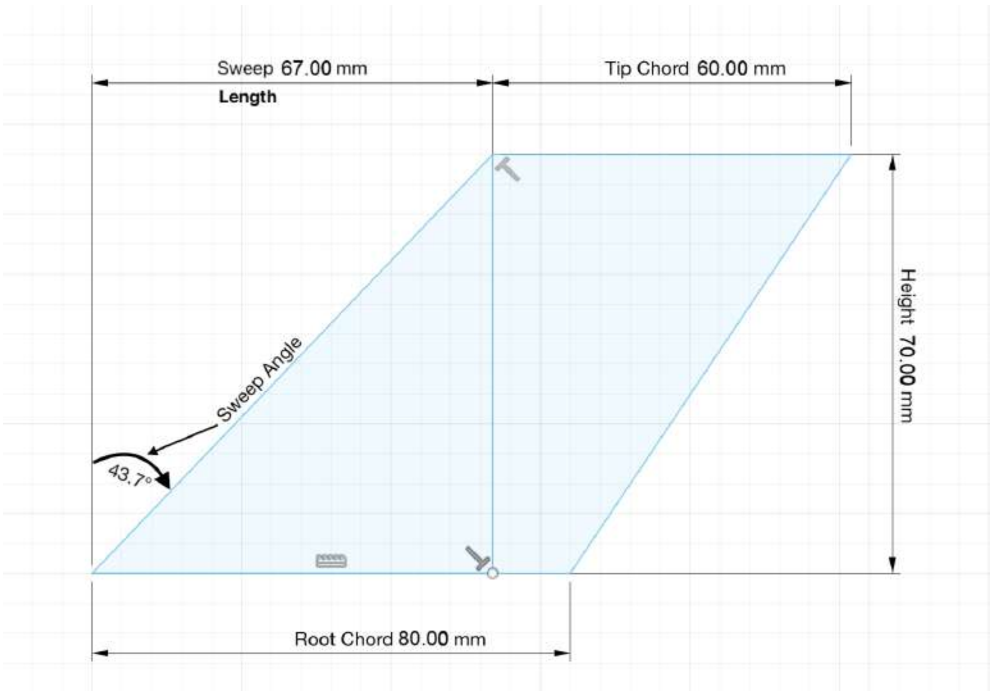

Figure 9: Fins of Hermes on Fusion 360

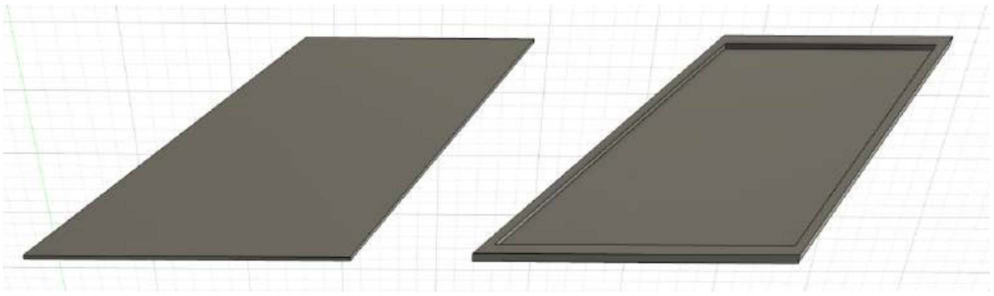

Figure 10: Hollow Interior Fin Design

In comparison, had a solid design been employed, the weight of a single fin would have been 28 grams, resulting in a saving of 10 grams per fin.

3.4. 3D Printing Process

3.4.1. Printer characteristics and Constraints

Regarding the 3D printer, the Prusa MINI+ shown in Fig. 11 was used. It is fitted with a sensorless homing mechanism, automated mesh bed calibration, interchangeable nozzles, network connectivity, USB printing and a full-colour LCD screen. The layer height of the printer ranges from 0.05 mm to 0.25 mm, with the nozzle set to 0.04 mm by default and the ability to accommodate filament diameters of up to 1.75 mm. The printer is compatible with a multitude of materials, including PETG, as previously stated. Additionally, the maximum permissible temperature for the nozzle is 280 $°C$, while the maximum permissible temperature for the bed is 100 $°C$. It is also notable that the print volume is $180 \cdot 180 \cdot 180$ mm. However, to enable printing of more complex designs and better results, this has been modified to $170 \cdot 170 \cdot 170$ mm [13].

Figure 11: The Original Prusa MINI+ Printer

3.4.2. Assembly of Body Tube, Nose Cone, and Shoulder

The assembly of the Hermes rocket is a process of aligning and connecting the main components in order to ensure structural integrity and aerodynamic efficiency. The four body tubes are stacked in a vertical configuration. The nose cone is affixed to the uppermost body tube via a precision-engineered shoulder joint. This design feature serves two purposes: firstly, to maintain stability during flight, and secondly, to facilitate the deployment of the recovery system. As the rocket is constructed from multiple components, they must be assembled. In the case of components that are to remain connected on a permanent basis, the use of a durable bonding agent such as epoxy glue is essential to ensure a robust and long-lasting attachment. However, in order to facilitate access to the internal components for maintenance or modifications, the modular assembly simplifies both the construction process and any future adjustments, thereby ensuring flexibility for testing and optimisation.

3.5. Joining Methods and Structural Testing

3.5.1. Epoxy Bonding

In order to achieve a good epoxy bond with the PETG material that was selected, it is important to use a glue that is specifically formulated for this purpose. The J-B Weld Plastic Bonder was selected as the optimal choice. The adhesive has a rapid setting time and is a two-part glue. The glue was employed for the fins, both for affixing the edge wall layer to the flat base layer and for attaching the entire fin to the body tube. Furthermore, the adhesive was employed for the nose, as previously stated, due to its two-part printing configuration.

3.5.2. Adapter-Based Screwed Connections

With regard to the adapter connections, the adapter is composed of two rings that are affixed to the interior of the tube. These rings are then glued at the base of one tube and the top of the other. The interior of the adapter features four holes through which the screw and threaded inserts were positioned. This configuration allows for the straightforward disassembly of the body tube, thereby facilitating access to the flight computer. The adapter is illustrated in greater detail in Fig. 12.

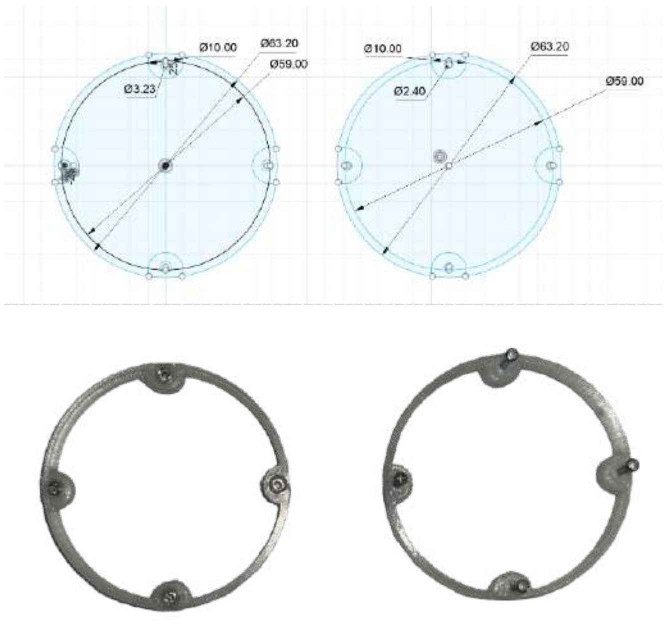

Figure 12: The screwed adapter on Fusion360 and printed

3.5.3. Adapter-Based Threaded Connections

Another concept for joining methods was fabricating a three-dimensional printed threaded adapter. To create this, two cylindrical components were designed, one male and one female, as shown in Fig. 13. The female component on the left has an internal diameter of 59.2 mm and the male component on the right has an internal diameter of 57.9 mm, providing a suitable clearance fit. Both components use an ISO metric thread profile with a triangular geometry and a flank angle of 60°, ensuring compliance with international standards. The thread size is M58x1.5, 57.9 mm is the nominal diameter and 1.5 mm is the thread pitch, ensuring a fine, precise fit. With regard to the female component, a 6H tolerance class was selected, representing a standard internal thread fit. In contrast, the male component employs a 6g class, ensuring a medium external tolerance. This combination ensures a reliable, slightly loose fit that accounts for potential inaccuracies inherent to the 3D printing process.

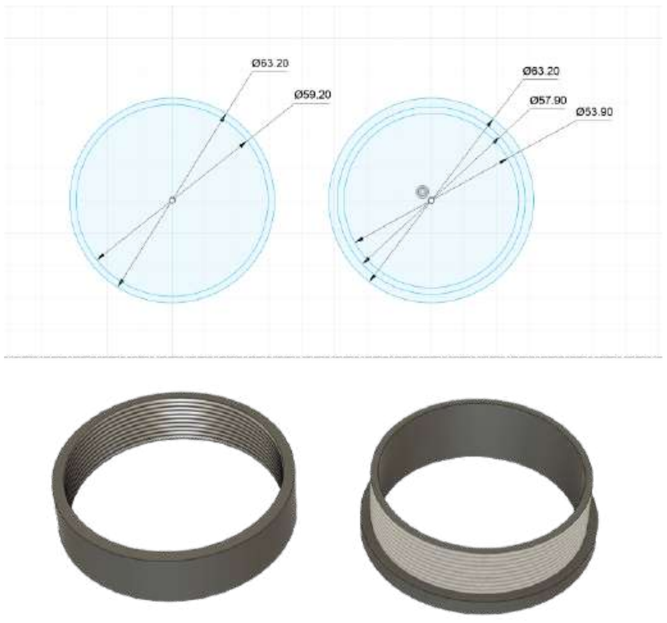

Figure 13: The threaded adapter on Fusion360

Following a comprehensive evaluation of the three proposed methods, namely epoxy glue, screwed-based adapter and threaded-based adapter, it was determined that the threaded adapter, due to its superior ease and convenience, was the optimal choice. The primary rationale for this determination was the ability to access all body tubes at any given moment, in contrast to the screwed connection. If modifications were to be made to the body tube in proximity to the engines, the necessity to unscrew every other component would be required.

4. Flight Computer and Sensor Integration

4.1. Overview of the Flight Computer System

The flight computer is the rocket's central component, which is responsible for collecting and storing data during the flight. It is also considered as the payload of the Hermes Rocket. Its primary function is to monitor and log important flight information, such as altitude, pressure, temperature and motion data, while also transmitting live telemetry to the ground station for real-time analysis. Additionally, it is equipped with a camera to capture video footage. This system ensures accurate data collection for post-flight analysis and performance evaluation.

The core components of the flight is the microcontroller, which in this instance is the Raspberry Pi Zero 2W. Additionally, a Hardware Attached on Top (HAT) was placed to the Raspberry Pi, which is equipped with Qwiic output, this will facilitate a more straightforward connection with the sensors. Additionally, the sensors will be located in various positions across the rocket, thereby saving space and enabling the rocket to be narrower and with an improved weight distribution. The flight computer has a variety of sensors in order to achieve its goals, this includes motion sensors like the IMU, pressure and temperature sensors, a GPS module and a camera module which can be seen in Fig. 14. Details of the specifications, functions and capabilities of the sensors can be found in Section 4.2. Finally the flight computer is powered by a lightweight battery system to ensure stable operation while maintaining the overall weight limits of the Hermes rocket.

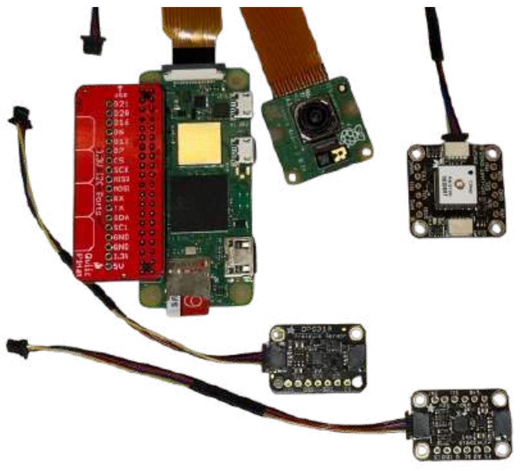

Figure 14: Flight Computer

4.2. Component Selection and Specifications

4.2.1. Microcontroller

The microcontroller that was selected is the Raspberry Pi Zero 2W as it has a compact size, it is low weight and has sufficient computational power to handle real-time data processing and transmission. Other options were the Arduino Nano, the Arduino Due, and the STM32F103C8T6 (also known as the Blue Pill). But, ultimately these options were unsuitable due to the Arduino Nano not being able to support additional sensors, the Arduino Due being to heavy, and finally the lack of compatibility with the required camera functionality of the Blue Pill. The Raspberry Pi, is equipped with a quad-core CPU and 512MB of RAM, which makes sure it is able to process multiple sensor inputs and handle real-time telemetry transmission. Finally, it has a low power consumption allowing it to operate efficiently on a 7.4W battery setup [14].

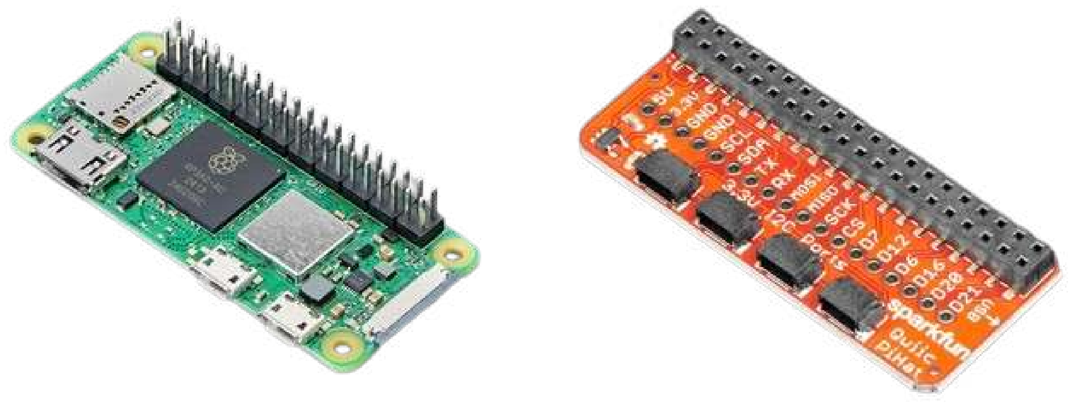

Figure 15: Raspberry Pi Zero 2W and SparkFun Qwiic/STEMMA QT HAT [14]

Furthermore, the Raspberry Pi offered the additional option of a HAT which can be seen in Fig. 15, it greatly simplified the connection with the sensors, eliminating the need for soldering. The selected HAT is the SparkFun Qwiic/STEMMA QT HAT for Raspberry Pi. The device facilitates the connection of the Inter-Integrated Circuit (I^2C) bus - including Ground (GND), 3.3 Volt (3.3V), Serial Data (SDA), and Serial Clock (SCL) - on the Pi to an array of Qwiic connectors on the HAT. The Pi HAT is equipped with four Qwiic connect ports, all on the same I^2C bus. Additionally, the device is compatible with any Pi or single-board computer (SBC) that features a standard $2x20$ General-Purpose Input/Output (GPIO) header [14].

4.2.2. Sensor Selection

As for the selection of the pressure sensors, the BMP180, BMP280, BMP388, and the DPS310 were considered. The technical specifications can be seen in Table 2.

By comparing the DPS310 and the BMP388 it can be seen that they are more precise in their pressure measurements and have a considerably broader range than the other sensors. Continuing with the altitude measurements, the DPS310 is the most precise of the sensors, offering a resolution of 20 mm, which makes it the optimal choice in this context. Furthermore, for the BMP388 and the DPS310 the sampling rate is considerably superior to that of the other sensors, and they are also more efficient during active operation. Also, the DPS310 and BMP388 offer the highest accuracy of ±0.5 $°C$, making them the optimal choice for the recording of temperature. While the BMP388 is also a strong contender, the DPS310 offers precision, resolution, and low power consumption, making it the most suitable option for flight computers where accuracy and efficiency are very important [14].

Feature	**BMP180**	**BMP280**	**BMP388**	**DPS310**
Pressure Range	300-1100 hPa	300-1100 hPa	300-1250 hPa	300-1200 hPa
Pressure Acccuracy	±1.0 hPa	±0.12 hPa	±0.08 hPa	±0.5 hPa
Altitude Resolution	∼10 mm	∼10 mm	∼10 mm	∼20 mm
Temperature Range	0 to 65 $°C$	-40 to 85 $°C$	-40 to 85 $°C$	-40 to 85 $°C$
Temperature Accuracy	±2.0 $°C$	±1.0 $°C$	±0.5 $°C$	±0.5 $°C$
Sampling Rate	∼3 samples/sec	1 - 25 Hz	1 - 200 Hz	1 - 128 Hz
Power Consumption	∼12 μA	∼2.7 μA	∼3.4 μA	∼3 μA
Interface	I^2C / SPI	I^2C / SPI	I^2C / SPI	I^2C / SPI
Size (mm)	3.6x3.8x0.93	2.0x2.5x0.95	2.0x2.0x0.75	2.0x2.5x1.0

Table 2: Comparison of Barometric Pressure Sensors [14]

The range of motion sensors available for selection was less extensive than that of pressure sensors. The principal options were the MPU6050 and the ICM-20948, which may be regarded as a moderately enhanced iteration of the MPU9250. The key differentiating factors between the two are as follows: the MPU6050 is a 6-axis IMU that provides data from accelerometers and gyroscopes, whereas the ICM-20948 is a 9-axis IMU that provides the same data and is also a magnetometer with a ±4900 μT range, enabling full 3D orientation tracking with enhanced precision. The ranges of the accelerometer ($range$: ±2 g, ±4 g, ±8 g, ±16 g) and the gyroscope ($range$: ±250 $°/s$, ±500 $°/s$, ±1000 $°/s$, ±2000 $°/s$) are identical. Moreover, the ICM-20948 exhibits markedly superior power efficiency, consuming only 1.23 mA in contrast to the 3.6 mA consumed by the MPU6050. The

ICM-20948 operates at a voltage range of 1.71 V to 3.6 V, whereas the MPU6050 necessitates a voltage range of 3.3 V to 5 V. Furthermore, both sensors have comparable sampling rates. Finally, the ICM-20948 has a smaller size and supports both I2C and SPI, and can be readily obtained with STEMMA QT/Qwiic connectors. Therefore, the ICM-20948 which can be seen in Fig. 16, with its 9-axis capability, magnetometer integration, and low power consumption, is more suitable for high-precision orientation tracking [14].

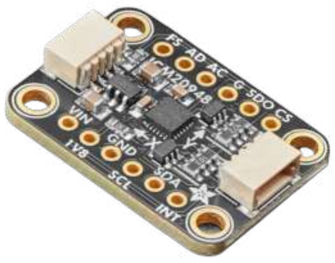

Figure 16: Adafruit TDK InvenSense ICM-20948 [14]

In regard to the GPS selection, given that the microcontroller in use is the Raspberry Pi with Qwiic connectors, the sole viable option is the Adafruit Mini GPS PA1010D as depicted in Fig. 17. The device is a compact and efficient GNSS module that supports the GPS satellite system and offers a position accuracy of ±3 metres and a timing accuracy of ±30 nanoseconds under open-sky conditions. It operates within a voltage range of 3.0 V to 4.2 V, with an active power consumption of approximately 20 mA. Furthermore, the device has also a built-in ceramic patch antenna, ensuring reliable operation in challenging environments. Moreover, the GPS is designed to withstand extreme conditions, as it can operate from $-40°C$ to $+85°C$ and has a maximum altitude of 18000 metres, making it very suitable for rocket flights with a high altitude. Finally, its compact dimensions ($11 \cdot 11 \cdot 2.2$ mm) and lightweight design make it an optimal selection for this project [14].

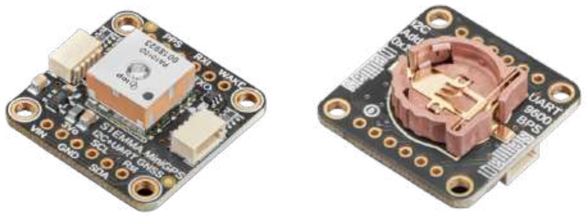

Figure 17: Adafruit Mini GPS PA1010D [14]

4.2.3. Raspberry Pi Camera

In regard to the camera selected for this project, the decision to utilise the Raspberry Pi Camera was an appropriate one, given that it is easily connected to the microcontroller, which has a designated port for this purpose. This choice was made in light of the fact that the microcontroller in use is the Raspberry Pi. Three options were available: the Raspberry Pi camera module 2, the camera module 3, and the camera module 3 with a wide-angle lens. The second generation has an 8-megapixel resolution, in comparison to the 12-megapixel resolution of both the Camera Module 3 options, which also support autofocus and high dynamic range. The field of view represents a further distinguishing feature between the models. The module 2 offers a horizontal field of view of 62.2 °, the standard module 3 a 66 ° field of view, and the wide lens a 120 ° field of view. The remaining features, including the CSI interface, weight, low power consumption, and dimensions, are largely consistent across the options. With regard to cost, the module 3 is available at a comparable price point. For this project, the camera module 3 with the wide lens as seen in Fig. 18 is the optimal choice due to its capacity to capture a broader perspective, which aligns well with the requirements of dynamic, high-altitude environments [14].

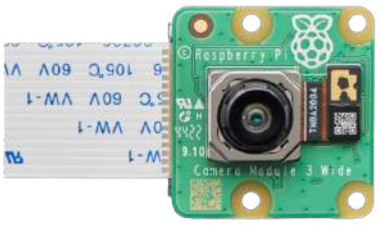

Figure 18: Raspberry Pi Camera Module 3 - 12MP 120 Degree Wide Angle Lens [14]

4.2.4. Power Supply

The power supply represents a crucial element of the flight computer system, ensuring the dependable functioning of all electronic components throughout the rocket's flight. The system has been designed to meet the electrical requirements of the Raspberry Pi Zero 2W, sensors and camera module, while also maintaining a lightweight and compact design. The total power requirement of the flight computer system was calculated based on the following components: The Raspberry Pi Zero 2W has a typical consumption of 1.2 W at 5 V, the DPS310 and ICM-20948 sensors have a combined power consumption of less than 5 mA, the GPS module has around 20mA consumption during operation and the Camera Module 3 has an additional load of 200 mA during active recording. To calculate the power consumption, the following steps were undertaken, as shown in Eq. (24), Eq. (25), Eq. (26), and Eq. (27).

- The current for the Raspberry Pi Zero 2W is calculated:

$$\begin{aligned}\text{Current (mA)} &= \frac{\text{Power (W)}}{\text{Voltage (V)}}\\ &= \frac{1.2}{5} = 250\text{ mA}\end{aligned} \tag{24}$$

- Adding up the current consumption of all the components:

$$\begin{aligned}\text{Total Current Draw} &= 240\text{ (Pi)} + 5\text{ (Sensors)} + 20\text{ (GPS)} + 200\text{ (Camera)}\\ &= 465\text{ mA}\end{aligned} \tag{25}$$

- Assuming the system operates continuously for 15 minutes (including flight computer testing before launch), the conversion into hours is:

$$\text{Flight Duration (hours)} = \frac{15}{60} = 0.25\text{ hours} \tag{26}$$

- The total energy consumed during this time is:

$$\begin{aligned}\text{Energy Required (mAh)} &= \text{Current Draw (mA)} \cdot \text{Flight Duration (hours)}\\ &= 465 \cdot 0.25 = 116.25\text{ mAh}\end{aligned} \tag{27}$$

In order to satisfy the requirements mentioned above, it is necessary to use a battery that has a voltage more than 5V and a total capacity of 200 mAh, in order to accommodate safety margins. Consequently, the battery that was selected is the Cellevia Battery LP903450 as seen in Fig. 19, which is a 3.7 V, 800 mAh lithium polymer battery. The battery was also selected as its weight is equivalent to that of the Raspberry Pi, thus serving the function of a counterweight. Furthermore, a step-up converter was used in order to upgrade the voltage output.

Figure 19: LP903450 3.7 V 800 mAh Lithium Polymer Battery

4.3. Sensor integration on Rocket

The positioning of sensors is of paramount importance for ensuring the stability of a rocket, given that the weight distribution is not uniform and one side of the rocket may be more heavily loaded than the other. This disparity in weight distribution can result in a tilt towards the more heavily loaded side during flight. The flexibility in sensor placement afforded by the cables extending from the flight computer allows for sensors to be positioned in a range of locations. To this end, a tube has been printed for the ejection charge to go through it and for placing the Raspberry, battery, IMU and pressure sensor. Also mounts have been developed to facilitate the positioning of the camera and GPS, thereby ensuring balance and preventing the rocket from tilting during flight. At the top and bottom of the tube another design was made that keeps the tube centred and for the bottom part two half circles were printed in order to protect the flight computer from the ejection charge.

4.3.1. Placement of Microcontroller and Battery

The flight computer is composed of several components, the heaviest of which are the battery and the Raspberry with the hat, which each weigh 26 grams. These components are positioned at the centre of the rocket. This is also evident in Fig. 20. The Raspberry Pi and the battery's integration within the rocket is facilitated by the utilisation of tape, a method that ensures its precise placement in the confined space of the rocket.

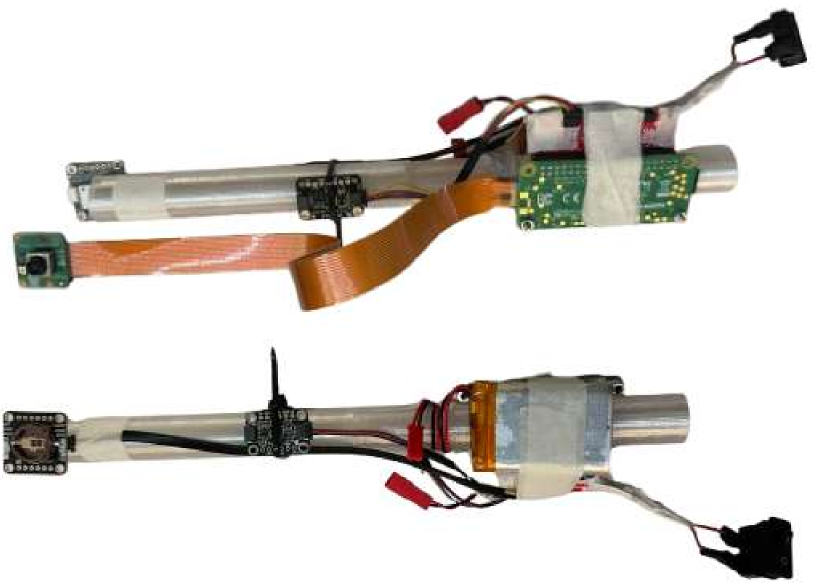

Figure 20: The tube with the Raspberry Pi and sensors

4.3.2. Placement of Sensors and Camera

The sensors of the flight computer are located at a point 10 centimetres above the battery and the Raspberry. The IMU accelerometer and gyroscope, as well as the DPS pressure sensors, are positioned at this location. Both sensors have a weight of 1 gram and in order to keep them in this position tire ups were used. The final sensor is the GPS, which has a weight of 3 grams, and the Raspberry Pi Camera, which also which has a weight of 3 grams. These modules have been placed at a height of 6 centimetres above the IMU and DPS sensors. As the camera has to be mounted on the body tube in order to have a view of outside of the rocket as demonstrated in Fig. 21. Each corner of the modules is equipped with four holes, enabling them to be positioned on the holes of the body, together with a screw and a nut for safety. Furthermore, a hole was made in the body tube of the rocket to accommodate the camera lens, enabling the external environment to be filmed.

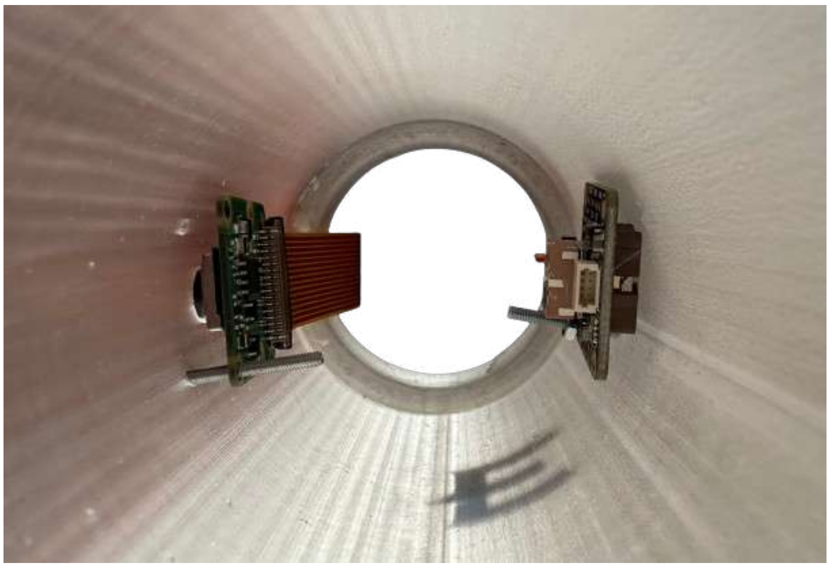

Figure 21: Camera and GPS

4.3.3. Challenges and Design Considerations

Mounting the rocket's sensors in the best position was a challenge, primarily due to their compact size and the need for lightweight mounts. Furthermore, the need for air to flow freely through the body tube in order to deploy the parachute via the ejection charge was a crucial factor. Initially, the concept of employing oversized mounts that would allow the sensors to slide along the side of the mount was considered. However, these mounts proved to be very heavy. Subsequently, the idea was to use the small holes in the corners of the sensors, with the use of the 3D printer. However, as these mounts were too small, they were prone to breaking. Ultimately, the decision was made to use screws, ensuring precise thickness, to achieve the desired outcome.

4.4. Telemetry System

The telemetry system is an important part of the Hermes rocket, facilitating real-time monitoring and post-flight analysis of critical flight data. There will be the possibility to transmit live data from onboard sensors to a ground station for the first few meters of the launch, before the Raspberry pi will disconnect from the Wi-Fi. The telemetry system is essential as it provides insight into the rocket's performance during the launch, flight, and recovery phases. This is of great importance to the project, as after the launch potential issues can be analysed and then changed before the next flight.

4.4.1. Live Data Transmission and Communication

In order to facilitate the transmission of sensor data to the ground station, an intermediate component is required. This component is the Raspberry Pi, which possesses the capability to transmit data via Wi-Fi or Bluetooth, in our case the Wi-Fi option was used. The Raspberry Pi's integrated code for data collection from the sensors also incorporates the command to transmit the data to the ground station, which has a dedicated code for this purpose, provided both devices are connected to the same Wi-Fi network. The functionality is further enhanced by the use of a mobile hotspot, as the launch site does not have Wi-Fi. After the Rocket, essentially the Raspberry Pi is out of reach from the mobile hotspot, the script will continue to run, meaning it will still store the data to the micro SD that is placed in the Raspberry.

4.4.2. Data Logging, Visualization, and Ground Station Integration

The ground station, in this case, a MacBook, receives real-time data transmitted via Wi-Fi from the Raspberry Pi flight computer using the User Datagram Protocol (UDP). This integration enables the live monitoring of flight parameters such as altitude, temperature, velocity, acceleration, vertical speed, orientation (pitch, roll yaw), rotation rate and finally rocket drift, providing immediate feedback to ensure mission success. Furthermore, the ground station processes and visualizes this data using custom Python scripts and libraries like Matplotlib, offering intuitive plots that facilitate real-time decision-making and post-flight analysis.

As illustrated in Fig. 22, the telemetry comprises a total of eight plots. The Temperature vs Time plot displays the onboard ambient temperature measured degrees Celsius and recorded by the the DPS310 pressure sensor. Next to it is the Altitude vs Time in both barometric and GPS. It compares the altitude estimated from the barometric sensor again measured in metres and calculated based on pressure while on the same plot we can also determine the altitude from the GPS sensor. The last plot on the top row is the Vertical Velocity vs Time plot, which computes the rate of change of barometric altitude between samples, offering us an insight into the rocket's ascent and descent dynamics. In the second row, we can see the Total Acceleration vs Time plot, which tracks the net acceleration magnitude that the rocket is experiencing, which is also derived from the IMU's accelerometer. Next to this is the Vertical Acceleration vs Time plot which shows only the z-axis acceleration component from the IMU. Finally on the second row is the Orientation vs Time plot, which illustrates the angular orientation of the rocket, derived from the IMU's accelerometer. On the last row is the Angular Rates vs Time plot which hows the real-time rotational speed (pitch rate, roll rate, yaw rate) measured by the gyroscope. Finally the Rocket drift vs Time plot the horizontal displacement of the rocket from the launch point in metres.

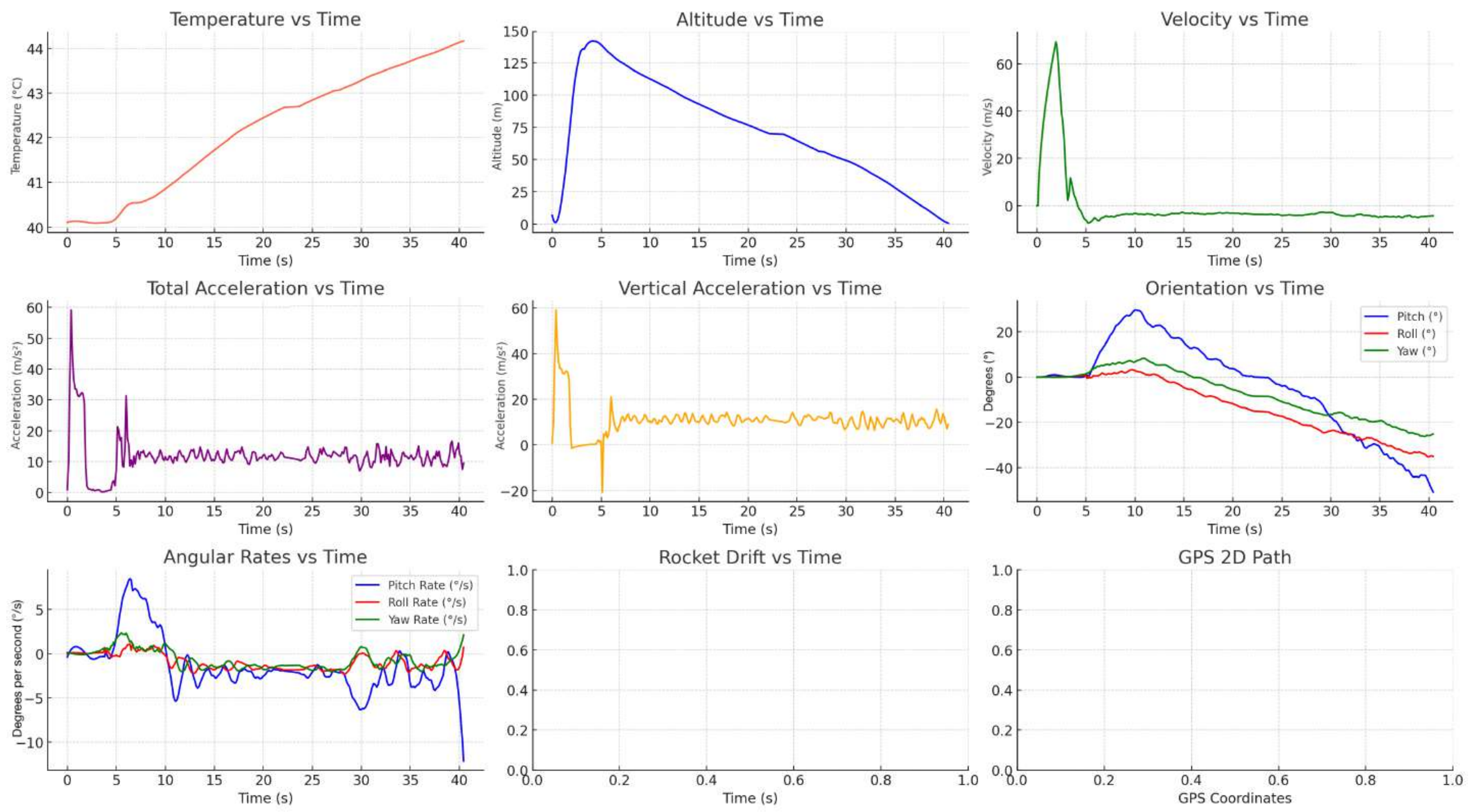

Figure 22: Data Visualisation

5. Propulsion System

5.1. Engine Selection and Performance

The selection of the engines had to be compatible with the rocket's weight, stability margin, target apogee, ejection delay, cost and availability on the market, as well as on the OpenRocket simulator that was used. Given the substantial rocket weight, there was a requirement for powerful engines. Consequently, as a result of the aforementioned considerations, the engines would have to be multiple Klima D-class engines or TSP E-class engines, as the majority of the others are not readily available in Europe. Following a cost-benefit analysis, the Klima D9-5 as shown in Fig. 23 solid rocket engine was selected. This engine has a total impulse of 20 Ns, an average thrust of 9 N, a maximum thrust of 25 N, a propellant weight of 16 g, a burning period of 2.1 s, and a specific ejection delay of 5 s, which ensures the recovery system deploys at apogee. The engine's dimensions are 70 mm in length and 18 mm in diameter, and a total weight of 27 g, of which 16 g is black powder propellant. A single engine would not be sufficient for the Hermes rocket to overcome Earth's gravitational pull and also reach the desired apogee of arounf 150 metres. Consequently, the use of three engines would be the optimal configuration [15].

Figure 23: Klima D9-5 engines

5.2. Thrust Curve Analysis

The thrust curve of the Klima D9-5 engine is an important factor in determining the rocket's flight profile. As illustrated in Fig. 24, the engine's peak thrust of 20 N is attained immediately following the start of the ignition process. This results in a rapid acceleration, enabling the rocket to overcome the forces of gravity and drag. Over the 2.1 s burn duration, the thrust gradually decreases, stabilising at an average of 9.2 N. This smooth thrust profile is conducive to minimising stress on the rocket's structure while maintaining stability. The total impulse of 20 Ns guarantees that the rocket attains its target apogee under nominal conditions. The Klima D9-5 engine was available directly in the OpenRocket motor library, meaning its predefined thrust curve was used in the simulation. This allowed for more accurate predictions of performance and flight dynamics. The thrust curve analysis also makes sure that the engine provides enough force to maintain a thrust-to-weight ratio greater than 5:1, a critical parameter for safe and stable flight [15].

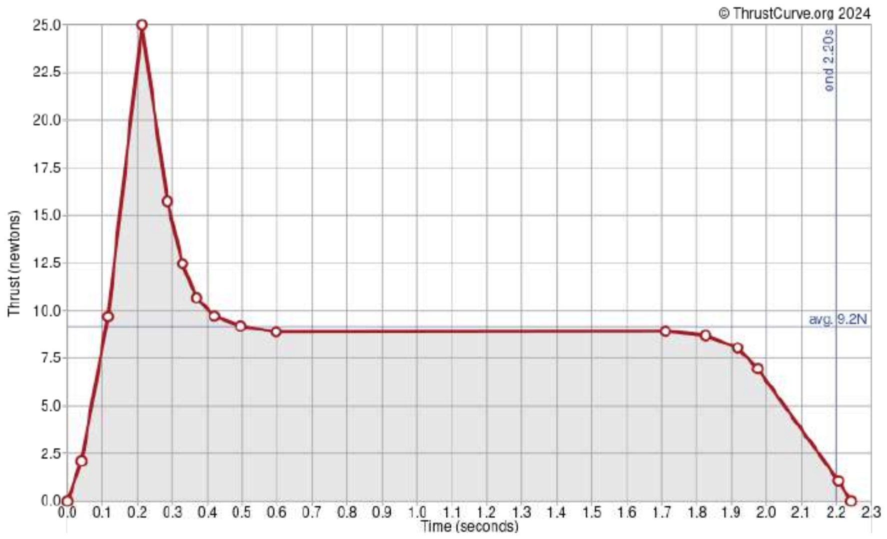

Figure 24: Thrust Curve of Klima D9-5 Engine [15]

5.3. Integration with Rocket Design

The D9-5 engines are mounted in a custom-design engine mount that can be seen in Fig. 25. The mount has been designed in a way that in fits precisely within the body tube where it was then stabilised with the epoxy glue. This, guarantees the alignment of the thrust vector during the launch phase with the rocket's centre of gravity thereby maintaining stability. In more detail, two engine mounts have been positioned at one and five centimetres from the bottom of the tube. The dimensions of the 3-ring cluster design have an outer diameter of 63.2 mm and a thickness of 5 mm.

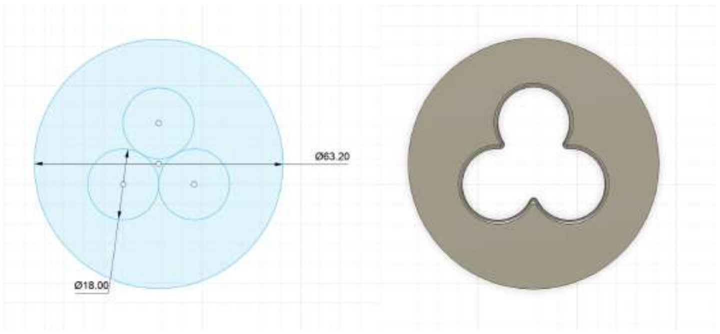

Figure 25: Engine Mount on Fusion 360

6. Rocket Simulation and Stability Analysis

6.1. Stability Principles

Stability is an important part in model rockets, it makes sure that the rocket maintains a predictable and controlled flight trajectory while ascending and descending. Having a stable rocket means that it will not deviate from its planned trajectory due to aerodynamic forces, but instead helps it to achieve this trajectory with precision. The three main objectives of the stability analysis are to optimise the rocket's flight trajectory, to keep it safe and finally to verify that the design parameters are aligned with the aerodynamic criteria. This is achieved by balancing the rocket's CG and CP to ensure an adequate stability margin, which is commonly expressed in terms of the rocket's body diameter (Cal). The optimal range for most model rocket is between 1-2 Cal, as it makes sure that it is stable enough under different flight conditions. Stability analysis is instrumental in refining design parameters, such as fin size and placement, or internal component distribution, thereby ensuring the rocket's capacity to withstand real-world disturbances, including wind and uneven thrust. The incorporation of these principles into the design process enables engineers to enhance performance, mitigate risks, and ensure the success of mission outcomes [16].

6.1.1. Centre of Gravity (CG) and Centre of Pressure (CP)

Rocket stability is governed by the relative positions of the CG and CP. When a free body in space rotates, it spins around an imaginary point where all its mass appears to be concentrated, this is the balance point, or else called the centre of gravity. On the other hand the point where all the aerodynamic forces act is called the centre of pressure.

6.1.2. Stability Margin and Caliber

Having established the definitions of the CG and CP, the subsequent section will address the consequences of these points on the stability of a model rocket. It should be noted that three distinct stability conditions can be identified, as illustrated in Fig. 26.

1. Positive stability: In this condition, the model rocket's CG is positioned ahead of the CP. The design of the model rocket is characterised by the presence of large fins set far back on the body tube. This results in the rocket flying straight and weathercocking into the wind at launch.

2. Neutral stability: In this case, the CP and CG are at the same position on the model rocket. This may occur if the rocket has a lightweight nose or fins that are too small, or both. In the absence of stabilising and restoring forces in the model rocket, this may result in unpredictable paths.

3. Negative stability: In this condition, the CG is behind the CP. In this scenario, the aerodynamic forces on the fins tend to cause the model to fly tail-first, which is not possible under power. As the nose has any pitch or yaw movement, a force exists to keep it swinging, resulting in the model pinwheeling end over end and eventually descending to the ground [1].

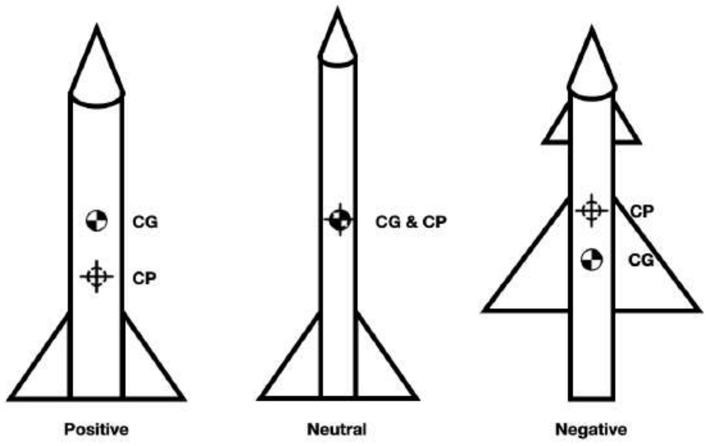

Figure 26: The three stability conditions with their CG-CP

It is important that the model rocket has the CG ahead of the CP. This is shown in Equation Eq. (28) and it represents the distance between the CG and the CP, expressed in terms of the rocket's body diameter.

- The definition of the caliber is derived from the following equation:

$$Cal = \frac{\text{Distance between } CG \text{ and } CP}{\text{Body Tube Diameter}} \tag{28}$$

There are four interpretation of Cal values:

- 0 cal: The CG and CP are aligned, thereby rendering the rocket highly unstable. This condition has previously been described as neutral stability.
- <1 cal: The rocket is marginally stable and may not be able to fly in a straight path, particularly when subjected to wind disturbances.
- 1-2 cal: The rocket exhibits ideal stability, which is the optimal range for most model rockets. This condition ensures that the rocket flies straight while maintaining sufficient manoeuvrability.
- >2 cal: The rocket may exhibit excessive stability, thereby impeding the process of adjustment and potentially resulting in a deviation from the intended trajectory in the presence of strong winds [16].

The location of the CG and CP of the Hermes rocket can be determined by observing the OpenRocket application. Specifically, from the tip of the rocket, the CG is situated at 570 mm and the CP is at 703 mm. This shows that the Cal value is 1.98, suggesting that the rocket maintains ideal stability and would only deviate from its trajectory in the presence of strong winds.

6.2. Weight Distribution

As previously mentioned it is vital to make sure there is optimal weight distribution within a rocket, in order to maintain stability and ensure the rocket follows the intended flight path without deviating excessively to either side due. In order to achieve this, the sensors and all other mass components within the rocket are placed in equal positions. This has been achieved based on the weight of each sensor and then by placing them opposite each other so they can cancel out as depicted in Fig. 27. To illustrate this approach, the IMU and the DPS310, both of which weigh 1 gram. A similar principle applies to the Camera and GPS, which each weigh 3 grams. The battery and the Raspberry, on the other hand, have weights of 26 grams each. However, the Raspberry also contains the step-up converter and the rocker switcher, which powers the flight computer. These components cancel out the weight difference. The engines are another component that is very heavy, with a total mass of 81 grams, but with the help of the engine mounts, they are positioned centrally. The adapters that connect the body tubes together are in total three, but they are placed in equal distances. Finally, the parachute, which is of equal width to the body tube, is located in close proximity to the first adapter, thus ensuring a centralised positioning of the system.

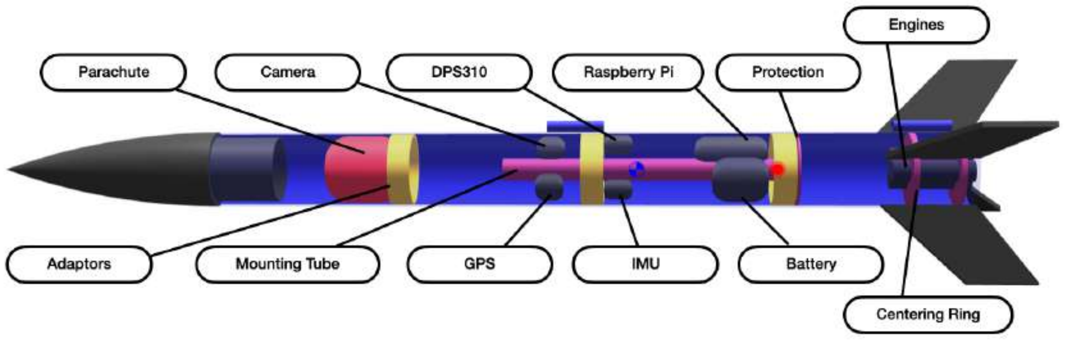

Figure 27: Weight Distribution

6.3. OpenRocket Documentation and Justification of Simulation Differences

6.3.1. How OpenRocket Works and Its Assumptions

The OpenRocket software was utilised for the purpose of simulating the flights of the rocket. OpenRocket is an open-source software that can be used for model rocket simulators. It facilitates the design and simulation of rockets prior to construction and launch. The utilisation of simulations enables the prediction of key parameters, optimisation of rocket design and assessment of conditions such as weight distribution, engine clusters, and parachute deployment, with the objective of determining their impact on the overall flight plan. These simulations are founded on aerodynamic theories and numerical methods to predict the rocket's flight characteristics. A more detailed analysis of the architecture of OpenRocket reveals that it is build as a modular Java system that has core functionalities like graphical user interface, the simulation engine as well as aerodynamic calculators, which are split into dedicated software package. These details have been extensively described in *Development of an Open Source model rocket simulation software* a master's thesis by Sampo Niskanen [5]. Rocket designs are represented through a component-based structure, in which each component like the nose cone, body tube, or fins is defined as a subclass of the abstract Rocket Component class. This structural design produces a tree-like model that replicates the physical rocket design, with additional interfaces such as Motor Mount and Clusterable allowing for the flexible modeling of multi-motor clusters and staging setups. Additionally, OpenRocket uses reflective programming techniques in order though to maintain modularity. The software design is structured in a way that the system is both adaptable and efficient, thereby supporting a wide variety of rocket configurations and simulation scenarios. The component architecture is illustrated in the diagram presented in Fig. 28.

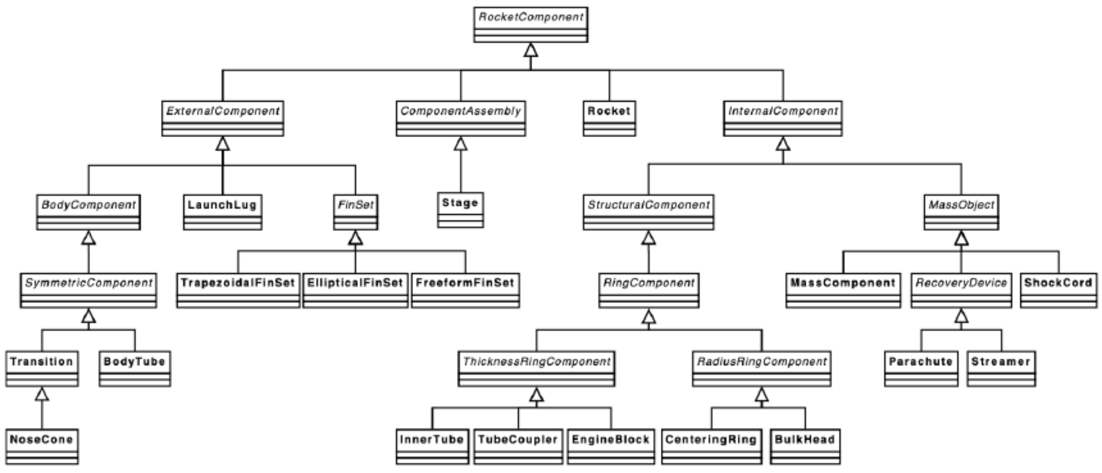

Figure 28: A diagram of the rocket component classes [5]

Furthermore, except for the core functionality of OpenRocket that it offers, it is also designed to be modular. As previously mentioned, the modular Java System allows the advanced users to add new aerodynamic models or simulation methods like swapping Runge-Kutta for Euler integration or incorporating CFD, without the need of rewriting the whole system. Additionally, another feature is the use of simulation listeners that are able to interact with the simulation in real time. They have the ability to monitor flight progress, collect custom data as well as modify the rocket's properties while in simulation, thus making possible to test active control systems. All these features make OpenRocket not only a design tool but also a flexible platform for research and advanced experimentation.

6.3.2. Core Equations and Calculation Methods

As outlined in the master's thesis *Development of an Open Source model rocket simulation software* by Sampo Niskanen [5], the three core equations implemented in OpenRocket are presented below. These include the stability calculation using the Barrowman equations Eq. (29), the drag force estimation Eq. (30), Eq. (31), and the motion simulation based on the 4th-order Runge-Kutta method Eq. (32).

1. Stability Calculations:
   The initial step in this process is to consider the core equations and calculation methods that are utilised by OpenRocket. For instance, the stability of a rocket is determined by the Barrowman equations. These equations calculate the centre of pressure based on the geometry and configuration of the rocket components. There is no external database utilised in this scenario, Instead, OpenRocket employs real-time calculation of the centre of pressure (CP) and centre of gravity (CG) from the rocket's input geometry. The normal force coefficient derivatives C_{Na} for each component are estimated using analytical formulas derived from the Barrowman method, and subsequently summed as:

$$X_{CP} = \frac{\sum X_i C_{Na,i}}{\sum C_{Na,i}} \tag{29}$$

   - Where:
     - X_i is the position of the component's centre of pressure.
     - $C_{Na,i}$ is the normal force coefficient derivative for the component.

2. Drag Force Estimation:
   As is the case in the Hermes project, the rocket reaches only subsonic speeds (Mach < 0.8). Therefore, to calculate the drag force, OpenRocket uses the classical equation:

$$F_D = \frac{1}{2}\rho u^2 C_D A \tag{30}$$

Where the total drag coefficient C_D is calculated by summing individual contributions:

$$C_D = C_{D_{\text{body}}} + C_{D_{\text{fin}}} + C_{D_{\text{base}}} + C_{D_{\text{parasitic}}} \tag{31}$$

- Where:
  - $C_{D_{\text{body}}}$ is the drag from the body tube.
  - $C_{D_{\text{fin}}}$ is the drag from the fins.
  - $C_{D_{\text{base}}}$ is the drag from the base due to low-pressure zones.
  - $C_{D_{\text{parasitic}}}$ is the drag from the launch lugs or other small protrusions.

The estimated values for each component are obtained using empirical data from wind tunnel experiments and established aerodynamic tables, based on slender-body assumptions derived from the Barrowman method. As the Hermes rocket operates entirely within the subsonic regime, supersonic and transonic corrections are not applied here.

3. Motion Simulation:
In relation to the motion simulation, the rocket flight is simulated by employing the 4th-order Runge-Kutta method to solve the equations of motion. But in this case OpenRocket relies on external data. Specifically, the thrust curves come from a database such as thrustcurve.org [15] which provide time vs. thrust profiles based on manufacturer or measured data. Additionally, the mass depletion is calculated based on total propellant mass and burn time from the motor file and finally it uses default atmospheric condition, unless the uses inputs custom values.

The core equation is as follows:

$$\frac{d^2x}{dt^2} = \frac{F_{\text{thrust}} - F_{\text{drag}} - F_{\text{gravity}}}{m} \tag{32}$$

- Where:
  - F_{thrust} is the thrust force from the engine.
  - F_{drag} is the air resistance or drag force.
  - F_{gravity} is the gravitational force.
  - m is the rocket mass, which decreases over time as the fuel burns.

6.3.3. Simulation Assumptions

The core equations and calculation methods alone are insufficient to yield the final result, and consequently, OpenRocket also bases its simulations on a combination of aerodynamic theories and numerical methods, incorporating several assumptions to simplify and optimise the simulation process. Firstly, the rocket is assumed to be axially symmetric. This assumption simplifies the calculations by presuming that the aerodynamic forces are uniformly dispersed along the rocket's axis. This is a common assumption in the majority of traditional rocket designs, where the fins and body components exhibit symmetry. In the context of ideal motor performance, the motor thrust is modelled based on thrust curves provided by the manufacturer. These curves represent optimal conditions and do not consider environmental variations, such as temperature. Consequently, minor discrepancies may emerge in the motor performance predictions. Additionally, OpenRocket simplifies the complex behaviour in the transonic and supersonic regimes by using predefined drag coefficients, thus facilitating high-speed aerodynamics. Unless modified by the user to reflect actual launch conditions, the simulations assume standard atmospheric conditions. Another assumption that, unless explicitly defined, are the wind conditions and turbulence, which are not factored into the simulation. However, these can be changed if the relevant data are available [5]. Finally, for the recovery phase like the parachute deployment in the case of this project, OpenRocket models it as an event triggered stage where the drag coefficient is getting increased significantly during deployment. The user has the option to input the drag coefficient and effective surface area manually or to select from default preset values in order to model the parachute's drag. In contrast to the thrust curves that rely on external motor databases, the parachute functions independently. When the parachute is deployed, the simulation tracks the descent phase by calculating the new drag values and recalculating the rocket's deceleration, descent rate, and terminal velocity accordingly.

6.3.4. OpenRocket Simulation Features and Output Parameters

In order to achieve the most accurate simulations on OpenRocket, it is possible to input parameters such as launch conditions, aerodynamic properties, and propulsion data. Following this, the user has the option to generate various plots, including the final trajectory, acceleration and forced acting on the rocket, drag coefficient vs Mach number, parachute deployment sequence and descent rate, and stability metrics such as CG and CP over time. It has been demonstrated that OpenRocket predictions, when substantiated by experimental validation, exhibit an accuracy of 10-15 percent relative to real flight data. Variations in performance are attributed to wind effects, surface irregularities, and discrepancies in motor performance.

6.3.5. Possible Reasons for Differences

Additionally to the experimental factors, the simulation model includes also assumptions and uncertainties. Specifically, the aerodynamic models assume axisymmetric, rigid-body rockets with smooth surfaces, neglecting the influence of manufacturing imperfections, surface roughness, or minor misalignments. In addition, environmental conditions such as wind and turbulence are only used if the user inputs them, in all other cases, the simulation utilises default settings that do not allow for site-specific variation. Moreover, the motor performance is simulated by using the ideal thrust curves, as previously described in Section 5.2, and does not account for temperature effects, nozzle erosion, or inconsistencies in ignition. These parameters cannot be adjusted or inputted at a later stage in the OpenRocket simulation. Another important note is that the empirical drag coefficients that are being used are derived from simplified fittings and therefore lack the ability to accurately capture complex fluid dynamics such as turbulent wakes or Reynolds number sensitivity. A summary of the key simulation assumptions and their potential impacts is provided in Table 3. Overall, these assumptions introduce systematic variations, which can result in discrepancies between simulated predictions and actual flight performance [5], [16].

Category	Assumption	Potential Impact
Rocket Geometry	Axisymmetric, rigid-body rockets with smooth surfaces	Ignores flexing, warping or off-center mass
Aerodynamic Modeling	Uses empirical drag coefficients from slender-body models	Not accounting for turbulent wakes or surface texture
Motor Performance	Uses idealised thrust curves from databases	Not accounting for temperature, nozzle wear, or misfire
Atmospheric Conditions	Assumes standard atmosphere (unless user inputs custom data)	Ignores site-specific wind, temperature, or pressure
Recovery Modeling	Applies user-specified or default parachute parameters	Errors in drag coefficient or area affect descent rate
Numerical Integration	Uses 4th-order Runge-Kutta with fixed time-step	Small numerical errors may accumulate over long flights

Table 3: Summary of key simulation assumptions in OpenRocket and their potential impacts on accuracy

7. Recovery System Design

The recovery of the rocket in an intact state is of great importance in this project, as the rocket is required to be able to fly multiple times and it would also prevent any damage to the flight computer. In this particular project, the parachute recovery system has been selected. This recovery was first used in model rockets by Orville H. Carlisle in 1954, and since then there has been a great deal of research into parachutes, with these systems still being used in many cases today. The parachute is affixed to the rocket's nose cone, and in conjunction with a shock cord, it constitutes a highly reliable recovery system [1].

7.1. Parachute Design for Safe Descent Rate

Parachutes are available in a variety of shapes, including circular, square, hexagonal, and octagonal designs. The prevalence of circular parachutes is low, as the fabrication process for hexagonal and octagonal parachutes is more straightforward. The parachute's design incorporates a targeted descent rate of 4-6 m/s to ensure a safe and controlled descent while also keeping an easy design using a semi-spherical octagonal. This rate is calculated to minimise the risk of damage to the rocket during landing while ensuring a timely recovery. The diameter of the parachute was calculated using the drag equation, as shown in Eq. (33), Eq. (34), and Eq. (35), with a resultant measurement of 77 cm [1].

- The drag equation is calculated as:

$$D = W = 0.5 \cdot \rho \cdot V^2 \cdot C_d \cdot A \tag{33}$$

- Where:
  - W is the rocket's weight.
  - ρ is the air density, which at sea level is typically 1.225 kg/m^3.
  - V s the desired descent velocity.
  - C_d is the drag coefficient, which for a hemispherical parachute is approximately 1.5.
  - A is the parachute's area.

In greater detail, the parachute area is calculated as follows [17]:

$$A = \pi \left(\frac{d}{2}\right)^2 = \frac{\pi d^2}{4} \tag{34}$$

7.2. Material Selection, Size, and Deployment Mechanism

The parachute has a fabric that is super soft, light and tear-resistant quality. The parachute's design incorporates eight shock cords, which contribute to its structural integrity. These cords are engineered to possess both tensile strength and resistance to heat from the ejection charge.

- Plugging the area into the drag equation and solving for d, we get:

$$d = \sqrt{\frac{8 \cdot m \cdot g}{\pi \cdot \rho \cdot V^2 \cdot C_d}} \tag{35}$$

The dimensional calculations were determined by the total mass and the desired descent rate. The shock cord, a synthetic material characterized by its elastic properties, was segmented into two sections: 60 cm from the body tube to the nose cone, and 6 cm from the swivel to the parachute itself as it can be seen in Fig. 29. This configuration was selected to minimize stress on the attachment points. The deployment of the parachute is contingent upon the utilization of a black powder ejection charge within the engines, which functions to disengage the rocket's nose cone from the body tube at apogee. This enables the parachute to be deployed. The system is initiated by the five-second delay inherent in the engines, a concept elaborated upon in Section 5. Prior to the integration of a swivel mechanism in the design, measures were implemented to avert the entanglement of the parachute chords during descent, thereby ensuring a seamless and regulated recovery.

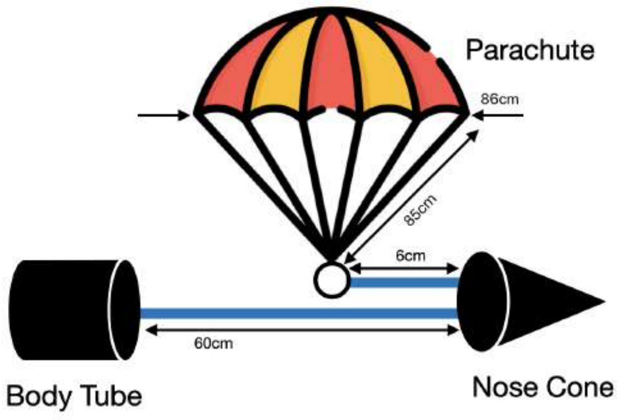

Figure 29: The parachute system with dimensions

8. Testing and Validation

As with all aerospace missions, there are some requirements that have to be set first and tests are designed to verify that each requirement is met. These requirements are about the rocket, the flight computer, the parachute and telemetry. Specifically, they ensure that the flight computer and sensors function correctly in the expected environment. The rocket structure is aerodynamically stable, the parachute recovery system deploys reliable and finally the telemetry systems stores flight data without significant or any losses.

ID	Requirement Description	Verification Method
R-1	The flight computer shall collect sensor data (pressure, altitude, temperature, GPS, accelerometer, gyroscope, magnetometer) at a minimum rate of 1 Hz	Sensor data logging test
R-2	The flight computer shall store data over a distance of at least 200 metres	Telemetry transmission range test
R-3	The camera shall store video feed over a distance of at least 200 metres	Video Camera storing range test
R-4	The parachute shall eject and deploy upon activation of the ejection system charges	Drop test and ejection test
R-5	The power system shall operate for at least the entire mission duration (15 minutes)	Battery endurance test
R-6	The aerodynamic stability margin shall be within $\geq$ 1.5 cal (calibers) to ensure a stable flight trajectory	Swing test

Table 4: Requirement Description and Verification Method

8.1. Pre-Flight Electronics Validation

Starting with the sensor data, telemetry transmission and camera test (R-1, R-2, R-3) as they can be verified simultaneously. The objective is to verify that the sensors record accurate data and the Raspberry Pi shall store data and the camera stores video feed while it is at least over 200 metres away from the ground station (which is the laptop). The procedure here is:

1. Power on the flight computer and connect it via ssh to a laptop.
2. Set up the Raspberry Pi for live data transmission and the camera.

3. Record sensor data and video feed while the system remains stationary.
4. Walk away from the ground station at increasing distances.
5. Complete the data collection and switch off the Raspberry Pi.
6. Check for data loss.

In order for the requirement to be successful the data should be recorded at ≥ 10 Hz and that there is a stable transmission at 200 metres with minimal to no data loss. On 20 March 2025, the test was successfully conducted by initially utilising live telemetry for the initial 60 metres, followed by the relocation of the flight computer after its disconnection from the Wi-Fi network and subsequent power interruption. Upon reconnection, it was determined that the flight computer continued to store data up at 1 Hz to the point of the power interruption and the video feed was saved successfully. Additionally this test was conducted every time before each flight.

8.2. Launch and Recovery Testing

In this test the objective is to ensure that the parachute eject and deploys successfully (R-4). The procedure would be:

1. Place the parachute and ejection system in the rocket.
2. Activate the parachute ejection charge.
3. Monitor the parachute deployment.

For the test to be successful, the parachute must deploy successfully. On March 26, 2025, the parachute was folded in accordance with the established protocol, prior to its deployment on the rocket. The parachute was then ejected by blowing air into the rocket, where the actual ejection charge will be located.

8.3. Battery Endurance Test

In this case the objective is the that the power system supports the mission duration (R-5). The procedure for this test is:

1. Fully charge the battery system.
2. Run the entire flight computer system, sensors, and transmission module for at least 15 minutes.
3. Monitor the battery over time to detect any disconnections.

For the test to be successful the system shall operate continuously. On March 21, 2025, the battery was first fully charged, then the GPS test was conducted, which is the test for the GPS to connect to a satellite for 15 minutes and after that the scripts that will also be used at the launch were ran, which are the data storing and video storing for another 10 minutes.

8.4. Ground Testing of Aerodynamic stability Test

Continuing with the aerodynamic stability (R-6) the objective here is that the rocket's stability margin is $\geq$ 1.5 calibers. This would be tested with a swing test. The procedure is:

1. Find the rocket's CG.
2. Attach a string to the CG and spin it to check for stable flight behaviour.

As here it would be difficult to check the exact caliber number the pass criteria would be that the rocket maintains a stable flight path during the test. Before each flight this test was conducted to ensure the weight distribution was correct.

9. Results and Discussion

9.1. Overview of Test Flights

In total, there were 12 engines and as three engines are required per launch there were overall four launches to evaluate the rockets flight performance, to access the sensors reliability and finally to validate the accuracy of the simulation predictions in OpenRocket. The goal of each flight was to capture data such as temperature, barometric altitude, GPS altitude, velocity, vertical speed, acceleration magnitude, orientation (pitch, roll, yaw), rotation rate, and GPS path. All the sensors on the flight computer were thoroughly tested prior the launch as discussed in previous sections. In Table 5, the conditions and weather data of each flight are summarized; these environmental parameters were also input into the simulation to improve prediction accuracy. This process improved the predicted descent rates and flight durations, which showed that even small corrections to environmental and propulsion parameters can significantly improve the reliability of the simulation outputs.

Parameter	**Flight 1**	**Flight 2**	**Flight 3**	**Flight 4**
Launch Date	05.04.2025	18.04.2025	04.05.2025	11.05.2025
Launch Site Elevation	35 m	2 m	175 m	19 m
Temperature	19 $°C$	17 $°C$	22 $°C$	15 $°C$
Atmospheric Pressure	994.47 hPa	1012.55 hPa	991.93 hPa	1015.64 hPa
Relative Humidity	55 %	60 %	45 %	68 %
Wind Speed	15 km/h	7.2 km/h	11 km/h	7 km/h
Wind Direction	258 °	120 °	201 °	144 °

Table 5: Flight Conditions and Environment Summary

9.2. Flight Data Analysis

This section presents the results of each flight, which were then compared with the simulation data. In the subsequent section, a comparison is drawn between the altitude, velocity, vertical speed, and acceleration magnitude plots, as these are the only variables that can be simulated. The remaining plots are either not available for simulation, as is the case with the temperature inside the rocket, or OpenRocket assumes ideal vertical flight, thus resulting in the absence of 6 Degrees Of Freedom (DOF) attitude or in-flight dynamics.

9.2.1. Flight 1: Data overview

Flight 1 was conducted on April 5, 2025. As illustrated in Fig. 31, the complete set of telemetry plots acquired from the first flight is presented. The visualisation of these plots, which utilises a 1 Hz sampling rate, offers a comprehensive representation of the rocket's behaviour throughout all phases of its flight trajectory. Before the flight we can see clearly that the GPS did not lock with a satellite and there were no data, meaning the GPS path and the Rocket drift plots are not available. The rocket launches immediately and hits the ground at 30 s. Starting with the temperature plot, we can see that it starts at around 42.5 $°C$. This temperature is likely the result of internal heating or the influence of nearby electronics. The temperature rises to a peak of around 50 $°C$ during the ejection charge. It is evident at this point that the heat has penetrated the protection system and transferred to the flight computer. The vertical velocity shows a sharp spike at the launch, then peaking at around 2 s with 42 m/s and then a clear drop that shows the parachute deployment and at 30 s it stabilises showing that the rocket hit the ground. In terms of vertical and total acceleration, there is a spike at the start, peaking at around 2 s and reaching 22 m/s^2 for total acceleration and around 15 m/s^2 for the vertical acceleration. As for the orientation plot, the pitch shows significant changes in the rockets nose angle, especially during the ascent part, as for roll it fluctuates more consistently showing some imbalance or aerodynamic forces and finally yaw remains relatively stable, with minor adjustments in heading. All three lines after 12 s stabilise, showing steady flight after ascent and recovery deployment. Finally about the angular rates plot, pitch again shows sharp fluctuations at launch, the roll rate has some periodic fluctuations probably due to recovery dynamics and lastly the yaw rate has smaller vibrations that the other two. As illustrated in Fig. 30, the last frame of the camera is visible. The issue here was that the camera was disconnected from the Raspberry Pi, which additionally was terminally damaged by the ejection charge penetrating the protection system. Consequently, it was necessary to design an improved protection system for the next flight.

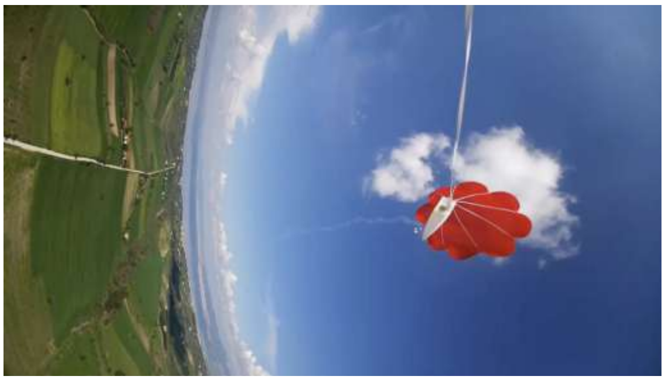

Figure 30: On-Board photo from Flight 1

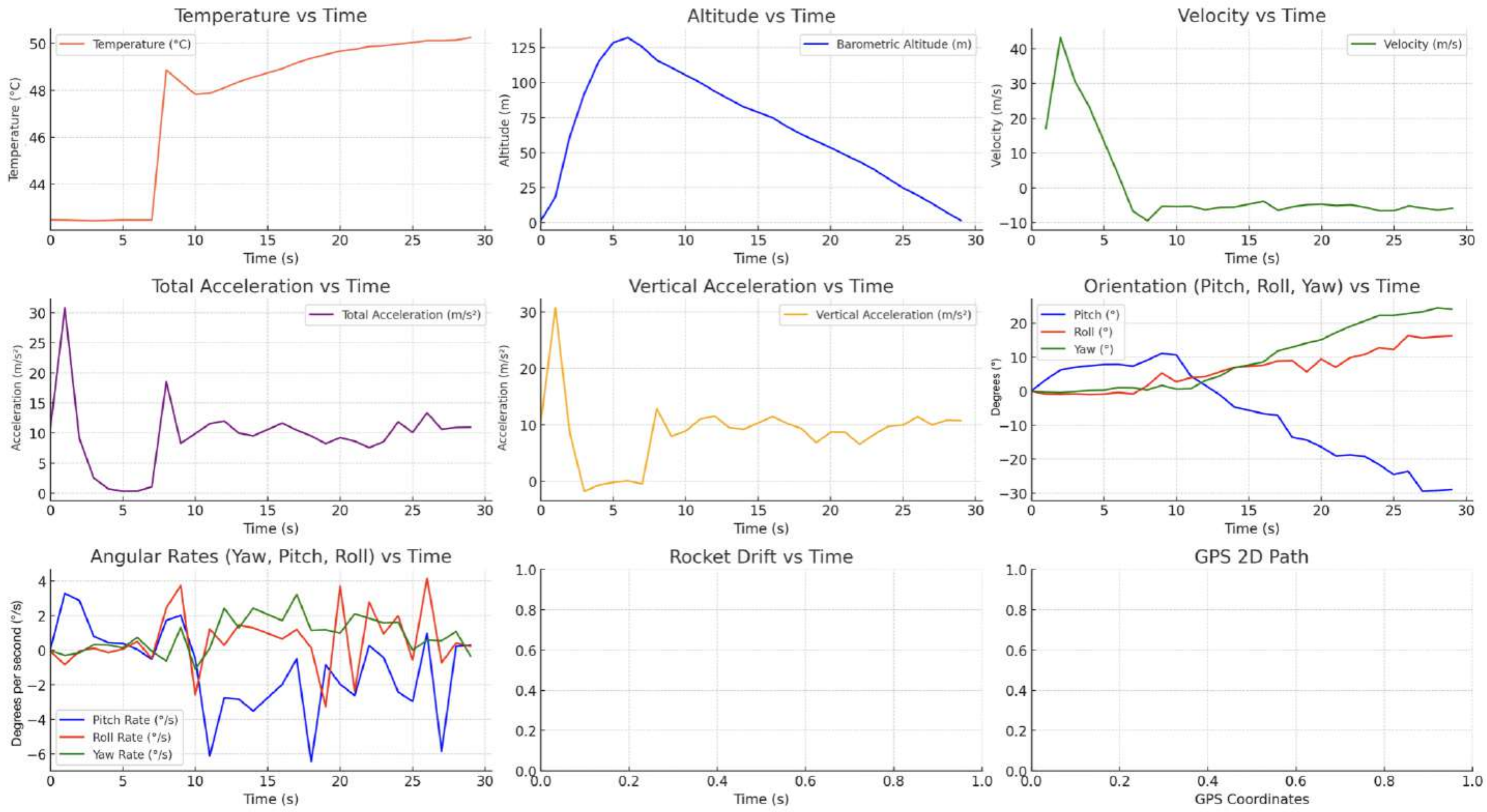

Figure 31: Telemetry data from the first flight

Prior to conducting a more detailed examination of the plots, it is important to acknowledge that the sampling rate was inadequate, potentially resulting in the highest point of any data set occurring between two samples. The altitude plot in Fig. 32, where the blue line represents the actual altitude achieved by the rocket and the red line represents the simulated altitude in OpenRocket, demonstrates that the simulated apogee was not attained due to the aforementioned reason. Additionally, the rocket's landing may have been delayed due to strong winds, which would be consistent with the observed data.

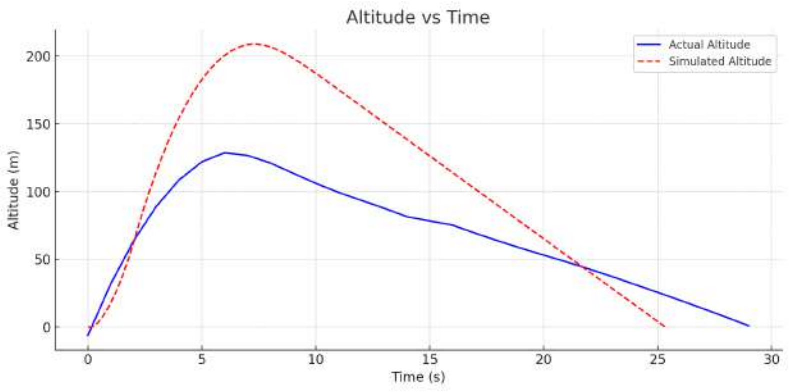

Figure 32: Flight 1 - Comparison of Actual and Simulated Altitude

As demonstrated in Fig. 33, the velocity plot reveals that the peak in the simulation occurs at approximately 55 m/s, while the peak in the actual plot is around 45 m/s. Following the deployment of the parachute, there was a substantial decrease in velocity, with the minimum vertical velocity being -12.5 m/s in the simulation, at which point it stabilises. In the actual plot, the minimum vertical velocity is -10 m/s, but it fluctuates slightly. The difference between the two plots may be due to sampling rate, uneven deployment of the parachute, or irregularities in air resistance.

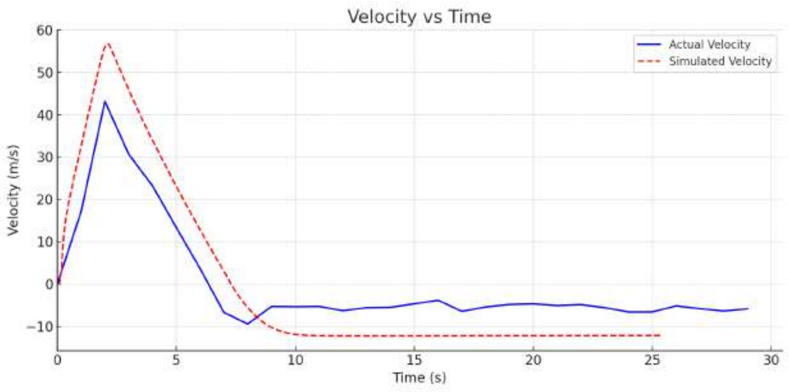

Figure 33: Flight 1 - Comparison of Actual and Simulated Velocity

Finally, with regard to the acceleration plots in Fig. 34, it should be noted that the actual data represents apparent accelerations, i.e., accelerations without gravity subtracted. This ensures that the acceleration curves peaks closer to the presentation of net forces by OpenRocket. However, it should be noted that the peak acceleration values in the actual data may differ from the simulation due to real-world effects such as sensor sampling limitations, and because the highest recorded values could have occurred between two samples. In more detail both plots for the simulated data follow a smoother trend, with sharp peaks at key events but in the actual data there are more fluctuations. Furthermore, for the simulated total acceleration there are some relative steady values after the burnout but on the other hand the actual total acceleration has significant variation in later stages. This is likely caused by forces such as wind or slight deviations in the flight path.

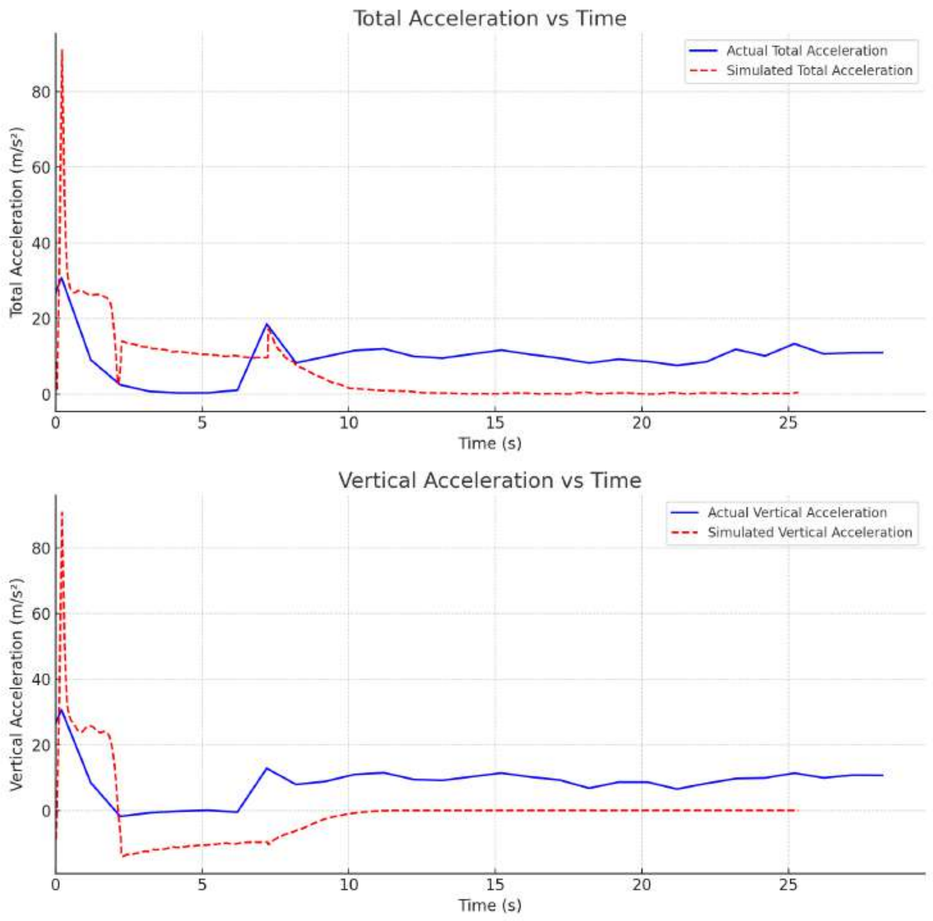

Figure 34: Flight 1 - Comparison of Actual and Simulated Total and Vertical Accelerations

9.2.2. Flight 2: Data overview

The second flight occurred 13 days after the first flight, as it was imperative to ensure the problems that had arisen were resolved. The most significant update to the software was the alteration to the 10 Hz sampling rate. But in order to keep the Raspberry Pi from overloading this also meant dropping the live telemetry system and reducing the camera quality from 1080p to 720p, but keeping the Frames per Second (FPS) at 30. Furthermore in Fig. 35, a more robust protection system for the ejection charge was devised to prevent the disconnection of the camera from the Raspberry Pi, as well as the failure of the Raspberry Pi itself.

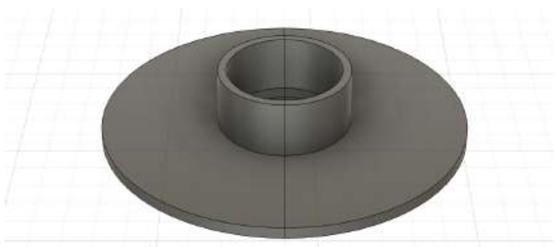

Figure 35: Ejection Charge Protection System

From Fig. 36 we can again see that the GPS did not work which leads us to the conclusion that the sensor is faulty. The most noticeable observation is that the data stopped being recorded after nine seconds of flight. After the flight it was noted that the button had been deactivated, indicating that it had been pressed during the flight. Unfortunately, this was not the most significant issue experienced during the flight. At approximately apogee, the parachute was ejected at a slightly delayed time, and upon doing so, the nose cone became detached from the remainder of the rocket. This resulted in the rocket experiencing a rapid descent from an altitude in excess of 100 metres, ultimately leading to a collision with the ground. This resulted in the braking of two body tubes, although the flight computer survived the crash. Prior to the analysis of the flight, it is essential to note that the temperature within the rocket was recorded at 34 $°C$ before launch, and at the ejection charge, there was a mere increase of 0.15 $°C$ in temperature. This indicates that the enhanced protection system was operational. As illustrated by the altitude plot, there is a consistent increase to a height of approximately 130 metres before the cessation of the data. The vertical velocity plot reveals a sharp increase during the launch phase, reaching a peak at 2.5 s with a velocity of around 60 m/s. This is followed by an unanticipated decline to 0 m/s, the reason for which is unclear. The plot then shows a sudden increase to nearly 30 m/s, representing the rocket's descent without a parachute. As illustrated in the total acceleration plot, an initial spike is observed at approximately 60 m/s^2; this is followed by a decline during the coast and descent phase, and ultimately, a final burst that is presumably due to parachute detachment. The vertical acceleration displays the same data as the total acceleration, with the exception of the

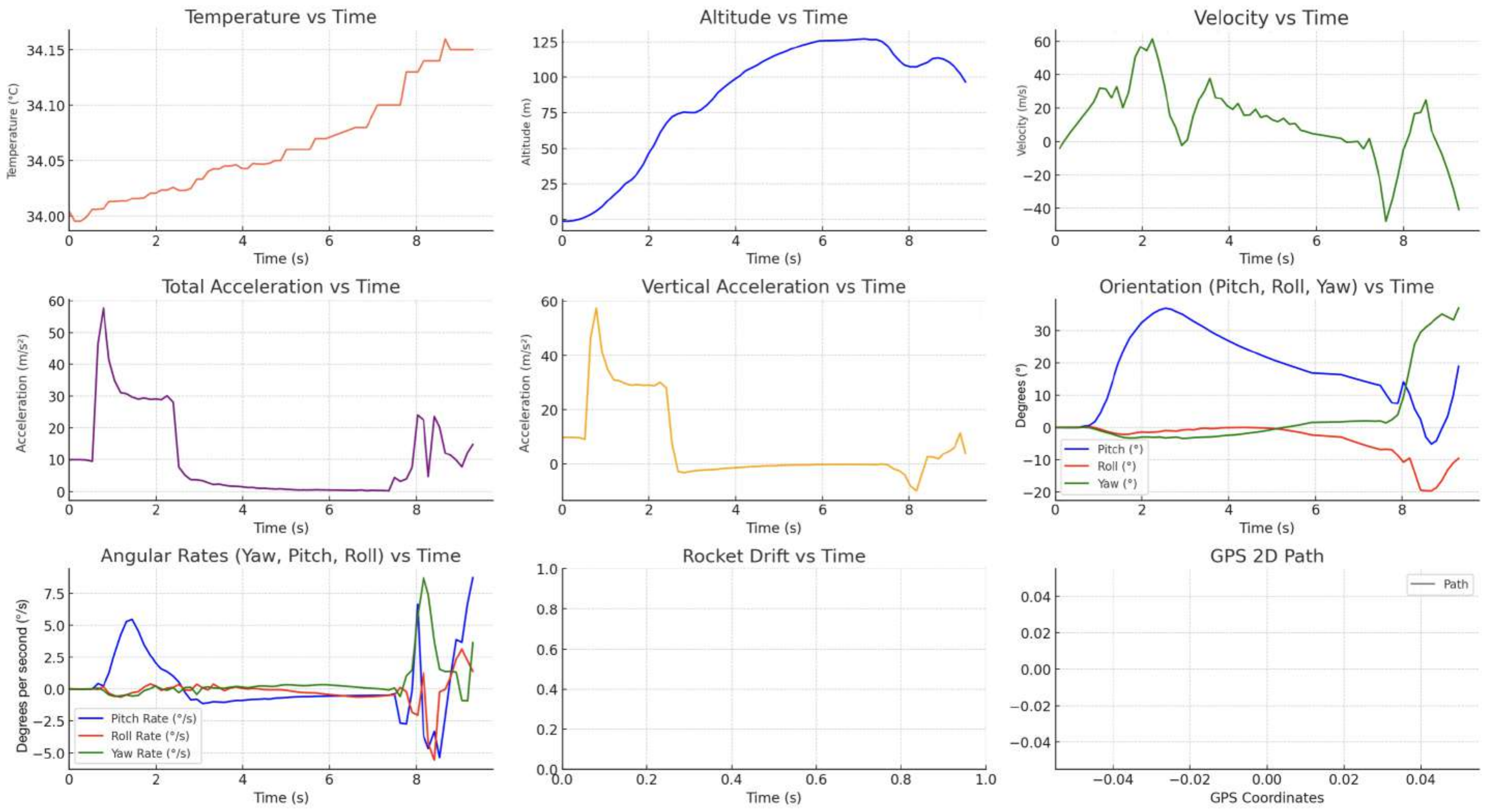

Figure 36: Telemetry data from the second flight

parachute ejection event where the change is less compared to the total acceleration. With regard to the orientation, the pitch exhibited significant fluctuations during the ascent, indicating that the rocket did not ascend in a completely straight trajectory. A deviation of 35 degrees was observed, suggesting an angular instability, which is likely caused by an imperfect weight distribution. Conversely, roll and yaw remained relatively stable until the parachute deployment. In conclusion, the angular rates indicate a sharp increase in pitch at the point of launch. This is followed by a significant increase in pitch, roll and yaw immediately prior to shutdown, suggesting dynamic motion following detachment. Before the comparison of the simulation and the actual data it needs to be added that there was a minute long video from the on board camera which then stopped moments before the launch. Continuing with the altitude comparison plot in Fig. 37 the rocket reached only 130 metres which is significantly lower that the simulated apogee. The ascent is initially similar but after 5.5 s, the actual altitude flattens and then drops suddenly at 9 s, explaining the parachute separation and free fall event.

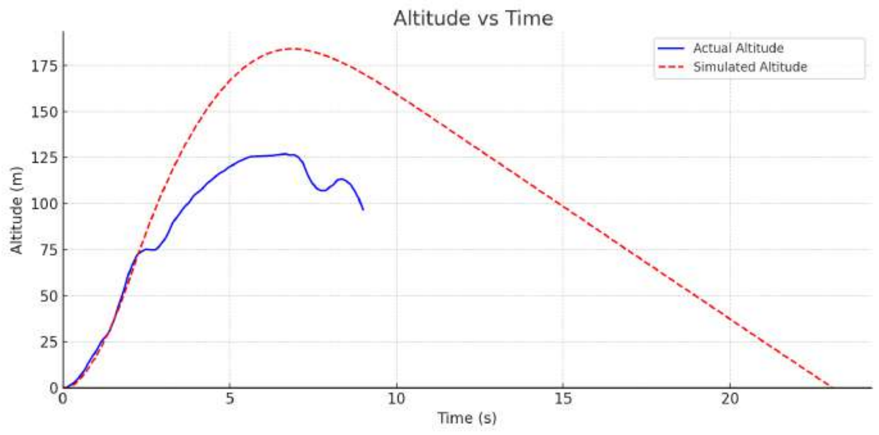

Figure 37: Flight 2 - Comparison of Actual and Simulated Altitude

As for the Velocity plot in Fig. 38 we can observe a similar launch profile, as they both peak at around 55-60 m/s shortly after liftoff. But then the actual data shows some fluctuations and a sudden drop at 2.5 s. After that we can see a descent of the velocity but then a sudden spike at 8.5 s, revealing the free-fall that was caused by the parachute detachment. Overall, the lines align well at ascent but at descent the are many deviations due to the mid-flight failure. For the comparison of the total acceleration plot in Fig. 39, we can see that both the actual and the simulated lines peak sharply at launch, reaching around 60 m/s^2 and nearly 90 m/s^2 respectively, as mentioned in the previous flight the actual data has the gravity subtracted meaning the peak should be around 65 m/s^2 and after the engine burn it should be at 0 like in the simulation. The rest of the difference is possibly due to a combine set or reasons like, sampling rate, overestimation of the thrust curve maybe the motor model assumes ideal pressure and burn efficiency and finally real world dynamics like vibrations and wind. The other spike at 7.5 s indicates the parachute deployment.

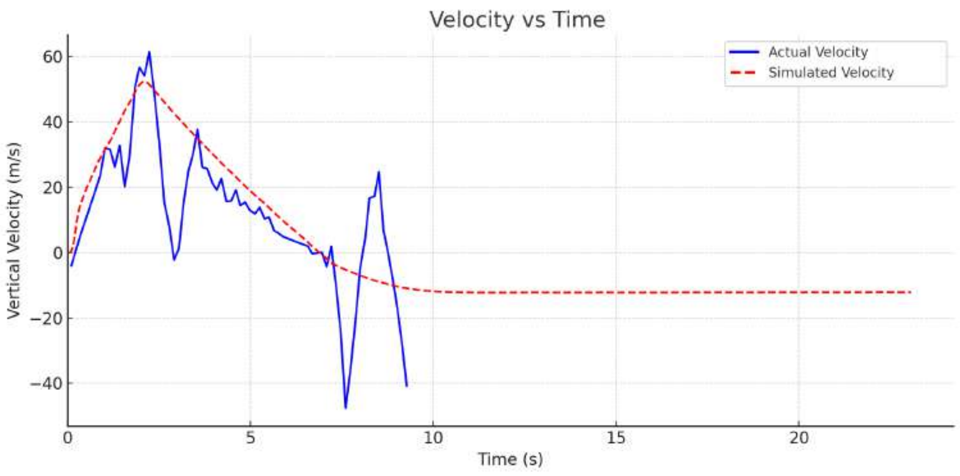

Figure 38: Flight 2 - Comparison of Actual and Simulated Velocity

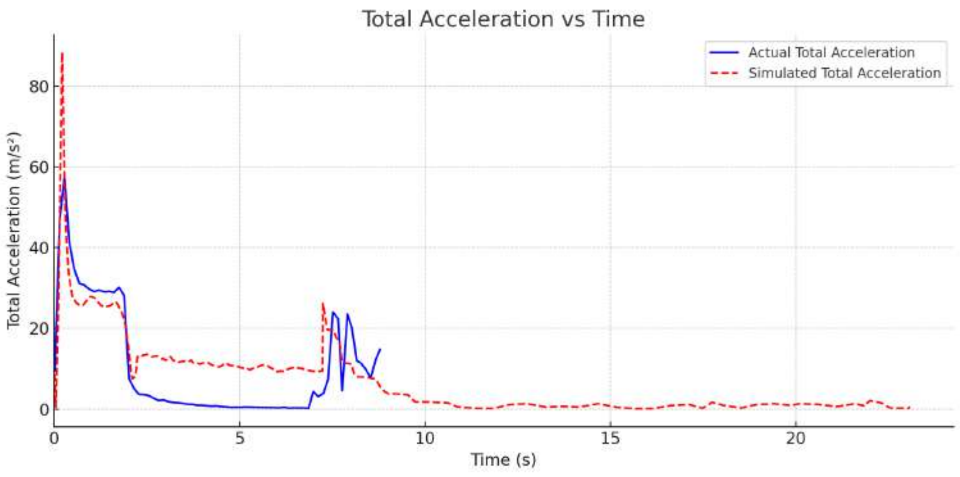

Figure 39: Flight 2 - Comparison of Actual and Simulated Total Acceleration

Finally the simulated vertical acceleration and the actual acceleration are looking very similar as depicted in Fig. 40. The reason that they are not peaking at the same number is for the same reason as mentioned in the total acceleration. After the motor burnout the vertical acceleration follows the same path as the simulated, the only difference is the subtracted gravity. The only major difference can be noted at the point of the parachute deployment and then the recovery system failure.

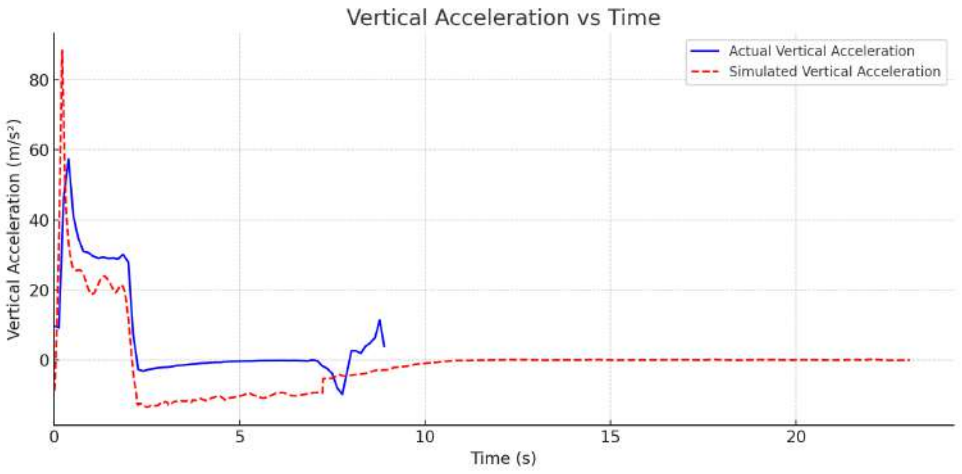

Figure 40: Flight 2 - Comparison of Actual and Simulated Vertical Acceleration

9.2.3. Flight 3: Data overview

Following the crash of the second flight, the production of two new body tubes and two sets of screwed adapters for the body tubes was required. The adapters were redesigned with the objective of reducing weight. Furthermore, as previously stated, the flight computer stopped recording due to the switch being turned off. Therefore, the decision was taken to remove the switch. Finally, the issue of the point of failure was addressed by redesigning the shock cord, which had been the cause of the previous disconnection from the nose cone. This leads us to the third flight that took place May 4, 2025. As illustrated in Fig. 41, the initial temperature reading prior to the rocket's launch was measured at 40 $°C$ due to sunny weather. At 7 s, an increase in temperature was observed, attributed to the ejection charge. This led to a peak temperature of 44 $°C$, suggesting that the protective system was operational. The altitude plot displays a peak close to 150 m which occurs before the ejection charge followed by a smooth descent, indicating a successful recovery. As illustrated in the velocity plot, the peak occurs at approximately 3 s at a velocity in excess of 60 m/s and subsequently at 2 s following the deployment of the parachute, at which point the velocity reduces to zero. As is evident in the total acceleration plot, an initial spike is observed at approximately 60 m/s^2 and is followed by a smaller increase to 30 m/s^2 due to the successful deployment of the parachute at 7 s. Finally, stable readings are observed at 10 m/s^2 up until landing. The vertical acceleration mirrors the total acceleration, with an initial thrust spike, followed by a decrease to 10 m/s^2 at 7 s due to the ejection charge. Concerning the orientation, it appears that the rocket undergoes minor alterations in any given direction up to close to 5 s where a 20-degree pitch spike is observed, meaning the rocket has started to tip off its vertical trajectory before the ejection charge.

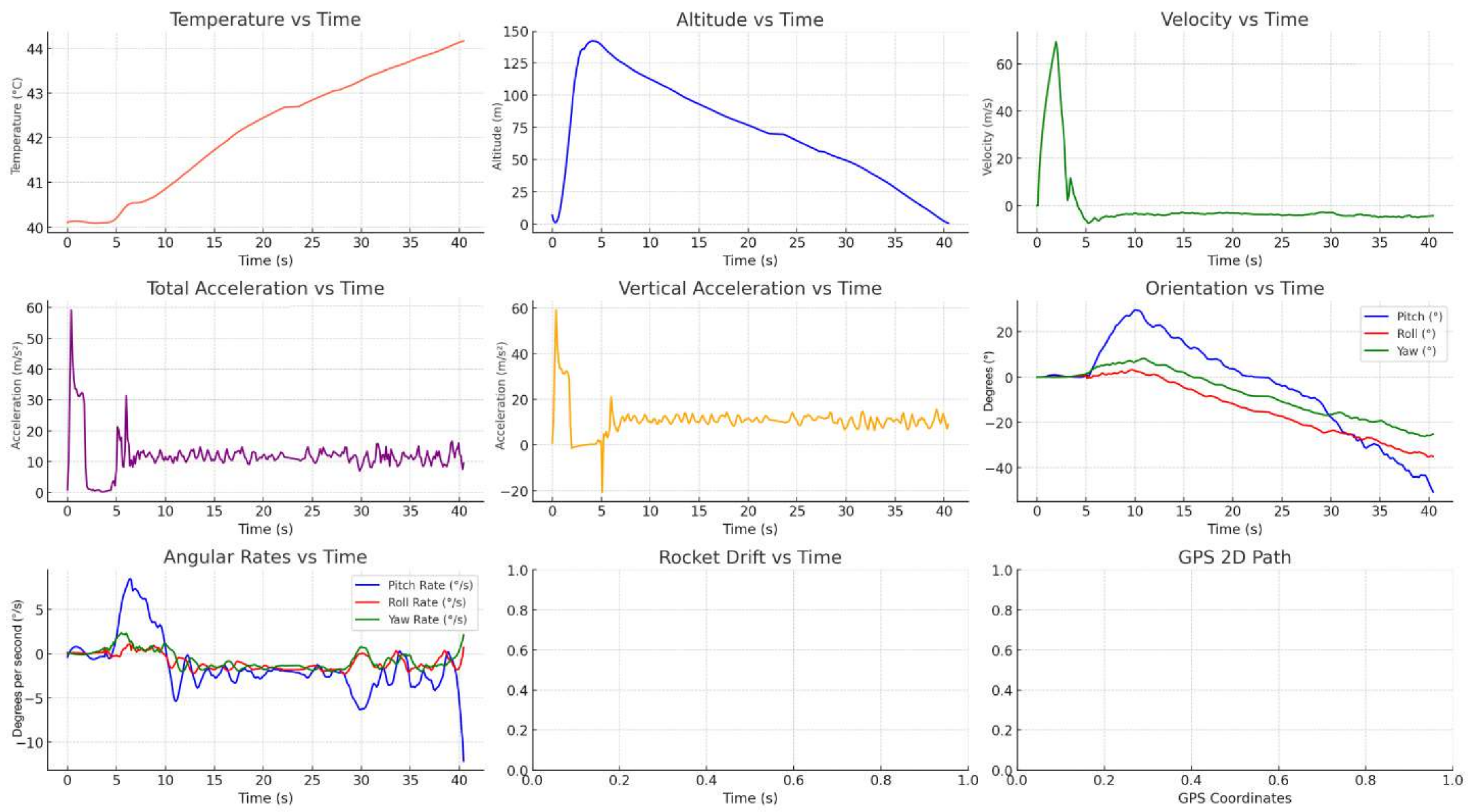

Figure 41: Telemetry data from the third flight

This is then followed by minor increases in roll and yaw, suggesting that the rocket experienced the ejection charge and a rotational movement until its landing. The angular rate plot captures minor changes, i.e. those less than $1/s$, which were mentioned previously, but the pitch rate starts to increase just before 5 s. During both the ascent and descent, a number of additional changes are observed, as previously explained. It is important to note that the rocket drift and GPS path plots remain blank, as the GPS is not connected at all for this flight due to malfunction of the sensor. In conclusion, the data obtained from the analysis of the plots indicates that there was a clean liftoff, coherent sensor behaviour, and dynamic mid-flight orientation changes. With regard to the camera that was on board, this time the video footage of the entire flight is available. On Fig. 42 in the middle of the image, the launch site can be seen.

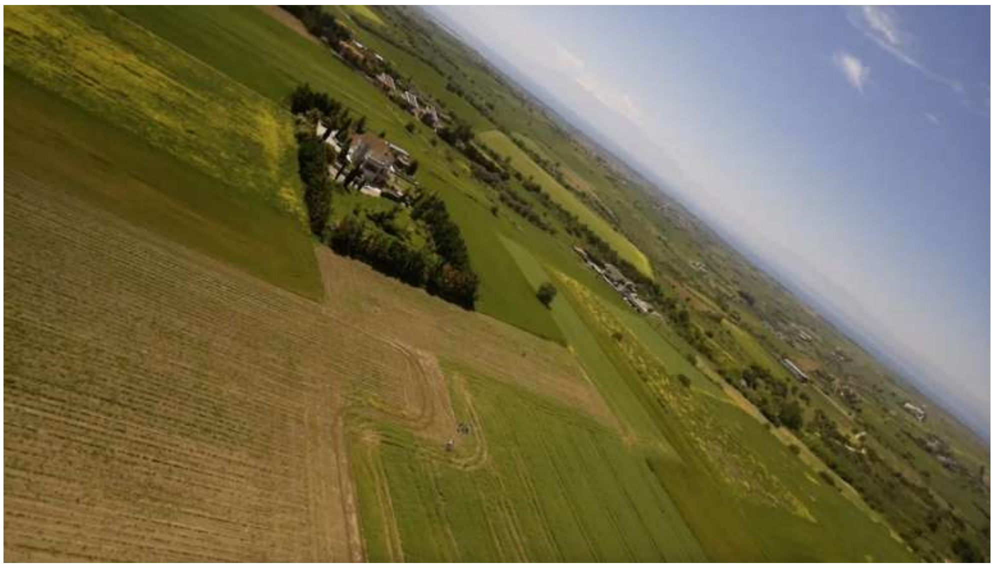

Figure 42: On-Board photo from Flight 3

As depicted in Fig. 43, the rocket achieved an altitude of approximately 150 metres, marking a new record. However, this remains below the simulated apogee, which is attained at the 7 s mark, just prior to parachute deployment. In real-world conditions, the apogee is achieved at around 5 s, with parachute deployment occurring during the descent phase. This could be due to several reasons, OpenRocket assumes ideal thrust and coast timing, but it could be possible that the engines burned inconsistently, the actual thrust was lower or burned faster. Another reason could be that the rocket has higher real drag than in the simulations leading to the rocket slowing down sooner. It is notable that at around 5 s in both the orientation and angular rates the pitch is changing leading to the rocket tipping off and not reaching its desired apogee. In the initial phase, the ascent is comparable; however, later on, the rocket requires a shorter duration to reach the ground, with a difference of approximately 10 s. The reason for this is that the rocket does not attain the simulated altitude, and consequently, it lands sooner.

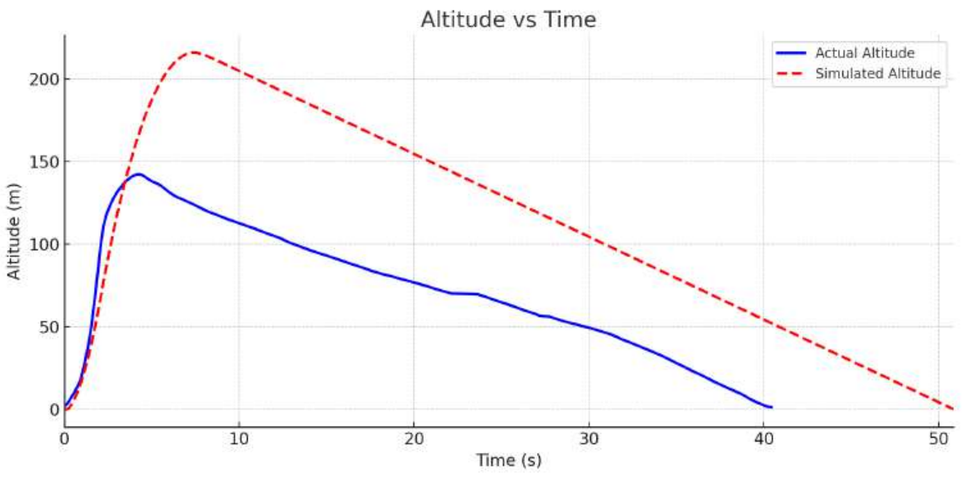

Figure 43: Flight 3 - Comparison of Actual and Simulated Altitude

Continuing with the comparison of the velocity plots in Fig. 44 we can observe a very similar spike but again the actual velocity peaks earlier and higher, closer to 70 m/s than the simulation 58 m/s, indicating that there is a initially a stronger thrust. Although the rocket has a bigger speed it also begins to decelerate much sooner that expected, suggesting either higher drag, instability or early motor burnout. As mentioned in the altitude plot this earlier deceleration contributed to a lower apogee. Overall the plot points out that real-world losses are not accounted in the simulation.

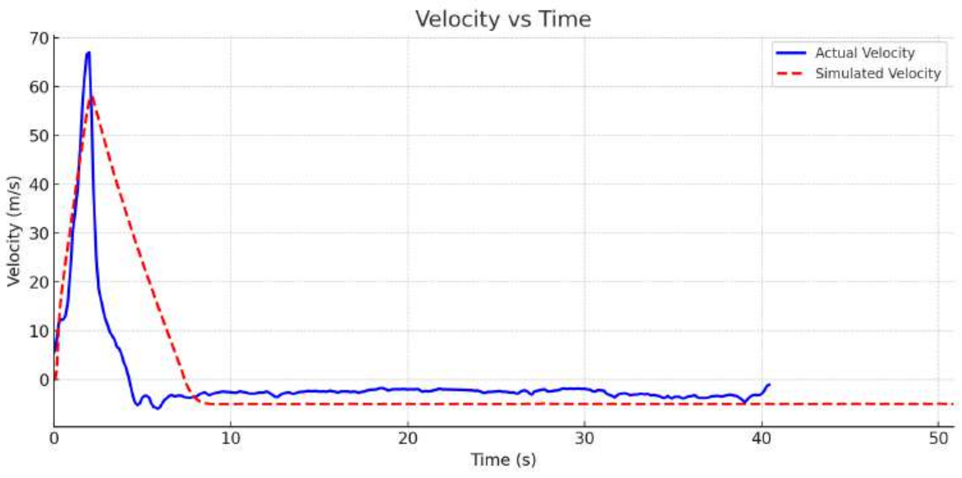

Figure 44: Flight 3 - Comparison of Actual and Simulated Velocity

As for the total acceleration overlay plot shown in Fig. 45, the actual acceleration peaks sharply just after the launch, reaching over 60 m/s^2 but as stated previously the exact peak could be missed due to sampling rate. The drop after the peak seems to be aligned with the simulation but the small increase after the motor burnout is delayed at 5 s. Afterwards the actual data shows post-burn disturbances and sustained oscillations around 10-15 m/s^2, this is likely happening from the vibrations and dynamic forces during coast and descent.

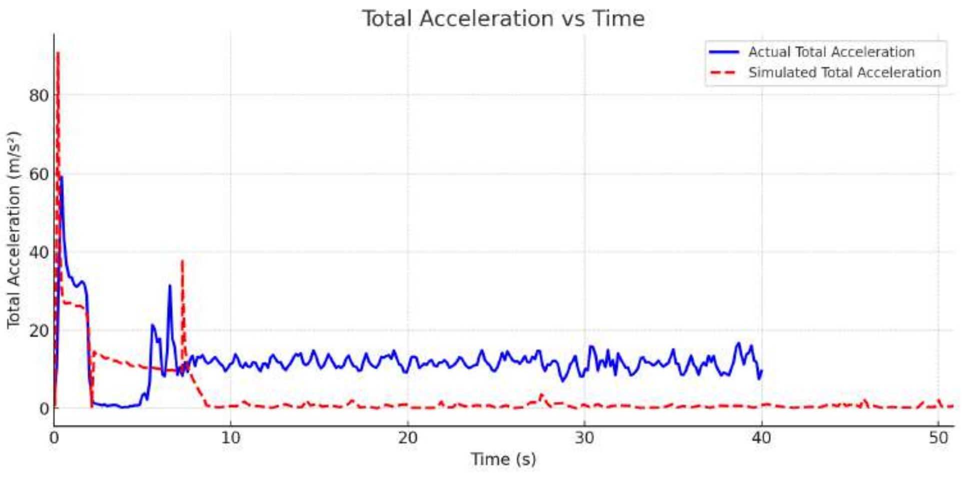

Figure 45: Flight 3 - Comparison of Actual and Simulated Total Acceleration

Finally, for the vertical acceleration depicted in Fig. 46, a similar pattern is seen. The actual acceleration peaks right after liftoff and then it is followed by turbulence and a more erratic behavior than the smoother simulated profile. Notably, at 5 s, the actual data drops rapidly, while on the other hand the simulated data levels off around zero. This is suggesting that the real flight experienced greater aerodynamic perturbations that could not have been captured in the simulation.

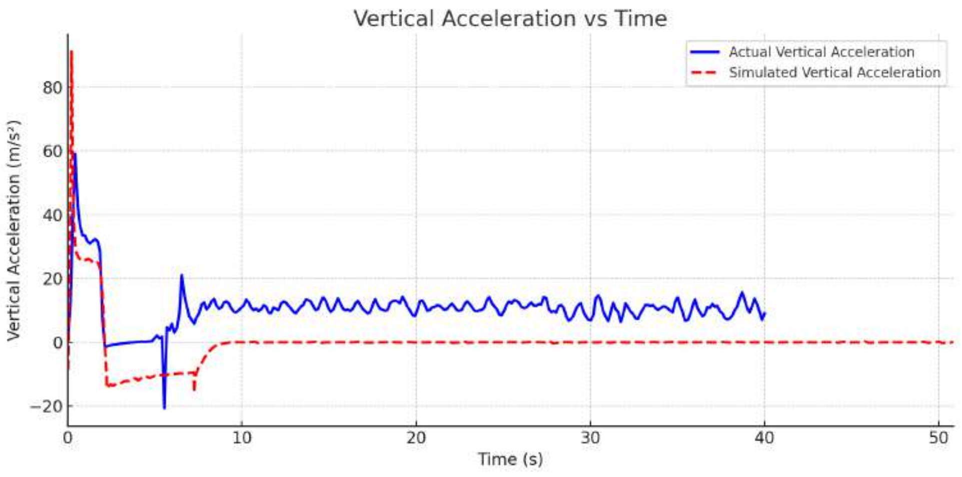

Figure 46: Flight 3 - Comparison of Actual and Simulated Vertical Acceleration

9.2.4. Flight 4: Data overview

The final flight saw the implementation of several modifications to ensure optimal outcomes before the conclusion of the entire experiment. Firstly, it should be noted that the code has been modified to display net acceleration, with the effect of gravity being deducted. Furthermore, the sampling rate has been increased to 100 Hz. This modification carries with it, the potential risk of compromised video quality; however, given the success of the previous flight in achieving full video footage, it was deemed more important to increase the sampling rate. Following the unusual indication in the previous launch, the fins were reattached with a view to achieving more precise alignment. The flight occurred one week after the third flight, and the complete telemetry can be observed in Fig. 48. The temperature plot shows the rocket starting at 35 $°C$, then increasing by 3 $°C$ following the ejection charge. Despite modifications being made to the rocket's fins, these proved ineffective in achieving the desired altitude of 135 m. However, due to the absence of significant wind, the rocket reached the ground in a shorter timeframe. As the velocity plot shows, the peak was reached at approximately 2 s and was close to 50 meters per second. Following the deployment of the parachute, this decreased to 0. The total acceleration plot demonstrates that, despite the increase in sampling to 100 Hz, the peak remained at 60 m/s^2. This was followed by a decrease to 0, but subsequently increased during the ejection charge. After this, the acceleration remained close to 0. It is important to note that the same observations can be made with regard to the vertical acceleration. During the rocket's ascent, it attained a negative pitch angle, which was subsequently stabilised until the deployment of the parachute. Roll and yaw remained relatively constant throughout the entire flight phase. As illustrated in the angular rate plot, the negative 7 degree pitch is apparent. Following parachute deployment, the angular rate decreased from 5 degrees to 0 degrees, leading to the descent of the parachute. Despite the increase in sampling rate to 100 Hz, the video feed was once again 100% successful, and it was even suggested that the correct fin reattachment helped to stabilise the footage as observed in Fig. 47.

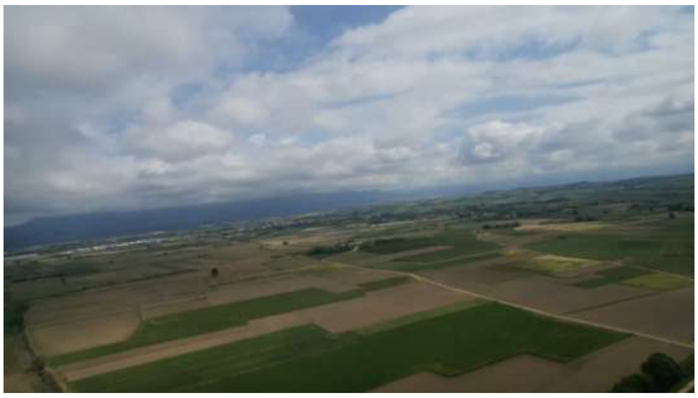

Figure 47: On-Board photo from Flight 4

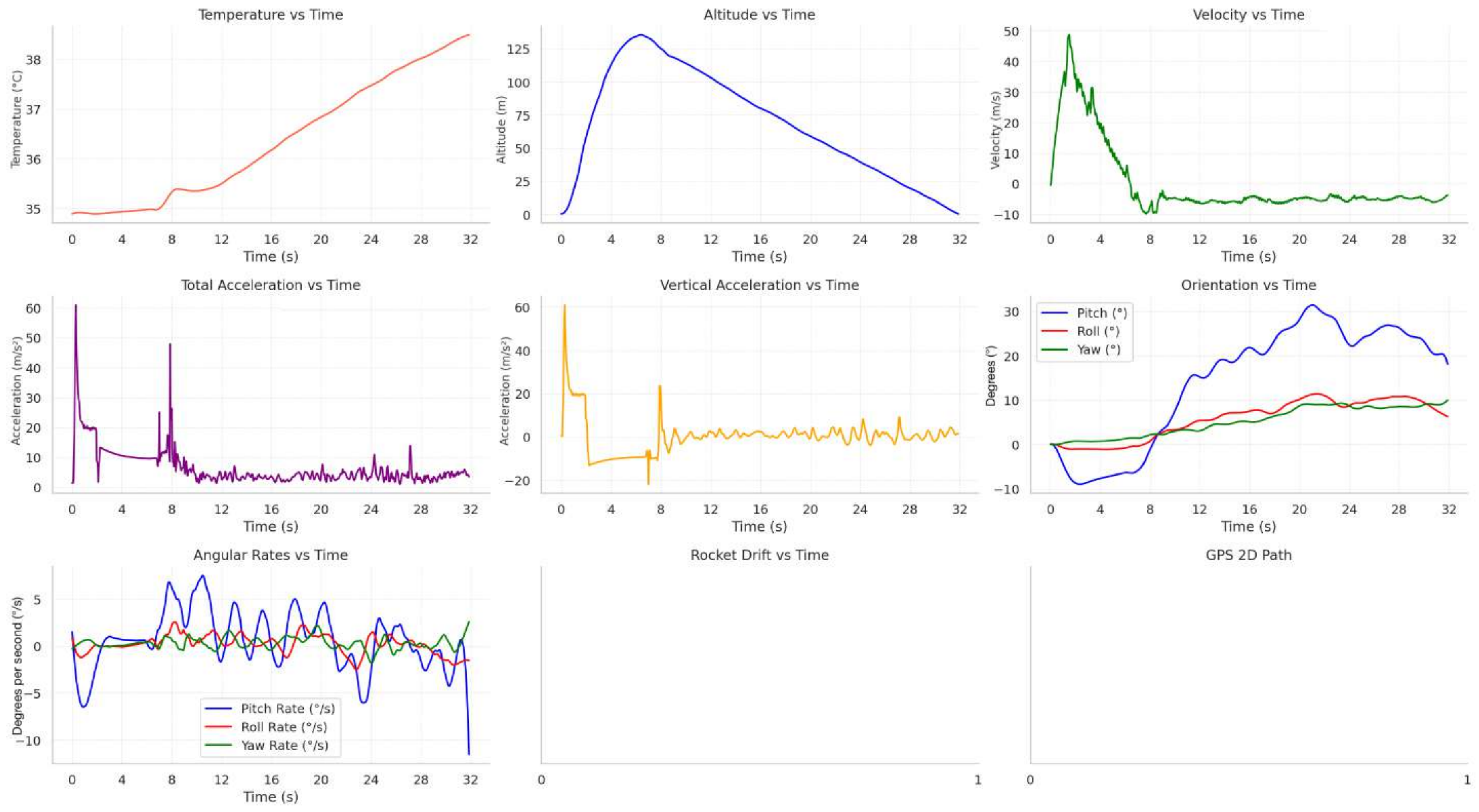

Figure 48: Telemetry data from the fourth flight

As Fig. 49 illustrates, the rocket achieved an altitude that was considerably lower than that of the simulation. However, it should be noted that both peak were achieved at the same time, suggesting that this is the maximum altitude that the rocket can achieve. Furthermore, in contrast to Flight 3, Fig. 43, the wind speed was lower on this occasion, as can be observed in Table 5. The flight duration was found to be shorter than that of the simulation, primarily due to the altitude attained.

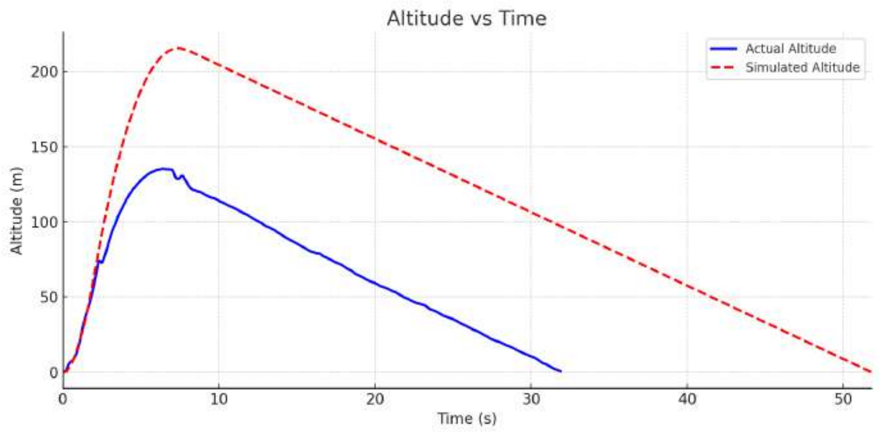

Figure 49: Flight 4 - Comparison of Actual and Simulated Altitude

This hypothesis is further supported by the overlaying velocity plot in Fig. 50, which indicates that the maximum velocity is attained a few seconds earlier and then decreases until the parachute deployment. Furthermore, during the ascent, the lines were found to be in exact alignment, and following parachute deployment, the velocity was recorded at -5 m/s, which is less than the -10 m/s simulated. This suggests that the actual parachute achieved higher drag performance than that modeled in OpenRocket.

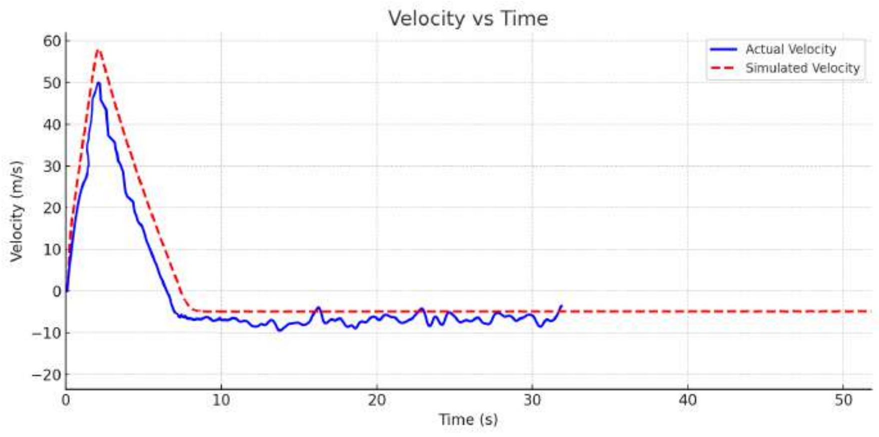

Figure 50: Flight 4 - Comparison of Actual and Simulated Velocity

With regard to the total acceleration overlay plot in Fig. 51, the actual acceleration peaks just after the launch at over 60 m/s^2. As previously stated, this is the maximum acceleration experienced by the rocket, given that the sampling rate is now at 100 Hz. It is evident that, with the exception of the increase during the ejection charge, the lines follow a similar trajectory to the simulated data. In reality, the increase was observed to reach almost 50 m/s^2, whereas in the simulations, it approximated 15 m/s^2. Following the deployment of the parachute, there is a decline in acceleration to zero, accompanied by fluctuations.

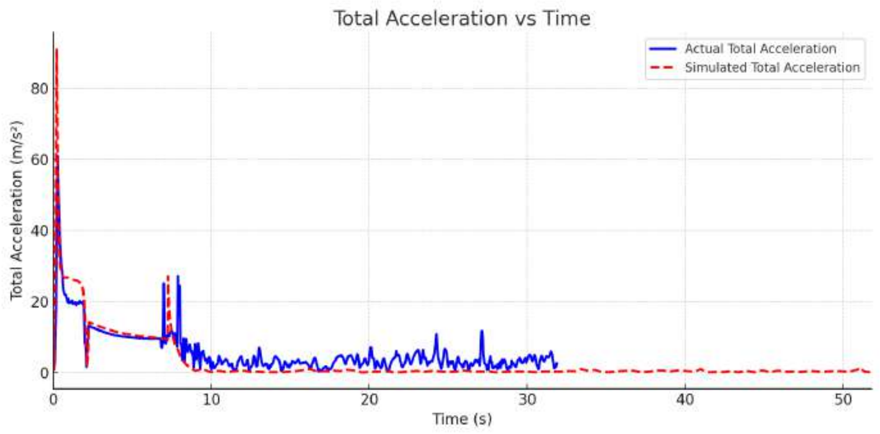

Figure 51: Flight 4 - Comparison of Actual and Simulated Total Acceleration

Additionally, the same observations can be made with respect to the vertical acceleration in Fig. 52. In this case, however, the variations are smaller than for the total acceleration and are the most accurate and closest to the simulations to date.

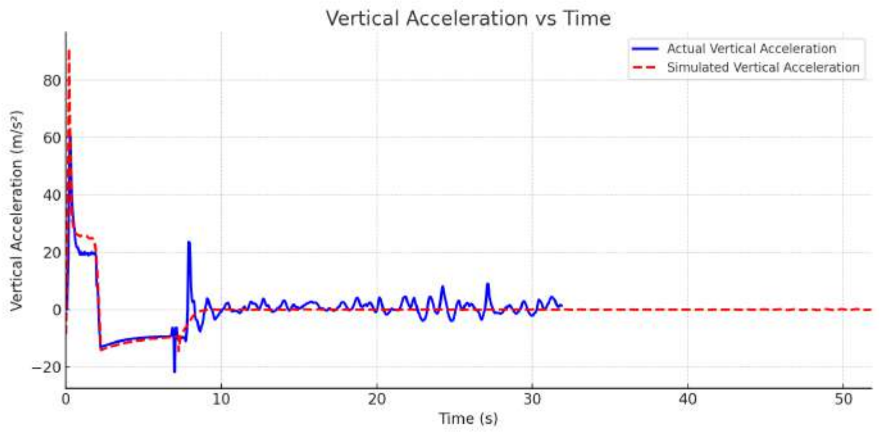

Figure 52: Flight 4 - Comparison of Actual and Simulated Vertical Acceleration

9.3. Performance Comparison of All Flights

This section presents a comparison of the key performance parameters across the four flights. Starting with the temperature plot in Fig. 53, it can be noted that during flight 1, there was an increase of almost 8 $°C$ when the ejection charge took place which also terminally damaged the Raspberry Pi. Subsequently, a better protection system was designed to ensure that the flight computer remained safe. This is confirmed to be working as on the next flights the increase was minimal.

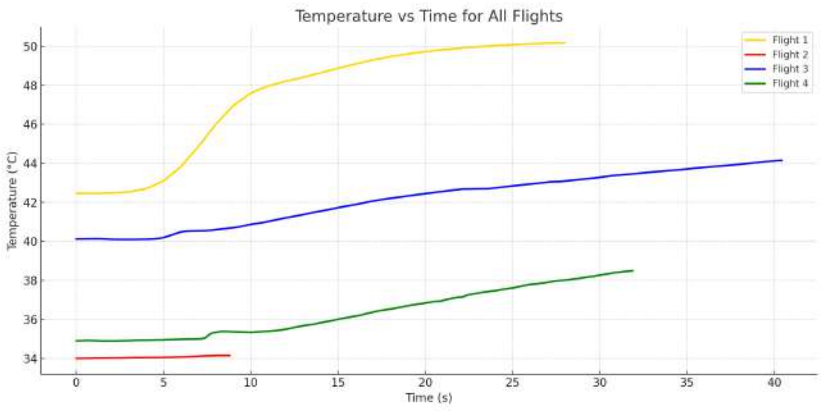

Figure 53: Temperature Comparison of All Flights

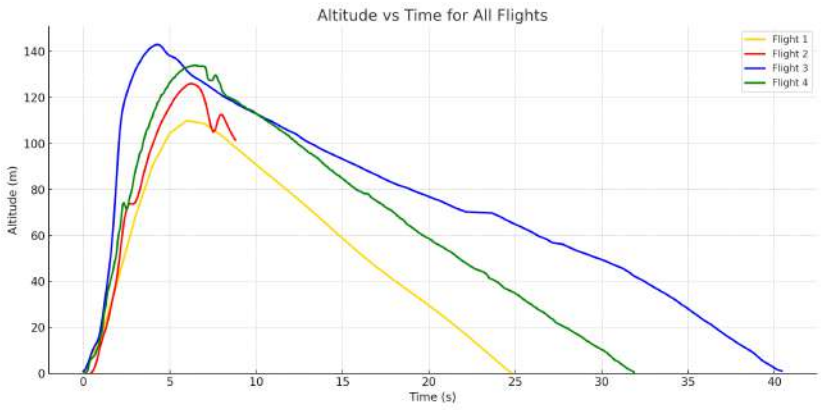

Figure 54: Altitude Comparison of All Flights

As indicated by the altitude plot comparison in Fig. 54, it is evident that the greatest apogee was achieved during the third flight which was 143 m. This occurred after the redesign of certain components, which resulted in a weight reduction. However, the other flights were not significantly distant, even following the failure during the second flight. It can be concluded from this outcome that the rocket attained its maximum potential. The simulated apogee for Flight 3 was 210 m, which is more than the 10-15 percent accuracy margin typically reported for OpenRocket predictions as mentioned in Section 6.3.4. This shows that real-world factors not fully captured by the simulation have a significant impact on the results. Moreover, the velocity plots in Fig. 55 show that the rocket attained a maximum velocity of approximately 50 m/s in the majority of its flights which is close to the previously mentioned 10-15 percent accuracy margin, with the exception of the inaugural flight, for which the sampling rate was found to be considerably low. Suggesting that this was the maximum velocity that the rocket could achieve.

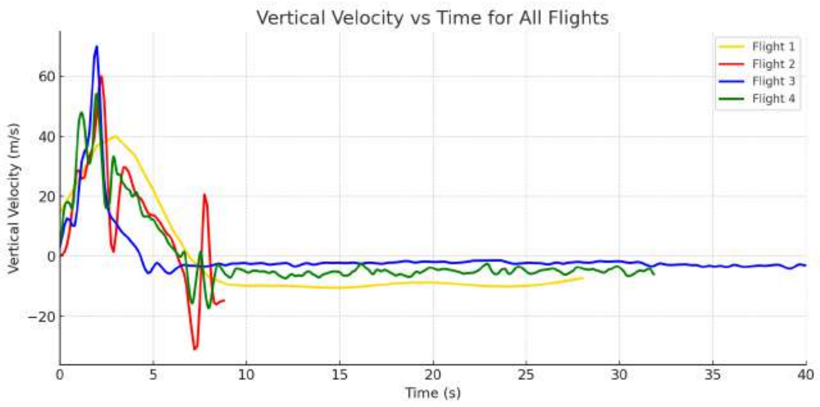

Figure 55: Vertical Velocity Comparison of All Flights

It is important to note that during the post-flight analysis, it was discovered that all prior simulations were conducted using the diameter of the parachute that was personally designed, rather than the one that was purchased and then used. This parameter significantly affected the prediction of the descent phase, particularly the duration that the rocket needed to reach the ground. Following the identification of this issue, the parachute diameter was updated for the simulations in Flight 3 and 4, where detailed mass, weight distribution and sensor placement were available. It is unfortunate that this was not possible for the simulations in Flight 1 and 2, as the major design changes between configurations, in addition to the lack of precise data (e.g. updated weight and sensor positions), prevented the recalculation of earlier simulations. Despite the aforementioned correction, a notable difference remained between the simulated and actual flight durations as shown in Fig. 56. Specifically, the difference between the simulated and actual times was approximately 10 s for Flight 3 and around 20 s for Flight 4, both are more than the 10-15 percent accuracy margin. This is likely due to the fact that the rocket in the actual flights did not reach the projected altitude that was shown in the simulation, resulting in a shorter actual descent path. Furthermore, environmental factors such as wind may have contributed to the differences observed between the two flights. However, in the simulations, wind effects were modelled to have minimal impact on total flight duration. This suggests that the observed differences are primarily due to altitude and real-world variability rather than modelled aerodynamic drag alone.

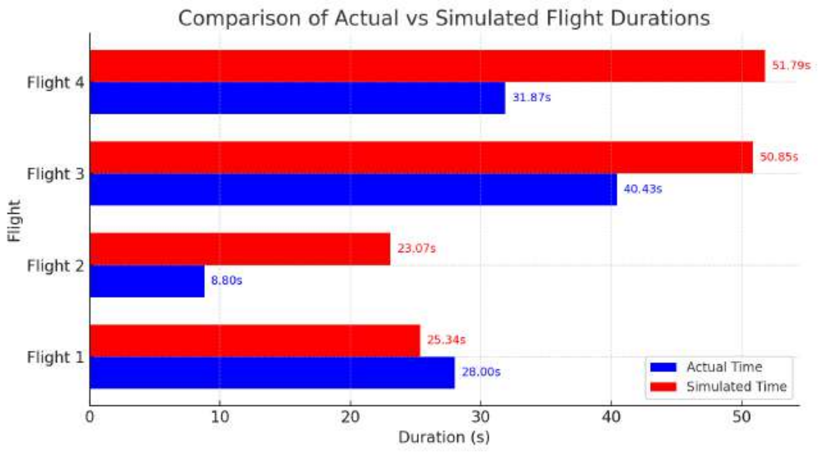

Figure 56: Actual vs Simulated Flight Durations

Finally, Fig. 57 presents the total acceleration plot, which, though challenging to analyse, demonstrates that following the increase in the sampling rate, the peak value is consistently 60 m/s^2. However, with regard to Flight 4, i.e. the green line, gravity is subtracted, meaning that the peak reached would have been at 70 m/s^2. It is also for this reason that, following the deployment of the parachute, it stabilises at zero. Additionally, during Flight 3 (illustrated by the blue line), the ejection charge occurred earlier than planned, suggesting a misfire of the engines. Finally, Fig. 57 presents the total acceleration plot, which, though challenging to analyse, shows that after improving the sampling rate, the measured peaks consistently reach around 60 m/s^2. The simulated peaks reach approximately 90 m/s^2, thus showing a deviation of about 50 percent, a value that is clearly larger than the typical 10-15 percent accuracy margin reported for OpenRocket.

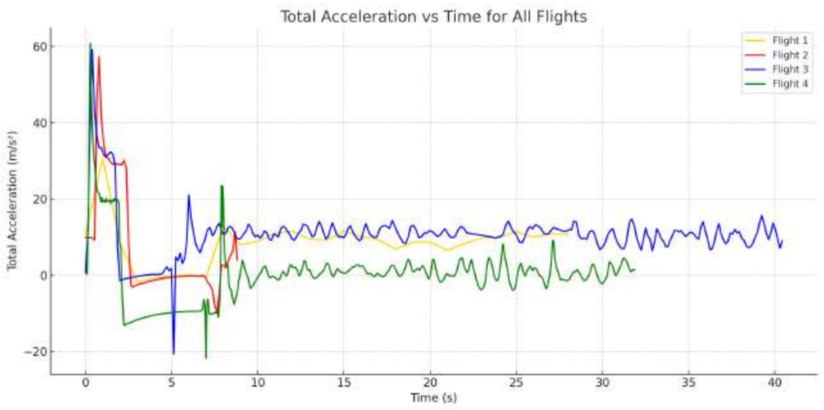

Figure 57: Total Acceleration Comparison of All Flights

9.4. Material and Structural Evaluation

Following the conclusion of the flights, it was determined that the selection of PETG material for the rocket was the optimal decision. Subsequent to the launch of the rocket, there wer no visible indications of warping or cracking on the rocket's exterior. Moreover, the rocket crashed from an altitude exceeding 100 metres during its second flight, yet only two of its tubes required replacement. Additionally, over half the rocket could be reused following the crash, which is a testament to the efficacy of the reusability design. In addition, although the parachute was suitable for reuse, it was ascertained that the shock cord required re-gluing prior to each flight. Something that did not happen before the second flight, which ultimately resulted in the crash. Subsequently, the structure was redesigned, incorporating the use of screws.

9.5. Assessment of Flight Computer

In the context of the evaluation of the flight computer during the first launch, the Raspberry Pi sustained terminal damage as a result of its protective system falling short of the required standard. Consequently, it was replaced with a more advanced protective system. The remaining sensors functioned through out the duration of the flights, with the exception of the GPS, which failed to operate from the initial testing phase.

9.6. Performance Analysis and Discussion

The theoretical framework that was described in Section 2 provided the essential basis for the understanding of the experimental results. To estimate the theoretical maximum altitude, the Tsiolkovsky rocket equation is applied Eq. (5) to see the achievable altitude by the rocket, assuming ideal conditions, but first the Specific impulse needs to be calculated from the the motor's total impulse and propellant mass using the Eq. (6). Below, there is an example for Flight 4:

$$\begin{aligned} I_{sp} &= \frac{I_t}{m_p \cdot g_0} \\ &= \frac{20.0\,Ns}{0.016\,kg \cdot 9.81\,m/s^2} \\ &= \frac{20.0}{0.15696} \\ &= 127.5\,s \end{aligned} \tag{36}$$

where:

- $I_t = 20.0\,Ns$ is the total impulse per motor,

- $m_p = 0.016\,kg$ is the propellant mass per motor,
- $g_0 = 9.81\,m/s^2$ is the standard gravitational acceleration.

Plugging the updated values into the rocket equation:

$$\begin{aligned}\Delta v &= v_e \cdot \ln\left(\frac{m_0}{m_f}\right)\\ &= 1250\,m/s \cdot \ln\left(\frac{0.736}{0.688}\right)\\ &= 84.4\,m/s \end{aligned} \tag{37}$$

where:

- $m_0 = 0.736\,kg$ is the initial mass (rocket + propellant),
- $m_{propellant} = 0.016\,kg$ per engine Section 5,
- $m_f = 0.688\,kg$ is the final mass (after burning 0.048 kg of propellant),
- $I_{sp} = 127.5\,s$ from Eq. (36),
- $v_e = I_{sp} \cdot g_0 = 127.5 \cdot 9.81 = 1250\,m/s$ is the effective exhaust velocity.

This is the ideal change in velocity the rocket achieves (ignoring drag and assuming perfectly vertical ascent). That means the maximum possible speed change the rocket can achieve. Interestingly, the maximum vertical velocity that was recorded during Flight 4 was approximately 50 m/s, which is much lower than the theoretical Δv of 84 m/s calculated using the rocket equation. This suggests that the rocket experienced a substantial reduction in its ideal velocity, likely due to the combined effects of aerodynamic drag, gravitational forces, other real-world factors or even a decreased motor performance. The maximum altitude can be estimated through the application of the energy conservation :

$$\begin{aligned}h &= \frac{(\Delta v)^2}{2g}\\ &= \frac{84.4^2}{2 \cdot 9.81}\\ &= \frac{7123}{19.62}\\ &= 363\ m\end{aligned} \tag{38}$$

Theoretically, based on the Tsiolkovsky rocket equation and the energy conservation the maximum altitude that can be achieved is 363 m. However, this is the upper limit under ideal conditions (i.e. when drag is ignored and perfectly vertical ascent is assumed). OpenRocket simulated an altitude of 212 m while the actual measured altitude of 143 m reflects more realistic scenarios. The model takes into account the forces of drag, the loss of mass due to gravity during the combustion process, and the impact of the environment, all of which have a significant effect on the achievable altitude. Furthermore, the actual flight performance is affected by additional real-world factors such as launch inefficiencies, wind disturbances, and motor performance variability. This explains why the theoretical predictions are consistently higher than the simulated and measured results. It is also possible to estimate the approximate velocity gain during the burn phase by applying Newton's Second Law Eq. (1), by using the average thrust and burn time:

$$\begin{aligned} a &= \frac{F_{\text{total}}}{m} \\ &= \frac{3 \cdot 9.0\, N}{0.736\, kg} \\ &= \frac{27.0}{0.736} \\ &= 36.7\, m/s^2 \end{aligned} \tag{39}$$

$$\begin{aligned} \Delta v &= a \cdot t \\ &= 36.7\, m/s^2 \cdot 2.1\, s \\ &= 77.1\, m/s \end{aligned} \tag{40}$$

The velocity calculated using the acceleration estimated at 77 m/s is slightly lower than the calculated using the rocket equation, which was 84 m/s. This difference is due to the fact that the rocket equation takes the rocket's changing mass into account as fuel is burned, whereas the acceleration estimate assumes the mass stays constant. However, the measured flight data showed that differences between the theoretical and simulated predictions were caused by effects such as imperfect thrust, air drag and wind. One crucial issue was the overestimation of descent speed, which was primarily due to assumptions about the drag coefficient (C_d) and parachute performance. This emphasises the necessity of calibrating simulation models with real flight data. Taking this approach would improve accuracy and ensure the practical utility of theoretical models.

10. Conclusion and Future Work

10.1. Summary of Findings

This project successfully designed, fabricated, reused and tested a 3D printed model rocket equipped with an integrated telemetry system. In total, four launches were conducted, the aim was to collect flight data such as temperature, altitude, velocity, acceleration magnitude, orientation, rotation rate and GPS data. The ground station received the telemetry in real time, but the data was also saved on the Raspberry Pi for post-flight analysis. The code for the flight computer was developed with the use of AI tools, and the script can be found in my GitHub project, along with all CAD files, telemetry scripts, camera scripts, analysis results, and videos, at this link `https://github.com/Filpass/Hermes-Rocket-Project`. A comparison with OpenRocket revealed that the simulated apogee was never reached in the actual flights. This was primarily due to differences between the simulated and actual acceleration and velocity, as well as the fact that OpenRocket's simplified assumptions do not fully account for complex real-world aerodynamic factors.

The measured velocity was found to be close the 10-15 percent accuracy margin reported for OpenRocket as mentioned in Section 6.3.4, but the differences in the altitude, acceleration and flight duration were significantly larger. Since velocity, altitude, acceleration and flight duration are interdependent, a variation in one of these parameters will result in changes to the others. This suggests that either the motors underperformed compared to their specified performance, the external aerodynamic effects are bigger than expected or the rocket's design did not achieve the anticipated level of efficiency. Importantly, after each flight, the quality and detail of the collected data improved, leading to better simulations and improved rocket performance. A similar level of deviation was reported in [18], where the same Klima D9-5 engines were used. This suggests that the differences observed may be a result of motor performance characteristics or external aerodynamic effects. By applying theoretical equations presented in Section 2, including the rocket equation, energy conservation and drag force estimation, this project provided a solid theoretical basis for analysing both simulation outputs and experimental data. A number of calculations were performed to provide ideal expectations (such as Delta V, altitude, and terminal velocity) that could be directly compared with OpenRocket predictions and measured flight results. Crucially, this connects the theoretical foundations in Section 2, with the practical experimentation of Section 9, showing how theoretical models can inform design, guide simulations and be improved through testing. This process validates the analytical methods while providing a basis for improving simulations and the real-world performance of rockets in the future.

10.2. Limitations and Challenges Faced

Many beginner challenges related to rocket launching had already been encountered during a previous research project conducted for the bachelor's thesis. This prior experience proved valuable in overcoming similar issues at the outset of the current work [18]. The main issue of the complete project was the limited space inside of the rocket, which had to be that way because if the rocket was bigger then it would need additional engines, which means that the price of the project would be even higher and as it was a self-funding project the price needed to remain low. The limited space inside the rocket meant that it was challenging to fit the flight computer in there, while also screwing the camera on the outside of the rocket, keeping the wires connected to the Raspberry Pi and also protecting the flight computer from the ejection charge of the engines. Additionally, limited prior experience with coding presented an added challenge during the project. After the first flight the Raspberry Pi was fried so there was the need to buy another one and design a better protection system. On the second flight the protection system was working as expected but the parachute disconnected from the rest of the rocket leading to a crash. Therefore, for the next flight, two body tubes were needed to be re printed and two sets of newly designed screwed adapters were also printed. As for the third flight, a small improvement in the fins and the software was made.

10.3. Recommendations for Future Work

The utilisation of a 3D printer proved highly beneficial for the project, enabling rapid design and testing of various concepts throughout the development process. Future research in the area of the flight computer would be to improve the system by using dedicated GPS antennas for stronger satellite lock and a telemetry system that does not rely on Wi-Fi connections, so it is not limited by range. With reference to the rocket, once again the utilisation of a 3D printer could be used to design and print landing legs for a smoother parachute landing. Additionally controllable surfaces on the fins like in Fig. 58 would help with a more stable flight. Finally, these findings indicate clear possibilities for the improvement of future models. The potential benefits of improving the correlation between simulations and actual rocket behaviour are significant. These improvements could be achieved by applying a range of methodologies, including the refinement of drag coefficients based on measured results, the fine-tuning of parachute settings, and the inclusion of more detailed wind and weather inputs. While the current model provides reliable initial predictions, further improvements should focus on advancing aerodynamic modelling. Specifically, the incorporation of methodologies that target limitations such as turbulent wake effects, Reynolds number sensitivity, and other complex fluid dynamics could result in a significant improvement in the precision of the simulation. The model could be improved by including CFD or refined empirical corrections. This would allow the model to better capture the finer details of subsonic and transonic flow regimes, thereby providing more accurate predictions for both the ascent and descent phases.

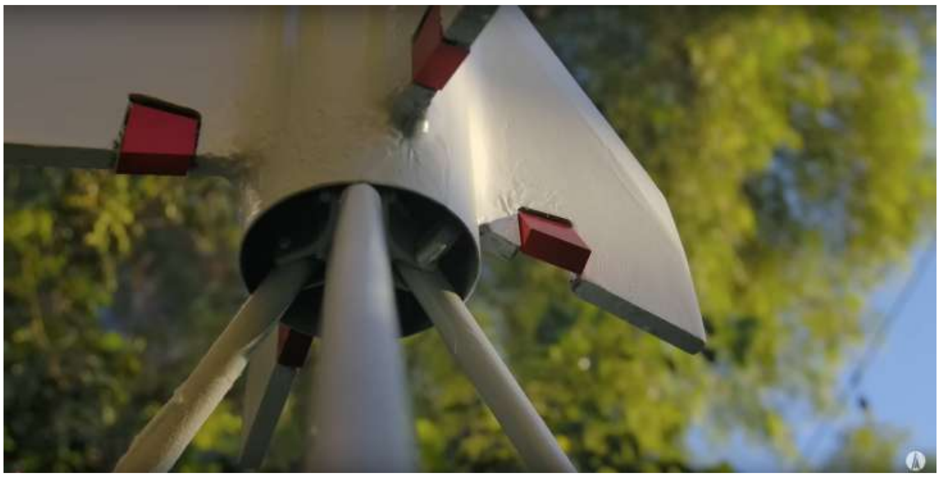
Figure 58: Controllable Surfaces on Fins by BPS.space [19]

10.4. Potential Applications of the Hermes Rocket

The Hermes Rocket shows how low-cost, lightweight model rockets can be used as testbeds for sensor validation, communication protocols and flight control experiments. From an educational perspective, it provides a practical learning environment for students interested in rocketry, programming, and data analysis. Furthermore, its flexibility allows the rocket to be adapted for early-stage avionics prototyping or used in activities to promote interest in aerospace engineering. Its open-source nature also facilitates extensibility for research in areas such as trajectory optimisation, fault tolerance and autonomous recovery systems.

References

[1] G. H. Stine and B. Stine, Handbook of Model Rocketry. John Wiley Sons, Apr. 2004.

[2] Y. O. R. Forum, National model and sport rocketry collection blog, https://forums.rocketshoppe.com/printthread.php?t=16676&page=5&pp=40, Rocketshoppe.com, 2017. (visited on 11/18/2024).

[3] N. Hall, Four forces on a rocket - glenn research center - nasa, https://www1.grc.nasa.gov/beginners-guide-to-aeronautics/four-rocket-forces/, Glenn Research Center | NASA, Nov. 2023.

[4] G. P. Sutton and O. Biblarz, Rocket Propulsion Elements, 9th. John Wiley & Sons, 2017, ISBN: 978-1118753651.

[5] S. Niskanen, Development of an open source model rocket simulation software, https://openrocket.sourceforge.net/thesis.pdf, May 2009.

[6] R. Barrowman and D. Boyce, "Air distribution from lateral ducts in barley," Journal of Agricultural Engineering Research, vol. 11, no. 4, pp. 243–247, 1966, ISSN: 0021-8634. DOI: https://doi.org/10.1016/S0021-8634(66)80031-8. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0021863466800318.

[7] D. Halliday, R. Resnick, and J. Walker, Fundamentals of Physics, 10th ed. John Wiley & Sons, 2013, ISBN: 9781118230718.

[8] Z. Wawryniuk, E. Brancewicz-Steinmetz, and J. Sawicki, "Revolutionizing transportation: An overview of 3d printing in aviation, automotive, and space industries," Sep. 2024. DOI: 10.1007/s00170-024-14226-y.

[9] M. Sheetz, Inside relativity space's 3d-printing rocket 'factory of the future', https://www.cnbc.com/2020/10/07/inside-relativity-space-hq-3d-printer-rocket-factory-of-the-future.html, CNBC, Oct. 2020.

[10] MarkForged, Pla vs abs vs nylon, https://markforged.com/resources/blog/pla-abs-nylon, Markforged, 2024.

[11] G. Slump, Pla vs abs vs petg: The differences, https://all3dp.com/2/pla-vs-abs-vs-petg-differences-compared/, All3DP, May 2021.

[12] A. Billington, Optimizing strength of 3d printed parts, https://3dpros.com/guides/optimizing-part-strength, 3dpros.com, n.d.

[13] Prusa 3D Printers, Prusa 3d printers, Accessed May 2025, n.d.

[14] Adafruit, Adafruit industries, unique fun diy electronics and kits, https://www.adafruit.com, www.adafruit.com, n.d.

[15] ThrustCurve.org, Klima d9 motor data, Accessed May 2025, n.d.

[16] OpenRocket, Openrocket, https://openrocket.info, openrocket.info, n.d.

[17] I. Apogee Components, Apogee rockets, model rocketry excitement starts here, `https://www.apogeerockets.com/Peak-of-Flight/Newsletter623`, Apogeerockets.com, 2023.

[18] P. Moschidis and P. S. Bithas, in Arduino Rocket Flight Computer, 2022, pp. 1–6. DOI: `10.1109/PACET56979.2022.9976324`.

[19] BPS.space, Building a guided rocket to hit mach 3, YouTube, Available at: `https://www.youtube.com/watch?v=tcqzLW3iI-s`, Accessed: May 2025, 6.11.2024.

A. Project Repository

The complete set of CAD files, telemetry code, camera code, analysis results and videos are available at: `https://github.com/Filpass/Hermes-Rocket-Project`

B. Component List and Costs

SL	Component Name	Quantity	Total (€)
1	Raspberry Pi Zero 2W	2	33.48
2	Raspberry Pi Hat	1	9.61
3	Adafruit DPS310	1	6.46
4	Adafruit ICM-20948	1	13.90
5	Adafruit Mini GPS PA1010D	1	27.85
6	Raspberry Pi Camera V3 Wide lens	1	42.32
7	3A USB step-down voltage regulator module DC-DC Converter	1	0.88
8	LP903450 3.7V 1600mAh Lithium Polymer Battery	1	14
9	Charger for Battery	1	3.70
10	STENNA QT/Qwiic cable	3	2.65
11	Raspberry Pi Zero Camera Cable	2	4.86
12	Spectrum PETG Fillament	1	21.90
13	Sandpaper	3	1.80
14	JB-WELD Plasticbonder	2	17
15	Klima D9-5 Engines (pck 6)	2	66.50
16	Klima Parachute	1	30.40
=	TOTAL	24	297.31

C. Launch Day Checklist

1. Flight Computer Setup

1.1 IP Configuration

- ☐ Find Raspberry Pi IP: ping raspberrypi.local
- ☐ Find Mac IP via system settings
- ☐ Update script IP: nano senddata2.py
- ☐ Default SSH: ssh pi@172.20.10.4

2. Sensor and Camera Execution

2.1 GPS Test (Optional)

☐ Connect to Pi: ssh pi@172.20.10.4

☐ Activate environment:source myenv/bin/activate

☐ Run GPS test: python3 gpstest.py

☐ Stop test with: Ctrl + C

2.2 Onboard Camera (Video Recording)

☐ Connect to Pi: ssh pi@172.20.10.4

☐ Run camera script: python3 camera2.py

2.3 Data Transmission (Telemetry)

☐ Connect to Pi: ssh pi@172.20.10.4

☐ Activate environment: source myenv/bin/activate

☐ Run telemetry: python3 senddata4.py

2.4 Ground Station (Live Telemetry Visualization)

☐ Open terminal on Mac

☐ Run: python3 receivedata.py

3. Weather Data Collection

☐ Wind speed and direction: `windy.com`

☐ Air temperature: from barometric sensor

☐ Atmospheric pressure: from barometric sensor

☐ Humidity: from local weather report

☐ Launch site elevation: from Google Earth

☐ Wind shear: `windy.com`

D. Declaration of Consent

Universität
Bremen

Nachname	Moschidis	Matrikelnr.	6254474
Vorname	Philippos Georgios		

Hinweise zu den offiziellen Erklärungen

1. Die folgende Seite mit den offiziellen Erklärungen

   A) Eigenständigkeitserklärung

   B) Erklärung zur Veröffentlichung von Bachelor- oder Masterarbeiten

   C) Einverständniserklärung über die Bereitstellung und Nutzung der Bachelorarbeit / Masterarbeit in elektronischer Form zur Überprüfung durch eine Plagiatssoftware

   ist entweder direkt in jedes Exemplar der Bachelor- oder Masterarbeit fest mit einzubinden oder unverändert im Wortlaut in jedes Exemplar der Bachelor- oder Masterarbeit zu übernehmen.

   Bitte achten Sie darauf, jede Erklärung in allen drei Exemplaren der Arbeit zu unterschreiben.

2. In der digitalen Fassung kann auf die Unterschrift verzichtet werden. Die Angaben und Entscheidungen müssen jedoch enthalten sein.

Zu B)

Die Einwilligung kann jederzeit durch Erklärung gegenüber der Universität Bremen, mit Wirkung für die Zukunft, widerrufen werden.

Zu C)

Das Einverständnis der dauerhaften Speicherung des Textes ist freiwillig.

Die Einwilligung kann jederzeit durch Erklärung gegenüber der Universität Bremen, mit Wirkung für die Zukunft, widerrufen werden.

Weitere Informationen zur Überprüfung von schriftlichen Arbeiten durch die Plagiatsoftware sind im Nutzungs- und Datenschutzkonzept enthalten. Diese finden Sie auf der Internetseite der Universität Bremen.

Universität
Bremen

Nachname	**Moschidis**	Matrikelnr.	**6254474**
Vorname	**Philippos Georgios**		

A) Eigenständigkeitserklärung

Ich versichere, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe. Alle Teile meiner Arbeit, die wortwörtlich oder dem Sinn nach anderen Werken entnommen sind, wurden unter Angabe der Quelle kenntlich gemacht. Gleiches gilt auch für Zeichnungen, Skizzen, bildliche Darstellungen sowie für Quellen aus dem Internet, dazu zählen auch KI-basierte Anwendungen oder Werkzeuge. Die Arbeit wurde in gleicher oder ähnlicher Form noch nicht als Prüfungsleistung eingereicht. Die elektronische Fassung der Arbeit stimmt mit der gedruckten Version überein. Mir ist bewusst, dass wahrheitswidrige Angaben als Täuschung behandelt werden.

☒ Ich habe KI-basierte Anwendungen und/oder Werkzeuge genutzt und diese im Anhang "Nutzung KI basierte Anwendungen" dokumentiert.

B) Erklärung zur Veröffentlichung von Bachelor- oder Masterarbeiten

Die Abschlussarbeit wird zwei Jahre nach Studienabschluss dem Archiv der Universität Bremen zur dauerhaften Archivierung angeboten. Archiviert werden:

1) Masterarbeiten mit lokalem oder regionalem Bezug sowie pro Studienfach und Studienjahr 10 % aller Abschlussarbeiten
2) Bachelorarbeiten des jeweils ersten und letzten Bachelorabschlusses pro Studienfach und Jahr.

☒ Ich bin damit einverstanden, dass meine Abschlussarbeit im Universitätsarchiv für wissenschaftliche Zwecke von Dritten eingesehen werden darf.

☐ Ich bin damit einverstanden, dass meine Abschlussarbeit nach 30 Jahren (gem. §7 Abs. 2 BremArchivG) im Universitätsarchiv für wissenschaftliche Zwecke von Dritten eingesehen werden darf.

☐ Ich bin nicht damit einverstanden, dass meine Abschlussarbeit im Universitätsarchiv für wissenschaftliche Zwecke von Dritten eingesehen werden darf.

C) Einverständniserklärung zur Überprüfung der elektronischen Fassung der Bachelorarbeit / Masterarbeit durch Plagiatssoftware

Eingereichte Arbeiten können nach § 18 des Allgemeinen Teil der Bachelor- bzw. der Masterprüfungsordnungen der Universität Bremen mit qualifizierter Software auf Plagiatsvorwürfe untersucht werden.
Zum Zweck der Überprüfung auf Plagiate erfolgt das Hochladen auf den Server der von der Universität Bremen aktuell genutzten Plagiatssoftware.

✓ Ich bin damit einverstanden, dass die von mir vorgelegte und verfasste Arbeit zum oben genannten Zweck dauerhaft auf dem externen Server der aktuell von der Universität Bremen genutzten Plagiatssoftware, in einer institutionseigenen Bibliothek (Zugriff nur durch die Universität Bremen), gespeichert wird.

☐ Ich bin nicht damit einverstanden, dass die von mir vorgelegte und verfasste Arbeit zum o.g. Zweck dauerhaft auf dem externen Server der aktuell von der Universität Bremen genutzten Plagiatssoftware, in einer institutionseigenen Bibliothek (Zugriff nur durch die Universität Bremen), gespeichert wird.

Mit meiner Unterschrift versichere ich, dass ich die obenstehenden Erklärungen gelesen und verstanden habe und bestätige die Richtigkeit der gemachten Angaben.

30.05.2025 [illegible]