

UNIVERSITY OF  
**Southampton**

# Lunar Hopper Mk. II Enhancements Summary Report

Group Design Project 59

## Supervisors

Prof. Antony Musker  
Dr. Graham Roberts  
Dr. Charles Ryan

## Authors

Daogaru, Marian  
Dufresne, Alison  
Elkins, Adam  
Studebaker, Adam

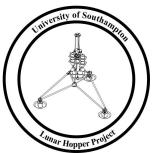
## Second Assessor

Dr. Angelo N. Grubisic

Thesis submitted in partial fulfillment for the requirements of the degree of Master of Engineering.

**University of Southampton  
Faculty of Engineering and Environment  
2015 - 2016**





# Introduction

The Lunar Hopper is a vertical take-off, vertical landing (VTVL) prototype rover for lunar exploration. The project commenced five years ago and has completed many important milestones. The Lunar Hopper Mk I was completed in Phase III and test flown. Though the test flight was unsuccessful, valuable data was gathered. Phase IV notably worked with a hybrid rocket development team to develop a new model, the Lunar Hopper Mk II, utilising the new engine. This model was designed and built during Phase IV but failed system flight-readiness tests on the day of attempted launch.

The main objective of this Phase V is to ruggedize the Mk II, ensuring that with or without flight testing, Phase V would be passing down an improved Mk II to Phase VI. This objective has been achieved through extensive structural, PDS and control redesigns carried out and outlined in this report. The next step in the project is to conduct a test flight, which has preliminarily planned for the end of May.

## Objectives

The main objective for this project is to design, build and test the prototype of a VTVL Lunar Hopper. The main objective of this phase of the project is to ‘ruggedize’ the Mk II design. This objective has been achieved through extensive structural, propellant delivery system and control redesigns carried out and outlined in this report. This was pursued as the primary objective, ensuring that with or without flight testing Phase V would be passing down an improved Mk II to Phase VI.

Secondary objectives were suggested/inherited from previous groups and were as follows:

- I. Investigate and debug communication drop outs and signal loss in the current communications system and fix or replace the system
- II. Design and develop a propellant delivery system to remove structural loading and leaks
- III. Redesign the Lunar Hopper’s structure and landing mechanism so that it operates independently of the current metal wire system.
- IV. Document and model the entire Lunar Hopper, including a detailed SolidWorks 3D model. As discussed previously, a large amount of documentation was produced during this phase, including a detailed SolidWorks model. This will be passed down to Phase VI team. This will be passed down to Phase VI team.
- V. Use the Lunar Hopper for outreach and promotion by performing demonstrations and generating multimedia for the University to use at open days.
- VI. Investigate and record the benefits and detriments of integrating an additional oxidiser tank onto the Lunar Hopper

## Approach

Initially, the team was split into two main areas – structure and electronics. Adam Studebaker and Alison Dufresne were to focus on structure while Marian Daogaru and Adam Elkins were to focus on electronics.

The team met throughout the week and with the project supervisors once per week. The Team Leader constructed a Gantt chart at the start of the year to develop the overall project plan. The initial project plan gave the team time to get organised and prepare for the project. The team read through the previous hopper reports and took an inventory of everything found in the hopper lab.

The Team Leader developed and updated a rolling actions list. Made in Excel, it recorded what the ongoing actions were, which team member was to be responsible, and the expected



completion date. Each action was either listed as incomplete, complete, or cancelled. The rolling actions list increased team productivity and communication. It provided structure to the meetings with the team supervisors and was regularly revisited to check the team's progress.

To track the progress of the individual systems, a 'systems overview' spreadsheet was created by the secretary. This file contained the objectives, derived requirements, subsystem information, mass and power budgets along with proposed changes to the Lunar Hopper subsystems.

## Resources

To be successful, the team had to use various resources not only at the University of Southampton but throughout the city of Southampton. At the University, room 13/1035 served as the main area for the project. Meetings were either held there or in one of the University's many meeting rooms. Additionally, the final build was assembled and integrated in the hopper lab. The Student Workshop and EDMC were both heavily used to build components and for manufacturing advice. Within the EDMC, Adam S. and Alison searched through scrap metal regularly to find suitable materials for free. In 13/1055, the design workshop, Alison used the University's 3D printers while Marian and Adam E. made use of freely supplied wood and acrylic to use with the University's laser cutters.

Outside of the University, the team used So Make It, a non-profit Makerspace in Southampton where the team leader is a member. There, the team could 3D print parts significantly faster than at the University for a fraction of the cost. A part that would cost around 30 pounds to print at the University could be printed for less than five pounds at So Make It.

## Constraints

The project overcame several major constraints. Time was a significant constraint, with under seven months allowed for achieving the objectives. In addition, the time constraint was heightened by the relatively small amount of manpower available for the project, considering the team size of four.

Budget was a second constraint. With a smaller group, the Project received a smaller budget (£680) from the University. However, the team managed to secure a grant through the Elevator Pitch (£1000). Although the available budget was considered enough to a certain extent, an even bigger budget would have enabled the team to outsource more manufacturing to external companies. Consequently, the funding constraints translated in more team time required for component manufacturing.

Lastly, project testing requires operating a 400 N rocket engine. The University does not possess an appropriate testing facility for this type of projects. Consequently, testing was heavily constrained. Moreover, in order to test the project, the team is required to travel to an external facility, which minimises the testing opportunities.



# Team organisation



*Lunar Hopper Phase V team. From left to right: Adam Studebaker, Marian Daogaru, Adam Elkins, Alison Dufresne.*

This year's lunar hopper team was one of the smallest groups at the University. Consequently, group organisation was vital for success. The team was split into four core administrative roles with multiple technical areas of specialty:

| Team member            | Admin Role                | Technical Roles  | Report Contribution   |
|------------------------|---------------------------|--|---|
| <b>Adam Studebaker</b> | Team Leader               | Chassis<br>Attitude Control System<br>Propellant Delivery System           | Group Organisation<br>Resources and Restrictions<br>Attitude Control System<br>Chassis<br>Report Editing  |
| <b>Marian Daogaru</b>  | Treasurer                 | Electronic Hardware<br>Flight Code<br>Communications                       | Budget<br>Electronics and Software<br>Report Editing  |
| <b>Alison Dufresne</b> | Secretary                 | Landing Mechanism<br>Propellant Delivery System<br>Attitude Control System | Project Heritage<br>Aims and Objectives<br>Attitude Control System<br>Propellant Delivery System<br>Landing system<br>Flight Test Plans<br>Report Editing |
| <b>Adam Elkins</b>     | Health and Safety Officer | Pilot Training Simulator   | Pilot Training Simulator  |

*Roles and responsibilities of each team member*

## Important results

### Structure

- Landing legs are now robust and functional
- Chassis has been redesigned, manufactured and assembled
- The new structure is more robust and rigid than the inherited model
- The new structure is simpler than the previous design

## Attitude Control System

- ACS is now securely supported by structure
- The ACS thrusters have been reprinted
- ACS manifold leaks have been eliminated

## Propellant Delivery System (PDS)

- The PDS has been redesigned
- Pipe bending has eliminated some complexity and mass from the system
- New system components have been procured

## Electronics

- The code has been improved in terms of response time, by creating a better algorithm, software syntax, and removing the obsolete code
- Previously detected software problems, such as the faulty ACS response, were solved
- A new implementation to define the true input from sensors was added

## Communications

- Communications are now reliable
- Zulu modules replaces Xbee communications module
- New circuits and breakout boards were manufactured to accommodate the new communications components
- Communications modules fulfils the system requirements

## Pilot Trainer

- The simulator has been designed and assembled

# Conclusion

The main objective of this phase of the project was to ‘ruggedize’ the Mk II design. This objective has been achieved through extensive structural, propellant delivery system (PDS) and control redesigns carried out and outlined in this report. The secondary objectives have almost all been achieved, with the exception of pending testing for the PDS.

The communication system issues have been rectified by replacing the unreliable components in the system. Reliable long distance communications have been achieved. The Lunar hopper control system software has been improved and previously detected software problems have been resolved. The pilot simulator has been designed and assembled.

The chassis has been redesigned and built. The ACS has a new structure, a refinished ACS manifold, and new 3D printed nozzles. The propellant delivery system has been redesigned, procured and is awaiting assembly. The landing mechanism has been redesigned to function independently of the steel wire used in Phase IV.

The Lunar Hopper Mk II has been fully documented, along with the media, test reports and so on. These documents have been combined onto a DVD which will be passed down to the Phase VI. The Lunar Hopper project has been successfully used for outreach and promotion by performing presentations at the open days and generating multimedia for the University to use.

An investigation into whether additional oxidiser tanks were needed was undertaken early in the



project and a feasibility study was generated which highlighted the lack in return for such an endeavour.

## Recommendations

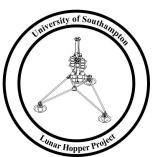
The first recommendation, is for a flight test to be conducted, should the flight test planned for June not occur. To do this, the Lunar Hopper systems will all have to pass flight-readiness test.

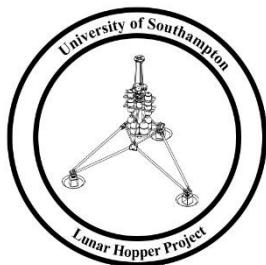
Further recommendations would be to reworked, test, and verify the ACS control theory.

Investigation into removing one of the ACS nitrogen tanks should be undertaken, with consideration to the structural changes which would have to be made to accommodate this.

Additionally, code changes could implemented to optimise the overall software operation, by reducing the computing time. Further development of the hopper and controller software can be made by introducing new pre-flight testing functions, such as testing the ACS. Lastly, the hopper could be programmed to have an autopilot function.

The electronics on the hopper can be upgraded by creating a more versatile power distribution board. Additionally, this board could be included in a more robust PCB design which should incorporate all the sensors attachment points. Swapping the Arduino for a Raspberry Pi would greatly improve the operating speed of the software, and vastly expand the capabilities of the hopper.





UNIVERSITY OF  
**Southampton**

# Lunar Hopper Mk. II Enhancements Main Report

Group Design Project 59

## Supervisors

Prof. Antony Musker  
Dr. Graham Roberts  
Dr. Charles Ryan

## Authors

Daogaru, Marian  
Dufresne, Alison  
Elkins, Adam  
Studebaker, Adam

## Second Assessor

Dr. Angelo N. Grubisic

Thesis submitted in partial fulfillment for the requirements of the degree of Master of Engineering.

**University of Southampton  
Faculty of Engineering and Environment  
2015 - 2016**



# **Group Project Academic Integrity Statement**

## **Declaration**

I, the undersigned, confirm that the material presented in this project is all my own work. References to, quotations from, and the discussion of work of any other person have been correctly acknowledged within the report in accordance with University of Southampton guidelines on academic integrity:

Signed: \_\_\_\_\_

Name: Adam Elkins (24485209)

Date: \_\_\_\_\_

Signed: \_\_\_\_\_

Name: Adam Studebaker (25424181)

Date: \_\_\_\_\_

Signed: \_\_\_\_\_

Name: Alison Dufresne (23322446)

Date: \_\_\_\_\_

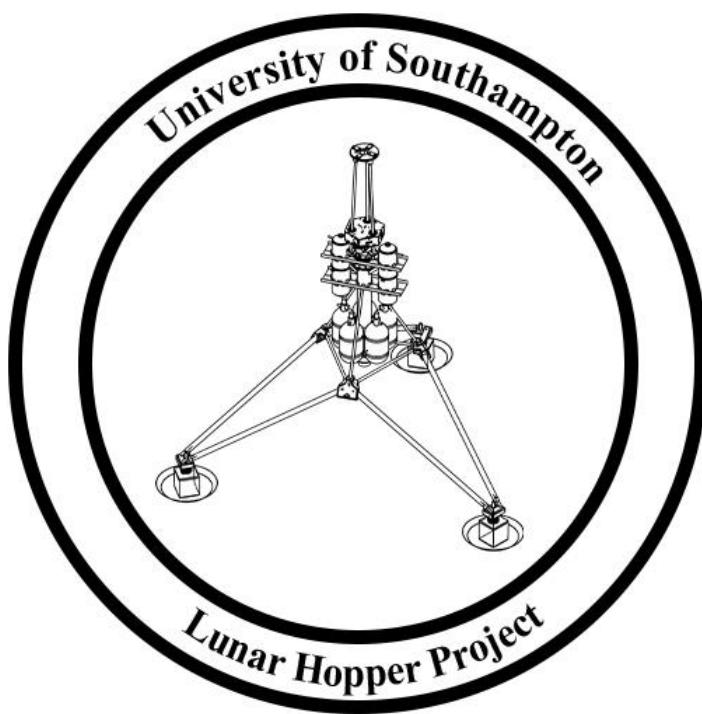
Signed: \_\_\_\_\_

Name: Marian Daogaru (25685252)

Date: \_\_\_\_\_

“We choose to go to the moon in this decade and do the other things, not because they are easy, but because they are hard.”

-Address at Rice University on Nation’s Space Effort; John F. Kennedy, 1962



## **Acknowledgements**

We sincerely thank our three project supervisors Prof. Tony Musker, Dr. Graham Roberts and Dr. Charles Ryan for their continuous support throughout the project. Their encouragements and their trust is us enabled us to be confident in our work, and be able to push ourselves even more. Their devotion for the project was a lighthouse after a long voyage.

We would like to thank the University for providing finances, facilities and the opportunity to be a part of such a great project. We would like to thank the staff of the EDMC and ISVR stores who provided essentially help and advice.

We would also like to thank our parents for their love and ongoing support throughout, without which, none of this would have been possible.

We acknowledge the help of Graham Williams from Solenoid Valve Ltd. for his help with changing several components. In addition, we would like to thank RF Solutions for their help with supplying 2 additional ZULU modules, free of charge.

# Table of Contents

|   |    |
|---|----|
| Acknowledgements .....                        | iv |
| Abbreviations .....                           | ix |
| Nomenclature .....                            | x  |
| 1 Introduction .....                          | 1  |
| 1.1 Project Heritage .....                    | 2  |
| 1.1.1 Lunar Hopper Phase I .....              | 2  |
| 1.1.1 Lunar Hopper Phase II .....             | 2  |
| 1.1.2 Lunar Hopper Phase III .....            | 2  |
| 1.1.3 Lunar Hopper Phase IV .....             | 3  |
| 1.1.4 Rocket Engine Heritage .....            | 4  |
| 1.2 Aims and Objectives .....                 | 5  |
| 1.3 Group Organisation .....                  | 6  |
| 1.4 Resources and Restrictions .....          | 10 |
| 1.4.1 Budget .....                            | 10 |
| 1.5 Mk II Overview .....                      | 13 |
| 2 Attitude Control System .....               | 14 |
| 2.1 System Requirements .....                 | 15 |
| 2.2 Phase IV Review .....                     | 15 |
| 2.2.1 Assumptions .....                       | 15 |
| 2.2.2 Attitude Control Thrusters (ACTs) ..... | 15 |
| 2.2.3 Structure .....                         | 16 |
| 2.2.4 ACS Manifold .....                      | 16 |
| 2.3 Design and Development .....              | 16 |
| 2.3.1 ACS Structure .....                     | 16 |
| 2.3.2 Nozzles .....                           | 18 |
| 2.4 Summary .....                             | 18 |
| 3 Propellant Delivery System .....            | 19 |
| 3.1 PDS Requirements .....                    | 20 |
| 3.2 Phase IV Design Review .....              | 20 |
| 3.3 Design and Development .....              | 22 |
| 3.4 Non Inherited Issues .....                | 23 |
| 3.5 Testing .....                             | 24 |
| 3.6 Summary .....                             | 24 |
| 4 Chassis .....                               | 27 |
| 4.1 Structural Requirements .....             | 27 |
| 4.2 Phase IV Design Review .....              | 27 |
| 4.3 Design and Development .....              | 28 |
| 4.3.1 Struts .....                            | 28 |

|       |  |    |
|-------|--|----|
| 4.3.2 | Nitrogen Tank Mounts .....             | 29 |
| 4.3.3 | Oxidiser Tank Mounts .....             | 34 |
| 4.4   | Summary .....                          | 40 |
| 5     | Landing System .....                   | 41 |
| 5.1   | Structural Requirements.....           | 41 |
| 5.2   | Phase IV Design Review.....            | 42 |
| 5.2.1 | Testing.....                           | 44 |
| 5.2.2 | Mathematical Analysis.....             | 45 |
| 5.3   | Design and Development .....           | 46 |
| 5.3.1 | Option Evaluation .....                | 46 |
| 5.3.2 | Spring Sizing.....                     | 47 |
| 5.3.3 | Finite Element Analysis .....          | 48 |
| 5.3.4 | Material Selection .....               | 50 |
| 5.3.5 | Manufacturing.....                     | 52 |
| 5.4   | Testing.....                           | 53 |
| 5.5   | Summary .....                          | 54 |
| 5.5.1 | Finalised Structural Design Mk II..... | 55 |
| 6     | Electronics and Software .....         | 56 |
| 6.1   | Hopper Software .....                  | 56 |
| 6.1.1 | System requirements .....              | 56 |
| 6.1.2 | Background software .....              | 56 |
| 6.1.3 | Software Issues .....                  | 58 |
| 6.1.4 | Software Updates .....                 | 60 |
| 6.1.5 | Code testing .....                     | 65 |
| 6.2   | Electronics.....                       | 66 |
| 6.2.1 | System requirements .....              | 66 |
| 6.2.2 | Components .....                       | 66 |
| 6.2.3 | Conclusion .....                       | 74 |
| 6.3   | Communication subsystem design.....    | 76 |
| 6.3.1 | Previous communication.....            | 76 |
| 6.3.2 | Radio Communication .....              | 77 |
| 6.3.3 | Communications Conclusion .....        | 84 |
| 7     | Pilot Training Simulator.....          | 85 |
| 7.1   | Background and Requirements .....      | 85 |
| 7.1.1 | Analysis of the problem .....          | 85 |
| 7.1.2 | Hopper Characteristics .....           | 86 |
| 7.2   | Software .....                         | 87 |
| 7.2.1 | fuelrate().....                        | 87 |
| 7.2.2 | simulator().....                       | 88 |

|        |   |     |
|--------|---|-----|
| 7.3    | Electronics .....   | 91  |
| 7.3.1  | Electrical Components.....  | 91  |
| 7.3.2  | Housing.....  | 91  |
| 7.4    | Accuracy Analysis and Testing .....                                     | 93  |
| 7.5    | Simulator Performance.....  | 93  |
| 7.6    | Summary.....  | 94  |
| 8      | Test Flight Plans .....   | 95  |
| 8.1    | Objectives .....  | 95  |
| 8.2    | Constraints.....  | 95  |
| 8.3    | Logistics .....   | 96  |
| 9      | Conclusions .....   | 98  |
| 9.1    | Attitude Control System.....  | 98  |
| 9.2    | Propellant delivery System.....   | 98  |
| 9.3    | Chassis .....   | 98  |
| 9.4    | Landing System.....   | 98  |
| 9.5    | Software & Electronics.....   | 99  |
| 9.6    | Communication .....   | 99  |
| 9.7    | Pilot Training Simulator .....  | 99  |
| 10     | Future Work.....  | 100 |
| 10.1   | Attitude Control System .....   | 100 |
| 10.2   | Propellant Delivery System .....  | 100 |
| 10.3   | Chassis .....   | 100 |
| 10.4   | Landing Mechanism .....   | 100 |
| 10.5   | Software.....   | 100 |
| 10.6   | Electronics .....   | 100 |
| 10.7   | Communication .....   | 101 |
| 10.8   | Pilot Training Simulator .....  | 101 |
| 11     | References .....  | 102 |
| 12     | Appendix A - Expenses .....   | 104 |
| 13     | Appendix B – Computational Modelling .....                              | 106 |
| 14     | Appendix C– Pressure Drop in Pipes Derivation Phase IV Report [4] ..... | 119 |
| 15     | Appendix D – Propellant Requirement Calculation from Phase IV [4] ..... | 121 |
| 16     | Appendix E –Technical Drawing for Corner Piece .....                    | 123 |
| 17     | Appendix F - Proposal for upgrading the Arduino to a Raspberry Pi ..... | 124 |
| 17.1   | What is Raspberry Pi? .....   | 124 |
| 17.2   | Changing the Arduino to a Pi? Is it worth it? .....                     | 124 |
| 17.2.1 | Pros .....  | 124 |
| 17.2.2 | Cons .....  | 125 |
| 17.3   | Conclusion.....   | 125 |

|        |  |     |
|--------|--|-----|
| 18     | Appendix G - Rate Gyro .....                                     | 127 |
| 19     | Appendix H – Test plans electronics .....                        | 128 |
| 19.1   | Test 1.....  | 128 |
| 19.1.1 | Objectives: .....  | 128 |
| 19.1.2 | Procedure: .....   | 128 |
| 19.1.3 | Results:.....  | 128 |
| 19.1.4 | Conclusions:.....  | 128 |
| 19.2   | Test Report.....   | 129 |
| 19.2.1 | Objective .....  | 129 |
| 19.2.2 | Item tested description .....                                    | 129 |
| 19.2.3 | Equipment description .....                                      | 129 |
| 19.2.4 | Test sequence .....  | 129 |
| 19.2.5 | Data discussion .....  | 129 |
| 19.2.6 | Conclusion .....   | 130 |
| 20     | Appendix I – ZULU module breakout board detailed schematics..... | 131 |
| 21     | Appendix J – Pilot Trainer Simulator Software .....              | 132 |

## **Abbreviations**

- ACS – Attitude Control System  
ACT - Attitude Control Thrusters  
CAA – Civil Aviation Authority  
Comms – Communication Subsystem  
EDMC – Engineering Design and Manufacturing Centre  
HDPE – High Density Polyethylene  
HTP – High-Test Peroxide  
IDE – Integrated Development Environment  
MECO – Main Engine Cut-Off  
Mk. / MK – Mark; variant of a project  
Oxi - Oxidiser  
P-III – Lunar Hopper Phase III Project  
P-IV – Lunar Hopper Phase IV Project  
P-V – Lunar Hopper Phase V Project (Current Project)  
PCB – Printed Board Circuit  
PDS – Propellant Delivery System  
PWM – Pulse-Width Modulation  
VTVL – Vertical Take-off Vertical Landing

# Nomenclature

| Symbol                         | Definition                      |
|--------------------------------|---------------------------------|
| <b>C</b>                       | Capacitance                     |
| <b>F<sub>available</sub></b>   | Thrust available                |
| <b>f<sub>c</sub></b>           | Cut-off frequency               |
| <b>R</b>                       | Resistance                      |
| <b>x</b>                       | Displacement                    |
| <b>F<sub>total</sub></b>       | Total force                     |
| V <sub>windmax</sub>           | Max wind velocity               |
| <b>F<sub>landingpad</sub></b>  | Force experience by landing pad |
| <b>F<sub>spring</sub></b>      | Force experience by spring      |
| <b>F<sub>hinge</sub></b>       | Force experience by hinge       |
| <b>M<sub>static</sub></b>      | Static moment                   |
| <b>M<sub>hardlanding</sub></b> | Hard landing moment             |
| <b>m</b>                       | Mass                            |
| <b>σ<sub>total</sub></b>       | Total stress                    |
| <b>σ<sub>axial</sub></b>       | Axial stress                    |
| <b>σ<sub>lateral</sub></b>     | Lateral stress                  |
| <b>k</b>                       | Spring stiffness                |
| <b>g</b>                       | Gravitational acceleration      |
| <b>h</b>                       | Height                          |
| <b>t</b>                       | Thickness                       |
| <b>U<sub>e</sub></b>           | Exit velocity                   |
| <b>ṁ</b>                       | mass flow rate                  |
| <b>ρ</b>                       | Density                         |
| <b>A</b>                       | Area                            |
| <b>L</b>                       | Length                          |
| <b>P</b>                       | Pressure                        |
| Δ                              | Variance                        |
| θ                              | Deflection angle                |
| θ <sub>s</sub>                 | Static deflection angle         |
| θ <sub>m</sub>                 | Max acceptable deflection angle |
| <b>v</b>                       | Velocity                        |
| <b>r</b>                       | Radius                          |
| <b>t</b>                       | Thickness                       |
| <b>F<sub>lateral</sub></b>     | The lateral force               |
| <b>F<sub>axial</sub></b>       | The axial force                 |

# 1 Introduction

Author: Alison Dufresne

The Lunar Hopper is a vertical take-off, vertical landing (VTVL) prototype rover for lunar exploration. The concept revolves around a novel method of travel for autonomous rovers on non-terrestrial surfaces.

Previous lunar rover designs (see Figure 1) utilise traditional wheel-based travel, which can be restrictive as speed of transit is limited by rough terrain and momentum losses are continuously incurred due to friction. Just as terrestrially we have moved away from land-based travel, designs for future non-terrestrial rovers are exploring faster, more novel ways of travel.

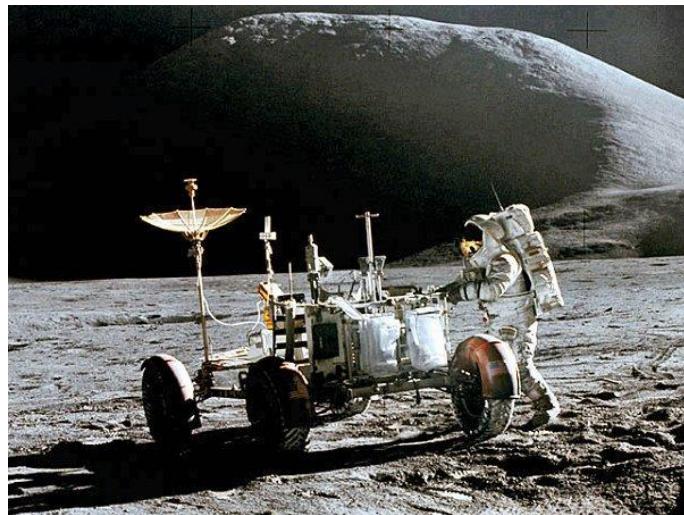


Figure 1: Apollo lunar rover [1]

The Lunar Hopper design overcomes the limitations of terrestrial travel by using a rocket engine coupled with an autonomous attitude control system capable of directing the rocket's thrust in order to launch and 'hop' from one location to another (see Figure 2).

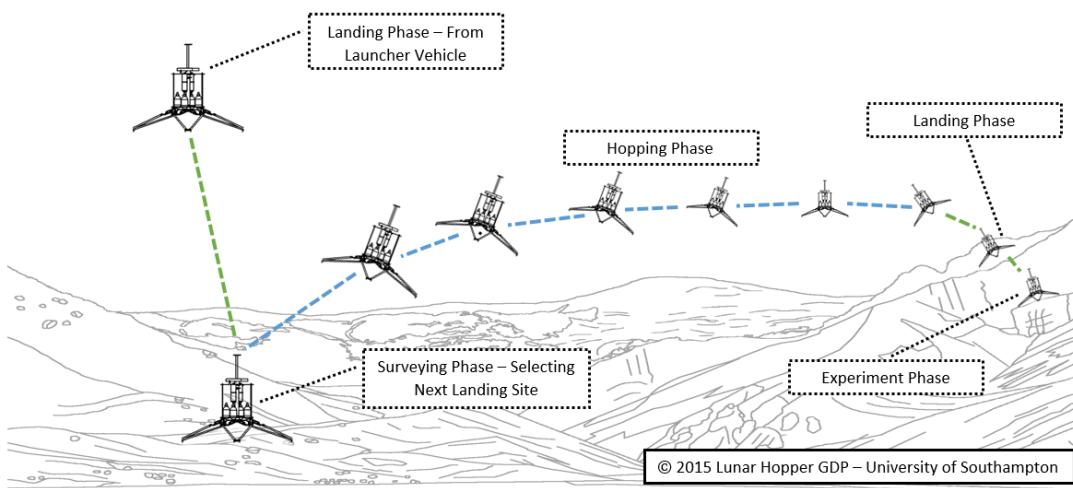


Figure 2: Illustration of the operational phases of the Lunar Hopper

This design could allow for pre-programmed rovers to autonomously 'hop' from location of interest to location of interest on the lunar surface, conducting large-scale science missions with minimal input from ground stations. The University of Southampton Lunar Hopper project aims to design, build and test the terrestrial prototype for this VTVL vehicle.

## 1.1 Project Heritage

Author: Alison Dufresne

### 1.1.1 Lunar Hopper Phase I

This project was initially inspired by the Google X prize which called for innovative solutions to lunar travel. During this phase of the project, the initial design for a VTVL vehicle with translating capabilities (the Lunar Hopper) was outlined and investigated. The first iteration of the Mk I was designed and produced based on utilising a 280 N bi-propellant rocket engine provided by DELTACAT.



Figure 3: Phase I Solidworks model of Lunar Hopper Mk I structure [2]

The initial version of the Attitude Control System (ACS) was designed to operate using a gimbal system which would tilt the engine to direct the thrust. This system was later replaced during Phase IV in favour of a cold gas thruster system, but the initial framework for the control law was developed during this phase. During this project, the H<sub>2</sub>O<sub>2</sub> tanks, which are also used on Mk II, were designed and manufactured.

### 1.1.1 Lunar Hopper Phase II

During the second year of this project, the Mk II structure was re-evaluated and upgraded. Shock absorbers were added to the landing pads, which are also used in the Mk II. The nitrogen tanks which are now used on the Mk II were purchased during this phase.



Figure 4: Inexpensive carbon fibre paintball tanks [3]

Static rocket tests were conducted which led to redesigned propulsion and propellant delivery systems, which have been subsequently replaced in the Mk II. A pilot trainer/hopper simulator was created. The ACS code and electronics were assembled and tested. The PDS was tested and validated. A test rig for the Hopper was designed. Flight testing was not conducted, however, due to health and safety clearance, weather and time. [3]

### 1.1.2 Lunar Hopper Phase III

At the end of the third iteration of the project, a successful model passed all necessary flight-readiness tests. A flight test was conducted (see Figure 5) but a gust of wind unbalanced the Mk I Lunar Hopper shortly after take-off, resulting in a somewhat dramatic crash, which damaged the structure beyond repair.

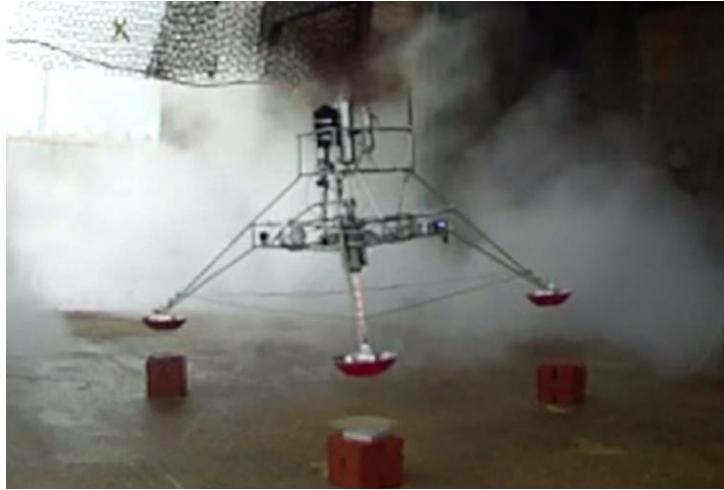


Figure 5: A previous iteration of the Lunar Hopper project powered by a 300 N [4]

### 1.1.3 Lunar Hopper Phase IV

The fourth iteration of the project decided to almost completely redesign the Lunar Hopper. Utilising the extensive knowledge collected by the previous groups they generated the Mk II design. The team worked concurrently with the hybrid rocket engine team to produce the Mk II (see Figure 6).



Figure 6: Mk II Phase IV Test Flight State

The Mk II differed from the Mk I in many ways. Most significantly, it ran in parallel to a team which designed and built a 400N hybrid rocket engine for the Mk II. The new engine and engine requirements led to considerable changes in all Lunar Hopper systems. A significant change was from the gimballed engine ACS, to Attitude Control Thruster (ACT) based ACS.

Unfortunately the model they created did not pass flight readiness tests on the day of launch, with several issues being identified like the unreliable communications modules and leaky manifolds. The model shown above was assembled on launch day, at the launch site but these issues proved too much of a risk for flight and therefore the engine was never fired, and the Mk II has not flown.

### 1.1.4 Rocket Engine Heritage

The Lunar Hopper Mk II utilises a hybrid (liquid oxidiser and solid fuel grain) rocket engine developed by a team of engineering students at the University of Southampton in 2015 for the Mk II [5].



Figure 7: 400 N Hybrid rocket engine developed by the University of Southampton

The hybrid rocket consists of a high-density polyethylene (HDPE) fuel grain, which is oxidised by 87.5% conc. hydrogen peroxide (HTP). The HTP is first passed through a catalyst pack to decompose it into supersaturated, super heat steam, which easily oxidises and ignites the HDPE in the combustion chamber producing ~400 N of thrust.

The engine was designed for a 20 bar maximum system pressure and a mass flow of ~0.16 kg/s, but during testing, low pressure was identified as one of the performance issues and subsequently the engine maximum pressure has been increased to 40 bar.

Leveraging the data from the Phase IV report [4] (see Appendix C) and hybrid engine test data, the requirements for a 15 s burn of this engine (including preheating phase) are 3.7kg of oxidiser and 0.23813 kg of nitrogen for pressurisation (see Appendix for calculations and verifications). The engine weighs 8.6 kg with fuel and catalyst included.

The main limitation of the engine is its thermal material constraints. In order to prevent the nozzle from melting, the burn time is restricted to 15s (at which point the nozzle should reach ~1700 K [5]). The engine is capable of throttling through control of the oxidiser tank pressure and thus, the mass flow rate of the oxidiser into the engine.

## **1.2 Aims and Objectives**

Author: Alison Dufresne

The main objective for this project is to design, build and test the prototype of a VTVL Lunar Hopper. The main objective of this phase of the project is to ‘ruggedize’ the Mk II design. This objective has been achieved through extensive structural, propellant delivery system and control redesigns carried out and outlined in this report. This was pursued as the primary objective, ensuring that with or without flight testing, Phase V would be passing down an improved Mk II to Phase VI.

Secondary objectives were suggested/inherited from previous groups and were as follows:

- I. Investigate and debug communication drop outs and signal loss in the current communications system and fix or replace the system
- II. Design and develop a propellant delivery system to remove structural loading and leaks
- III. Redesign the Lunar Hopper’s structure and landing mechanism so that it operates independently of the current metal wire system.
- IV. Document and model the entire Lunar Hopper, including a detailed SolidWorks 3D model. This will be passed down to Phase VI team.
- V. Use the Lunar Hopper for outreach and promotion by performing demonstrations and generating multimedia for the University to use at open days.
- VI. Investigate and record the benefits and detriments of integrating an additional oxidiser tank onto the Lunar Hopper.

### 1.3 Group Organisation

Author: Adam Studebaker

This year's lunar hopper team was one of the smallest groups at the University. Consequently, group organisation was vital for success. The team was split into four core administrative roles with multiple technical areas of specialty:

| Team member            | Admin Role                | Technical Role   | Report Contribution   |
|------------------------|---------------------------|--|---|
| <b>Adam Studebaker</b> | Team Leader               | Chassis<br>Attitude Control System<br>Propellant Delivery System           | Group Organisation<br>Resources and Restrictions<br>Attitude Control System<br>Chassis<br>Report Editing  |
| <b>Marian Daogaru</b>  | Treasurer                 | Electronic Hardware<br>Flight Code<br>Communications                       | Budget<br>Electronics and Software<br>Report Editing  |
| <b>Alison Dufresne</b> | Secretary                 | Landing Mechanism<br>Propellant Delivery System<br>Attitude Control System | Project Heritage<br>Aims and Objectives<br>Attitude Control System<br>Propellant Delivery System<br>Landing system<br>Flight Test Plans<br>Report Editing |
| <b>Adam Elkins</b>     | Health and Safety Officer | Pilot Training Simulator   | Pilot Training Simulator  |

Table 1: Roles and responsibilities of each team member.

Administratively, the Team Leader was responsible for the overall project timeline and the rolling actions list, which is explained further below. The Treasurer maintained an updated budget and regularly presented a full balance sheet. He tracked all incoming and outgoing cash flows, ensuring that the team did not overspend. The Secretary took regular meeting minutes and passed them on to all of the team members and supervisors regularly, keeping everyone up to date. She also booked meeting rooms when necessary. The Health and Safety Officer assessed the risks involved with all of the team's operations. He completed all of the necessary health and safety paperwork, discussed with the team ways to minimise risk and ensured that we followed university rules and regulations.

Each of the team members also had areas of technical specialty. Over the course of the project timeline, it became clear that such a small team would have to be flexible and dynamic in order to succeed. Throughout the year, many team members had to go outside of their initial technical role or change completely to help one another and successfully complete the hopper.

Initially, the team was split into two main areas – structure and electronics. Adam Studebaker and Alison Dufresne were to focus on structure while Marian Daogaru and Adam Elkins were to focus on electronics. Adam S. and Alison worked closely together to break the hopper structure and physical build down into manageable sections. Adam S. took charge of the chassis while Alison focused on the landing mechanism. After these were complete, they worked jointly on the hopper's attitude control system (ACS) and propellant delivery system (PDS). Alison kept a regularly updated mass budget to ensure that the team stayed within their weight limit. Marian Daogaru designed and assembled all of the electronics on the hopper. He went through all of the hopper's code piece-by-piece to ensure that every line worked consistently. Marian also provided valuable input to Adam S. and Alison while they were constructing the physical hopper. Adam Elkins designed a pilot trainer that could accurately simulate the dynamics of the hopper inflight. He derived all of the necessary equations for a single degree of freedom flight simulator, wrote the necessary code, and integrated everything into a standalone box.

The team met throughout the week and with the project supervisors once per week. Adam S. constructed a Gantt chart at the start of the year to develop the overall project plan (Figure 8).

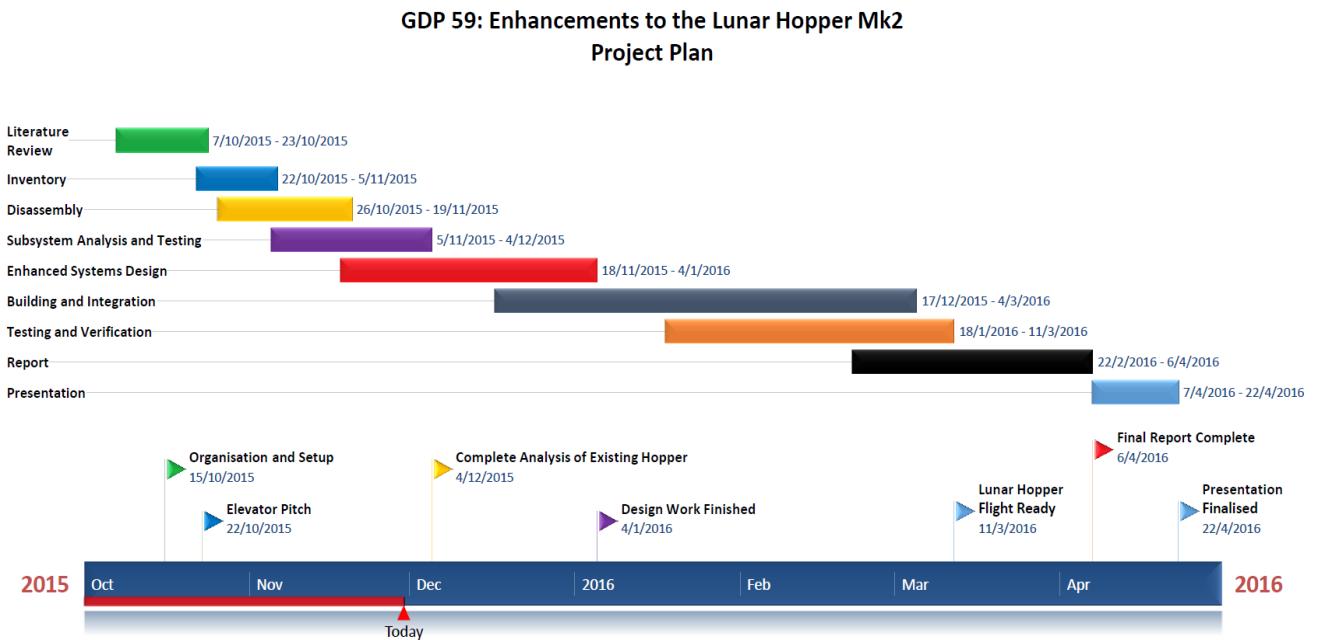


Figure 8: Initial Project Plan

The initial project plan gave the team time to get organised and prepare for the project. The team read through the previous hopper reports and took an inventory of everything found in the hopper lab.

| Item                     | Quantity | Comments            | Location                  |
|--------------------------|----------|---------------------|---------------------------|
| Pipe Cutter              | 1        |                     | 3rd shelf from top        |
| Screwdrivers             | 12       | Inside black binder | 2nd shelf                 |
| Soldering Paste          | 1        | Inside black binder | 2nd shelf                 |
| Soldering Wire           | 1        | Cylinder            | 2nd shelf                 |
| Electrical Tape          | 1        | Roll                | 2nd shelf                 |
| Desoldered               | 1        |                     | 2nd shelf                 |
| Rope                     | 1        | A bundle            | 2nd shelf                 |
| Computer Workstation     | 1        |                     | Corner of room            |
| Lunar Hopper Mk1         | 1        |                     | Side of room              |
| Laptop & Laptop Bag      | 1        |                     | Table/corner              |
| Thermocouple Connectors  | 3        | Green               | 2nd shelf                 |
| Muvi Action Cameras      | 2        |                     | 2nd shelf from top        |
| Muvi Extreme Sports Pack | 1        |                     | 2nd shelf from top        |
| Pro Throttle             | 1        |                     | Bottom shelf left         |
| Spare Arduino Uno        | 1        |                     | 2nd shelf teddy box       |
| Spare XBEE Module        | 1        |                     | 2nd shelf teddy box       |
| Keypad controller        | 1        |                     | 2nd shelf spare foam pads |

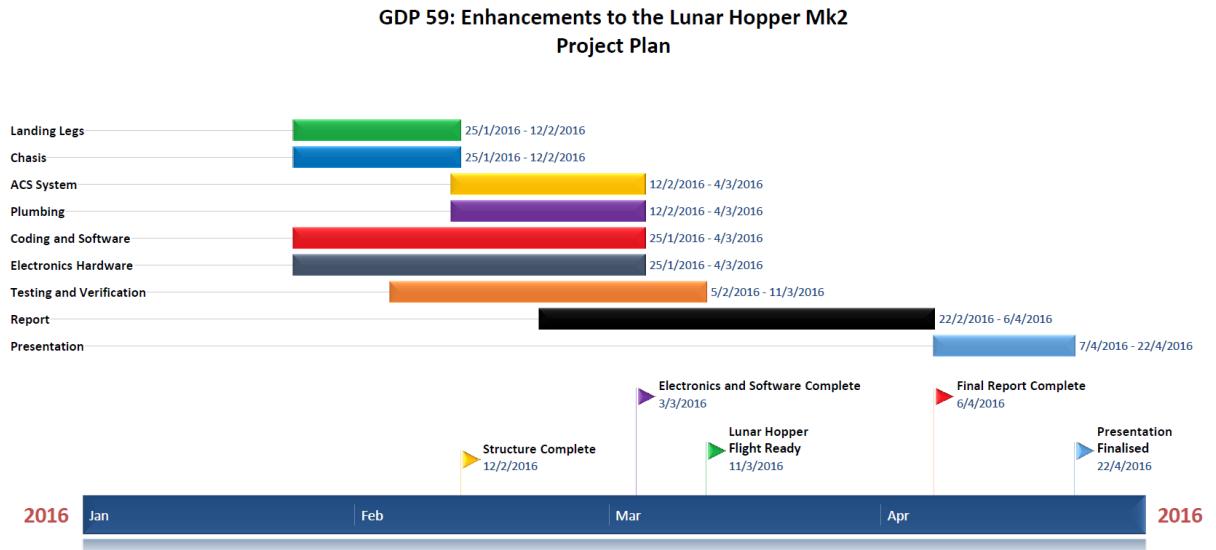
Figure 9: Sample section of the inventory

A sample section of the inventory is shown above. It was compiled into an Excel sheet to make it easier to find items the team already had in their possession. By creating this list, the team was able to avoid unnecessarily procuring new components that they would not need. The inventory listed what the item was, how much of it the team had, any comments about the item, and where it was located in the lab.

After getting set up, the team focused on the Elevator Pitch to generate funds for the project. The team's successful pitch raised 1000 pounds, which was incredibly useful throughout the year. The Elevator Pitch is discussed further in the Budget section of the report.

Following the Elevator Pitch, the plan was to begin disassembling the existing hopper and to conduct subsystem analysis and testing in order to analyse the previous design. This procedure was initially followed with the leg design, but it was taking far too long to make significant progress. It became clear that the team would need to adapt and change the way it worked on the hopper.

At the end of the first semester, the team met with the project supervisors to discuss how they had progressed relative to the project plan. The team was notably small and wanted to increase their productivity and efficiency. Part of this effort included generating a new project plan (Figure 10).



*Figure 10: Updated Project Plan*

The new project plan broke the work down into subsystems and gave a brief overview of when they would need to be completed. Within each of the subsystems, there was an additional project plan that included identifying issues, design work, the build and integration of the new enhanced system, and finally testing and verification. But this was not sufficient on its own because the work needed to be broken down into further accountable tasks.

Consequently, Adam S. developed and updated a rolling actions list. Made in Excel, it recorded what the ongoing actions were, which team member was to be responsible, and the expected completion date. Each action was either listed as incomplete, complete, or cancelled. The rolling actions list increased team productivity and communication. It provided structure to the meetings with the team supervisors and was regularly revisited to check the team's progress.

Figure 11 shows a sample section of the rolling actions list.

| Ongoing Actions   | Person Responsible | Comments  | Status     | Expected Completion |
|---|--------------------|---|------------|---------------------|
| Test new valve  | Marian             | Purchased and delivered, delayed because incorrect part sent, risk assessment complete, comment on the opening time | Complete   | 11-Feb              |
| Finish breakout board to convert hole spacing for radio module                                      | Marian             | Parts Arrived   | Complete   | 18-Feb              |
| Finalise Materials and Manufacturing Decisions  | Adam S             |   | Complete   | 25-Feb              |
| 3D print leg foot attachment pieces   | Alison             |   | Complete   | 3-Mar               |
| Make leg/chassis attachment pieces  | Alison/Adam S.     | All 3 complete  | Complete   | 3-Mar               |
| Look at top manifold and how it is connected  | Adam S.            |   | Complete   | 3-Mar               |
| Look at redesigning hinge link to add preloading  | Adam S/Alison      | Leg to hinge part has been complete. Preloading can be added by redesigning hinge to structure link                 | Cancelled  | N/A                 |
| Generate technical drawing for new tested and verified leg component                                | Alison             | New component must be remade identically  | Cancelled  | N/A                 |
| Design and produce housing for 3D printed 'top cone'  | Marian             |   | Cancelled  | N/A                 |
| Read up on integrated circuits for the Zulu comms module  | Marian/Adam E      |   | Incomplete | ongoing             |
| Test Zulu module and measure latency  | Marian             | digital comms on analog   | Incomplete | ongoing             |
| Complete the systems engineering documents (objectives, requirements, testing and validation plans) | Alison             | Generate documentation to easily track project progress   | Incomplete | ongoing             |
| Debug ACS code  | Marian/Adam E      |   | Incomplete | ongoing             |
| Design Arduino C code for pilot trainer   | Adam E             | Initial C code worked through. Being finalised  | Incomplete | 4-Feb               |

Figure 11: Sample section of the rolling actions list

At this stage, the team agreed that a pilot trainer would be necessary to have the new hopper complete a successful flight. Adam Elkins was assigned to work on the pilot trainer while the remaining three team members continued to work on the hopper itself.

To track the progress of the individual systems, a ‘systems overview’ spreadsheet was created by the secretary. This file contained the objectives, derived requirements, subsystem information, mass and power budgets along with proposed changes to the Lunar Hopper subsystems.

|    |   |   |             |          |  |
|----|---|---|-------------|----------|--|
| A5 | Redesign the Lunar Hopper's structure and landing mechanism so that it is reusable and operates independently of the current metal wire system. | Design Study for New Leg Design           | DS-02       | OneDrive |  |
|    |   | Tech drawing for new leg design           | TECH-STR-01 | OneDrive |  |
|    |   | Validation through testing of leg design  | VTR-03      | OneDrive |  |
| B4 | Use the Lunar Hopper for outreach and promotion by performing demonstrations and generating multimedia for the University to use.               | Video                                     |             | OneDrive |  |
| A6 | Fire the hybrid rocket motor and plot thrust versus throttle setting.   | Open day presentation                     |             | OneDrive |  |
| B5 | Investigate the potential of using an ultrasonic probe for altitude measurement.  | Test Report Document                      | TR-05       | OneDrive |  |
|    |   | Design Study for adding ultrasonic oprobe | DS-03       | OneDrive |  |
|    |   |   |             |          |  |
|    |   |   |             |          |  |
|    |   |   |             |          |  |

 Acronym Guide   Objectives and Document Tracing   Requirements   Budgets   Subsystem breakdown   Design changes proposed

Figure 12: Sample section of the 'Systems Overview' file

This document also contained links to what documentation needed to be generated and where it would be stored. Due to the quantity of documentation generated, only relevant reports have been included in the Appendix, but a comprehensive folder of documentation is to be passed onto the next group.

## 1.4 Resources and Restrictions

Author: Adam Studebaker

Time, money, and testing were all major restrictions the team had to face while completing the Lunar Hopper this year. The entire project had to be completed within 7 months without going over the set budget. The team was not allowed to do all of their testing on campus, which meant that money had to be reserved to travel offsite. Many of the team's tests relied supervision or equipment that meant that the project depended on other people's availability.

To be successful, the team had to use various resources not only at the University of Southampton but throughout the city of Southampton. At the University, room 13/1035 served as the main area for the project. Meetings were either held there or in one of the University's many meeting rooms. Additionally, the final build was assembled and integrated in the hopper lab. The Student Workshop and EDMC were both heavily used to build components and for manufacturing advice. Within the EDMC, Adam S. and Alison searched through scrap metal regularly to find suitable materials for free. In 13/1055, the design workshop, Alison used the University's 3D printers while Marian and Adam E. made use of freely supplied wood and acrylic to use with the University's laser cutters.

Outside of the University, the team used So Make It, a non-profit Makerspace in Southampton where the team leader is a member. There, the team could 3D print parts significantly faster than at the University for a fraction of the cost. A part that would cost around 30 pounds to print at the University could be printed for less than five pounds at So Make It.

### 1.4.1 Budget

Author: Marian Daogaru

Phase V of Lunar Hopper had a total budget of £1620. Initial GDP funding consisted of £300/GDP with £80 per student in the GDP, totalling to £620. £1000 were awarded to the Lunar Hopper GDP following a successful presentation in the Elevator Pitch competition, the maximum amount that could be awarded to a project. This additional budget was considered necessary to improve the project from what could be achieved with the initial budget, in terms of both functionality and overall improvement. The elevator pitch was initially divided in half, with £500 reserved for improving the existing components, and the remaining £500 reserved for additional testing.

The pitch award, not only almost tripled the budget, but also enable the GDP to gain future exposure. Two conditions were given to the group as part of being awarded the additional money. Firstly, the group had to take part in several UCAS Open Days, where the members

would present their work and what a GDP involves to prospective students. Secondly, the members of the group would have to produce a promotional video to be used in the future by the University, enabling the University to gain significant amounts of exposure not only to prospective students, but also to the general public.

Initially, the budget was divided between subsystems, with more funds being allocated to critical sections of the hopper that required drastic rework, while the less critical subsystems received a smaller budget. Table 2 describes the initial distribution of the budget.

| <b>Subsystem</b>    | <b>Sum (£)</b> |
|---------------------|----------------|
| <i>Chassis</i>      | 220            |
| <i>Electronics</i>  | 165            |
| <i>PDS</i>          | 95             |
| <i>Testing</i>      | 30             |
| <i>Delivery</i>     | 50             |
| <i>Margin (10%)</i> | 60             |
| <b>Total</b>        | <b>620</b>     |

Table 2: Initial budget partition.

However, Table 3 provides the updated budget, with the additional funds from the Elevator Pitch. The same reasoning as before was applied in this case, with more funds being diverted to the critical subsystems. In both cases, the distribution has been based on a team decision, as to which subsystems will require critical work to achieve the project objectives.

| <b>Subsystem</b>         | <b>Sum (£)</b> |
|--------------------------|----------------|
| <i>Structure</i>         | 345            |
| <i>Electronics</i>       | 295            |
| <i>Tools</i>             | 76             |
| <i>Additional Nozzle</i> | 100            |
| <i>Additional Valves</i> | 240            |
| <i>Delivery</i>          | 140            |
| <i>Testing</i>           | 270            |
| <i>Margin</i>            | 150            |
| <b>Total</b>             | <b>1616</b>    |

Table 3: Revised budget including Elevator Pitch funding.

Following the project implementation, a secondary distribution of funding was created, to enable the team to achieve the objectives. The funds were diverted from subsystems that were given a certain budget based on expected expenses, but did not require them, or from subsystems that were deemed less critical. As such, Table 4 provides the final budget for each subsystem that was worked on, based on total spending throughout the year. A final list of all spending can be found in Appendix A - Expenses. The remaining budget shall be used after report submission for further investments in the project.

| <b>Subsystem</b>   | <b>Sum (£)</b> |
|--------------------|----------------|
| <i>Structure</i>   | 512.6          |
| <i>Electronics</i> | 272.55         |
| <i>Tools</i>       | 65.45          |
| <i>Comms</i>       | 96.58          |
| <i>PDS</i>         | 340.3          |
| <i>Testing</i>     | 250            |
| <b>Total</b>       | <b>1537.46</b> |
| <b>Remaining</b>   | <b>82.53</b>   |

Table 4: Subsystem expenditures throughout the project.

While the overall funding was considered adequate for this project, several aspects of the project could have been improved with a bigger budget. The main improvements come from better but more expensive materials that could have been used throughout the design. Also, a higher budget would have allowed the team to outsource more manufacturing to external companies, thus reducing the time that each member had to spend manufacturing different components for the project.

## 1.5 Mk II Overview

Author: Alison Dufresne

The new Lunar Hopper Mk II design has not deviated greatly from the Mk II design. The main objective of the project is to design, build and test a VTVL vehicle. This creates certain requirements which need to be fulfilled. The most obvious requirement for the Lunar Hopper is for a propulsion system. This propulsion system creates a need for a propellant delivery system (PDS), communication system and a so on. The Mk II Lunar Hopper is essentially made up of the following systems;

- Propulsion
- Propellant Delivery (PDS)
- Attitude Control (ACS)
- On Board Computer (OBC)
- Communications (Comms)
- Structure

Every system is integral to the project objective and must work in tandem to ensure mission success. The communication system is responsible for delivering the pilot commands to the Hopper, the OBC is responsible for interpreting these commands into functions for the sub systems. The propulsion is manually controlled via the Comms system, the OBC and the PDS. The ACS system operates autonomously with the OBC once the ‘start’ command has been received. The PDS is a sub system of both the ACS and propulsion system which is operated by the OBC to provide the main engine thrust requested by the pilot as well as the necessary rectifying force to offset destabilisation disturbances (such as wind).

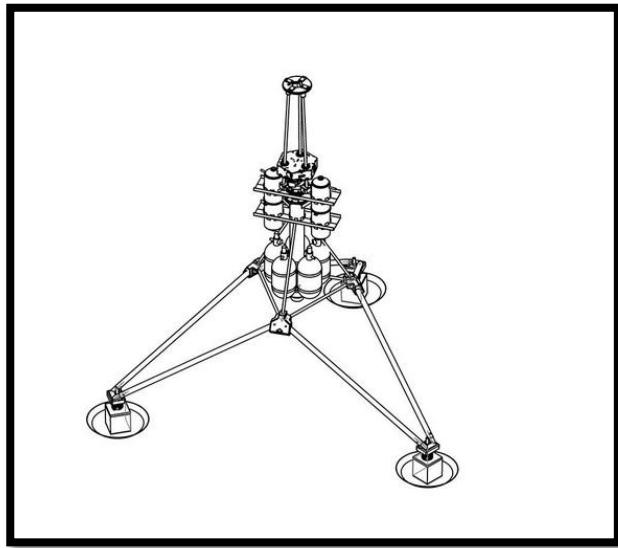


Figure 13: SolidWorks model of new Mk II design

The final design (Figure 13), is visibly different from the Mk II, but the main components and subsystem have remained the same, with some notable exceptions. These exceptions, along with all minor changes from the previous system, are discussed in detail in this report.

## 2 Attitude Control System

Authors: Adam Studebaker, Alison Dufresne

The attitude control system (ACS) provides control and stability for the lunar hopper throughout its flight. As VTVL is the objective of the Lunar Hopper, and disturbances in-flight are unavoidable, an ACS system is integral to the mission success. Without it, the Lunar Hopper becomes a projectile with no chance of steady controlled flight. The ACS consists of attitude control thrusters (ACTs), structure, propellant delivery system, software and electronics. In combination these components are used to produce 2-axis stabilisation for the Hopper by counter-acting external forces through autonomous activation of the ACTs. Four ACTs are required for 2-axis stabilisation.

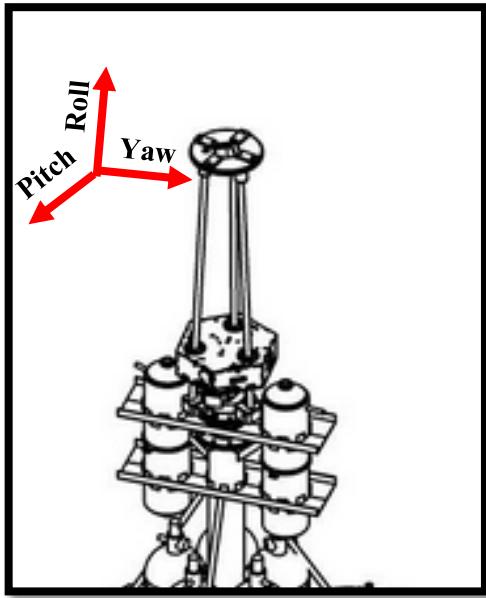


Figure 14: ACS with labelled axes

The control system for the ACS utilises gyroscopes and accelerometers to measure the Lunar Hopper's physical deviation and direction from pitch and yaw axes. The data from these is interpreted by the on board computer (OBC) and translated into a relay board which activates the appropriate solenoid valves, of which there are four, one for each ACT.

Phase IV designed an ACS which utilised a long moment arm to reduce the necessary thrust required per ACT to produce an adequate rectifying force. It consists of a half-metre-tall structure mounted to the top of the rocket motor [4]. At the very top, there are four nitrogen cold gas thrusters operated by a corresponding four solenoid valves. These four thrusters allow the lunar hopper to maintain control about its pitch and yaw axes without adding a large amount of complexity. Nitrogen gas was originally chosen as the propellant because it is dense, easy to store, and inert, so it will not readily react with its surroundings [4].

The previous hopper team designed the ACS control law that keeps the hopper stable in pitch and yaw during flight. Tomasz Noga developed a MATLAB model that tuned and optimised the control law's gains [4]. The current team did not prioritise making a new model for this year's hopper because of the team's limited size. The old model can be tuned to the new hopper, as the vehicles' disposition of mass is largely similar [6]. This old model was tested and verified last year using an ACS test rig [4]. The team is planning to adjust the control law to this year's vehicle before its flight. Tomasz is in contact with the team and can provide advice when the time comes. His section from the previous team's report can be found in Appendix B – Computational Modelling.

## 2.1 System Requirements

A set of requirements were derived in an effort to keep the hopper steady and under control while it remains within its operational envelope. The requirement can broadly be divided into:

Structural requirements

1. Fix the ACS the appropriate distance from the centre of gravity (1 m)
  - a. As all calculations from Phase IV are based on this distance
2. Fix the ACS so that the rectifying force from the ACTs is transmitted fully to the Chassis
  - a. Any absorption of the ACT forces will compromise the system operational requirement, which were leveraged from Phase II analysis [3]

Thrust Requirements

1. ACTs must be capable of two-axis stabilisation
  - a. This is derived from the vertical take-off vertical-landing objective of the project
2. ACTs must be capable of producing a sufficient rectifying moment
  - a. Phase IV calculations [4] have calculated this force to be 7.66 N based on the assumptions in discussed in Section 2.2.1

Electronics

1. ACS solenoids must be powered with 33.6 Watts
  - a. This has been calculated from the voltage and current required, which is discussed in Section 6.2.2.4
2. An autonomous control system for the ACTs must be implemented

## 2.2 Phase IV Review

The ACS system design review highlighted many serious issues which are discussed below.

### 2.2.1 Assumptions

The design of the previous system was based on a large number of assumptions which introduce potential errors in calculations [4].

Previous assumptions:

- Moment arm 1 m
- Distance between centre of pressure and centre of mass of 0.5 m
- Surface area 1 m<sup>3</sup>
- 0.5 drag coefficient
- Wind of 5 m/s

In reality the moment arm was underestimated, which reduced the rectification force it would be capable of delivering. The previous team also overestimated the distance between the centre of mass and centre of pressure, as well as the surface area of the hopper. The ACS system will need to correct for disturbances from two different sources. Wind will push the hopper, inducing a destabilising force. Additionally, as the hopper operates it will expend its propellant, which will cause the CM to move and alter the hopper's flight dynamics.

### 2.2.2 Attitude Control Thrusters (ACTs)

The attitude control thrusters rely on a 3D printed nozzle which screws into the solenoid valves. These nozzles were developed by the Phase IV team to achieve 4.7 Newtons of thrust. However, testing proved that they produce approximately 2.3 Newtons [4]. The previous team concluded that this was because of the surface finish of the 3D printed design [4]. Consequently, the hopper cannot fly in as adverse conditions as the previous team initially thought.

$$F_{available} = 2.3 \text{ N}$$

$$\text{Max Disturbance} = \frac{F_{\text{available}}}{0.5} = 4.6 \text{ N}$$

$$V_{\text{windmax}} = \frac{4.6}{7.7} * 5.5 = 3.3 \text{ m/s}$$

Previously, they had planned to correct for disturbances less than 7.7 N, or at a wind velocity of 10 Knots (5.5 m/s) [4]. Without new nozzles, the hopper's operational envelope would be reduced, allowing for only 6.4 Knots (2.5 m/s) winds.

### 2.2.3 Structure

The ACS boom which was designed in the Phase IV design was never manufactured (hence the reduced moment arm). The entire ACS was supported by the gas feed lines (three stainless steel 6 mm hollow pipes). These provided very little support and did not constrain the system effectively. It also put unnecessary stress on the nitrogen pipes, which should ideally remain independent of the structure.

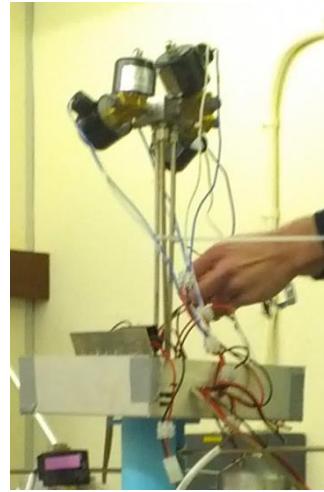


Figure 15: Inherited ACS

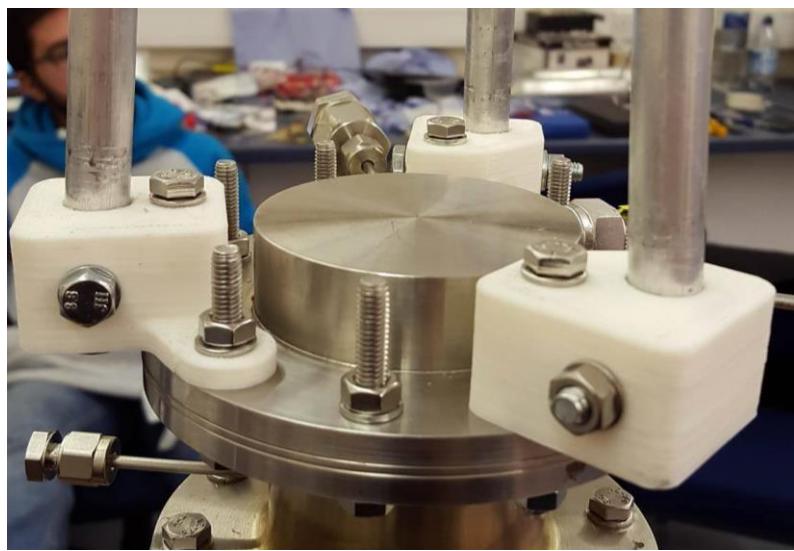
### 2.2.4 ACS Manifold

The ACS manifold is responsible for combining the nitrogen from the three tanks and directing it to the four ACTs. There were two issues with this manifold. The first being leaks and the second being inappropriate female threading/fittings which prevented the solenoid valves from aligned when fully connected (see Figure 15). Both of these issues are discussed in further detail in Section 3.3.

## 2.3 Design and Development

### 2.3.1 ACS Structure

The first issue to tackle was structure. The main design concept was derived from the chassis structure. In the Phase IV ACS design, there was a single vertical boom going from the rocket motor to the ACS manifold (which was never manufactured). The new design uses three triangulated struts attached to the top of the rocket motor with three 3D printed brackets. Each bracket is 3D printed with holes at 87 degrees so that the ACS struts are angled inwards. This increases rigidity and makes the entire ACS structure much stronger. Originally, the ACS brackets were designed to be made out of metal, but 3D printing allowed the design to be much more intricate without making it harder to build. Additionally, it significantly reduced the weight of the components. The only concern was whether or not the 3D printed components would be strong enough. To prove that they were, prototype parts were made and tested. They ended up being much stronger than required, so much so that the metal rod used to apply the force would bend before the 3D printed part would show any sign of structural failure.



*Figure 16: Prototype 3D printed ACS brackets*

At the top of the tower, there is a 3D printed ACS manifold holder. The ACS manifold holder attaches to the three struts and firmly secures the ACS manifold while allowing access underneath and to all four sides of the manifold itself. The following is an image of the ACS structure:



*Figure 17: ACS Structure*

### 2.3.2 Nozzles

To resolve the ACT issues, it was decided to reprint the nozzles. With the University's acquisition of higher-resolution printers than were available during Phase IV, more uniform and higher quality nozzles could be printed using the Phase IV design. Due to the relatively small size and volume, a decision was made to print many of these and to re-test to gauge consistency.



*Figure 18: Example of some of the ACT nozzles printed in Phase V*

### 2.4 Summary

The ACS structure has been completely redesigned to be rigid, strong, and lightweight for all phases of the hopper's flight. The ACS manifold has been refinished to ensure that it is leak-free and operates properly. New nozzles have been printed that are of a higher quality and surface finish than the previous ones. This will help rectify the thrust performance reduction that the previous team discovered and allow the hopper to operate within a wider operational envelope. Each of the structural requirements was met. The ACS is 1 metre from centre of gravity and the ACS boom's structure has been built rigidly so the thrust generated by the ACTs is effectively transferred to the rest of the Lunar Hopper body. The two thrust requirements were verified by the previous team, but more testing is required to verify the current system so it can be safely flown. The electronics requirements were met and are discussed further in Section 6.

### 3 Propellant Delivery System

Author: Alison Dufresne

As outlined in Section 1.1.4 and Section 2 respectively, the Lunar Hopper relies on a hybrid rocket engine for propulsion and cold gas nitrogen thrusters for attitude control. The propellant delivery system (PDS) refers to the system of tanks, pipes, valves and fittings which allows for controlled use of the engine and thrusters. This section discuss the non-electric physical PDS system on the Lunar Hopper Mk II. The related control system, which operates the solenoid valves, is discussed in Section 6.1 and 6.2.

The Lunar Hopper PDS is broken down into two subsystems: ACS and Main Engine. The ACS PDS is responsible for delivering the nitrogen from the tanks to the ACTs. The PDS schematic on the left in Figure 19 illustrates the Phase IV design for the ACS PDS of the Mk II. As can be seen, the ACS system is composed of three nitrogen tanks which are connected to the ‘ACS manifold’ and the ACTs. Although only two nitrogen tanks are necessary for storage, Phase IV decided to use the third tank to minimise the mass offset which will be incurred due to sloshing and unequal draw rates. Phase V has also chosen to utilise this third nitrogen tank for similar purposes as will be discussed further in Section 4.3. It was also deemed convenient as the previous system (including the manifold) is designed to utilise the three nitrogen tanks and therefore would minimise the amount of redesigning required for this system. The ACS PDS relies heavily on the function of the ACS manifold which combines the nitrogen from the tanks and then splits it into four separate channels, each connected to one solenoid valve which operates the attitude control thrusters (ACTs). As discussed in Section 2 the ACTs are located 1 m from the Lunar Hopper’s centre of gravity. To maintain this low centre of gravity, the nitrogen tanks are mounted just above the engine nozzle from the ACTs. The pipe pressure for the ACS system is 56 bar, which is controlled by a regulator on each of the nitrogen tanks. [6]

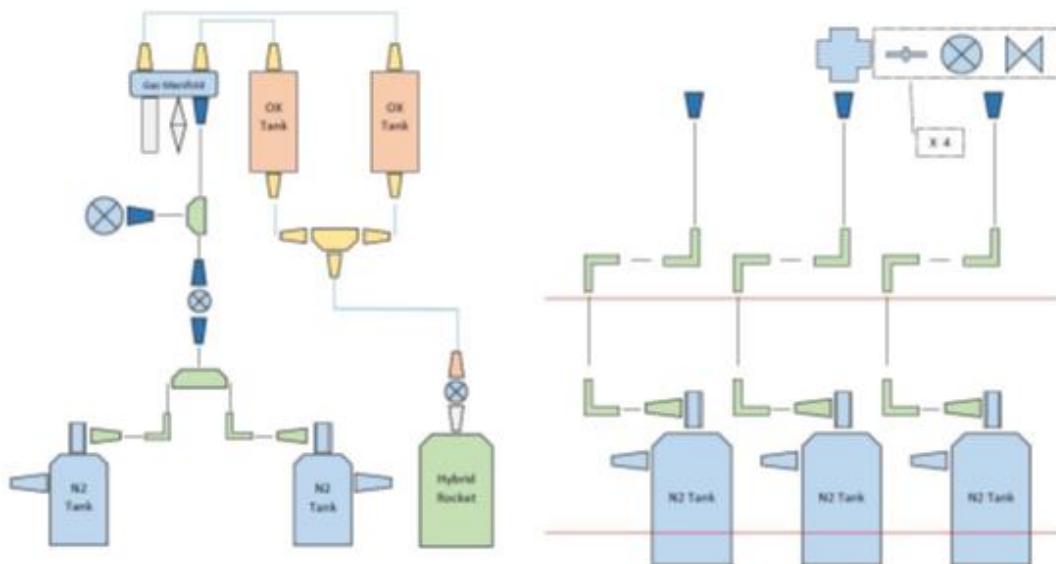


Figure 19: ACS Propellant Delivery System (left) and Engine PDS (right) for Mk II [4]

The second subsystem is the Main Engine (Figure 19). The main function of this system is to transport the oxidiser, which is stored in two separate tanks, to the catalyst pack inlet at the top of the engine at the appropriate flow rate. To achieve this, the oxidiser tanks must be pressurised.

The system is pressurised using nitrogen gas stored in two tanks, which require controlled delivery to the oxidiser tanks. The control is more complex than for the ACS system as the engine needs to be throttled. The system is remote controlled by the pilot which means the

communication system must be fully operational. The pilot must first send a ‘start’ command, which is translated by the OBC (discussed more thoroughly in Section 6.1.4.8) through a calculation of required mass flow rate, into a desired pressure in the oxidiser tanks (~16 bar for ‘start’ conditions). To achieve this, the OBC would open the solenoid valve between the nitrogen tank and the oxidiser tanks until the pressure transducer reads ~16 bar. The pressure transducer is connected to the manifold which is also connected to a solenoid vent valve and a mechanical relief safety valve. This valve is opened when the MECO command is received by the OBC. The oxidiser tanks inherited are limited to 40 bar [2] and should the pressure reach this, the mechanical relief valve will vent the nitrogen. These are essential safety features in the PDS.

### 3.1 PDS Requirements

The PDS system requirements are derived from the propulsion and ACS on board the Lunar Hopper.

ACS derived PDS requirements;

1. Must not incur a pressure drop of >1% in the pipes
  - a. The ACS system is designed to operate at 58 bar. Any significant pressure drop will compromise the system and potentially render it ineffective
2. Must provide nitrogen to all four thrusters
3. Must be rated to at least 60 bar
4. Must be capable of delivering 0.223 kg nitrogen to the ACT at 58 Bar pressure from the tanks [5]
  - a. Derived from ACS requirements. (See Appendix C)
5. Must store 0.223 kg of nitrogen
  - a. As derived from ACS requirements. (See Appendix C)

Main engine derived PDS requirements;

1. Must deliver ~3.7 kg hydrogen peroxide to engine at a variable flow rate of up to 0.216 kg/s for pre heat and a steady 0.16 kg/s during nominal flight. [4]
  - a. Derived from vertical take-off vertical landing requirement and inherited from the hybrid rocket engine used
2. Must be capable of delivering 0.298 kg of nitrogen to the hydrogen peroxide tanks for pressurisation [4]
3. Must be rated at least >60bar
4. All material must be compatible with 87.5% HTP
  - a. Derived by the oxidiser used in the hybrid engine
5. All fittings must be compatible with the current tanks and manifolds
  - a. This is due to time and budgets constraints which make replacing these undesirable
6. Must not incur a pressure drop of >1% in the pipes
  - a. The system is designed to function at variable pressures up to 40 bar

### 3.2 Phase IV Design Review

The Mk II PDS from Phase IV was one of the systems which did not pass flight readiness tests on the day of attempted flight test. It was inherited partially assembled, as shown in Figure 20. There were many documented issues, the most visible of which being a haphazardly assembled ACS at the top which was supported by three 6 mm hollow pipes that were responsible for delivering the nitrogen gas to the ACTs. Hanging above the base plate, the nitrogen tanks were unconstrained in the pitch and yaw axis, which would have caused major dynamic stability issues should flight have occurred. The oxidiser tank pipes, on the other hand, were excessively

long, which lead to them being wrapped around the oxidiser tanks. It was intended to change them to stainless steel during the Phase IV but this did not occur.



Figure 20: Inherited Mk II PDS

Though some complexity is inevitable, the use of many small pipes and 90-degree fittings made the systems difficult to understand and ultimately heavy. Inconsistent systems of measurement (both imperial and metric) diameters and fittings were used which were prone to leaks.

However, in some cases it is unavoidable (the  $H_2O_2$  tank pipe has an OD 5/16ths of an inch and replacing the tanks would be extremely expensive). Fittings from various companies were being used. The fittings had reportedly caused issues previously and only five spare compatible ferules (which are necessary to seal the fitting to the pipes) were found.

The major issues in this inherited system are as follows:

- ACS manifold fittings had leaks due to a rough surface area
- ACS manifold was constraining the solenoid valves so they were misaligned (as can be seen at the top of Figure 20)
- Nylon piping was being used for the hydrogen peroxide which was not rated to 60 bar
- Nylon piping was excessively long
- Nylon has low compatibility with HTP

At the end of the review it was decided that issues such as pipes of insufficient lengths and tanks being restricted were most easily eradicated through complete replacement. In order to simplify the system it was also decided to replace the majority of the fittings. The system was to be redesigned utilising only the functional components and fittings.

From the previous system, the functional components which were kept are as follows:

- Tanks (both nitrogen and HTP)
- Fittings on tanks (nitrogen)
- ACS Manifold
- Engine Manifold
- Pressure Transducer
- Mechanical Relief Valve
- Solenoid Valves

The constraints which affected Phase IV, were experienced in Phase V as well. As this report is written, the PDS system has been designed, procured and assembly techniques have been tested. The full assembly and testing of the PDS system is planned for completion by the end of April.

### 3.3 Design and Development

The first major change which took place in the system was a change from straight pipes to shaped pipes utilising a pipe bender. This allowed the PDS to be completed quickly post structural design and manufacture, which would not have been possible had pipe bending not been used.



*Figure 21: Prototype pipe bent by hand*

Starting with 3 m of stainless steel piping, the desired path was measured and marked using lengths of string which were then used to mark the pipes with the ‘centre of bend’ and radius of bend. The pipe angles were tailored to the structure as required. As the pipe bender worked up to 180 degrees, it provided less constraints and a much higher degree of flexibility than previous design. Pipe bending removed the need for 90-degree fittings (of which there were 11). This has reduced weight by 0.495 kg and reduced pressure losses while reducing the overall cost of the system (assuming a pipe bender is available to work with). As discussed in the critical design review, the pipes were constraining tanks and manifolds.

The redesign of the ACS structure (as discussed in Section 2) removed the need for the ACS PDS to take structural loading. This, combined with adequate lengths of piping replacing the short restrictive inherited pipes, removed the constraints on the nitrogen tank positions.

All piping on the Lunar Hopper has been replaced, including the nylon piping. This improves the ruggedness and long-term reliability of the PDS system as well as helping standardise the unit metric system used throughout. The new piping chosen is 6 mm stainless steel 316 with 1 mm wall thickness and is rated up to 450 bar. For this small variation in wall thickness there is a small variation in pipe incurred pressure drops. Using the equations and the assumptions made in Phase IV report (see Appendix B for full details), the pressure drop was calculated for the new pipe. The first component of the pressure drop is the frictional losses. To calculate this, the exit velocity was recalculated using the new pipe inner diameter (D).

$$U_e = \frac{\dot{m}_{throat}}{\rho A_{pipe}} = 14.097 \text{ m/s}$$

This was then used to calculate the frictional pipe losses, using all of the same variables and constants as in Phase IV analysis where L is the length of pipe, which is assumed to be 1 m.

$$\Delta P_{friction} = \lambda \frac{L}{D} \frac{\rho}{2} U_e^2 = 0.45 \text{ bar}$$

The pressure drop due to vertical elevation will be the same as calculated in Phase IV (as it is independent of pipe diameter).

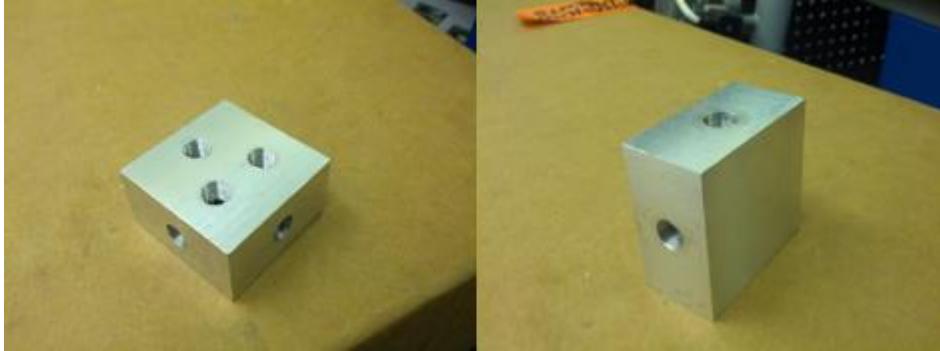
$$\Delta P_{elevation} = \rho g L = 0.006535 \text{ bar}$$

$$\Delta P_{total} = \Delta P_{elevation} + \Delta P_{friction} = 0.457 \text{ bar}$$

This, as expected, is a less than 1% pressure drop across the pipes, which meets requirements.

The leaky manifold inherited from the previous group was inspected and the extremely rough surface finish of the piece was identified as the cause of the leaks. Using a mill at the university

EDMC, the surface finish was improved. The issue with the misaligned solenoid valves was has been rectified with the use of two extra fittings, one which connects the ACS to a piece of pipe and another to the solenoid valve, which will allow the solenoid valves to be rotated to an appropriate position, until aligned correctly.



*Figure 22: New surface finish on the ACS manifold*

Tank fittings have all remained unchanged from Phase IV design due to their effectiveness. The other fittings, however, have all been replaced. Re-using the inherited fittings would require procuring new compatible ferrules, which proved difficult as the company from which the previous phase procured the components from do not sell individual ferrules. As so many new fittings and pipes were needed, the decision was made to order entirely new fittings (see Figure 17).



*Figure 23: New PDS fittings*

### 3.4 Non Inherited Issues

Early on in the project, whilst working on the structure, one of the members of the team snapped the piping on one of the oxidiser tanks. This presented a rather significant issue as no appropriate piping was available to replace it. The material compatibility is essential to the functionality and safety of the Lunar Hopper. Replacement piping was eventually procured for the oxidiser tanks, but unfortunately in an imperial measurement (5/16 in). As the OD of the pipe used previously was the same, and the difference in OD is only 0.063 mm, it was accepted that though it is not the preferred 8 mm OD, it would be appropriate.

To weld the pipe onto the damaged oxidiser tank, the tank had to be filed down to provide a smooth surface to weld. The weld material used (4041 wire filler) had to also be compatible with HTP and therefore was chosen carefully. Hydrogen peroxide materials compatibility for various material used on the Lunar Hopper is contained in Table 4 [7]

Table 5: Material compatibility with HTP

| Material            | Compatibility with 50% conc. | Compatibility with 100% conc. |
|---------------------|------------------------------|-------------------------------|
| Epoxy               | N/A                          | Excellent                     |
| Aluminium           | Excellent                    | Excellent                     |
| ABS                 | N/A                          | Excellent                     |
| Stainless Steel 316 | Excellent                    | Excellent                     |
| Nylon               | Severe Effect                | Severe Effect                 |
| Viton               | Excellent                    | Excellent                     |

The last task in rectifying the problem was extensive cleaning of the tanks and pipes. Any small amount of incompatible material (such as the filings from the tank) would cause partial decomposition of HTP within the pipes and therefore is unacceptable. To achieve this the tanks must be cleared out with multiple runs of de-ionised water with small amount of cleaning agent [8] (roughly ten rinse through) followed by more pure de-ionised water rinses (roughly twenty).

### 3.5 Testing

Preliminary testing to verify that the leaks in the manifold were eliminated was carried out, though unpressurised and with a liquid (water) instead of gas. The system must now also be verified under pressurised conditions before it can be certain that the leaks have been eliminated.

The following tests must be conducted prior to flight;

- Leak tests for each system, subsystem and component
- Full system pressure testing
- ACT pressure testing

### 3.6 Summary

The following ACS PDS (see Figure 25) was designed and developed during this phase.

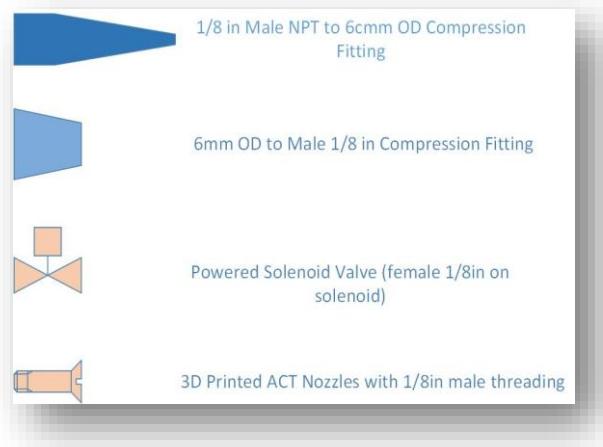


Figure 24: ACS PDS figure index

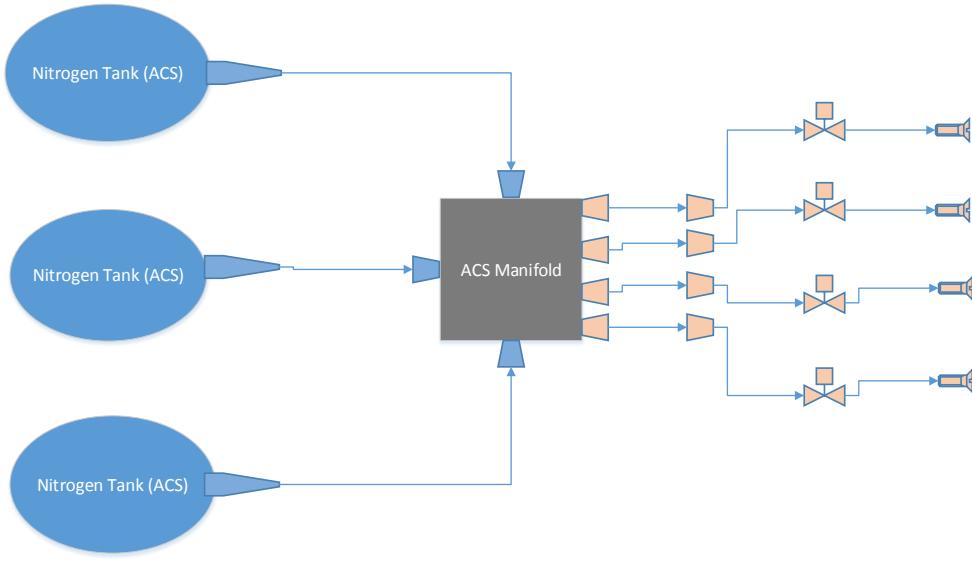


Figure 25: MK II updated ACS PDS

This design has been simplified from the previous system by utilising pipe bending and removing all of the 90-degree fittings. This has also decreased the system mass by almost half a kg (0.495 kg). The system no longer takes structural loading or restricts the position of important subsystem components such as tanks. New, more robust components have been procured. Preliminary testing has shown no leaks in the ACS manifold, however, full verification through pressure testing is pending. These achievements can be seen as significant improvement of the inherited system. Full system leak testing and pressure testing are to be conducted by the end of April.

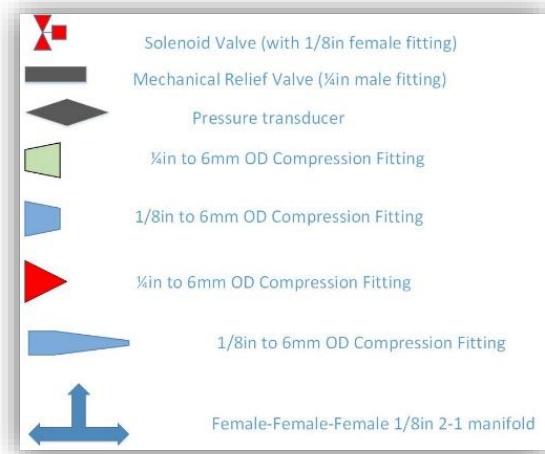


Figure 26: Main Engine PDS figure index

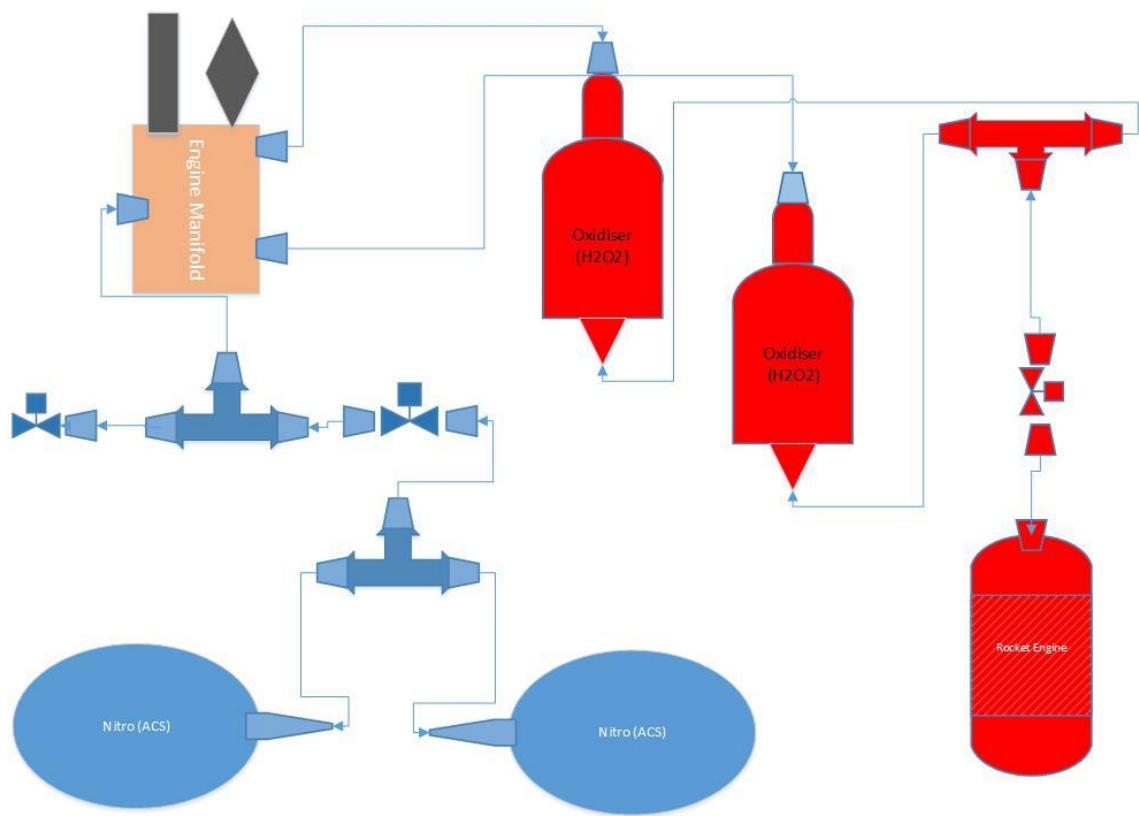


Figure 27: New engine PDS diagram and index

## 4 Chassis

Author: Adam Studebaker

The Lunar Hopper's chassis is the core of the Hopper's structure during flight. It supports the rocket motor, various tanks, the ACS, and more. The chassis includes everything structural that is above the hopper's landing legs. This year focused on four main goals for the new hopper chassis – make it rigid, make it light, make it simple, and make it affordable.

### 4.1 Structural Requirements

There are several structural requirements for the chassis. They are:

1. Centre of mass must align with the thrust axis of the rocket engine
2. Structure must secure all tanks rigidly
3. Structure must support the attitude control system (ACS)
4. Structure must house electronic sensors.

### 4.2 Phase IV Design Review

Last year's design had three main issues that needed to be addressed. It was heavy, it was not rigid, and it was not symmetrical. Initially, this caused a great deal of concern amongst the team. How could the design be made stronger and more rigid without adding additional weight? It became clear that the entire chassis would have to be redesigned from the ground up.

The previous hopper team investigated several different designs before choosing their final chassis shape. They looked at square, triangular, and hexagonal frames, and then chose the triangular design as it had the greatest strength to weight ratio [4]. They also discovered that, in order to move the centre of pressure down, the nitrogen tanks would need to be moved to the bottom of the hopper with the oxidiser tanks on top [4]. As the centre of pressure is moved lower, the wind's destabilising effect is diminished. The current hopper team agreed with last year's research and decided to pursue another triangular structure with the same vertical positioning for the tanks.

The earlier design used a top and bottom plate to support the hopper's weight [4]. Their final design is shown below:

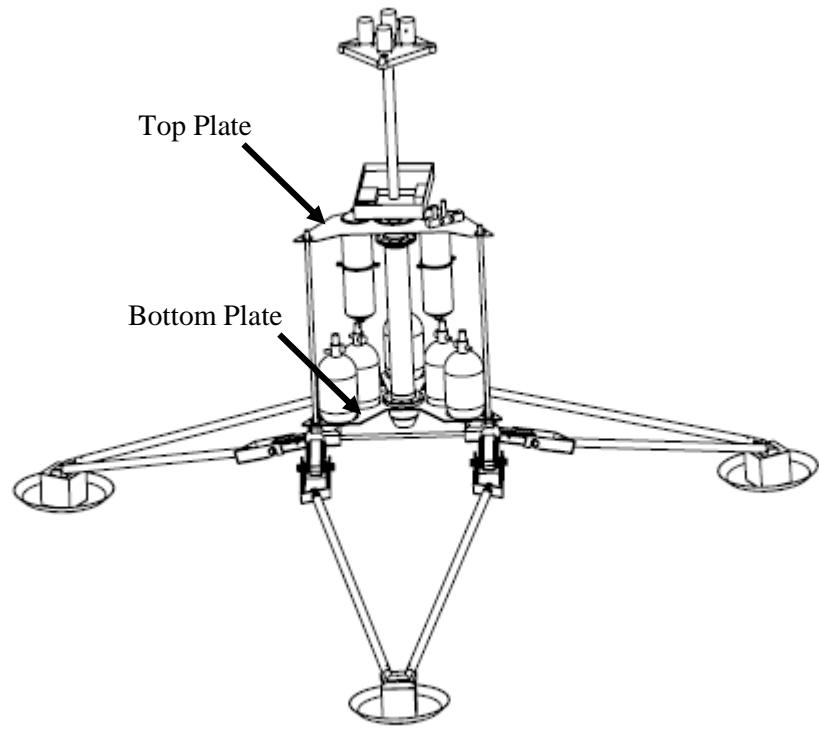


Figure 28: Previous Hopper assembly [4]

These plates were warping under the weight they were supporting. The entire frame lacked triangulation, which meant that a side force applied to the top of the frame would cause the entire chassis to deflect. All of these issues are further explored in the following chassis design and development section.

### 4.3 Design and Development

#### 4.3.1 Struts

For this year's design, the vertical struts shown above were brought in toward the centre of the hopper and connected to the rocket motor itself. As a result, the structure was much more rigid and no longer required heavy and complex top/bottom plates.

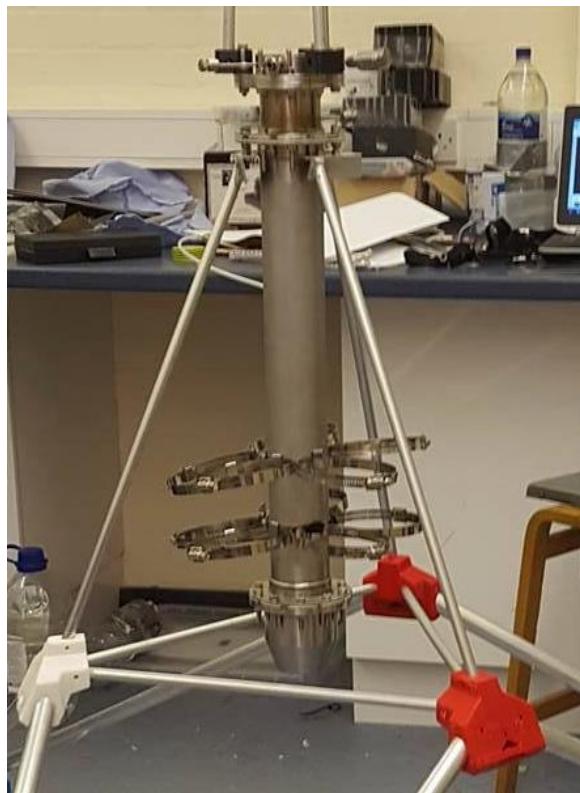


Figure 29: Triangulated Struts

The struts attach to the leg corner pieces to form the base of the chassis structure and to three aluminium metal brackets at one of the rocket motor flanges. The metal for these brackets was sourced from the EDMC scrap bins, so they did not cost the project any money. They were initially cut to be much larger than necessary and then they were trimmed down to reduce weight. Each chassis strut is placed so that it aligns with the ACS struts. The bracket itself is made wide enough to attach to two bolts. By doing so the bracket is constrained from moving rotationally and much more rigid. Because the flange reaches extremely hot temperatures [5], the brackets are attached to two bolts and then offset from the flange itself. This helps the bracket remain cool and, as an added benefit, each of the struts can have its height adjusted with respect to one another. This increases the flexibility of the design and makes it easier to integrate with the leg assembly. The brackets are also able to have their angles independently adjusted to counteract any error caused through the manufacturing process.

#### 4.3.2 Nitrogen Tank Mounts

The old design used the bottom plate to act as a nitrogen tank holder and mount [4]. After manufacturing, the previous team found that the tanks would not be able to sit in their allocated slots. The bottom corner pieces and PDS constrained them so they could no longer fit. That team used worm drive hose clamps to hold some of these tanks to the vertical struts, but they could not do it for all of them and the tanks were not rigidly secure. Many of the tanks were being held in position by the PDS itself, which meant that the PDS was inadvertently becoming part of the hopper structure. A new tank mount would have to be made that would fix all of these problems while fitting within the new frame.

Inspired by the previous team's attempt to use worm drive hose clamps, a mount was built using nothing but hose clamps. This new mount attached all of the nitrogen tanks radially around the rocket motor.

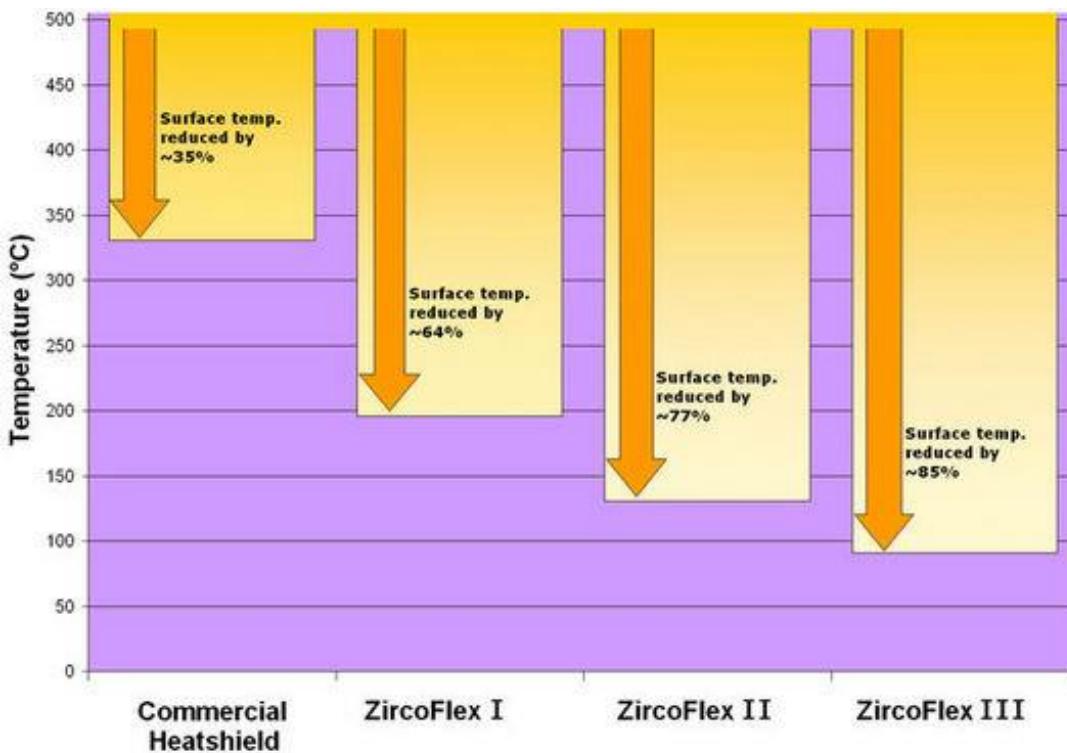


Figure 30: One nitrogen tank attached to the rocket motor



Figure 31: Nitrogen tank mounts

It is made up of two assemblies of six hose clamps each, along with clips that connect each of the clamps to the rocket motor. These clips were manufactured by using material from spare hose clamps. The mount relies on the tank and rocket motor to provide a high level of rigidity while remaining extremely lightweight, low cost, and simple. The mount allows access to any individual tank without affecting the rest of the assembly. The tanks can be rotated, raised, or lowered and still secured rigidly. A thermal protection sheet, ZircoFlex III, was placed between the rocket motor and the mount assemblies to protect the tanks from the heat coming off of the rocket motor. The thermal protection sheet is capable of reducing temperature by 85% and is rated up to 500 degrees Celsius [9].



*Figure 32: Temperature reduction chart for ZircoFlex III [10]*

Based on previous experience, this region's peak external temperature is expected to reach no more than 100 degrees Celsius a few minutes after the rocket motor is shut off [6].

The manufacturing procedure for the nitrogen tank mount clips is detailed below:



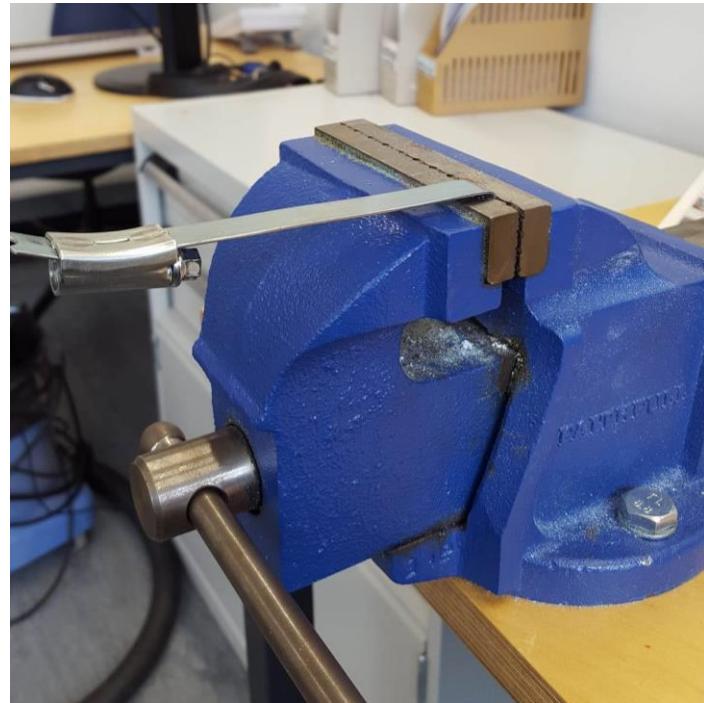
*Figure 33: Measured and marked hose clamp strip*

To begin, one of the hose clamps is flattened into a single strip of stainless steel. Two markings are made that are distanced apart 1.5 times the width of the hose clamps that the clip will eventually hold together.



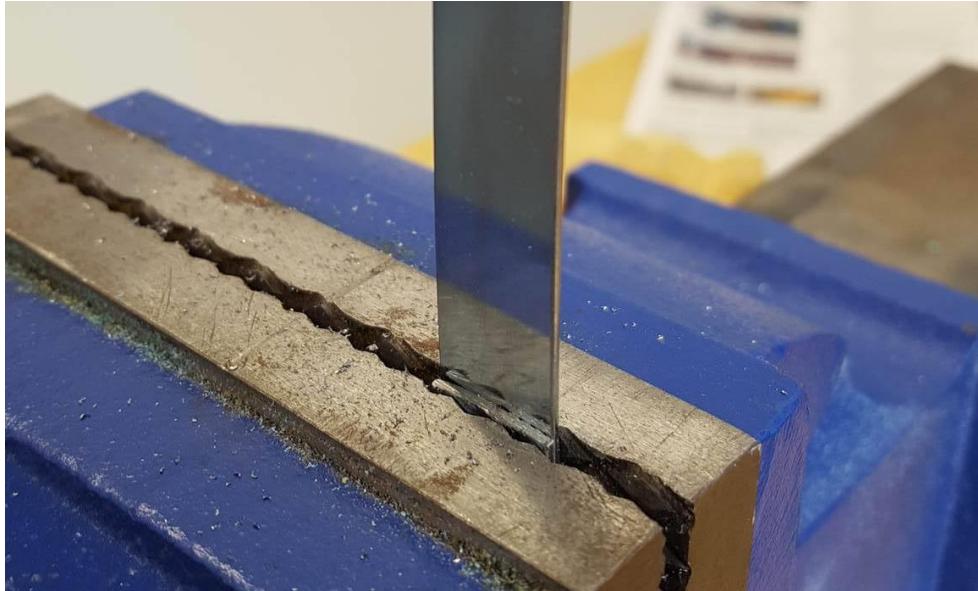
*Figure 34: Hose clamp strip in vice*

Next, the strip is placed in a vice such that the top measurement is aligned with the top surface of the vice. A sheet metal bender may also be used.



*Figure 35: Initial bend applied to hose clamp strip in vice*

The strip is bent as far as the vice or metal bender will allow.



*Figure 36: Completed first bend*

The strip is realigned in the vice or metal bender and compressed down to complete the first bend. The second bend can now be made by repeating the step shown in Figure 35.



*Figure 37: Bent metal clip*

The end result is shown above. It can now be cut with tin snips as shown below and the entire process can be repeated.



*Figure 38: Clip being cut from hose clamp material*

### 4.3.3 Oxidiser Tank Mounts

Similar to the nitrogen tank mount, the oxidiser tank mount attaches directly to the rocket motor. The previous design used the tabs on the tanks as attachment points [4], but the mount designed this year clamps down to the tank at any point. This allows the oxidiser tank placement to be much more flexible. Last year's team used the hopper's top plate to support the oxidiser tanks. With this new design, the top plate can be completely removed.

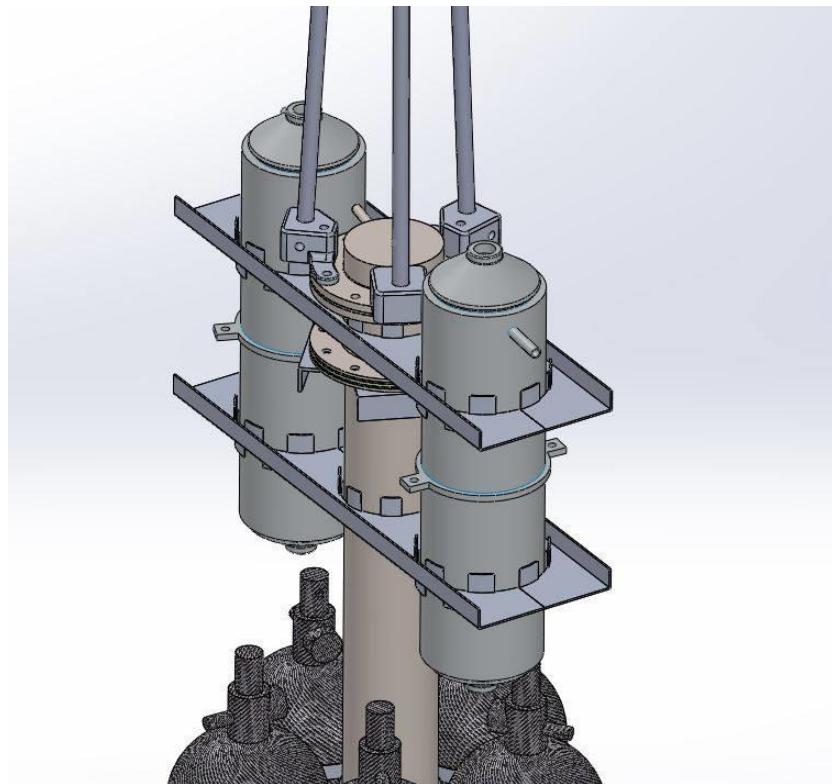


Figure 39: Oxidiser tank mount

Since the bottom part of the oxidiser tank mount can be moved at any point along the rocket engine, it can be aligned with the centre of gravity (CG) of the Lunar Hopper. This is particularly useful for housing the electronic sensors as they are meant to be placed as close to the CG as possible. The oxidiser tank mount has been designed to have enough space between the rocket motor and the oxidiser tanks to house the electronics.

The mount uses four interchangeable parts, two of which are shown in Figure 39. It clamps on to the rocket motor and tanks using twelve worm drive hose clamps, the same kind used on the nitrogen tank mount. Six are placed on each assembly, three around the tabs on the top and three around the tabs on the bottom. The final design is strong, rigid, and extremely lightweight. It was made using 1.5 mm thick 5052-H32 aluminium. The top mount will be attached to the outer part of the catalyst chamber, which is expected to reach about 500 degrees Celsius within seconds [6]. 5052-H32 aluminium has a melting point significantly above that, between 607 and 649 degrees Celsius [11]. Yet this still caused some concern, as the material properties of the aluminium may be significantly reduced in such a high temperature environment [12]. The team considered manufacturing the top mount out of steel, but during this time the Hopper was nearing its weight limit and reducing mass was a very high priority. Consequently the team chose to go with the aluminium design. Heat is applied for a short duration of time and the oxidiser tank mount has a large surface area that will radiate the heat away. If issues with the top mount arise, the bottom mount will still hold the oxidiser tanks upright and ensure that there is not a catastrophic failure. If the top mount proves to be unreliable, future teams should investigate making it out of steel.

An FEA analysis of the piece's structure showed that the section between the rocket motor and mount's edge is most susceptible to failure. To reinforce this area, the sides of the piece are bent upward. This increases the second moment of area substantially and made the entire assembly much more rigid without increasing its mass. Throughout the entire flight operation the oxidiser tank mounts have little to no displacement. The entire piece is then cut in half to give the assembly the ability to fit around the tanks and rocket motor without sliding on from the top or bottom. The FEA analysis for the static loading case is shown below:

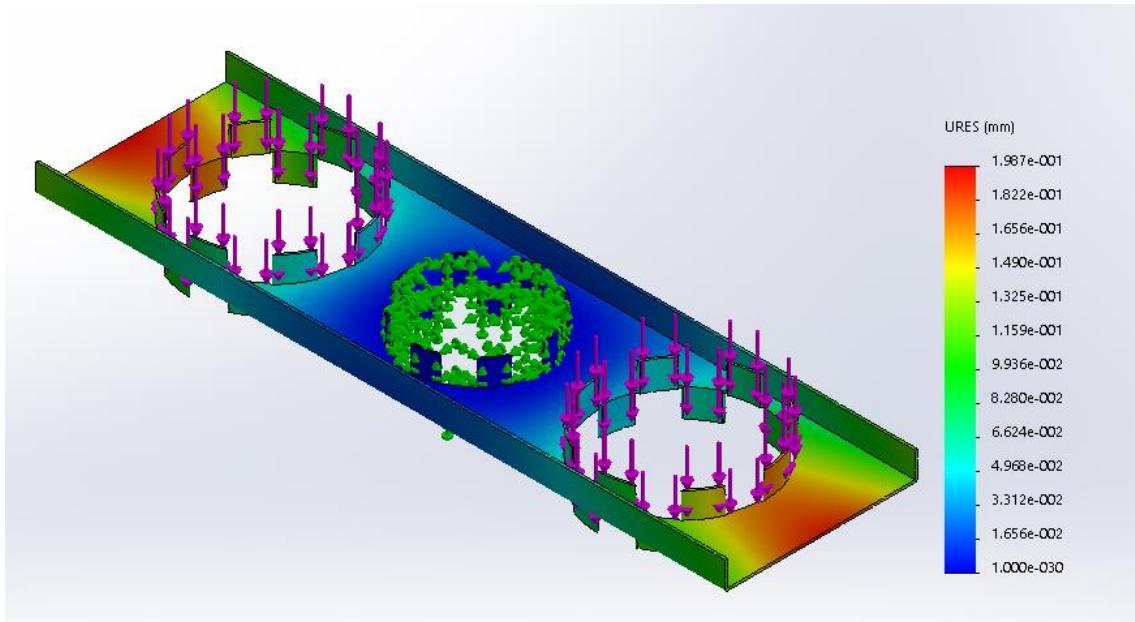


Figure 40: FEA static displacement for oxidiser tank mounts

There are two of these fully assembled mounts on the hopper, so each one will have to support approximately half of the weight of the oxidiser tanks and oxidiser. This was estimated to be approximately 36 N of static force. Figure 40 shows that the mount will displace less than 0.2 mm under these conditions. The greatest displacement occurs in the red regions, while the least amount of displacement takes place in the blue areas.

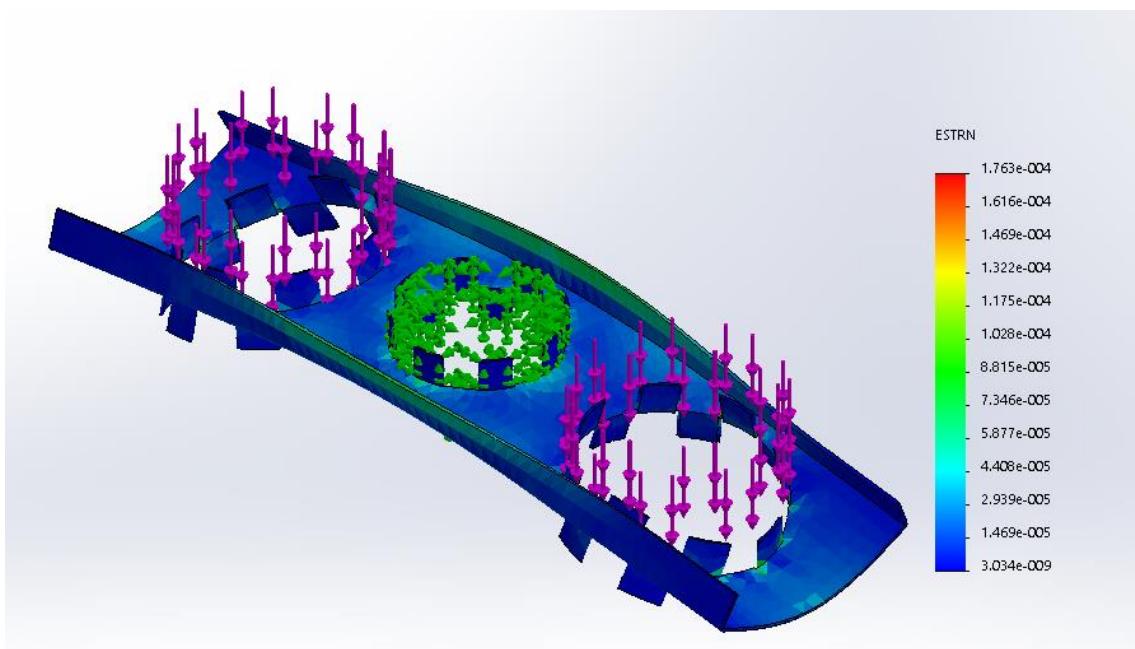
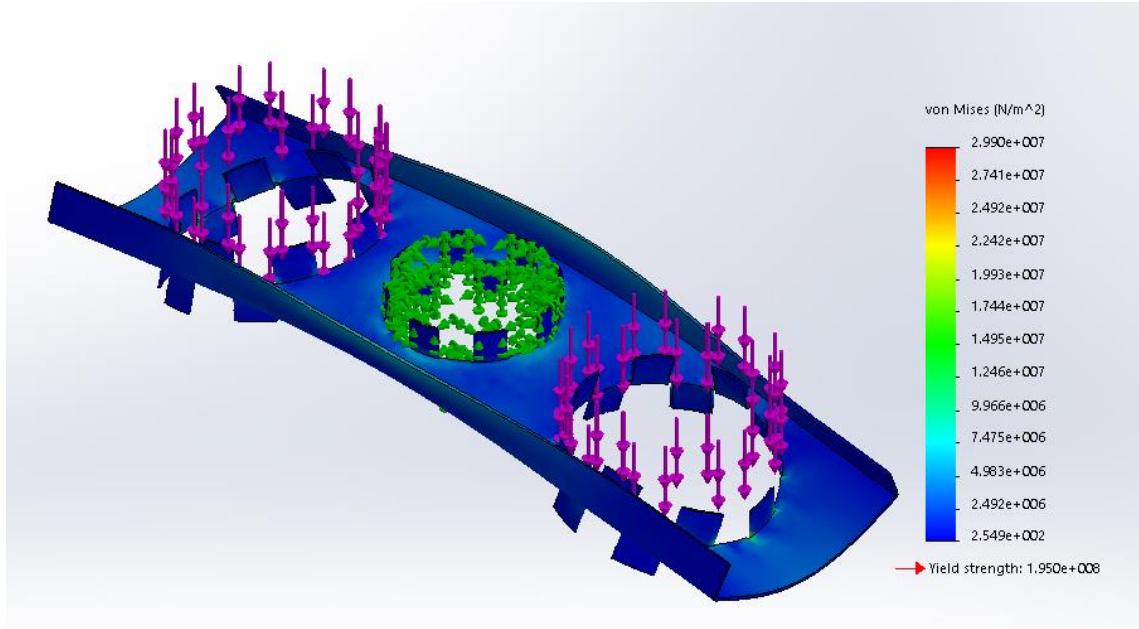


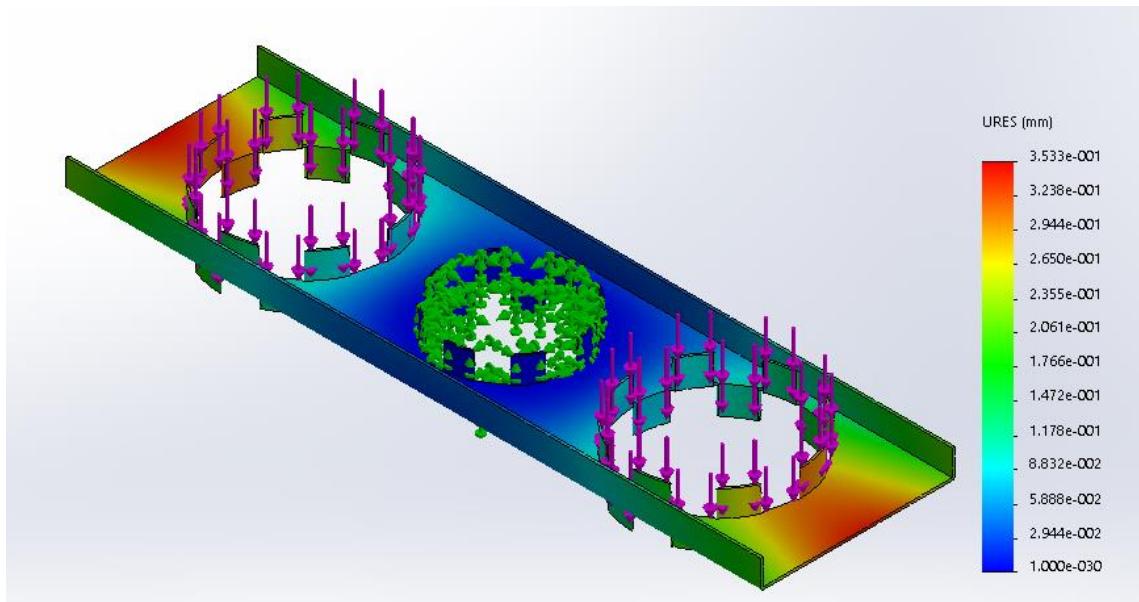
Figure 41: FEA static strain for oxidiser tank mounts

Figure 41 shows which areas of the oxidiser tank mount will have to endure the most stress. The analysis showed that the middle of the side walls around the rocket motor are under the highest amount of stress. The assembly was initially cut in half crosswise through just the rocket motor, but this split the side walls in half. To reinforce this area, it was made continuous by splitting the assembly lengthwise instead. The height of the side wall was increased as well.

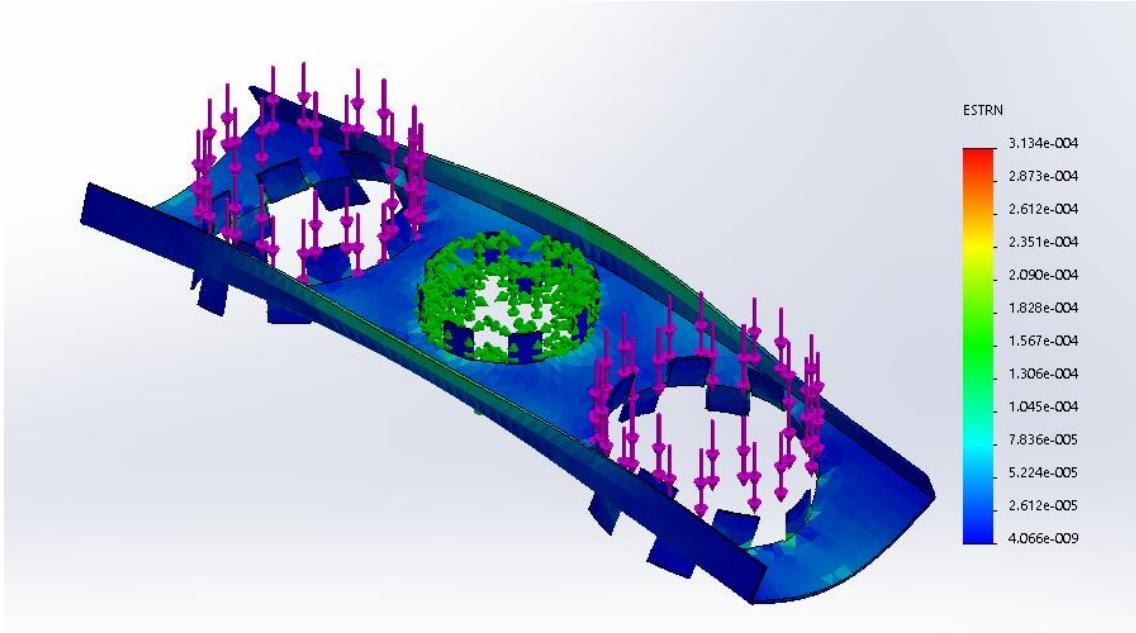


*Figure 42: FEA static stress for oxidiser tank mounts*

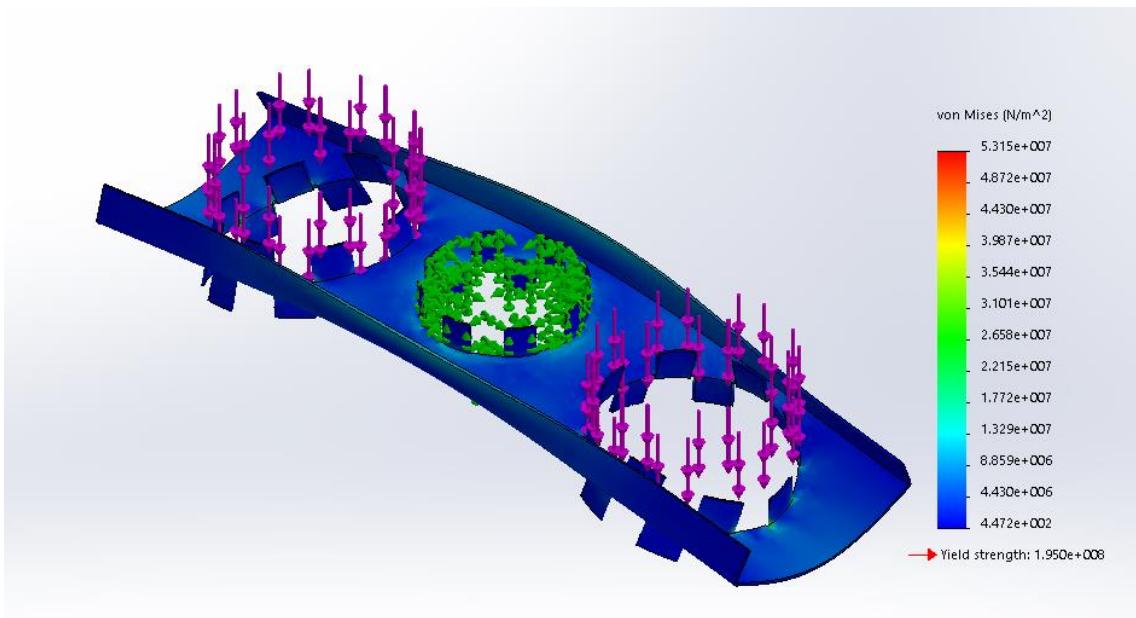
The FEA for the flight phase assumes that the maximum acceleration of the hopper will be  $1 \text{ m/s}^2$ . The final target acceleration will be derived from running the pilot trainer, but until then the team agreed that it will be less than  $1 \text{ m/s}^2$ . The flight phase FEA includes the force from the Hopper's acceleration, and it yields similar results to the static case. The results can be found below:



*Figure 43: FEA for displacement of oxidiser tank mounts during flight*



*Figure 44: FEA for strain experienced by oxidiser tank mounts during flight*



*Figure 45: FEA for stress experienced by oxidiser tank mounts during flight*

The oxidiser tank mount is potentially one of the more complex components on the hopper chassis. Consequently, it can be expensive if it is manufactured externally. This year's hopper team was quite small and heavily time constrained, so the decision was made to get it manufactured. Adam Studebaker collaborated with the workshop that made it to make sure it was done in such a way that it could be replicated by future years. Below are detailed manufacturing instructions so it can be built at the University using tools found at the Student Workshop.



Figure 46: 1.5mm 5032-H32 aluminium, mill finish

Start by shearing the starting piece of sheet metal to the correct size, 405 x 140mm. Then bend two long edges 90 degrees to form the 15mm tall sides.

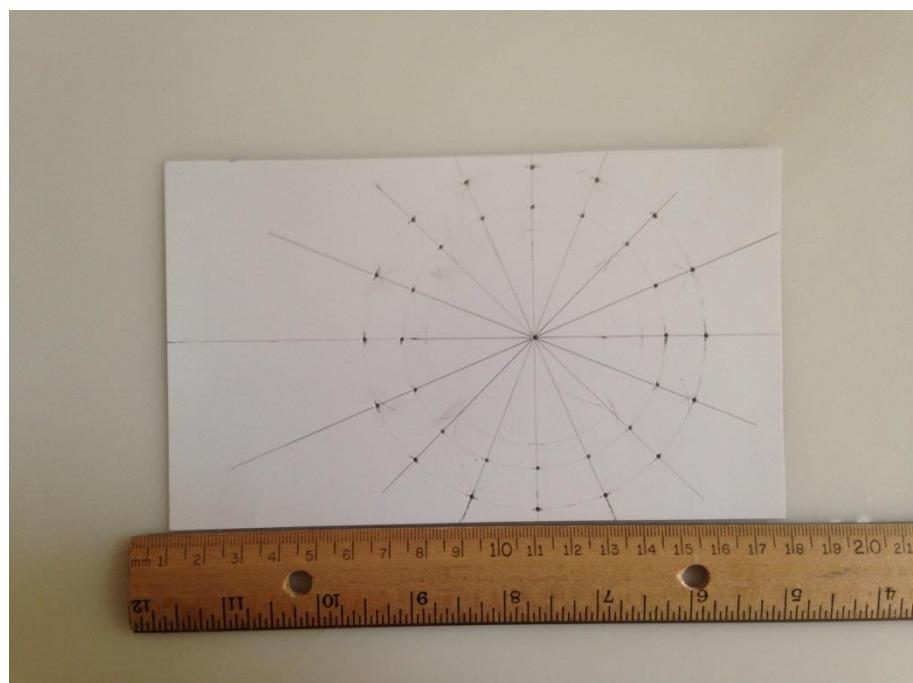


Figure 47: Template for rocket motor and oxidiser tank holes

Using a template or some other form of measurement, mark and punch out the inside diameters of each of the three main holes. Note that this does not include the tabs.

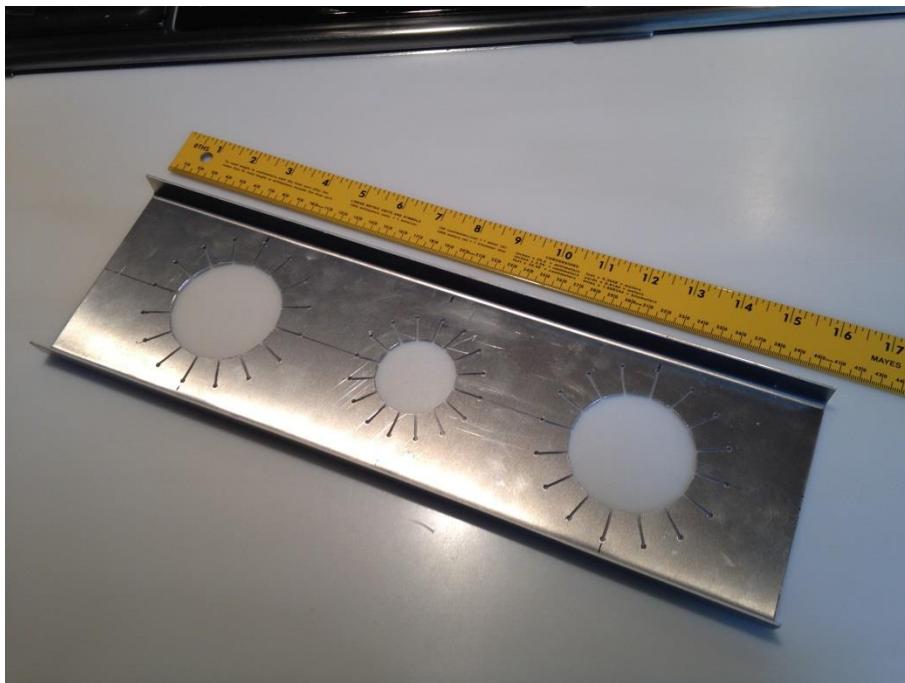


Figure 48: Sheet after making all holes and cuts

Drill 48 2.4mm holes. There are 16 of them equally spaced around each of the 95mm tank diameters and around the 75mm rocket diameter. Saw through from the large centre holes to the small 2.4mm holes. There are 48 cuts in total that can be done with a jigsaw or a hacksaw.

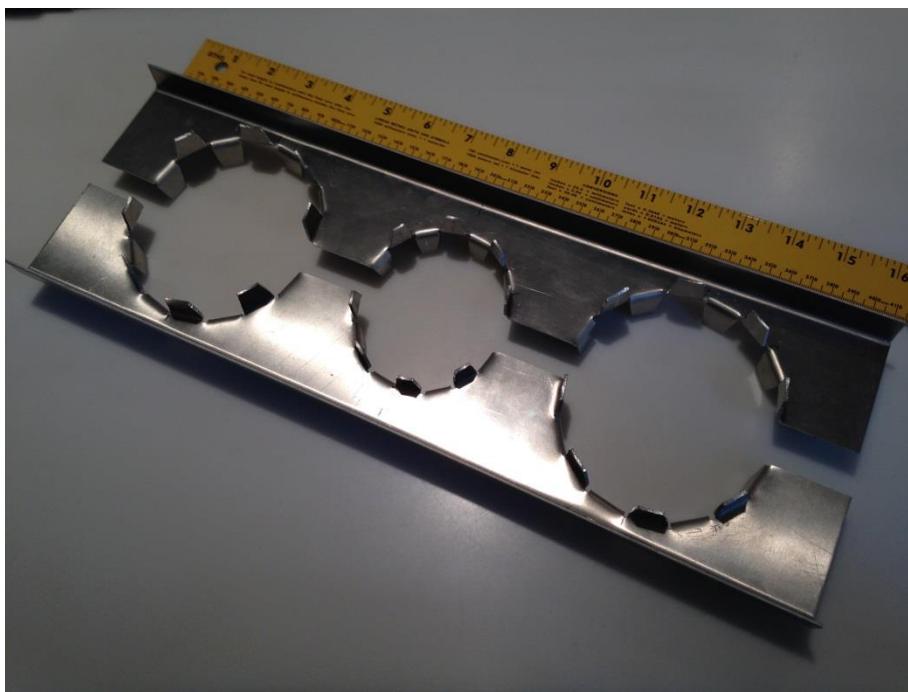


Figure 49: Finished oxidiser tank mount

After completing all 48 cuts, the entire sheet can be cut in half along its centreline lengthwise. The tabs can then be bent 90 degrees along the 2.4mm holes' centrelines alternatively above and below the plate. The tabs will naturally want to bend along the radius of the holes at the base of the tab. They can be bent with a pair of lineman's pliers like the ones shown below:



Figure 50: Lineman's pliers used to bend tabs

#### 4.4 Summary

The chassis for the lunar hopper has been built from a completely new design that eliminates the need for a top or bottom plate or vertical bars. The new design uses triangulation and radial mounts to secure all of the hopper components rigidly. It is lightweight, low cost, simple, and adaptable. All of the structural requirements listed in Section 4.1 were met. They are restated and discussed below:

1. Centre of mass must align with the thrust axis of the rocket engine

The new chassis uses radial mounts and equilateral struts to maintain symmetry throughout the design. This ensures that the centre of mass remains aligned with the thrust axis.

2. Structure must secure all tanks rigidly

The chassis design mounts all of the tanks to the rocket motor itself and maintains rigidity by securing everything with worm drive hose clamps. The team focused on ruggedizing the design, ensuring that the final design is reliable and completely rigid. .

3. Structure must support the attitude control system (ACS)

The ACS system's structure was completely redesigned and tested to ensure that it will remain structurally supported throughout the Hopper's flight.

4. Structure must house electronic sensors

The bottom oxidiser tank mount can be attached at any point along the rocket motor, so it can be placed at the centre of gravity (CG) and house the electronic sensors. There is room on the mount for everything that needs to be mounted.

## 5 Landing System

Author: Alison Dufresne

The landing system has been iteratively developed throughout the Lunar Hopper project. The original design from Phase I was designed for ideal flight, in which case the Mk I would gently glide to ground with minimal shock experienced where it would be allowed to skid to a halt. It used a simple design which supported the Mk I effectively but did not include any shock absorption (see Figure 51).



Figure 51: Phase I leg design [3]

Phase II accounted for shock on landing by incorporating shock absorbing foam pieces and metal plates to distribute the force in case of uneven landing. These foam pieces were validated for shock absorption [3] but not fully integrated into the Mk I. Phase III integrated the landing foam and metal bowl (see Figure 52). The Mk II design developed during Phase IV is assessed and evaluated in Section 5.2



Figure 52: Phase III Mk I landing mechanism

### 5.1 Structural Requirements

The applicable structural design requirements from system are as follows:

1. Centre of mass must align with the thrust axis of the rocket engine
  - a. Mission critical as ACS is not designed to rectify the deviation from vertical of the Hopper which would occur due to this off-balance
2. Must support >40 kg static load (391 N) when integrated with chassis, without excessive deformation
  - a. Mission critical that static deformation does not compromise the structural integrity of the landing mechanism
3. Must absorb shock load from a 1 m drop (~392 Joules)

- a. Secondary requirement to ensure the re-usability of the structure even in worst case which is defined as a 1 m uncontrolled free-fall
- 4. Must not exceed the weight of the previous design (~6.2 kg)
  - a. This requirement is derived from the allocation of mass budgets which are mission critical if not adhered to
- 5. Must keep centre of gravity low while maintaining the engine nozzle >0.5 m from the ground structure
  - a. Nozzle height above ground is derived by the maximum expected exhaust gas which would damage the structure
- 6. Manufacturing tolerances will be no greater than  $\pm 0.25$  mm for aluminium components
- 7. 3D printing part tolerances should be equal to  $\pm$  the resolution setting chosen on the printer

## 5.2 Phase IV Design Review

The Mk II leg and landing mechanism design (see Figure 53) developed by Phase IV utilised the landing pads developed in previous phases with an addition of a hinge mechanism (see Figure 54). This hinge mechanism connected the landing legs to the chassis structure and was designed to deflect on impact to absorb the shock. The design relied on hinge connections which were supposed to allow the legs to deflect at the corner of the chassis on impact, in order to absorb the force.



Figure 53: Mk II original leg design [4]

The inherited landing system on the Mk II consisted of the following ninety one components;

- Leg bars (6)
- ‘Hinge Link’ (12)
- Hinge Bar (12)
- Torsional Springs (12)
- Metal Bowls (3)
- Foam pieces (3)
- PLA leg connection piece (3)
- ABS corner pieces (3)
- ‘Hinge Adapter’ (12)
- Horizontal corner connecting bars (3)
- Nuts and bolts (24)
- Steel wire (1)

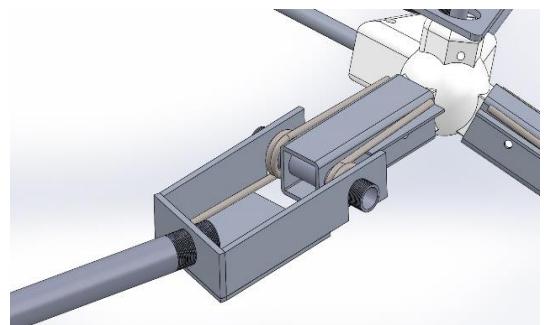


Figure 54: Solidworks Illustration of the Mk II leg ‘Hinge’

The landing mechanism was in a disassembled state at the beginning of Phase V. The system had been highlighted as a system which had serious inherited issue. The image of the Mk II from the day of attempted flight (see Figure 55) shows that the system was partially functional,

but not as designed. The system relied on steel wire, which was looped through a screw on each of the three landing pads, which forced the leg bars to attach to the chassis without engaging the springs.



*Figure 55: Mk II Landing system 'fix' on the day of launch*

The state of the inherited landing system made it difficult to assess. To begin, an attempt at assembly was made. The landing pads and leg bars connected easily but the hinge link immediately presented issues. The first issue was due to manufacturing and tolerance. This part, shown in Figure 56 was used to connect the leg bar to the springs and corner piece of the chassis. Due to extremely low tolerances ( $>1$  mm), the part did not constrain the springs or the leg bars effectively.



*Figure 56: Original 'hinge link' [4]*

In order to adequately assess the performance of the design itself, it was necessary to redesign and manufacture an alternative to this piece. The technical drawing of the new design is shown in detail in Appendix D. Figure 57 the simplicity of the new design (right), which required only three drilled angled holes, when compared to the previous version (left).

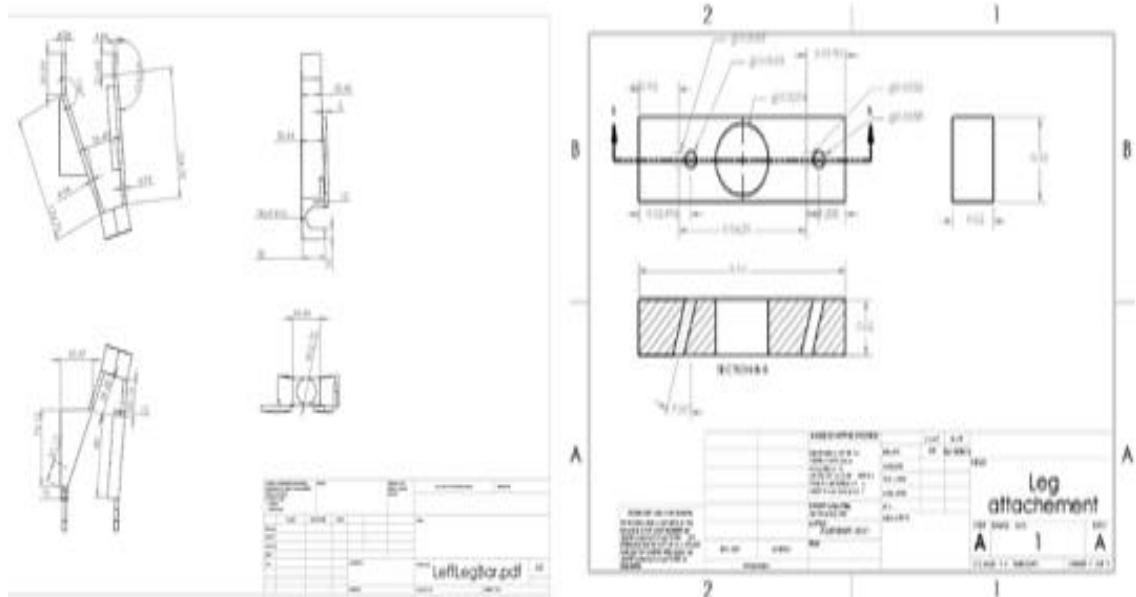


Figure 57: (Left) Previous part drawing for leg attachment piece. (Right) Phase V redesign of leg bar attachment piece

Once a prototype of this part was manufactured, assembly and testing of the hinge system could be carried out.

### 5.2.1 Testing

Utilising the newly manufactured prototype piece, spring testing was carried out. For a more realistic and comprehensive test the load was applied at the base of the leg (where it connected to the landing pads) with the springs constrained effectively. The force applied and deflection angles were measured and plotted against each other (see Figure 58). Failure was defined as leg bars deflecting to horizontal position, which for the Phase IV project equated to a deflection 20 degrees. As can be seen, very little force was applied before full system failure occurred.

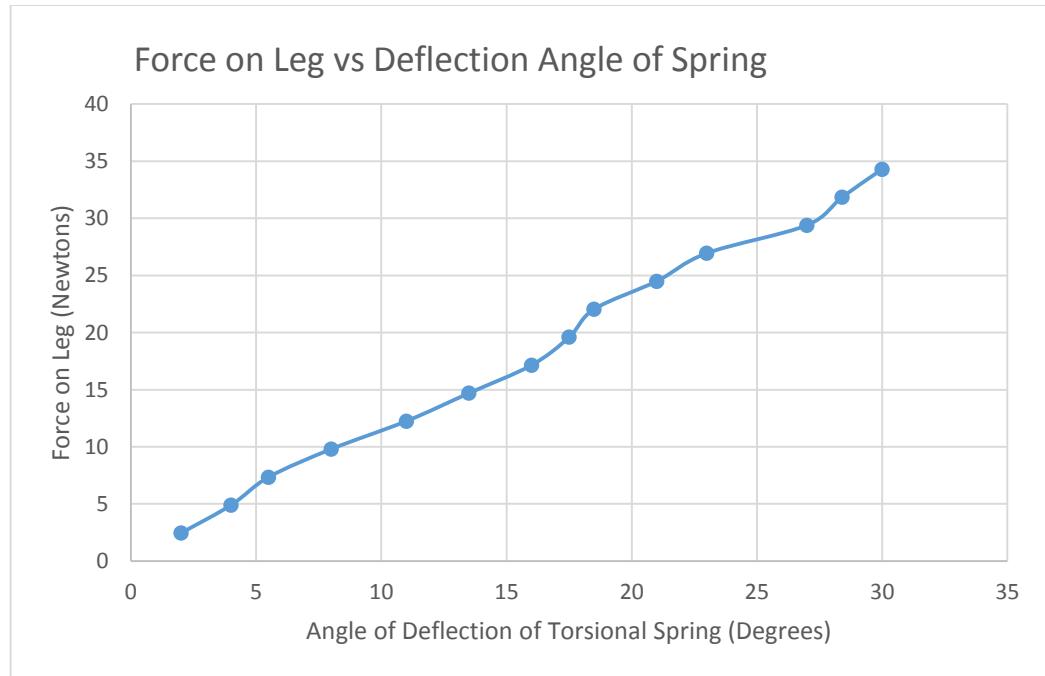


Figure 58: Spring arm deflection angle against load applied

The springs deflected 20 degrees when under 25N of force which is less than 40% of the expected static load of 65 N they would be subject to (See next section for validation).

### 5.2.2 Mathematical Analysis

For the Mk II design to function it must support the static load of the Lunar Hopper while keeping the nozzle a safe distance off of the ground. Previous design intended to maintain a distance of 0.46 m between the nozzle and ground. The inclination of the leg bars was 20 degrees from horizontal. For this design to work, the spring must not deflect more than a few degrees for static force and less than 20 degrees for hard landing. First this was analysed mathematically using simple beam theory using the variables and equations below. The results are contained in Table 6.

Total static force:

$$F_{total} = mg$$

Static force per landing pad:

$$F_{landingpad} = \frac{F_{total}}{3}$$

Static force per hinge link:

$$F_{hinge} = \frac{F_{landingpad}}{2}$$

Static force per spring:

$$F_{spring} = \frac{F_{hinge}}{2}$$

Minimum static torque per spring:

$$M_{static} = \frac{F_{spring}}{4}x$$

Maximum torque (hard landing case, where all force on one landing pad which activates four springs):

$$M_{hardlanding} = F_{spring}x$$

The spring rate (the most common measure of a torsional spring performance):

$$k = \frac{M_{static}}{\theta_s}$$

$$k = \frac{M_{hardlanding}}{\theta_m}$$

Where:

- Leg Length ( $x$ ) = 0.9 m
- Mass of Hopper ( $m$ ) = 40 kg
- Acceleration due to gravity ( $g$ ) =  $9.8 \frac{m}{s^2}$
- Failure deflection ( $\theta_f$ ) =  $20^\circ$
- Ideal Static deflection ( $\theta_s$ ) =  $5^\circ$
- Max allowable deflection angle ( $\theta_m$ ) =  $15^\circ$

Table 6: Results of preliminary analysis for spring moment

| Results                  |              |
|--------------------------|--------------|
| $F_t$                    | 392 N        |
| $F_{leg}$                | 130 N        |
| $F_{hinge}$              | 65 N         |
| $F_{spring}$             | 32.5 N       |
| $M_{static}$             | 29,250 NM    |
| $M_{hard landing}$       | 87,220 Nmm   |
| <b>Ideal spring rate</b> | 5880 Nmm/Deg |

Spring sizing during Phase IV relied on the energy equation, which would have been applicable for non-torsional springs but was not for the chosen torsional springs, as it doesn't consider the moment arm which the force is applied through. The torsional springs which were procured in Phase IV had a max torque of 12,536 Nmm (just over 42.9% of predicted static load and less than 15% of 'hard landing' load).

### 5.3 Design and Development

Section 5.2 outlines the issues with the Mk II landing system. This section discusses the design and development of the new system. To begin with, alterations/alternatives were discussed. The following ideas were explored;

- Replacement torsional springs
- Reduced moment arm (leg length)
- Addition of a weight bearing 'stopper' to prevent over deflection of inherited springs
- Pre-loading the spring
- Replacing leg 'hinge' design with integrated standard springs (non-torsional).

#### 5.3.1 Option Evaluation

Replacement of the springs with adequate torsional springs was explored due to its simplicity of integration with the current design. The company which the previous group procured springs from only supply springs with max torque up to 30,000 Nmm and other suppliers seem limited at 40,000 Nmm. This was prohibitively low with the minimum (static) torque being 29,259 Nmm and the hard landing case being 87,220 Nmm. After contacting several companies it became clear the torsional springs of the required stiffness would either be completely different dimensions (square springs being the only design found off-the-shelf capable of withstanding the torque) or specially manufactured to provide the required stiffness. Using springs of a different size or configuration would require a complete redesign of the corner pieces and hinge link and specially manufactured springs proved to be out of our budget (considering that a minimum of six would be required for the an updated version of this design to function). The additional weight was also considered which led to the decision to scrap the hinged leg design completely.

Pre-loading the springs was considered before spring testing was carried out and was based on the assumptions that the Phase IV spring sizing was only slightly miscalculated and assuming the issue was caused by the loose hinge link or shallow inclination from horizontal.

Reducing the moment arm was scrapped after a small amount of mathematical analysis (re-using the equations in section 5.2.2). It was calculated that, with  $\frac{1}{2}$  the moment arm ( $x = 0.45$  m), the torque would still be too high (14,625 Nmm) to consider using the inherited springs (which have a max torque of 12,536 Nmm). This method was also considered to incorporate off-the-shelf springs but, as these springs have different dimensions, it would require a redesign and manufacture of the ‘hinge link’ components (of which there are three) and, to prevent the ground clearance reducing to unacceptable levels, the bottom leg-to-landing-pad attachment piece (which inclines the leg bars 20 degrees from horizontal).

The addition of a stopper was also considered but only briefly. The design would utilise a stopper component which would prevent the springs deflecting past 20 degrees on static loading and incorporate shock absorption for landing. This concept was good in terms of recycling the old components but again, was only considered prior to the realisation that the inherited springs were inadequate.

The chosen redesign concept was to replace the torsional springs with simple compression springs located in the landing pads. Compression springs were considered and favoured due to the simplicity of design, low cost and ease of procurement. This would increase the stress experienced at the leg-to-corner piece attachment but absorb the shock with ease. Though this is similar design work as the replacement springs would incur, it is lighter, cheaper and less complex than the previous system. For this design to work, the corner piece and components of the landing pads would have to be redesigned.

### 5.3.2 Spring Sizing

To verify the use of compression springs, mathematical analysis was as follows;

$$F = kx$$

$$K.E. = \frac{1}{2}m v^2 = P.E. = mgh$$

By estimating the rough desirable size of the spring deflection for static loading ( $x$ ) and hard landing cases, the appropriate spring stiffness ( $k$ ) can be found (see Table 7). As one of the requirements is for the landing mechanism to absorb the shock from a 1 m uncontrolled drop, the springs were sized for ‘worst case’ in which only one foot would hit the ground first and therefore must not fail when  $\sim 391$  N are shock loaded. The spring dimensions were important for integration into the foot design and for spring sizing. Acceptable parameters were generated. Table 8 shows the desired attributes vs the chosen spring.

*Table 7: Results of spring*

|                           | Static Conditions | Hard Landing Conditions |
|---------------------------|-------------------|-------------------------|
| Force (N)                 | 65                | 392                     |
| Allowable Deflection (mm) | 5 - 10            | 25 - 30                 |
| Spring Constant (N/mm)    | 6.5 - 13          | 11.2 - 13.07            |

Table 8: Spring dimensions [13]

|                  |                    |
|------------------|--------------------|
| Spring Type      | Compression Spring |
| Outside Diameter | 50.8 mm            |
| Free Length      | 76.2 mm            |
| Wire Diameter    | 4.88 mm            |
| No of Coils      | 7                  |
| Max Solid Length | 36.58 mm           |
| Rate             | 11.65 N/mm         |
| Max Safe Load    | 458.14 N           |

Once the compression spring design was chosen, it was evident that the entire corner connection, between the legs and chassis, had to be redesigned and could be vastly simplified. Instead of eight components at this point it was decided to simplify to one. One piece was to be designed to connect the leg bars to the chassis. Two pieces were designed to enclose the spring and ensure even load distribution throughout.

### 5.3.3 Finite Element Analysis

A process of design, validate, verify was adopted. The validation was achieved through Solidworks FEA simulations. As the strength of the corner piece is integral to the functionality of the legs, extensive finite element analysis (FEA) was conducted to compare the designs and choose the ones which would be prototyped.

The analysis was carried out using the forces and moments calculated in Section 5.2.2. The static force was set to 130 N acting on the angled hole on top with 65 N acting vertically on the angled holes for each leg bars. The results of the static loading on the final corner piece design are shown below. The design did not include the metal bolts.

The first FEA study was done on version 1 of the aluminium corner piece. It was quickly realised the aluminium corner piece would certainly withstand static and hard landing loads without significant deformation as the leg bars and chassis struts, due to their geometry and the direction of the force applied, are structurally weaker.

FEA was important in the comparison of 3D printed pieces with different geometries. The results of the FEA for the final part is shown in Figure 59, Figure 60 and Figure 61. The deformation of the part is less than 0.0003 mm and 0.003 mm for hard landing (well within acceptable amounts).

Model name:Corner Piece V12 SciFi  
Study name: Static1 [-Default-]  
Plot type: Static displacement Displacement1  
Deformation scale: 4083.15

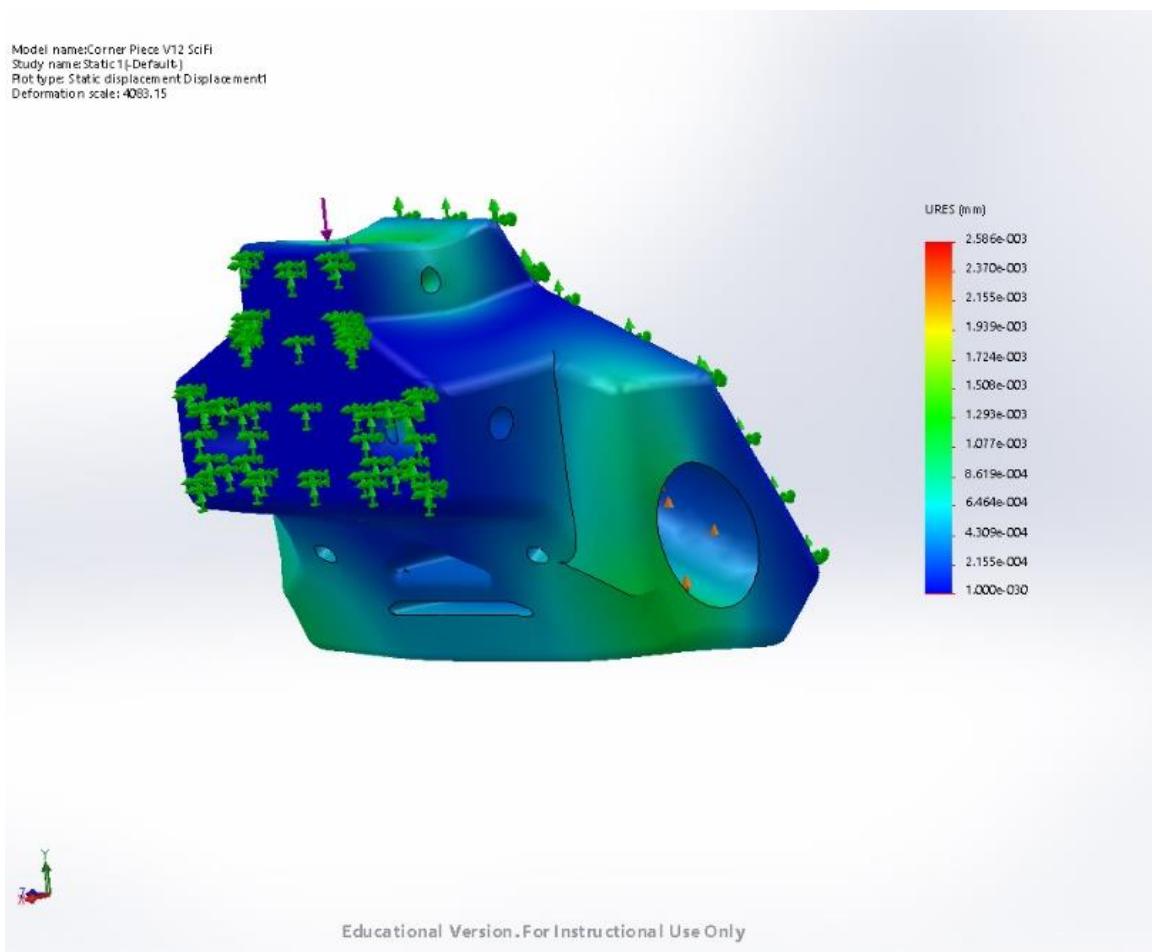


Figure 59: Displacement incurred from static force

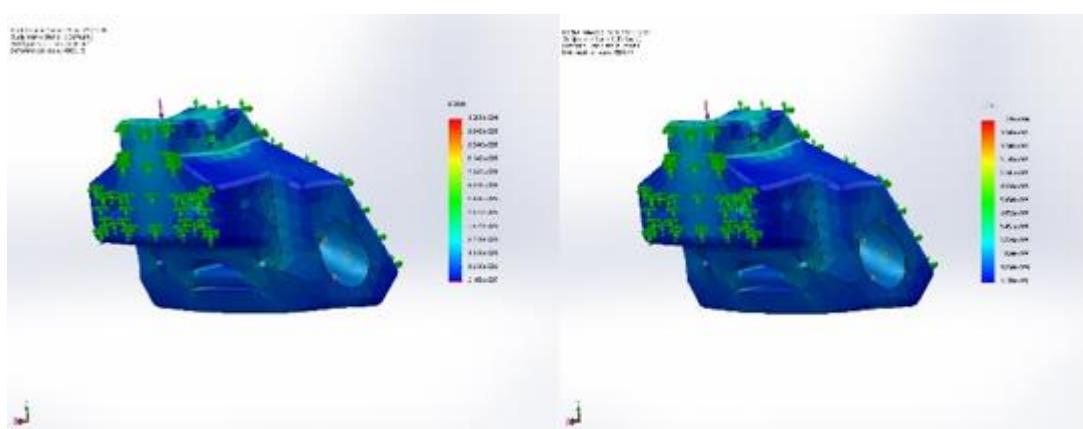


Figure 60: Stress (right) and strain (left) in corner piece from static loading

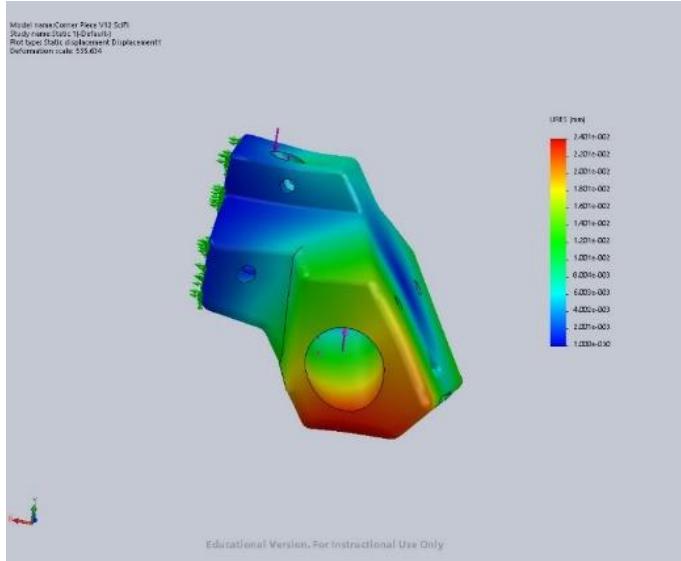


Figure 61: Hard landing deformation of final piece

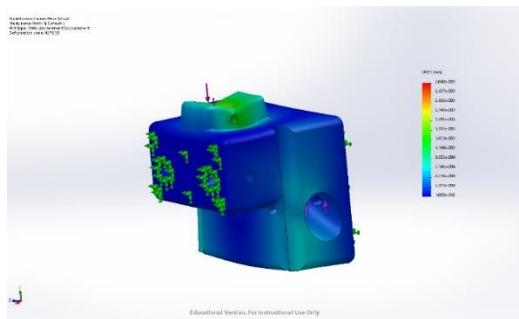


Figure 62: Static displacement for unchosen piece

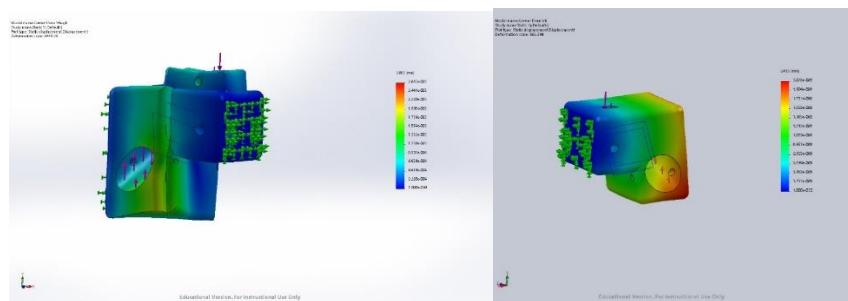


Figure 63: Hard Landing displacement for hard landing case

As can be seen from the FEA section, the component design went through many iterations with various large and small variations. For significant design changes, a new version was saved. The corner piece went through 12 significant changes, the landing pad attachment piece went through 4 and the spring casing 2.

### 5.3.4 Material Selection

The material selection process for the new parts was a relatively simple process. Taking into consideration the research done throughout the project and in particular, the recommendations of Phase IV ([4]), the options for material selection were quickly narrowed down to aluminium 6061 or ABS plastic. The relative pros of each are compared in Table 9.

Table 9: Pros of possibly materials

| <b>Aluminium (Pros)</b>                               | <b>ABS (Pros)</b>   |
|---|---|
| Stronger in tensile, compression and shear conditions | 3D printable allowing for vastly more complex shapes to be easily manufacture |
| Compatible material with leg bar allowing for welding | Light weight  |
| Relative to metals, easy to machine and manufacture   | Easily available  |

An important factor when considering the material of the corner piece, is the amount of stress which will occur at this point. To assess this, the compressive and axial forces at the end of the aluminium leg bar were calculated using the following variables and equations;

Variables:

- $r = 12.7 \text{ mm}$
- $t = 1 \text{ mm}$
- $x = 0.9 \text{ m}$
- $\theta = 20^\circ$
- $F_{static} = 392 \text{ N}$

Axial Compressive Force:

$$F_{axial} = F_{static} \cos(\theta) = 392 \cos(20) = 368.36 \text{ N}$$

Lateral Force:

$$F_{lateral} = F_{static} \sin(\theta) = 134.071$$

Compressive Axial Stress

$$\sigma_{lateral} = \frac{F_{lateral}xr}{\pi r^3 t}$$

Lateral stress at corner piece

$$\sigma_{axial} = \frac{F_{axial}}{2\pi r t}$$

Total stress

$$\sigma_{total} = \sigma_{lateral} + \sigma_{axial}$$

Table 10: Results of stress analysis

| Results                   | Static (MPa) | Hard Landing (MPa) |
|---------------------------|--------------|--------------------|
| <b>Compressive Stress</b> | 47.36        | 284.13             |
| <b>Lateral Stress</b>     | 105.94       | 635.64             |
| <b>Total Stress</b>       | 153.29       | 919.78             |

The dimensions of the corner piece design can be guided by the experienced stress in this leg bar. As the corner piece is designed to somewhat seamlessly integrate the leg into the corner, the stress will be along these lines.

Two designs were developed (see Figure 64), one design for aluminium (with manufacturability in mind) and one for ABS. Both of these designs went through an iterative design process which involved significant amounts of FEA.



Figure 64: (left) Aluminium version (right) ABS version of corner piece

While both designs were pursued for some time, once prototype testing had occurred, ABS was chosen as the corner piece material. The driving factor in this selection was the material weight as the aluminium version of the part weighed roughly ten times as much as the 3D printed version.

### 5.3.5 Manufacturing

The majority of the manufacturing was conducted at the University's 3D printing lab, with the high quality UP! printers. Prototypes were printed at Makerspace Southampton which provided significantly cheaper printing albeit at the expense of quality. The ability to print cheap and quick prototypes was invaluable for developing complex pieces which require high accuracy for assembly. The aluminium piece was manufactured at the University's EDMC using the mill.

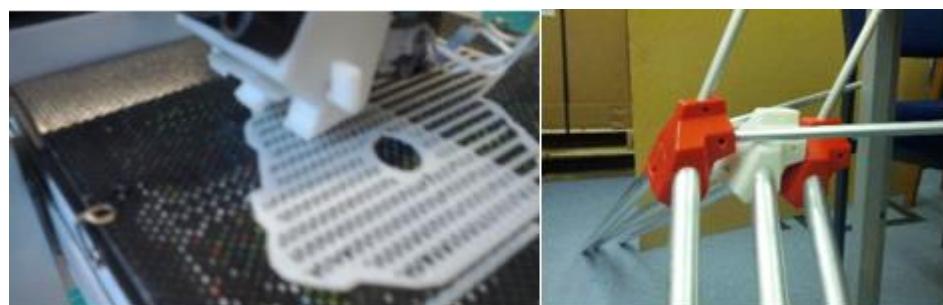


Figure 65: 3D printing and prototype pieces

### 5.3.5.1 3D Printed Piece Reinforcement

After accidentally breaking an important structural piece (during assembly and not operational conditions) the idea of re-enforcing the material came about (as well as the idea to put together a straight forward ‘assembly manual’ to pass down to the Phase VI team). A 3D printed part strength depends on the printer quality, settings and material. Print layer orientation is also very important and needs to be considered thoroughly in terms of the direction of loads applied.

The printed parts were treated with acetone but some layer separation was impossible to repair with this method. As it was shown with a proto-type piece (printed at a mere 20% fill) a strong surface layer can prevent damage in the part and prevent failure through layer separation. Research indicated that a cheap and effective way to increase the strength of a 3D printed part is to use casting resin to bind layers and create a strong outer surface layer. There are many types of casting resin. Polyurethane and polyester are the most common resin bases. These are both applicable, with polyurethane having more desirable physical properties [14]. Polyester-based clear casting resin was selected, however, due to the ease of availability, substantially lower cost and greater amount of flexibility with the sizes and shapes being cast. As the team did not have much previous casting experience, the ease of application which polyester offers was the main driving factor for its selection. [15]



Figure 66: Separation in layers of a prototype ABS corner piece



Figure 67: Post casting of corner piece

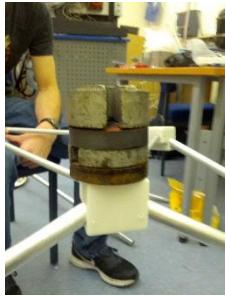
## 5.4 Testing

Preliminary testing was carried out on prototype pieces, none of which had been treated with casting resin at the time. Two tests were conducted to validate static loading conditions. The first test was conducted by constraining the corner piece in a vice as load was applied to the landing leg which was connected without a bolt (which will not be as strong as bolted design discussed in Section 5.3.3). The load was applied until a crack was initiated. The parts showed no failure at 65.3 N but crack initiation began at 98 N and 78 N for the prototype part. The latter being a part which already had layer separation pre-testing due to low print quality. These results were worryingly low, and consideration of the increased performance from appropriate bolts was not sufficient for validation of hard landing survival. Unfortunately, at the time of writing



Figure 68: Illustration of preliminary corner piece testing

this the prototype casting is curing and therefore the improvement cannot be tested.

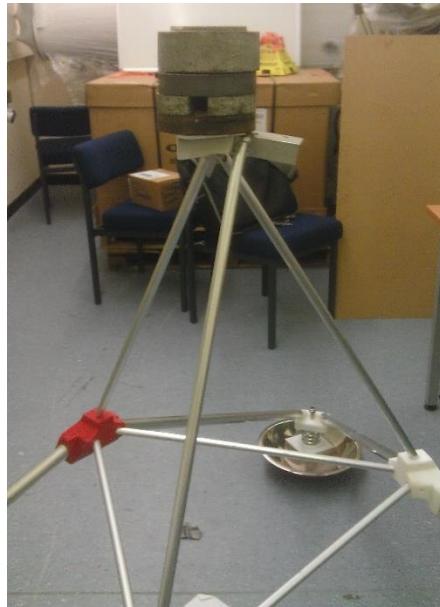


*Figure 69: Testing of corner piece prototype*

These were tested with weights of 1 kg, 2 kg, 5 kg and 38 kg. The load was first applied over the upper surface (see Figure 68) with the landing pads attached and resting on the ground. The intention was to apply force until the prototypes broke, which was considered likely to occur prior to operation loading (13 kg), as the prototypes were printed at a mere 20% volume and therefore expected to have a fraction of the strength of the final piece. Fortunately, this did not occur for any prototype or final piece tested

in this manner. The pieces were also ‘shock loaded’ with the 5 kg weight being dropped directly onto the top of the piece from 15 cm above. This resulted in the landing pad springs being activated proving the piece could transfer the shock load without failure. Shock load testing has not been conducted for the final pieces as each one is to be resin cast first.

Before integrating the chassis with the final landing system, a static system test was conducted which consisted of the 38 kg mass being mounted onto a flat surface on top of the chassis struts. As the chassis is significantly lighter than 38 kg, this was seen as sufficient validation to start careful integration.

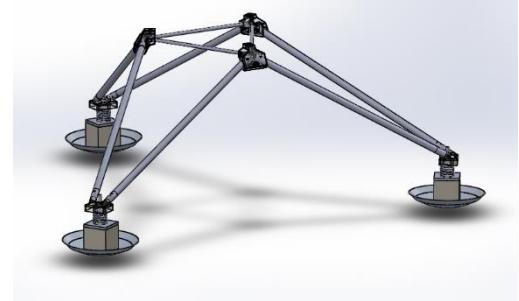


*Figure 70: Preliminary testing of corner piece static loading*

## 5.5 Summary

The final design (see Figure 71) of the landing system developed during Phase V consists of the following components;

- Leg bars (3)
- Corner pieces (3)
- Horizontal corner connecting bars (3)
- Landing pad (3)
- Metal bowls (3)
- Spring casing (3)
- Spring casing and leg bar connector (3)
- Compression springs (3)
- Nuts (21)



*Figure 71: SolidWorks model of landing system*

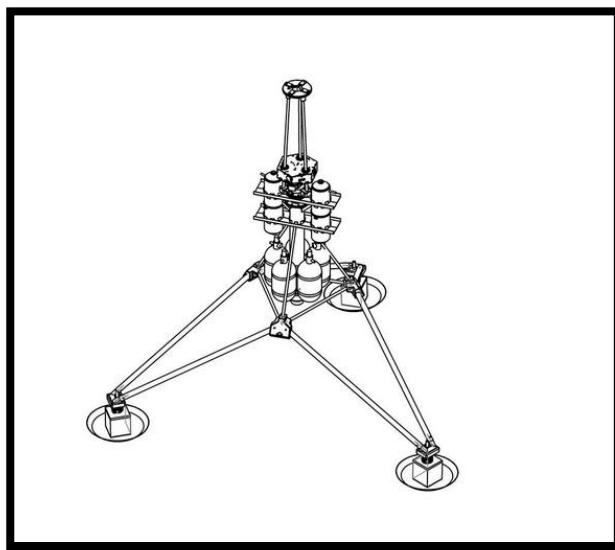
The inherited model contained ninety one components, the new system has forty five. This highlights the significant simplification which has occurred in the design over Phase V. The new system has been designed, manufactured, tested and integrated into the Mk II (see). In term of improvements over previous designs;

- 1.70 kg lighter than the previous system.
- Withstands static force
- Predicted to withstand hard landing
- Substantially simpler
- Roughly half as many components as the previous design
- More simple to procure, manufacture and assemble



### 5.5.1 Finalised Structural Design Mk II

The finalised integrated design is shown in Figure 72.



*Figure 72: Solidworks design of final structure*

## 6 Electronics and Software

### 6.1 Hopper Software

Author: Marian Daogaru

#### 6.1.1 System requirements

The Lunar Hopper Project being in the fifth year of development and second model design, several requirements have been already defined. Moreover, due to previous project work, some of the already defined requirements had to be changed, in order to become more realistic in terms of achievability. As such, the software requirements for Phase V are:

1. The hopper is required to operate autonomously, requiring user input only for start, emergency stop and engine throttle commands.
  - a. This requirement is derived from the general project requirement of vertical take-off and vertical landing. During preliminary design, the team concluded that a manual operation of all systems would be unfeasible for a human, due to slower response from a human pilot to external factors compared with an autonomous software.
2. The software frequency is required to be at least 50 Hz.
  - a. Requirement derived in Phase IV [4]. Simulation showed that at least 40 Hz are required for the system to maintain dynamic stability.
3. Software must operate with an efficiency higher than 99%.
  - a. Previous projects experience frequent crashes in the software, crashes that were rendering the hopper unresponsive until a hard restart was applied.
  - b. If a crash would trigger during the flight, it would present a real danger to the project team and the surrounding structures, and possibly bystanders.
4. The software must incorporate an ABORT (or MECO) command.
  - a. This command will effectively stop any further commands to be given to the hopper.
  - b. The command will stop any loop in the code being executed, until a hard restart is applied.
5. The system must interpret sensor readings and operate the ACS if the inclination registered is higher than 2 deg. Furthermore, the system must execute the ABORT command if the inclination on any axis is higher than 15 deg.
  - a. These requirements were derived in Phase IV. [4]
  - b. 2 deg inclination on any axis was deemed the threshold at which vibration induced inclination/movement would be overtaken by general pitch or yaw of the entire assembly.
  - c. At 15 deg, the ACS would be unable to restore the hopper to vertical position.

#### 6.1.2 Background software

The Phase IV hopper team developed the majority of their software based on previous work from Phase III. With the software being currently in the third year of development, with similar requirements between the current project and previous projects, the majority of the code can be reused. Moreover, although not truly field tested, as Phase IV did not launch, it was tested to a level that assures the user regarding its fidelity. As such, parts of the existing hopper software were accepted to the software used for Phase V as well.

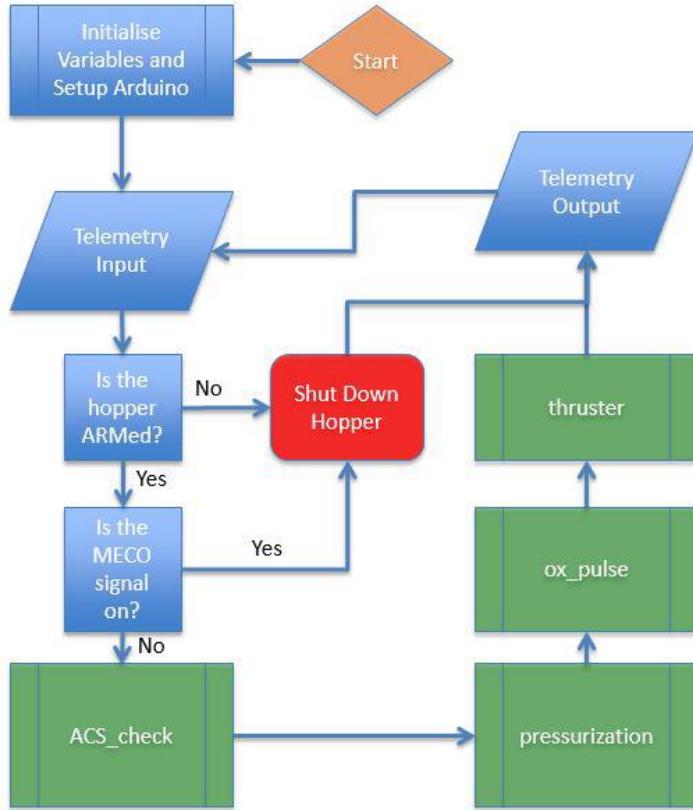


Figure 73: Phase IV main loop logic of the hopper software.

Figure 73 [4] presents the overall loop logic for the Hopper command software of Phase IV. Initially, the hopper waits until it receives a start command which is then followed by Arduino set up and calibration of sensors. During this step, the main sensors, the gyroscopes and the accelerometer, presented in Section 6.2.2, would calibrate and the reference values for each sensor would be registered. The references are required to be recorded during each operation as they are depended on the status of Arduino input voltage.

Following the calibration of sensors, the hopper would interpret the commands sent by the pilot and act upon it. In the case of the Main Engine Cut-Off (MECO) command being sent, or should the hopper record an inclination higher than the maximum allowed inclination, the hopper would automatically shut down by rejecting any commands from the pilot and venting all pressure from the Oxidiser tanks. If shut down did not occur, the hopper would first check ACS and record pitch and yaw inclination, followed by pressurization of oxidiser tanks according to pilot input. Next step was to continue firing the main engine at the new throttle input, and to adjust the hopper's inclination, depending on what it was registered earlier. Finally, the hopper would transmit an acknowledgment message to the pilot confirming the completion of current tasks, and combine with a message of the current status of the hopper. The process described would then repeat by receiving a new message until touchdown or MECO was activated.

It is important to note that the main hopper controller is an Arduino Mega which only supports code originated from its own Arduino Software programming language – Arduino Software Integrated Development Environment (IDE). The IDE accepts primarily C and C++ code. In addition, when creating the Arduino code, 2 main functions must be used at all times:

- Setup – where the first initialisation of the software is created. During this function, the internal clock of the Arduino starts. The code that rests within this function will be executed just one, in the beginning, unless it is called later.
- Loop – this function has the property, as the name suggests, that it will loop the sub-functions within after all run, the process starting from the beginning of the loop.

For both functions mentioned, the code is executed in order, from the top to the bottom, as shown in the following snippet:

```
void setup()
{
    function_1(); //executed first
    function_2(); //executed second
    function_3(); //executed last
}
void loop()
{
    //loop starts from here
    f_1(); //executed first, 4th, 7th, ...
    f_2(); //executed second, 5th, 8th, ...
    f_3(); //executed third, 6th, 9th, ...
    //loop stops here and goes back to the start
}
```

Phase III & Phase IV code iterations went for a modular code design in which additional functions were created to provide various functionalities. Therefore, this method allowed independent code to be written for:

- *MECO and Disarm* – emergency engine cut-off and hopper disarm by venting all gas;
- *ACS\_check* – check if ACS sensors and thruster valves are functioning properly;
- *ACS\_calibration* – calibrate the gyros and the accelerometer;
- *ACS* – main ACS function that controls the ACS functionality according to sensors reading;
- *Pressurisation* – pressurise the oxidiser tanks to required pressure by Ox<sub>2</sub>\_blip or throttle command;
- *OX<sub>2</sub>\_blip* – used to warm the catalyst bed and the fuel by opening the oxidiser valve for a short duration
- *Thruster* – primary thruster function, operating the oxidiser valve when required by telemetry data;
- *telemetryNetworkControl* – pings the remote pc with status updates;
- *TelemetryDataTx* – transmits telemetry data (e.g. pressure status, gyro angles, time, etc) to the pic at constant intervals;
- *TelemetryCmdRX* – checks if the hopper has received telemetry data from the remote, and decodes the received data.

The hopper software was also divided in 2 parts. Firstly, the functions described previously were part of the hopper module software which was uploaded to the Arduino installed on the hopper. This code also enabled the hopper to communicate with a remote station, a PC, using radio communication, from which the pilot could control the hopper using the secondary software installed on the PC.

### 6.1.3 Software Issues

During Phase IV, the majority of the software was tested extensively. However, several major issues with the code were observed. Extensive testing on the Phase IV software was done by

Phase V team to confirm the existence and severity of the problems reported. Following this testing, it was decided that the majority of the code from Phase IV was obsolete, with only small parts of the software to be reused.

One major issue observed, that was consider mission critical, was consistent crashes of the both the hopper module software, and also the remote station. These crashes were considerably frequent, up to 10% of the times the hopper was tested for launch. This meant that, 1 in 10 times, the hopper was susceptible to fatal failure.

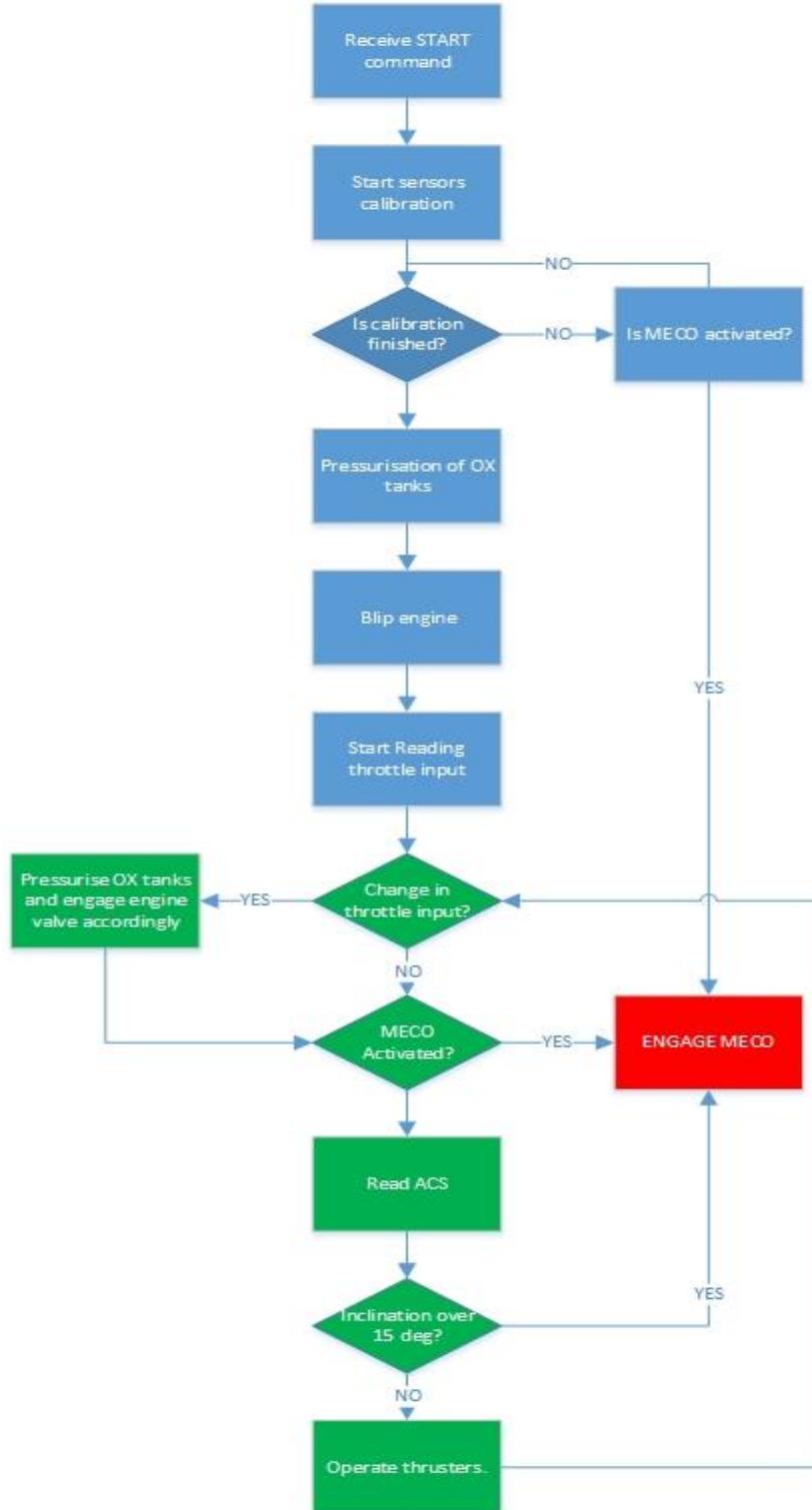


Figure 74: New software algorithm chart.

Additionally, Phase IV discovered that the ACS related part of the code was only reacting to one side of the hopper, not on both axis at the same time. As such, just one axis of the hopper was acted upon, with the second side being unresponsive; moreover, the second side was locked on full thrust from ACS. With one side locked on full thrust, the hopper would pitch towards that direction as soon as it would start to lift-off, crashing the hopper. Smaller problems were noted, such as the *Ox\_blip* not responding when connected to the valve, and the pressure transducer used to measure the pressure in the oxidiser tanks was giving wrong pressure readings.

#### 6.1.4 Software Updates

Although the majority of software, as described earlier, could still be reused, several updates were required in order to resolve the issues presented earlier, and to also include the new changes in the project. Figure 74 presents the new algorithm flow chart that the Arduino code will follow during normal operation. It can be observed that compared with the previous version of the software, this current version is considerably simpler.

##### 6.1.4.1 Obsolete telemetry

The main change in the software came with the decision to replace the existing communication device. The new comms device, the ZULU module described in (Section 6.3.2.2), although much simpler compared with the previously used hardware (Xbee module), enabled the team to solve the crashing issue. Initial testing using Phase IV communication subsystem (comms) proved unsuccessful in solving the crashing problem. Testing was conducted using the previously created test procedure. One Xbee module was connected to the remote PC and one to the Arduino, followed by initialisation of the previous remote software, communication was achieved after several attempts. With pairing achieved, sensors readings were obtained soon after. However, consistent crashing was observed when the line-of-sight was obstructed or vibrations were applied to either modules. These anomalies were further proven by testing. In addition, the computer software was unable to achieve communication in majority of the cases. Thus, the decision to not continue further with the remote PC software and the Xbee was taken. Consequently, the telemetry receiving and sending function (telemetryNetworkControl, TelemetryDataTx, TelemetryCmdRX) became obsolete, and thus decrease the time required one loop to run by 2 ms.

Using the new ZULU module, the code would only need to read a respective pin from the module in order to receive the pilot's commands. This would greatly increase the reliability, as the system would be read the direct command, rather than an interpretation of a command. However, the ZULU has a drawback, as it is not able to send back telemetry to the pilot. The loss of telemetry was accepted for the functionality achieved with the new setup.

##### 6.1.4.2 Remote software

With the remote PC software removed, a need for a remote controller was created. The ZULU module was also chosen for the dual capability to act as both a transmitter and a receiver. The controller software is presented in Figure 75. It can be seen that only a START and MECO command are present in the software. These two commands are registered by the controller through interaction with switches. The throttle output to control the engine does not appear in this algorithm as the analogue output from the throttle is fed directly to the transmitter. As the analogue output of an Arduino peaks at 5V in PWM (Section 6.3.2.3), this voltage is over the maximum voltage allowable in the ZULU module (3.3 V maximum voltage [16]). Due to potential damage, connecting the throttle directly to the transmitter was deemed necessary. Direct analogue input from potentiometer and data results was discussed in Section 6.3.2.5.

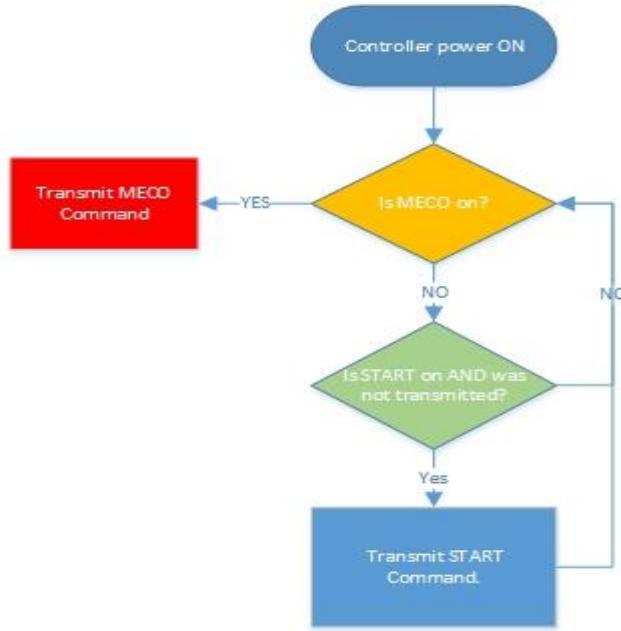


Figure 75: Remote controller algorithm flowchart.

#### 6.1.4.3 MECO

The MECO (*meco\_high*) function retained the same functionality as in Phase IV. When called, this function will stop the hopper from operating. It is an essential algorithm in the program, being the only method of ensuring that in case of emergency the hopper will not behave erratically. In case this method is called, no more commands can be given to the hopper. Furthermore, the oxidiser valve will close, with zero thrust being generated by the engine, followed by the venting of gas from both oxidiser tanks and all paintball tanks. As expect, this function must be called on only in extreme situations, when the hopper is beyond recovery. With zero thrust generated, the hopper will then be expect to crash, potentially damaging the hopper.

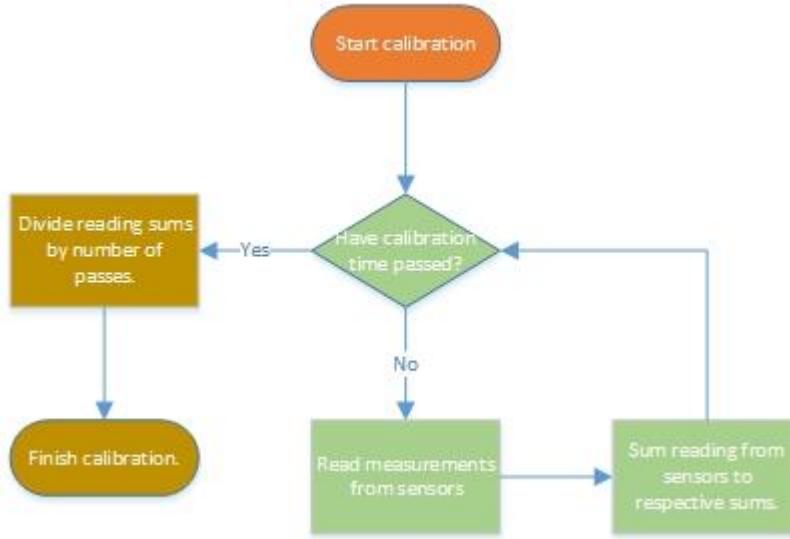


Figure 76: ACS\_Calibration algorithm.

#### 6.1.4.4 ACS\_Calibration

After START command was received the *ACS\_Calibration* starts. Compared with Phase IV, this method is extremely similar. The need for calibrating the sensors, especially the gyros, comes from their design: gyros, and in general electronics that produce an analogue signal,

produce a signal that has is not smooth, but noisy. This noise is represented by small variation when reading the sensor, even when there is no movement. Figure 76 presents the algorithm of *ACS\_Calibration*. By reading the sensors while the hopper is stationary, the 0 level is registered and stored in memory for later use. First stage of calibration is the respective summation of the sensors readings, and a count of how many time the sensors were read. After a certain duration is passed, the sums obtained are divided by the counts number. Consequently, a median value, the 0 level, for each sensor is obtained.

#### 6.1.4.5 ACS

Figure 77 presents the algorithm of the main ACS function. This method is used to stabilise the hopper during flight. After calibration finishes and the main loop enters into flight mode, the ACS starts a new cycle of stability. At the beginning of a new cycle, the function waits a short period in order to allow the hopper to adjust to previous ACS commands and to enable ACS thrusters to act based on commands from a previous cycle.

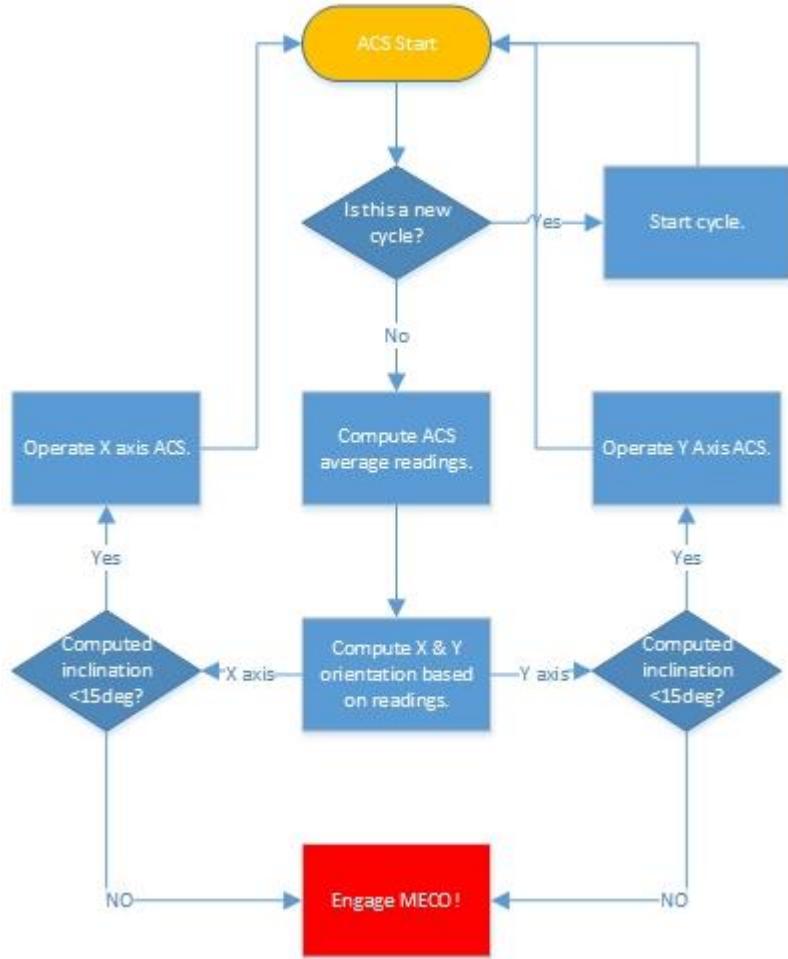


Figure 77: ACS algorithm.

After the resting period passed (15 ms), the method reads the sensors input 10 times to obtain an average value of the X & Y axis gyros and the 3 axis accelerometer. Having these values, the inclination of the hopper is computed. Next, if the inclination on either X or Y axis is higher than  $15^0$ , *MECO* is automatically activated. If the inclination is lower than  $15^0$ , and higher than  $2^0$  on either X or Y axis, the thrusters are activated or closed, respectively. If no considerable change in pitch or yaw is measured, the system is inactive until a new cycle begins and the process restarts. While the majority of the algorithm is similar to Phase IV, an anomaly was previously observed, as described earlier, in which just one axis of the hopper would engage. This issue was fixed by creating a different logical path, with identical priority for either axis

section of code. Stationary testing proved that the system would oscillate around the neutral position, due to noise in the system.

#### 6.1.4.6 Ox\_pulse

In Figure 78 the logical pathway of *Ox\_pulse* function is displayed. This function is used to heat up both the catalyst bed and the fuel grain. It is run immediately before the hopper loop goes into flight mode.

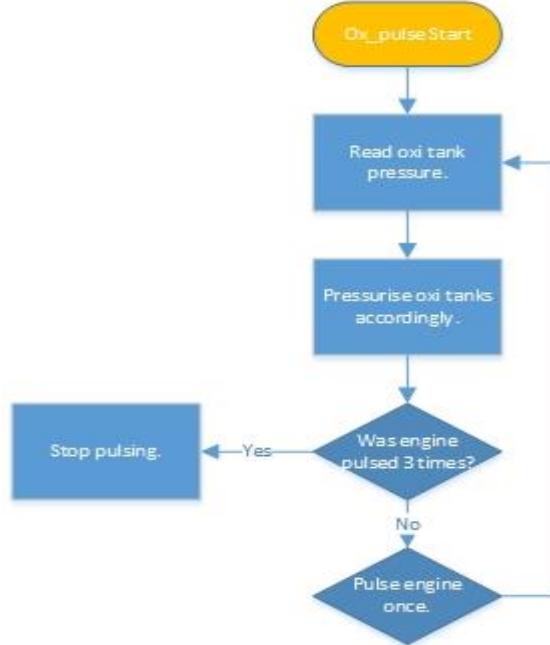


Figure 78: *Ox\_pulse* algorithm.

Initially, the pressure in the oxidiser tanks is read to determine if they require pressurisation, as the oxidiser requires a certain pressure to react with the catalyst bed. Next, the oxidiser valve is opened for a short duration (0.3s), allowing oxidiser to interact with the catalyst bed and the fuel. Proceeding on the valve closure, the system waits for a longer period (5s), to allow heat dissipation from the inside to the outside of the fuel grain. After 1 complete pulse was achieved, the oxidiser tanks are re-pressurised, and the system is pulsed 2 more times. The need for pulsing came from the engine design in order to start the engine directly into firing mode, rather than cold gas mode, as it is a hybrid engine [5]. Moreover, the pulse time duration was chosen as 0.3s (0.9s in total) due to the small mass of oxidiser stored in oxidiser tanks. Lastly, this method is a new addition to the code, solving the issue in which *Ox\_blip* (Section 6.1.2) would not react properly with a valve.

#### 6.1.4.7 Pressurisation

Pressurisation is an adapted function from Phase IV. The older version proved to not function properly, providing incorrect pressure readings from the pressure transducer. Testing showed that one consistent source of errors was the equation that translated the analogue value, obtained from the transducer, to pressure value. Other sources of errors came from ambiguous logical paths and faulty *if* statements. Thus, in the interest of time and functionality the new function, *Pressurisation*, was created. Moreover, parts of the old function were moved to other current function to increase performance.

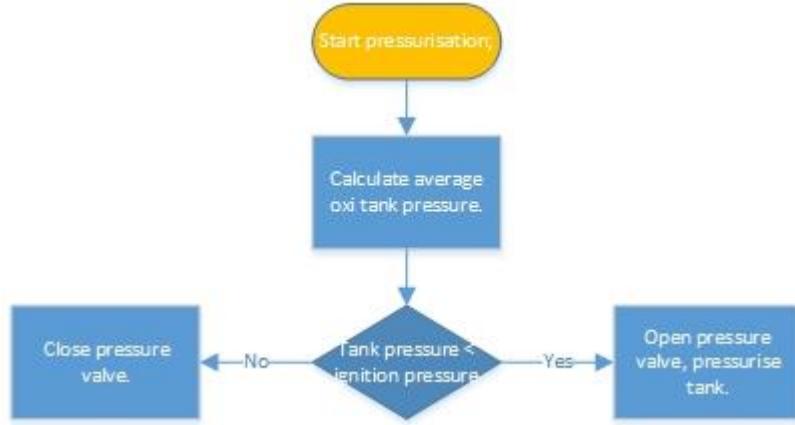


Figure 79: Pressurisation algorithm.

Figure 79 presents the current algorithm of the function. Initially, the function measures the average value of the pressure in the oxidiser tanks; average due to noise in the system. If the pressure is lower than ignition pressure, pressure at which the oxidisers will properly react with the catalyst bed, defined in engine specification [5], then the oxidiser valve is opened and the tanks are pressurised. When the pressures are equalised, the oxidiser valve shuts.

#### 6.1.4.8 Throttle

The throttle function is new addition to the code, based on the hardware used. In addition, it incorporates some of the functionality from the old *pressurisation* function. Figure 80 presents the logical flowchart employed by the function to control the engine throttling capabilities.

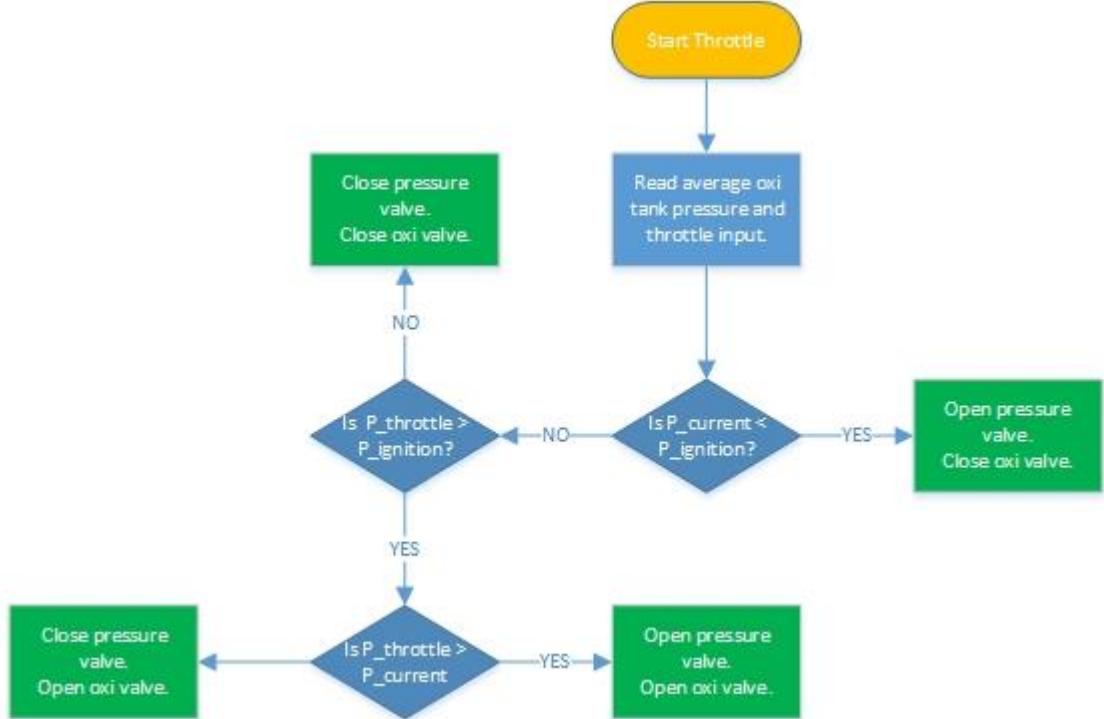


Figure 80: Throttle algorithm.

After the hopper enters flight mode, *throttle* becomes the main function, along with ACS. As the function is accessed during each pass in the loop, the pressure in the oxidiser tanks and the throttle input are read on each pass, measuring any change in both pressures. Following the measurements, a logical algorithm is used to determine which valve should be opened. With 2 valves used in the propulsion system, and each valve operating in 2 possible position, 4 options are available. The default stance is represented when the oxidiser tanks pressure ( $P_{current}$ ) is

either smaller than ignition pressure ( $P_{\text{ignition}}$ ), or higher than ignition pressure and throttle pressure ( $P_{\text{throttle}}$ ). These stages require the oxidiser valve to be opened (no oxidiser going in the engine), and the tanks are brought to ignition pressure. When the  $P_{\text{throttle}}$  is increased and surpasses  $P_{\text{ignition}}$ , oxidiser valve is opened and thrust is being generated by the engine.

Moreover, if  $P_{\text{throttle}}$  is higher than  $P_{\text{current}}$ , then both the oxidiser valve and the pressure valve are opened, allowing continuous pressure to be applied to the oxidiser.

During algorithm creating, a safety measure was introduced in the code, in order to protect the oxidiser tanks and the pressure transducer. As the transducer is rated up to 40 bars, if the pressure in the tanks reaches 39 bars, the pressure valve will close, regardless of throttle input. When the system is in the setup:  $P_{\text{ignition}} < P_{\text{throttle}} < P_{\text{current}}$ , the engine runs in the “blowdown mode”, with the pressure being reduced only by using the fuel. This method was considered to pose significant problems during descent. Consequently, opening the vent valve was considered. However, as no testing was done on the engine to prove that this method would produce better results, it is embedded in the code as a sub-routing that is not used.

#### 6.1.4.9 Ref25

This is a new function added to the software. It uses a LT1009 2.5V reference voltage regulator to detect the analogue value of 2.5V compared to the 5V output pin of the Arduino. While connected to an analogue pin, a signal could input 0 to 1023 bits, each bit representing 5V/1024 voltage. However, due to imperfections in the circuit, loads, power supply, the 5V rated output will most likely not output exactly 5V, but rather a value  $V_{\text{out}} < 5V$ . As such, now each bit of information would measure  $V_{\text{out}}/1024$ . Consequently, if an analogue value must be read but it remains constant and the voltage across the Arduino drops, the value obtained will rise. As such, this function reads the 2.5V input, and determines what is the actual voltage of each bit. Next, the voltage value of 1 bit is multiplied by the analogue value read by the pin in question to obtain the real voltage of that pin.

```
V25 = analogRead(pin25); //reads the value of the 2.5V reference voltage
```

```
V1 = 512 / V25; //obtains the voltage of 1 bit
```

```
V = V1 * analogRead(pin); //determines the actual voltage of the read pin
```

#### 6.1.5 Code testing

Initial testing was performed in order ascertain the communication problems and existing hopper software. Following existing procedure for testing described in Phase IV [4], limited communication was achieved. Moreover, the frequent crashes encountered led to the conclusion that a new communication method must be employed.

While a system test was not possible, considering that such a test would involve firing the engine, a reproduction of the hopper system was done by replacing valves with LEDs, and later by resistors, and the throttle and pressure transducer by potentiometers. With the assembly in place, the “hopper” was activated. During the initial testing using this method, the existing problems were observed.

Next step was represented by individual testing of different functions. While the majority of the functions were easily tested using different arrangements of resistors and LEDs, *ACS* and *throttle* were tested in detail to ensure proper functionality. Appendix H – Test plans electronics presents data recorded during *ACS* testing by measuring the voltage, and thus the analogue input, of a gyro during all stages of hopper software. Using data the current *ACS* function was created.

The requirements for this subsystem were met through the code design, and the extensive testing of the code.

## 6.2 Electronics

Author: Marian Daogaru

### 6.2.1 System requirements

The electronics subsystem is one of the most critical systems on the hopper. It is composed of different components that provide different functions. Preliminary design determined the following requirements for the subsystem to be considered fully functional:

1. Provide a controller for the hopper assembly capable of autonomous operations of at least 4 min, with minimum input from pilot.
2. The hopper controller must be able to read both analogue and digital inputs, and output digital commands. In addition, the controller must be able to output consistent voltage of 5V and 3.3V.
3. The system must be power autonomous, being able to provide power to all electronical components, including the solenoid valves.
4. The subsystem must use motion sensors to determine the hopper position during interval smaller than 20 ms, with an accuracy of at least 0.1deg.
5. A remote controller must be used to operate the hopper, with a maximum of 4 commands necessary to fully achieve vertical take-off.
6. The controller must be able to transmit engine throttle commands, and the hopper must be able to receive and interpret them.
7. Provide a rigid housing for all electronic equipment used in the subsystem, except solenoid-valves and attitude sensors.

### 6.2.2 Components

#### 6.2.2.1 Micro-controllers - Arduino

Phase IV used two microcontrollers on the hopper module – one to control the hopper and one to control the MECO sequence. Due to mass, power and physical space constraints, it was decided that one Arduino Mega 2560 was able to provide both functionalities, given improved algorithms in the code. The Mega was positioned inside the box (Section 6.2.2.6), onboard the hopper. It was responsible for the hopper control, MECO commands, and radio communication with the pilot remote.



Figure 81: Arduino Mega 2560.

Arduino Mega was also chosen as it was already flight proven by previous iterations of the project. In addition, the project team and the supervisors were only familiar with Arduino micro-controllers. Although the Arduino was chosen and deemed capable of meeting the

requirements, in terms of autonomy, operating speed and input/output capabilities, other microcontrollers were investigated, most notably a Raspberry Pi. Two major advantages of the Raspberry Pi were identified over the Arduino:

1. A Raspberry Pi used directly an operating system (OS). This cancels the need for additional equipment to operate it. Moreover, an OS offers more programming advantages, with code being able to be created in different programming languages.
2. A Raspberry Pi is a multi-core microprocessor. It has 4 identical cores, each individually more powerful than an Arduino Mega. With multiple cores, parallel programming can be implemented, allowing the user to perform different tasks at the same time.

However, due to its increase complexity, all heritage software already existing in Arduino code, and team experience with Arduino programming, the Arduino was chosen to act as the main controller for the Hopper. A proposal to upgrade to a Raspberry Pi can be found in Appendix E –Technical Drawing for Corner Piece. Following the development of the radio communication, an Arduino Uno was chosen to act as the remote controller. The microcontroller was responsible for acknowledging user input and transmitting it directly to the receiver on the Hopper.

#### **6.2.2.2 Rate Gyroscopes**



*Figure 82: CRS05-1 Rate Gyro.*

In Figure 82, the rate gyros used in the project can be observed. Two identical gyros were used, one for pitch and one for yaw, to detect inclination changes. The CRS05-01 are single axis Silicon Sensing gyroscopes, capable of detecting  $\pm 50\text{deg/s}$ . Thus, it can detect small changes required by the hopper, up to  $0.1\text{deg/s}$  resolution. These gyros were chosen as they were already flight tested in Phase III and Phase IV. Lastly, an important feature of these gyros is the analogue interface they provide, enabling neutral position acquisition with different input voltage. Technical data sheet can be found in Appendix G - Rate Gyro [17].

#### **6.2.2.3 Accelerometer**

The accelerometer chosen was a standard off the shelf 3-axis accelerometer, used in previous phases of the project and flight proven. The sensor had a dual mode of operation, being able to operate in  $\pm 1.5\text{g}$  or  $\pm 6\text{g}$  mode, depending on the resolution desired. In addition, it provides high resolution at  $1.5\text{V}$  ( $800\text{mV/g}$ ). Moreover, it can operate at  $2.7 - 6\text{ V}$  supply, making it ideal for an Arduino  $5\text{V}$  output. The sensor is able to operate a wide range of temperatures:  $-40^{\circ}\text{C}$  to  $80^{\circ}\text{C}$ , being able to withstand the temperatures during launch. Lastly, the sensor was still chosen due to its small dimensions ( $25.4\text{mm} \times 12.7\text{mm}$ ) and low mass:  $1.1\text{g}$ . [18]



Figure 83: Accelerometer MMA7361L.

#### 6.2.2.4 Batteries

With the hopper having similar electronical components as in Phase IV, the power requirements remained similar. However, during testing of the previous project, it was discovered that the hopper would start to malfunction after a short period of time due to power issues. The team noted that one possible explanation was low quality of the batteries.

Firstly, during initial testing of the hopper, the recommended batteries from last year were used: one 3S @ 850mAh & 20C and one 4S @ 2500mAh & 25C. It was thus observed that due to power consumption in the main battery, the voltage provided to the controller would decrease with time. This is a quality of a poor battery. Additional testing (Appendix H – Test plans electronics) demonstrated that extensive use of the battery would damage it, and also would provide wrong readings. By connecting the Arduino to a power supply and using different LEDs to simulate the valves, the setup proved that 2 batteries are required in order for the sensors not to interfere with the valves. It was also proved that a small decrease in voltage applied to the Arduino after calibration was performed, would create a false rotation movement reading given by the gyros. During calibration, a neutral position (the stable vertical position of the hopper) would be recorded as a voltage. Any deviation from this voltage implies a movement in the sensor, associated with the hopper. Table 11 illustrates these findings. It can be seen that during calibration, when no valves were operating, the output voltage remains at half input voltage value (as expected from technical specifications). When the system went into flight mode, the voltages dropped and a loop was created in which sensors were outputting a false rotation and the valve were trying to compensate, while the system was stable.

|                             | Output (V) | Input (V) |
|-----------------------------|------------|-----------|
| <i>Initial 1</i>            | 2.27       | 4.57      |
| 2                           | 2.29       | 4.6       |
| 3                           | 2.27       | 4.59      |
| <i>During calibration 1</i> | 2.32       | 4.67      |
| 2                           | 2.33       | 4.67      |
| 3                           | 2.32       | 4.67      |
| <i>During Operation 1</i>   | 2.22       | 4.3       |
| 2                           | 2.22       | 4.3       |
| 3                           | 2.14       | 4.3       |

Table 11: Output and Input voltage measurements from a Gyro during normal operation mode with LED simulated valves.

As such, the decision to use 2 batteries was maintained, with one battery powering the Arduino and the sensors, and one only for valves.

The power requirements of the electronics components on on-board the hopper are presented in Table 12.

| Component           | Operating Voltage (V) | Maximum Current (mA) |
|---------------------|-----------------------|----------------------|
| Arduino Mega        | 12                    | 50                   |
| Pressure transducer | 6-30                  | 20                   |
| Solenoid Valve      | 12                    | 2800                 |
| Rate Gyro           | 5                     | 35                   |
| Accelerometer       | 5                     | 0.4                  |
| Zulu                | 3.3                   | 18.5                 |

Table 12: Power requirements for the hopper.

It can be seen that each solenoid valve draws 2.8A. Due to their application, all solenoids are in parallel. At any given time, except if MECO is engaged, only 4 valves are opened simultaneously. Thus, the current drawn is:

$$2.8 * 4 = 11.2A$$

The current drawn by the other components is:

$$50 + 20 + 2 * 35 + 0.4 + 18.5 = 158.9 mA$$

Considering that both the solenoids and the Arduino had similar power ratings, a decision to use 2 identical batteries was made. The identical batteries would enable easier mounting in the hopper, and also would reduce the risk of wrong connection of components. Thus, 2 x 12V @ 2200mAh & 25C were chosen to supply to required power. The max current provided by the batteries is 2.2Ah & 25C = 60A, current which is higher than what can be expect to be drawn by the components. Moreover, this would provide an autonomy of  $\frac{2.2Ah}{11.2A} = 11.78min$  to the solenoids and  $\frac{2.2Ah}{0.1589A} = 13.85h$ ; more than enough for a 10s flight + several minutes of standby.

### 6.2.2.5 Peripherals

As expected, this project was required to interconnect all electronics, consisting of one Arduino Mega, 2 gyros, 1 accelerometer board which incorporates 3 accelerometers, a pressure transducer, 2 batteries, 1 ZULU module and 7 solenoid valves. With the addition of the ZULU and the swap in batteries, the solution employed in Phase IV had to be changed.

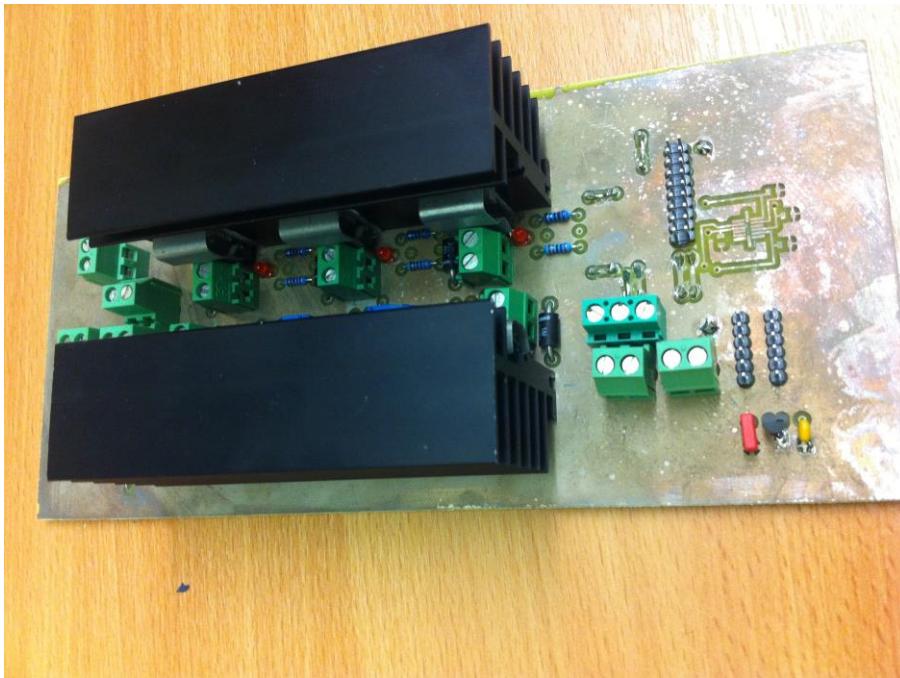
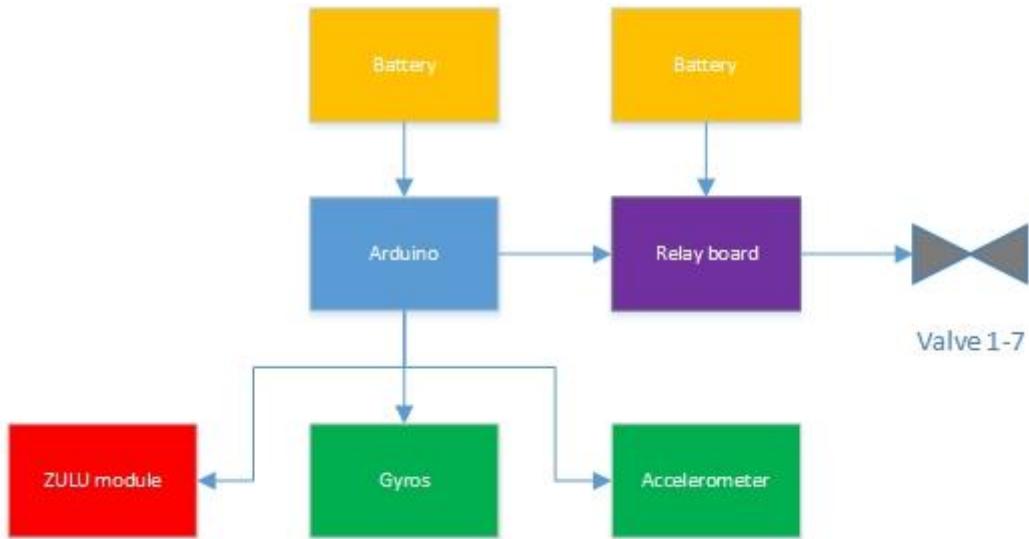


Figure 84: Phase IV PCB.

Figure 84 displays the Phase IV custom made PCB used to distribute the power to the valves, and to allow the interaction between the Arduino and the sensors. This PCB presented several advantages regarding its functionality. Firstly, it was an integration of all power distribution between all components. Moreover, it was a compact piece of equipment, with all subcomponents in on the same part. However, it allowed no options for upgrades. In addition, the lack of proper labeling and schematics proved extremely difficult to back-track and reverse-engineer the component for the new needs of the current project. Consequently, it was decided to not use this component in Phase V.

The redesign of the electronics connection was then shifted towards a more modular design, with adaptability being a priority. The existing PCB would be split into different components, each with its own functionality. This also enabled a better adaptability to design change, with individual components, that may become obsolete, to be easily changed to newer components without the need to change the entire design.



*Figure 85: Electrical circuit flowchart.*

Figure 85 presents the distribution of power in the system and the interaction between components. To achieve the functionality needed, and to allow adaptability, it was concluded that the connection between different components should be done using an additional electrical component. Firstly, the Arduino and the valve were connected via a relay board, consisting of 8 individual relays. This would allow easy swap between relay, their connection and the power distribution between them. Moreover, this would enable an easier use of the valves, as each relay acts as a switch. In addition, power distribution from one battery to the relay was achieved using a custom made power distribution board. The board was tested in order to ascertain its capabilities during high load.

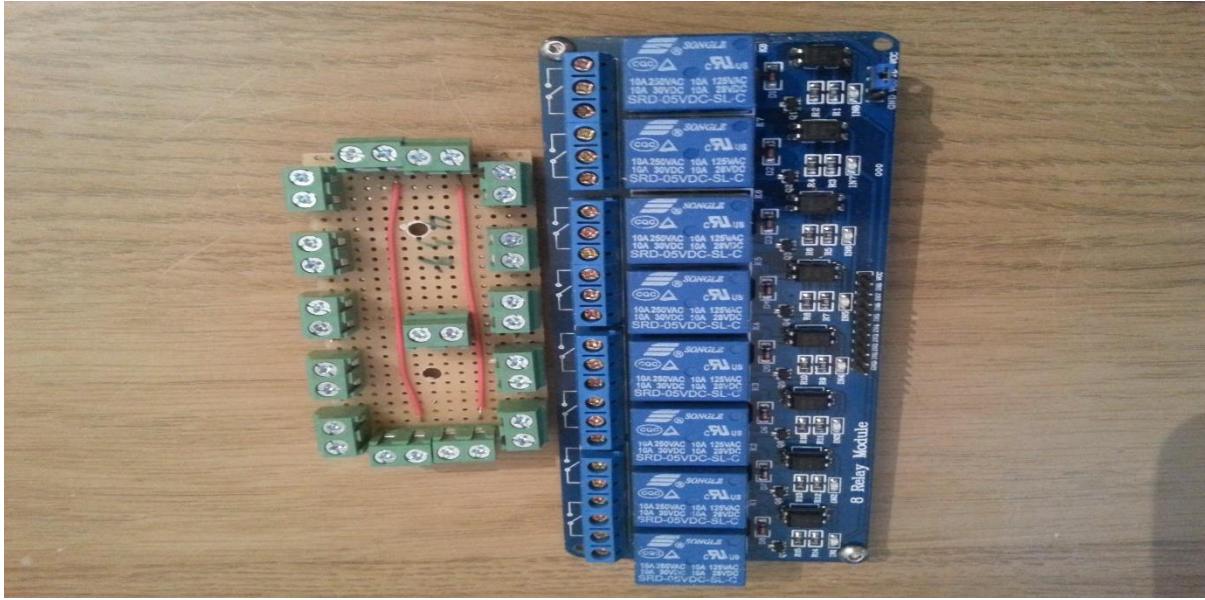


Figure 86: Custom made power distribution board (left) and relay board (right).

The power board was initially designed to be created as a PCB, with strips of metal to carry the load. Figure 87 presents the initial design of the PCB. However, it was considered that the strips of metal (copper) would not be able to resist the high currents. The design was then shifted for a simpler method, and a more customisable type of manufacturing, as displayed earlier.

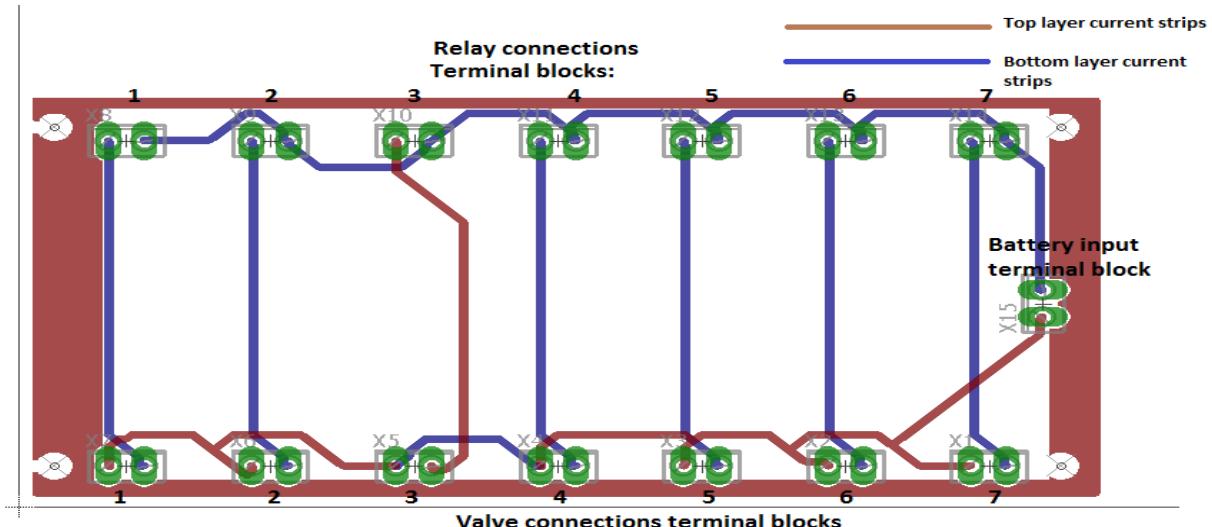


Figure 87: Power board PCB design.

For the interaction between relay board, sensors, comms and the Arduino, a blank Arduino shield was used Figure 88. Individual connections were created between the pins required and either pins where the connection was made, or secondary wires to connect directly to the relay board and comms.

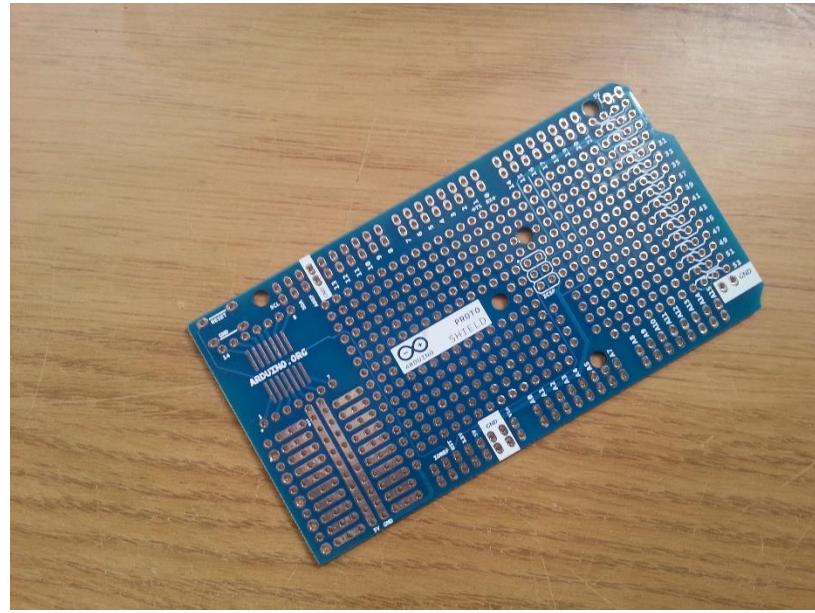


Figure 88: Blank Arduino shield.

#### 6.2.2.6 Electronics Housing – the Box

Previous iteration of the project lack a rugged housing for electronics. With one of the main aims of the project being ruggedisation of the hopper, the electronics housing (the box) needed to adhere to this requirement.

Two main factors constrained the design of the housing:

1. Positioning with respect to the center of mass;
  - a. Although the box was not expected to be heavy, less than 1 kg, its position would still be able to affect the center of mass; and thus destabilize the hopper.
2. Physical mounting space;
  - a. The engine provided little space for mounting around it. The majority of the space was already reserved by the oxidiser tanks or nitrogen tanks.

Taking these factors in consideration, it was decided that the optimum position of the box would be over the engine. The position had small influence on the center of mass radial displacement, important feature for stability. Moreover, beside the ACS structure, the top of the engine was relatively empty.

The ACS structure proved also useful as the mounting structure for the box. The housing required also to be easily accessible, for changes to be made with the box mounted. As such, it was decided that, again, for continuous improvement design and to allow easy changes to be made to the housing, a modular design must be created, with different parts easily interchangeable. With these concepts in mind, the design started by investigating options for mounting the box on the ACS struts. Using bolts to attach the components was considered the easiest solution. However, this option would leave no room for later improvements. It was then considered that using 3D printed components, placed on the struts, held in tension would provide an optimum design. The tension would be applied directly by the housing, with some of the parts being attached to the 3D printed parts. This solution would not fix the housing a position directly, but it would be held in one particular position due to geometry. This position could be later changed, if alterations to the box were to be made, without affecting the ACS structure.

The decision for the shape of the housing was also based on the constraints: a round shape would provide the mass distribution; however, it would be hard to manufacture a cylinder. A square / rectangle would be easier to manufacture, but the mass distribution would cause

potential problems. As such, it was decided that a hexagonal shape was able to better meet the requirements. With the configuration decided, the interface between the 3D printed parts and the box was created. The base plate, the plate closer to the engine, would be fixed to the 3D parts, using bolts for easier replacement. The top plate would be “locked” by a 3D printed locking mechanism. Figure 89 presents the 3D parts designed (top) and the manufactured parts. It can be seen that both the base plate and the top plate would be able to slide around the parts, then be attached.

A secondary design challenge arose from the ACS structure: with a tripod configuration at an angle, it would be impossible to slide a rigid structure along the struts. As such, the baseplate and the top plate were split into smaller parts, each being able to be individually attached to the 3D printed parts. Considering all constraints, rapid prototyping for box manufacturing was employed. Laser cutting was used to produce all the components, as precision was paramount for successful integration. Figure 90 displays the design of the box, integrated with the ACS struts. Lastly, as it can be seen in Figure 91, the final manufacturing of the housing used simple methods of attachment between components, for fast installation and interchangeability.



Figure 89: 3D printed components. Cad models (top) vs printed parts (bottom). Baseplate holder (left) and the locking mechanism (right) for top plate.

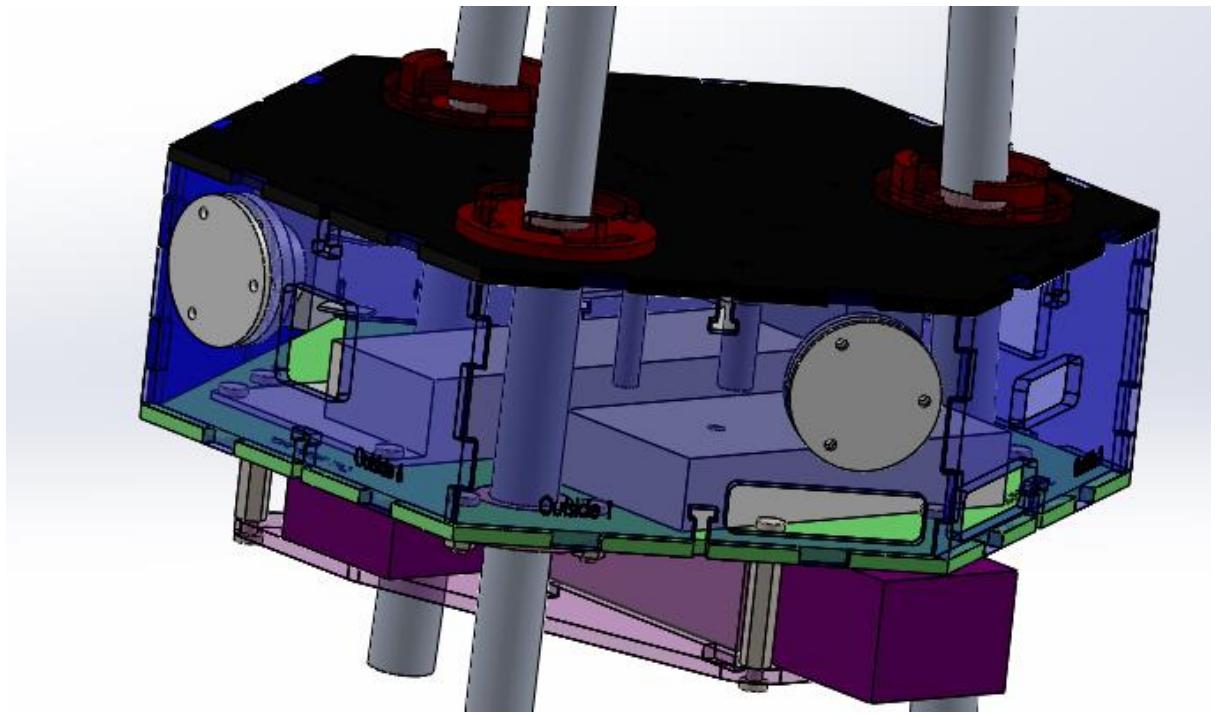


Figure 90: CAD design of electronics housing.

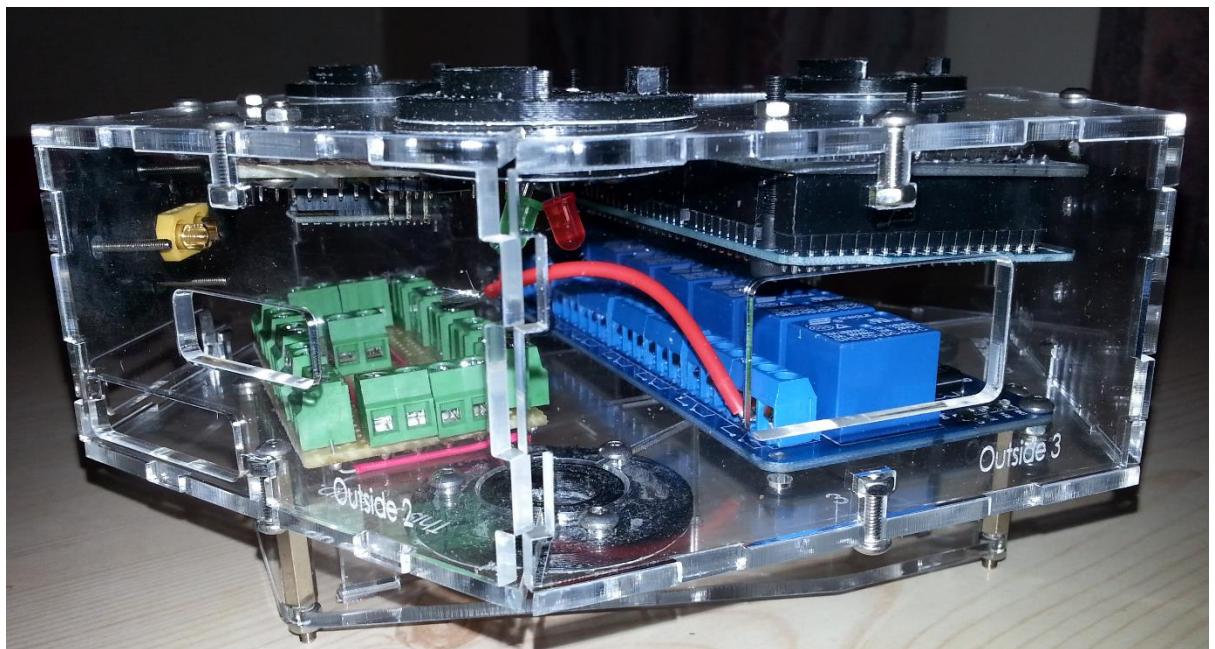


Figure 91: Post manufacturing integrated electronics housing.

### 6.2.3 Conclusion

Meeting the subsystem requirements was achieved by design and test.

1. The Arduino is able to operate autonomously for extended periods of time, provided proper software design. In addition, it is able to output 5 V and 3.3 V. Lastly, it is able to read analogue and digital inputs.
2. The batteries provide extended autonomy to the system: 11.38 min for continuous valve operation and 13.85 h for Arduino + peripherals.
3. The software design enabled the system to acquire position data at intervals smaller than 20ms. This was determined by testing normal software operation, described in Section 6.1.5. The selected gyros' accuracy was designed by the manufacturer to achieve 0.1 deg accuracy, value measured during testing.

4. The design of the communication subsystem was coupled with the electronics to enable just necessary 3 input to fully control the Hopper:
  - a. START – to start the Hopper
  - b. MECO – emergency stop of the Hopper
  - c. Throttle – control the throttle capability of the engine
5. A modular housing was designed and manufactured, able to contain all the necessary electronics.

## 6.3 Communication subsystem design

Author: Marian Daogaru

### 6.3.1 Previous communication

The Communication subsystem, commonly referred as comms, represents one of the most important subsystems in a remote controlled vehicle, such as the lunar hopper. It provides the means necessary to operate the hopper and to obtain data from it. In Phase IV, a dual link between the hopper and a remote controller was created. The remote used a PC as the main operating hardware coupled with an XBEE radio module (Figure 92). Alongside the hardware, a custom made software was used to operate the communication between the remote and the hopper. The program and the XBEE Module allowed 2-way communication between the remote and the hopper, with telemetry being able to be sent from the hopper.

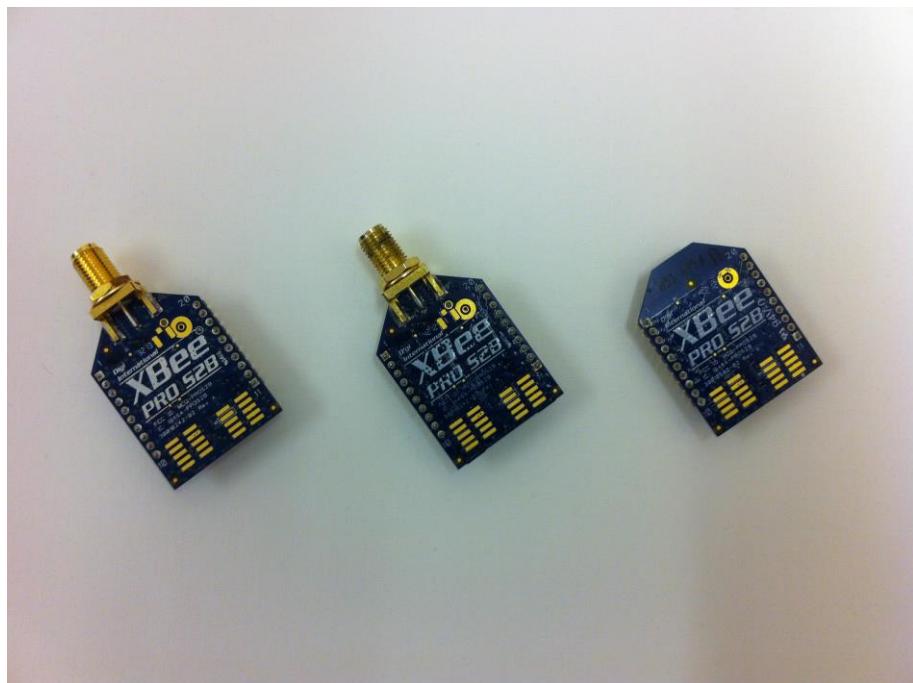


Figure 92: Different variation of XBEE modules. [4]

The hopper used a combination of two Arduinos and two XBEE modules to communicate. One XBEE paired with an Arduino MEGA were operating the main hopper software, while the other two components were only operating the MECO Command.

The overall design had considerable advantages in terms of communication potential between the hopper and the remote. Two-way communication was achieved. Pilot commands were sent to the hopper, and it was returning acknowledgement messages, along with task completion messages and inclination reports. Although these features were desirable, they were not mission critical. Using these additional features meant that the hopper software was slowed by the decoding, creating and sending new telemetry, during each short interval. With time sensitive function in the code, these features posed a serious operating risk. In addition, as noted by Phase IV team and testing done by Phase V, the software had a tendency to crash, becoming inoperable until a hard reset was applied to both the PC and the hopper. Due to time constraints, inadequate operating manual for the software, and the constant crashes detected, it was decided that this system was not be used in Phase V. An easier and more reliable communication method had to be created.

### 6.3.2 Radio Communication

For the purposes of this project, with current technology in embedded electronics, several options for reliable communication are available. Each method presents distinct advantages, ideal for different types of applications.

| Method          | Pros   | Cons   |
|-----------------|--|--|
| WIFI / Internet | <ul style="list-style-type: none"> <li>• No range restriction;</li> <li>• No line of sight issue;</li> <li>• Potential for unlimited data stream;</li> <li>• N-way communication;</li> </ul>   | <ul style="list-style-type: none"> <li>• Requires internet access;</li> <li>• Power intensive;</li> <li>• Slow data message transmission;</li> </ul> |
| Bluetooth       | <ul style="list-style-type: none"> <li>• Direct 2-way communication between paired devices;</li> <li>• Fast and secure;</li> <li>• Limited power consumption</li> </ul>  | <ul style="list-style-type: none"> <li>• Line of sight issues;</li> <li>• Short range;</li> </ul>  |
| Radio           | <ul style="list-style-type: none"> <li>• Potential for long range capabilities;</li> <li>• Fast and secure;</li> <li>• Limited power consumption;</li> <li>• Direct 2-way communication;</li> <li>• Small line of sight issues;</li> </ul> | <ul style="list-style-type: none"> <li>• Can be disturbed or jammed;</li> <li>• Signal decays with distance;</li> </ul>                              |

Table 13: Current determined methods of communication.

Considering the options presented in Table 13 and the launch location, along with the results obtained by previous projects with radio communication, it was decided to continue with a radio communication module. Moreover, due to time constraints and functionality, it was concluded that a “plug and play” type of communication device was optimal for the application of the project. “Plug and play” devices are ready to use electronical components that do not require software or hardware installation or manufacturing, besides the extensions to basic users’ needs.

#### 6.3.2.1 Pilot Output – Communication requirement

At current stage of the project, the hopper was remotely controlled by a pilot. To minimize the requirements on the pilot, it was decided that controlling the Hopper during the flight sequence using a remote controller with 3 basic outputs was the optimal solution:

- 1) START – this command will start the preflight checks in the hopper, calibrate all sensors and preheat the engine. Outputting 1 when engaged, and 0 when not;
- 2) MECO – Main Engine Cut-Off. Emergency stop of the software, and the hopper, if the pilot deems it fit. Outputting 1 when engaged, and 0 when not;
- 3) Throttle controller – Main engine control, throttling the response of the engine. Signal must be variable. Output is ranging from 0, for no thrust, to 676 or 1023 for full thrust when read by an analogue port on an Arduino supplying 5V.

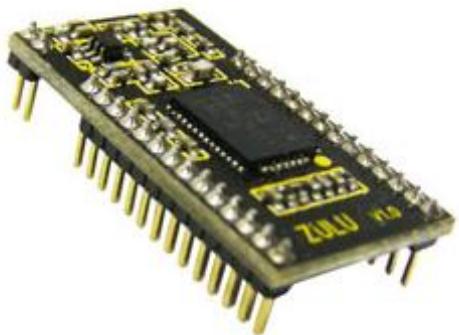
Considering these 3 outputs that have to be transmitted from the pilot to the hopper, a need for communicating both digital signals (ones and zero, high or low) and analogue signals (different voltage outputs depending on pilot actions) arose. While, in theory, Phase IV of the project was able to achieve these requirements, in practice the unreliability of the software and the equipment deemed these requirements impossible to achieve, as explained previously.

#### 6.3.2.2 Communication module choice

The choice for a suitable communication module had to meet all the previously mentioned requirements:

1. Easy to use – plug and play type
2. Use radio technology
3. Able to send both analog and digital signals
4. Able to send messages over long distances – at least 100 m
5. Low power consumption
6. Small size and low mass

Using these requirements, and considering suggestions from project supervisors, it was decided to use two ZULU-T868 Smart Radio Telemetry Modules [16]. The module is a transceiver, being able to act as a transmitter or as a receiver, depending of wire-up configuration at start-up. Thus, one module was used as transmitter and one as receiver.



*Figure 93: Zulu module. Courtesy to RF Solution.*

The ZULU module is able to transmit 2 analogue and 8 digital concurrently. Moreover, depending on the antenna used, meteorological conditions and line of sight, it can transmit data up to 2 km [16]. The module is considerably easy to use: commands are transmitted to a pin, the transmitter sends the command, and the receiver outputs the same command on respective pins. Its small size and low power consumption (18.5 mAh), combined with the ease of use after a breakout board is created, makes it an optimal choice for the project. In addition, although it is not a direct 2-way communication, with the receiver not being able to transmit data back, it incorporates a “message received” acknowledgement function; the receiver transmits to the transmitter that it has obtained the message.

#### **6.3.2.3 Zulu board design**

The ZULU modules were designed to fit a 1.27mm pitch. This meant that the distance between each pin (see Figure 93) was 1.27mm. The most used pitch for electronics however is 2.54mm. Thus, a breakout board had to be created, in order to allow the transition from 1.27mm to 2.54. In addition, the transmitter and the receiver require different configurations to operate their specific tasks.

Secondly, a breakout-board would allow the user to interact more easily with the module, and in a more reliable manner, rather than using potentially directly loose connections from the modules to a microcontroller. The pins from the ZULU would be easier to access if distributed to a normal pitch. The design of module did not permit for a true analogue output to be implemented directly. As such, the output was a Pulse-Width Modulation (PWM) signal. The output of a PWM is controlled by switching between high and low (input voltage and 0V typically), at a high frequency. A duty cycle describes the percentage of time when the output is high in the given period of PWM: a 75% duty cycle corresponds to the switch being high 75% of the period and off 25%; Figure 94 presents 3 different duty cycles. It can be seen, that for the same period, any value of the duty cycle can be attributed. 100% duty cycle corresponds with the output voltage = input voltage.

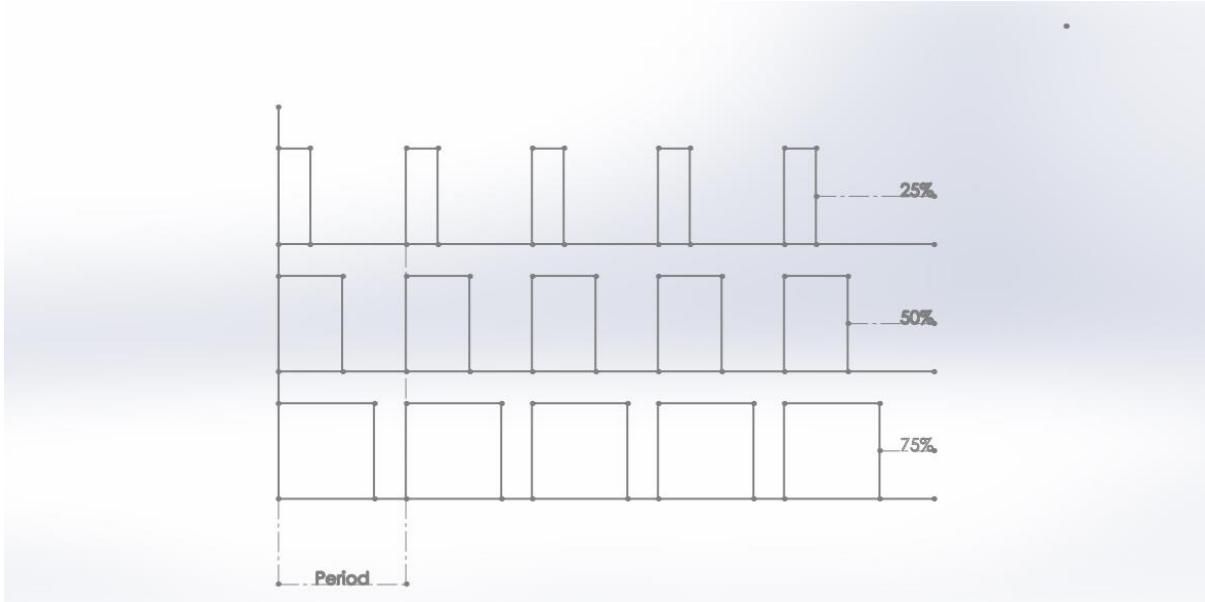


Figure 94: PWM signals for different duty cycles.

In particular cases, when the frequency of the PWM is extremely high, the receiving device (in this case the Arduino) would not have the capability to read such signal, directly reading it as analogue. However, as the PWM is not a true analogue signal, and the objective was to create a highly responsive device to the output of ZULU, a smoothing in the circuit was applied.

**A low-pass filter** was created to smooth the PWM incoming signal from the ZULU to the Arduino. A low-pass filter reduces the signal frequency, down to preferably half of the sampling frequency of the device. The low-pass filter allows only signals below a cut-off frequencies to pass, with higher frequencies being attenuated. With a 10 kHz sampling frequency on the “analogRead” command, a 5kHz signal frequency would be desirable. However, in order to reduce high frequency noise that could potentially perturb the engine during normal operation, an even lower frequency is desirable. The signal generated by the low-pass filter should be interpreted as a command from a human pilot. Consequently, it was decided that a cut-off frequency of 10Hz would allow the system to react properly to a human input. The simplest low-pass filter is an RC filter consisting of a resistor and capacitor. The signal comes through the resistor, then it is split between the capacitor connected to ground and a load. The capacitor’s reactance drops at high frequencies, the signal goes to ground, and blocks the low frequency signal, thus going through the load. The cut-off frequency equation is:

$$f_c = \frac{1}{2 \pi R C}$$

$R$  being resistance of the resistor and  $C$  the capacitance of the capacitor. Several standard resistors and capacitors were considered for the filter, and their response measured. Table 14 presents the values for  $f_c$  using different resistors and capacitors. It can be seen that a 10 kΩ resistor and a 1 μF capacitor provide the smallest value of  $f_c$ . Additionally, the response of each combination was tested for response time and noise reduction. All combinations displayed similar time responses. However, a response in noise attenuation was observed with the increase in cut-off frequency. As such, it was decided that the 10 kΩ resistor and 1 μF capacitor combination was optimal for this application.

| Resistor ( $k\Omega$ ) / Capacitor ( $\mu F$ ) | 0.68  | 1    |
|--|-------|------|
| 2.2  | 106Hz | 72Hz |
| 5  | 47Hz  | 32Hz |
| 10   | 23Hz  | 16Hz |

Table 14: Cut-off frequency for a low-pass filter using different RC combinations.

Following all previously discussed consideration, the breakout boards for the transmitter and receiver were created. Manufacturing was done using a 2.54 mm pitch prototyping board in conjunction with 1.27 mm to 2.54 mm pitch adaptor boards. In addition, the schematics provided by manufacturer (Figure 95) were used as a template for the design.

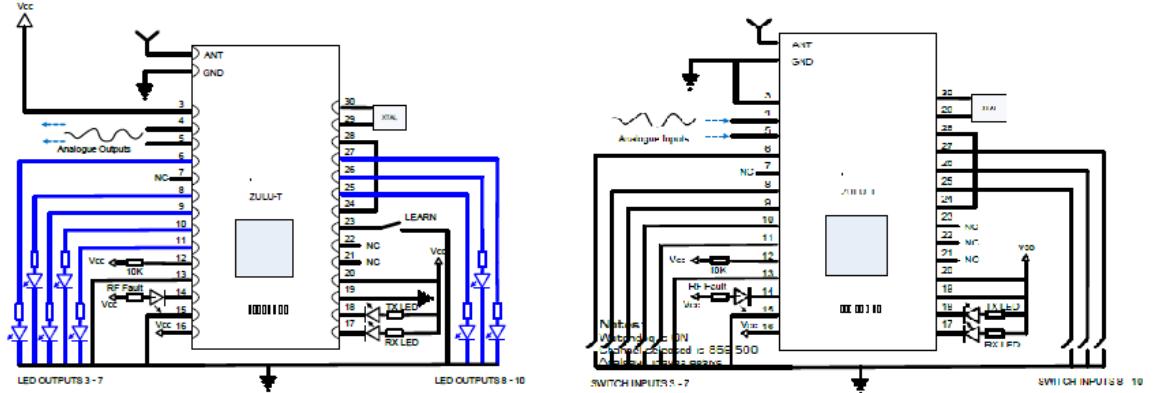


Figure 95: Pin mapping for ZULU module in receiver mode (left) and transmitter mode (right). [16]

The final model of the manufactured receiver and transmitter can be seen in Figure 96 and Figure 97. Appendix I – ZULU module breakout board detailed schematics. presents the detailed circuit diagram based on these figures.

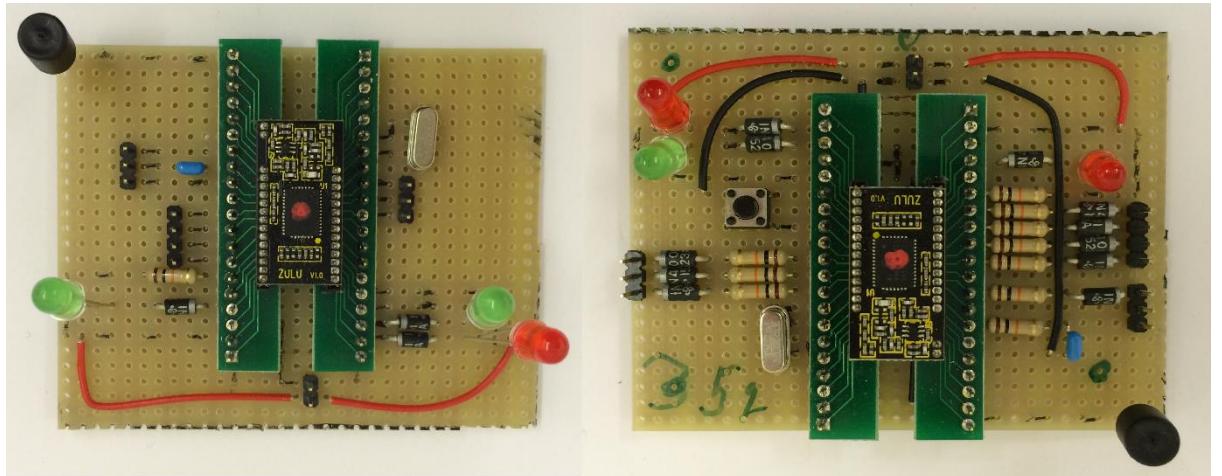


Figure 96: Top view of transmitter (left) and receiver (right) breakout boards.

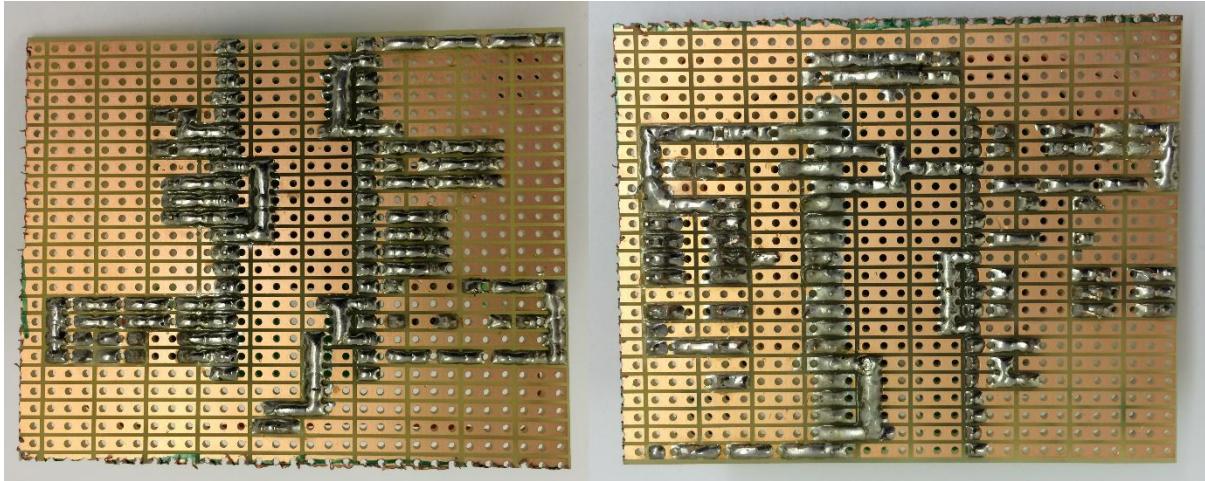


Figure 97: Back view of the transmitter (left) and the receiver (right) breakout boards.

#### 6.3.2.4 Zulu board design problems

While the design of the transmitter and receiver was relatively straight-forward, as explained previously (in Section 6.3.2.3), several problems were encountered after board production and during initial testing.

Firstly, the board designs were based on the schematics provided by the manufacturer [16]. In the technical specification, it was described that Pin 3 (Figure 95 left) must be connected to Vcc for Pin 4 & 5 to act as digital pins, or GND for Pin 4 & 5 to act as analogue pins. However, the receiver schematics detailed in the manual presented Pin 3 connected to Vcc, while mentioning that the said pins were acting as analogue, information which is contradictory to the other specification. During testing, it was concluded that the schematics was indeed faulty, and Pin 3 should be connected to GND for analogue.

A different problem was created by using a  $1 \mu\text{F}$  capacitor in the transmitter to smooth the analogue signal. This was recommended, again, by the manual. However, this capacitor was proved during testing that it was not smoothing the signal, it was damping the signal. As such, the transmitter was not receiving any input to transmit, even when it was supposed to receiver full analogue signal voltage. It was thus deemed necessary to remove the capacitor from the circuit, allowing real life data to be transmitted successfully.

Lastly, the module accepts as optimal voltage in 3.3 V, rather than 5 V typically. Moreover, due to voltage difference between the transmitter and receiver, it is particularly difficult to determine the exact output voltage of the 3.3 V output pin. As such, the response read from the ZULU module should be between 0 V and 3.3 V, due to ZULU technical limitations. Based on overall voltage in the Arduino (i.e. 5 V nominal might be 4.9 V) different analogue readings may occur for the same output by the ZULU: if the ZULU is outputting 2 V when the Arduino's voltage is at 5 V, the analogue value will be read as 410 bits. However if the same 2 V output is read when the Arduino is outputting 4.8 V, 427 bits are read by the analogue port. Thus, the 2.5 V reference voltage is used to determine the exact output of the module.

#### 6.3.2.5 Zulu Module Testing

Testing of the ZULU modules was done in 4 parts.

Firstly, the digital transmission capabilities were tested. The module is only required to send 2 digital signals at a given time. The signals are a HIGH position, sent when the pilot activates either START or MECO. As such, the system was not expected to be able to transmit pulsed digital signals. Initially, the transmitter was given similar commands as what it should be expected during flight, but pulling 2 pins high. The receiver was able to receive this change a reproduce it accurately. Consequently, it was determined that the ZULU modules were able to

fulfill the digital transmission requirements. However, in the interest of future developments in the project, pulsed digital signals were tested. Using an Arduino to simulate the signal, 0.5Hz, 1Hz, 2.5Hz, 5Hz, 10Hz and 25Hz pulsed signal were delivered to the module. Next, an oscilloscope was connected to the transmitter and the receiver to record the signals. Results from the oscilloscope indicate that the module can transmit pulsed signal up to 2.5Hz, as seen in Figure 98.

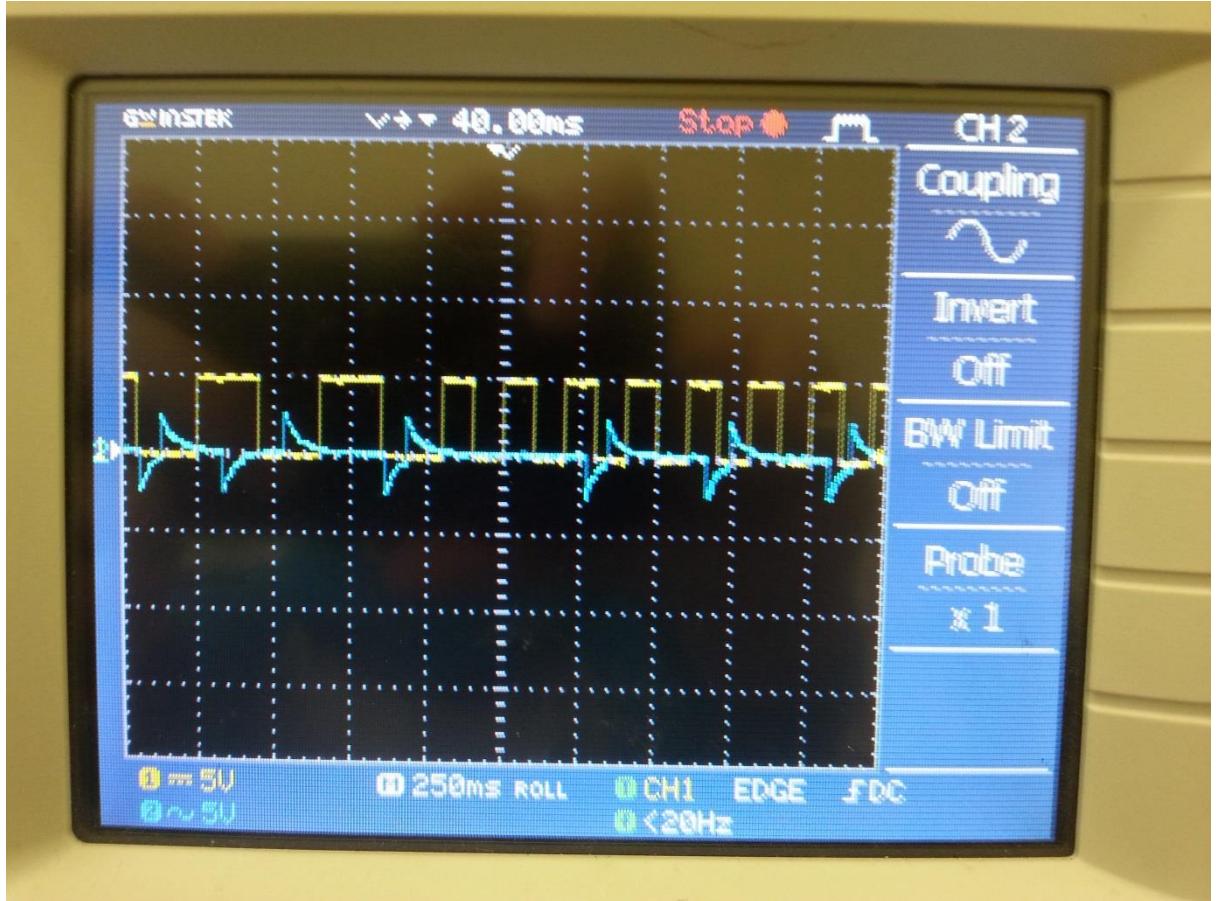


Figure 98: Voltage measurements of the transmitter (yellow) and receiver (blue) during pulsed signal testing.

After this frequency, signals are still being transmitted, but only a fraction of the signals are being received. However, for the purposes of the project, the module is capable of fulfilling the requirements.

The second test involved the testing of the analogue signal while the 2 modules were stationary at close distance (less than 20 cm apart). The purpose of this test was to determine the signal decay and the lag in the system. With the 2 assemblies powered on, firstly, the potentiometer was held at maximum voltage output. From Figure 99 it can be seen that while the output signal (blue line, the output of the potentiometer) has considerable noise a power drops (up to 17bits), the input signal was relatively constant. The input signal was also divided in 2 parts: an instantaneous value (green line) and an averaged value of the signal (red line), obtained after reading the input 10 times. For this test, it can be seen that these 2 values are almost identical, with a maximum 2 bit difference between them. The small difference between the general input and output voltages is due to the overall voltage across each individual Arduino. Next, in Figure 100, using the same setup, the potentiometer was slowly taken to 0 and back to full voltage, followed by a series of fast movements and staggered movements. It can be seen again that the values recorded are nearly identical, without any major discrepancies.

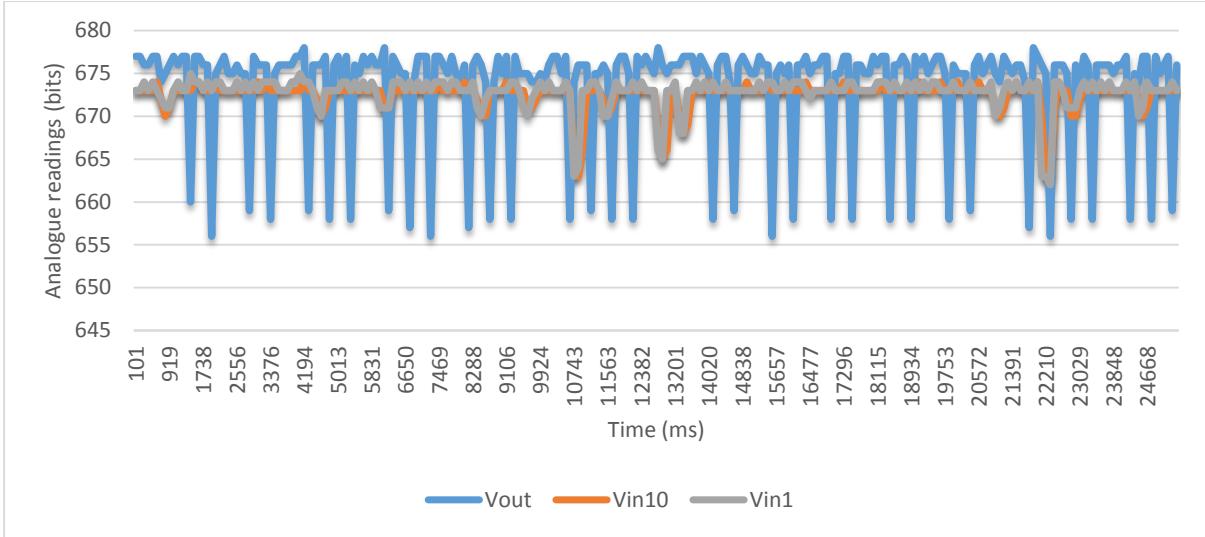


Figure 99: Voltage measurements for analogue transmission during stationary, maximum position test.

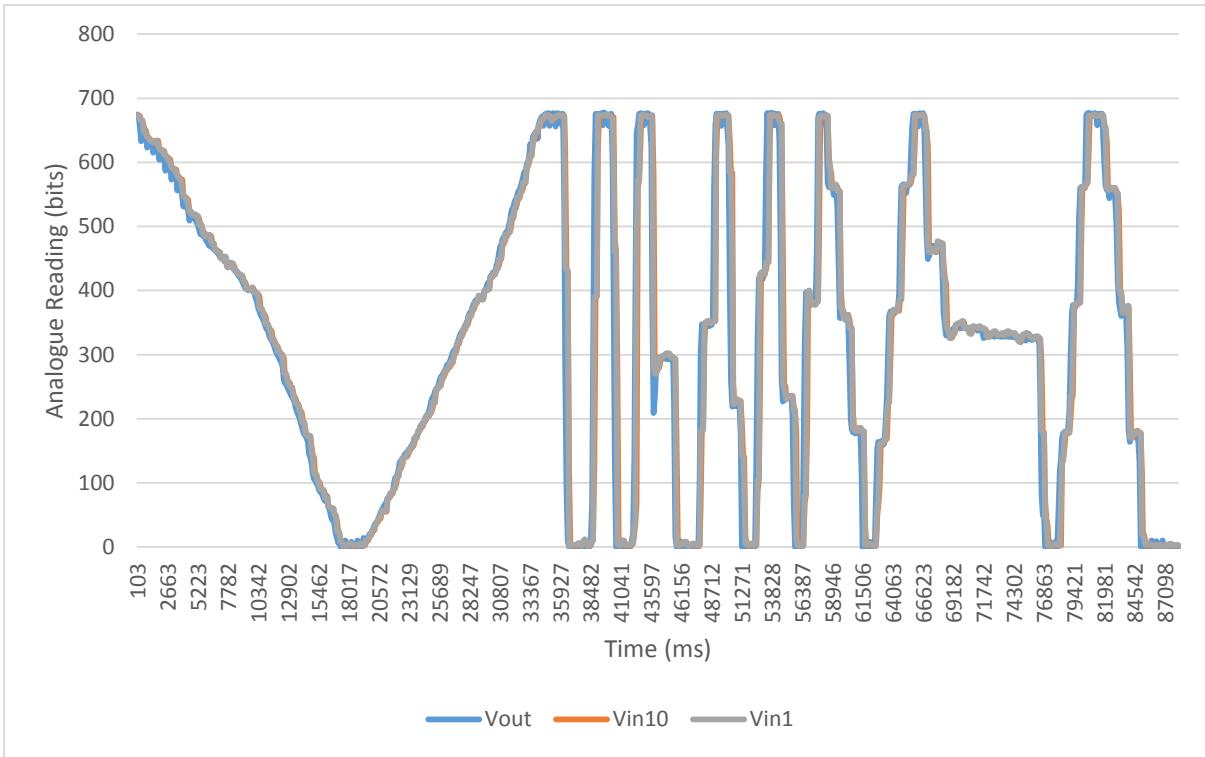
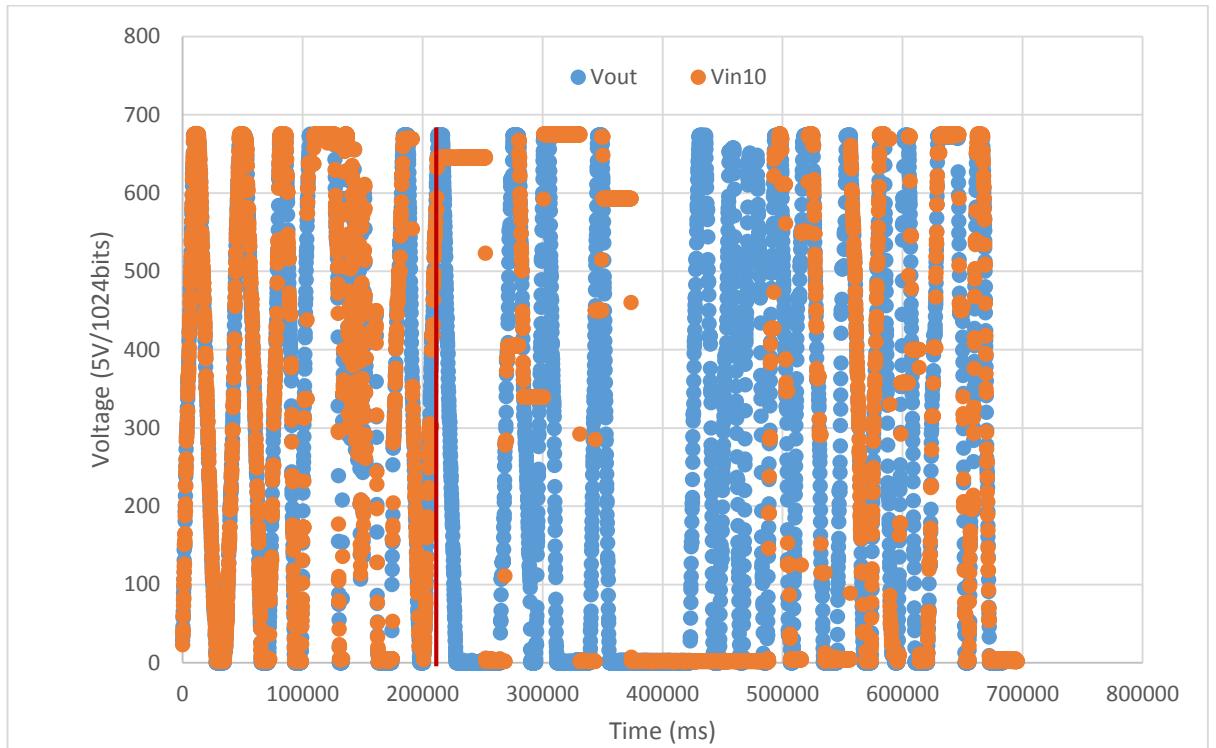


Figure 100: Voltage measurements for analogue transmission for variable positions test.

The third test involved a range test to determine if the current setup was able to operate at 100m. With the 2 modules attached to individual laptops, they were displaced by an initial distance of 70m. No loss of signal was detected, nor lag or signal decay, besides the expected difference in voltage. After this first success, the modules were displaced by 100m. Again, no loss of signal was detected, and no signal decay.

Lastly, the maximum range attainable by the current setup was tested. Using the 2 modules connected to 2 different laptops, they were displaced until signal loss was observed. The test was done in flight conditions: clear weather, open space. However, due to terrain constraints, with distance increase, elevation increase occurred. Moreover, several trees were disrupting the line of sight. Nevertheless, these were acceptable factors in range testing. As in previous experiments, the potentiometer was used to determine analogue signal. The potentiometer

handle was slowly moved from 0 to max voltage, and back to 0, with the motion repeated. Moreover, fast movements were also tested.



*Figure 101: Range test data for analogue communication.*

Figure 101 present the data during range test. It can be noted that the signals coincide until signal loss occurred. The input voltage (orange line) follows the output voltage (blue) with little deviation during normal operation. However, the signal loss point is not a fixed boundary, and thus, several spots of signal loss can be observed at 200000 ms mark (red line in Figure 101). Moreover, it can also be seen that the signal loss was not a complete loss, with sparse messages being received after maximum distance point. Moreover, after returning to signal area, the receiver was able to receive back messages. The final distance at which the signal was considered unusable was determined as 182m. This distance is, as mentioned before, not considering the increase in elevation. Furthermore, the distance was affected by the objects interfering with the line of sight. Lastly, the maximum distance obtained was also a factor of the antenna used: in this experiment, the smallest recommended antenna was used, with a 0dB gain, low power, and Omni-directional specifications. Consequently, the test was a success, demonstrating that the assembly is capable of sending data over long distances with a consistent accuracy.

### 6.3.3 Communications Conclusion

Overall, the reliability of the communication subsystem was considered to be improved. Firstly, the subsystem is deemed more reliable, with no software crashes or freezes encountered during the entire manufacturing and testing phase. Moreover, the system is considerably easier to use. As such, potential improvements to the system can be implemented relatively easy, and the learning curve for the next phases of the project is shallower, compared to this phase. Lastly, the current comms is significantly lighter, due to removing one Arduino Uno from the hopper plus the necessary peripherals, less power demanding, and smaller in size.

## 7 Pilot Training Simulator

Author: Adam Elkins

### 7.1 Background and Requirements

Based on experience from earlier phases of the lunar hopper project, supervisors recommended development of a real-time simulator to train team members to control the hopper in flight safely. This simulator would not be required to simulate the action of the RCS thrusters, but otherwise needed to simulate as many of the different operating characteristics of the hopper as possible, including the reduction of mass as propellant is expelled and engine thrust variations over time.

The Lunar Hopper is designed to be controlled remotely by a pilot; the pilot must increase thrust so that the hopper takes off, reduce it to a hover for a few seconds then reduce thrust further to bring the hopper to a soft landing. During the test flight various properties of the hopper will be changing, including propellant mass and engine thrust as noted above, so a simulator was required to firstly identify who in the team would be best at flying the hopper and, secondly, to train that person to fly the hopper well. The simulator therefore had to incorporate these characteristics in its calculations but, since the hopper's attitude control thrusters would operate automatically, the simulator was only required to simulate the vertical axis of flight.

The simulator itself was to comprise software that would run on an Arduino, independently of any other hardware. During the test flight, the pilot will control the thrust from the hopper's engine through changes in the oxidiser tank pressure: a change in tank pressure would result in a change in oxidiser flow rate and a consequent change in thrust. This pressure demand would be converted into an electrical signal to the Arduino by a potentiometer; in the simulator the Arduino software then needed to simulate the response of the hopper and relay the result to the pilot. Two analogue voltmeters were chosen for this. Additional controls were added to start and reset the simulator and an internal battery compartment was also provided.

The simulator software would perform the calculations for thrust and weight and integrate the equations of motion numerically over a small increment of time  $dt$ . In each cycle of the code the sum of forces was calculated and the resultant expressed as the acceleration of the hopper; this acceleration was then used to solve the equations of motion. The altitude was calculated by solving the equation

$$s = s_0 + ut + \frac{1}{2}at^2$$

while the change in velocity was calculated using

$$v = u + at.$$

These were implemented in the software by computing  $s += v*dt + 0.5*a*dt*dt$  and  $v += a*dt$  (using the compound notation where  $x += b$  is equivalent to  $x = x + b$ ). They were computed in this order so that the altitude equation could be computed based on the velocity computed by the previous cycle, with the velocity equation updating this value afterwards. The code was initially developed and tested in C since this language is very similar to the language used by the Arduino, before being copied to and installed on the Arduino for further development and testing. A full copy of the code is included in Appendix I.

#### 7.1.1 Analysis of the problem

A free body analysis of the hopper showed that there were two phases of 'flight' that needed to be modelled. The first was while the hopper was on the ground and engine thrust was less than the hopper's weight. In this case the hopper would be in static equilibrium; the engine thrust

would be less than the weight of the hopper and effectively reduce it, while a reaction force from the ground would balance the remainder.

In the second case the engine thrust would be greater than the weight and the hopper would start to lift off; static equilibrium would no longer apply and the hopper's flight would be described by the equations of motion. The onset of this phase would occur while the hopper was on the ground, either when thrust was increased to match the hopper's weight or as the hopper's weight reduced to less than thrust as propellant was expelled.

### 7.1.2 Hopper Characteristics

#### 7.1.2.1 *Change in mass as propellant burned*

As noted above, the hopper's mass would reduce during flight as propellant is burned and expelled from the engine, reducing its weight; if the throttle was set to a level just below that where the hopper would take off the hopper would stand on the ground for a few moments with its engine burning before slowly lifting off as the weight dropped below thrust. This was relatively simple to model in the software: an initial propellant mass was declared; the simulator calculated an element of propellant mass by multiplying the propellant flow rate (in kg per second) by the time increment  $dt$ , and subtracting this from the total propellant mass. An if statement checked the remaining propellant mass and, if it was calculated to reach zero, returned whatever propellant was calculated to remain and returned zero thereafter.

#### 7.1.2.2 *Propellant ullage*

The engine developed by the hybrid rocket team for phase 4 used a bang-bang control system to increase the pressure in the oxidizer tanks, but relied on a blowdown mode to allow the pressure to drop [4]. In this approach an increase in demand pressure would cause the pressure valve to open, resulting in an almost instantaneous increase in oxidiser tank pressure and a corresponding increase in thrust; however, if the pressure demanded was reduced it would be necessary to wait for the pressure in the tank to reduce as oxidiser is forced out to the engine. This complicated the simulation as while the thrust would increase almost instantaneously in response to the pilot commanding higher pressure, there would be a delay in the response to any command to reduce pressure as the gas in the oxidizer tanks expanded. This potentially had implications on whether or not the hopper would even be controllable and a full simulation was needed to determine if any further action was needed. Although the purge valve incorporated in the phase 4 design (and retained in phase 5) could be used to vent tank pressure quickly if needed this had not been done due to concerns about the effect of the additional delta-V this venting could have on the hopper while in flight along with the loss of pressurised gas that would result.

The gas expansion in the oxidiser tanks was assumed to be adiabatic— $PV^\gamma = \text{constant}$ —and this initial adiabatic constant was calculated. The volume of expansion for each time increment was calculated by multiplying the element of oxidiser mass consumed by its density ( $1370 \text{ kg m}^3$ ), assuming the oxidiser flow rate was proportional to tank pressure. With this and the adiabatic constant known the tank pressure, which will now have dropped, was calculated by

$$P = \frac{\text{constant}}{(V + dV)^\gamma}$$

and this new pressure was used to calculate the flow rate, and thus the drop in thrust and pressure, in the next cycle of the code.

#### 7.1.2.3 *Monoprop mode*

From the command being given to open the valves allowing oxidiser into the engine, there is a short time before ignition in which the engine 'fires' on oxidiser only before the fuel grain itself starts to burn; this is known as monoprop mode [4]. Although some thrust will be generated in this mode it is too small to cause the hopper to take off so does not need to be simulated

directly, however oxidiser will be expelled during this phase reducing vehicle weight and there will be a short period after the valve being opened before the hopper will be able to respond to throttle commands from the pilot.

## 7.2 Software

### 7.2.1 fuelrate()

The fuelrate() function was written to read the value of the analogue pin connected to the potentiometer and express it as a pressure demand setting. This function was also extended to express the blowdown mode of the engine along with an initial approximation of thermal effects on the fuel grain. In order to achieve this the total propellant flow rate (flowrate) is divided into flow rates for oxidiser (oxrate) and HDPE fuel (HDPERate).

The function first reads the value of the pin (identified by the pre-declared variable throttlePin) using analogRead(), which returns a value between 0 and 1023; this value is then multiplied by 40 and divided by 1023 to obtain a pressure demand in bar.

An if statement evaluates if the demand pressure is greater than the current simulated pressure and if so, raises the current pressure to the demand pressure (simulating opening of the pressure valve). This simulates the near-instantaneous increase in pressure in the oxidiser tank when the demand is increased. The function then calculates the adiabatic constant of the gas in the tank, obtains the oxidiser flow rate (modelled as proportional to pressure), computes the volume taken by the oxidiser expelled in each increment of time, adds this volume to the gas volume and calculates the new tank pressure from the adiabatic constant obtained earlier. This happens whether the earlier if statement evaluates to true or not, since oxidiser flows out of the tank under both conditions.

The monoprop mode is also simulated in this function by estimating the time that will elapse between valve opening and fuel grain ignition; during this time the fuel rate and  $I_{sp}$  are set to 0 (as the engine will not produce enough thrust for flight and HDPE combustion does not occur); this is achieved by another if statement. After this time has elapsed—in the else branch of the if statement—the fuel flow rate is calculated based on data obtained from the engine test firing carried out by the hybrid development team. After the if branches are evaluated the total flow rate is calculated and returned.

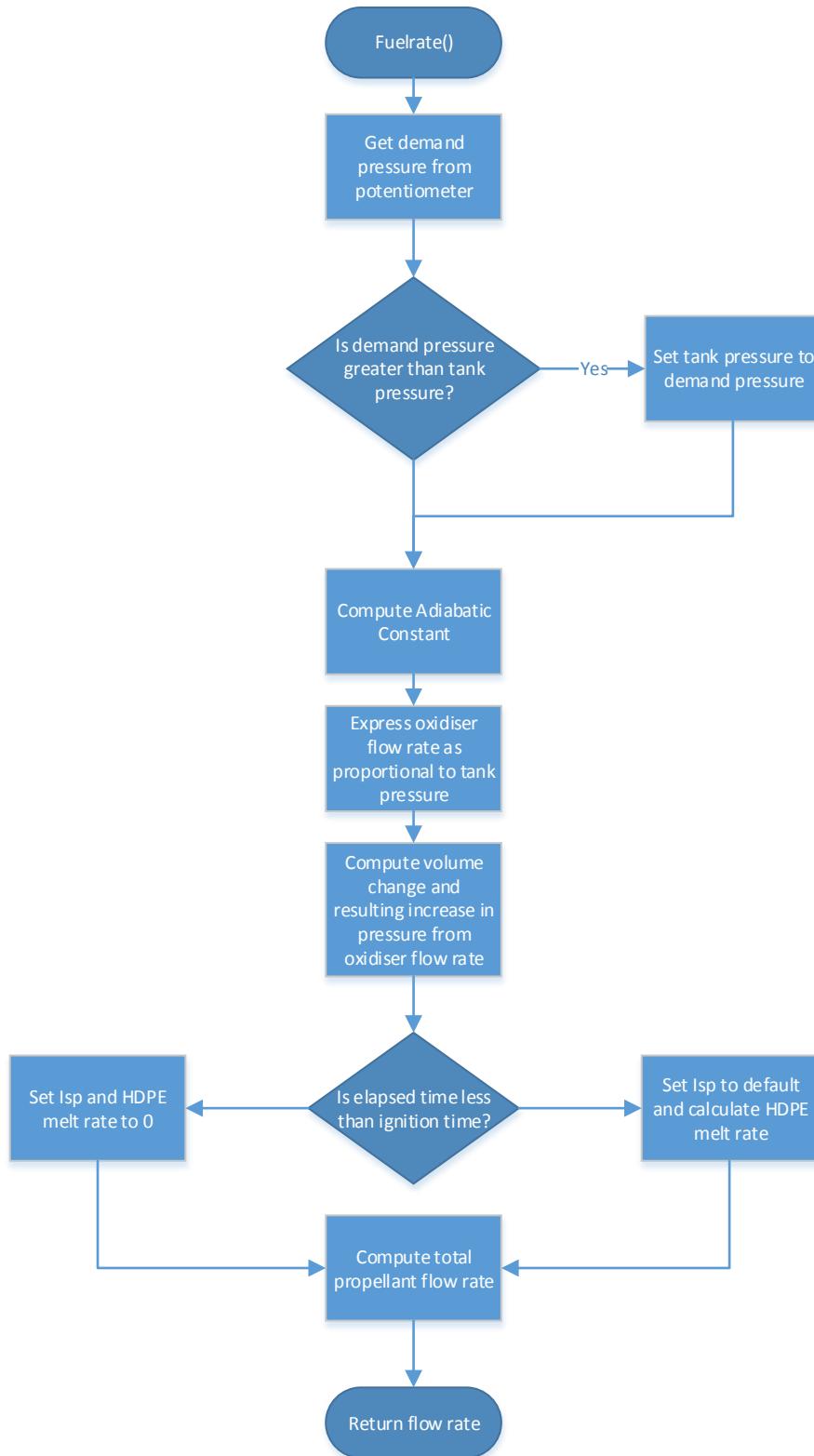


Figure 102: flow chart of processes within fuelrate()

### 7.2.2 simulator()

The `simulator()` function carries out most of the calculations for the simulator itself. The function first calculates the element of propellant to be used for each increment of time by calling `fuelrate()`, then calculates one of two things: if the propellant remaining is less than the propellant to be used the simulator substitutes in the remaining propellant mass then sets the remaining mass to zero; otherwise, the simulator subtracts the propellant to be used from the total propellant mass, thus simulating the consumption of propellant over time.

The simulator then calculates the behaviour of the hopper itself. Two approaches were tried for this. The first was an implementation of the rocket equation in its differential form to calculate an element of velocity change  $dV$  due to thrust. This approach required that equivalent velocity changes were calculated for both mass and ground reaction, which in the case of ground reaction at least, were unphysical.

The second approach calculated forces themselves. First the thrust was calculated from the  $I_{sp}$  (obtained from the hybrid rocket team's test),  $g$  and the propellant flow rate, then the weight of the hopper was calculated from the fuel and structural masses. A pair of nested if statements determined if the hopper was on the ground and if the thrust was less than the weight; if both of these were true the ground reaction was then calculated as the difference between thrust and weight, otherwise it was set to 0.

The simulator then calculated the hopper's resulting acceleration by summing thrust and weight and subtracting ground reaction, then dividing these by the total mass (fuel plus structure). If a ground reaction had been calculated the sum of forces, and thus the acceleration, would have been zero; this simulated the static equilibrium of the hopper standing on the ground. If the ground reaction was zero, the acceleration would be the resultant of the thrust and weight. This acceleration was then fed into the basic equations of motion to obtain the resulting change in altitude and acceleration using compound operators.

Testing the two methods showed that they produced identical results; since the first method was unphysical and had some confusing notation it was discarded and the second method was carried forward for further development.

An if statement was introduced to check the computed altitude and set it to zero if it became negative to avoid unphysical results. Within this if statement another if statement was used to check the velocity and set it to zero if it was not already, thus representing landing (or a crash) and avoiding another unphysical result. After these if statements the calculated elapsed time was increased by the pre-set increment.

The final part of this function dealt with outputting the results (altitude and velocity) to PWM pins to display using the voltmeters. This section used several if statements to limit the domain of the variables to suitable values ( $0 < s < 5 \text{ ms}^{-1}$  for altitude,  $-2.5 < v < 2.5 \text{ ms}^{-2}$  for velocity) then scaled the values to suit the voltmeters and sent the results to the appropriate PWM pins using `analogWrite()`.

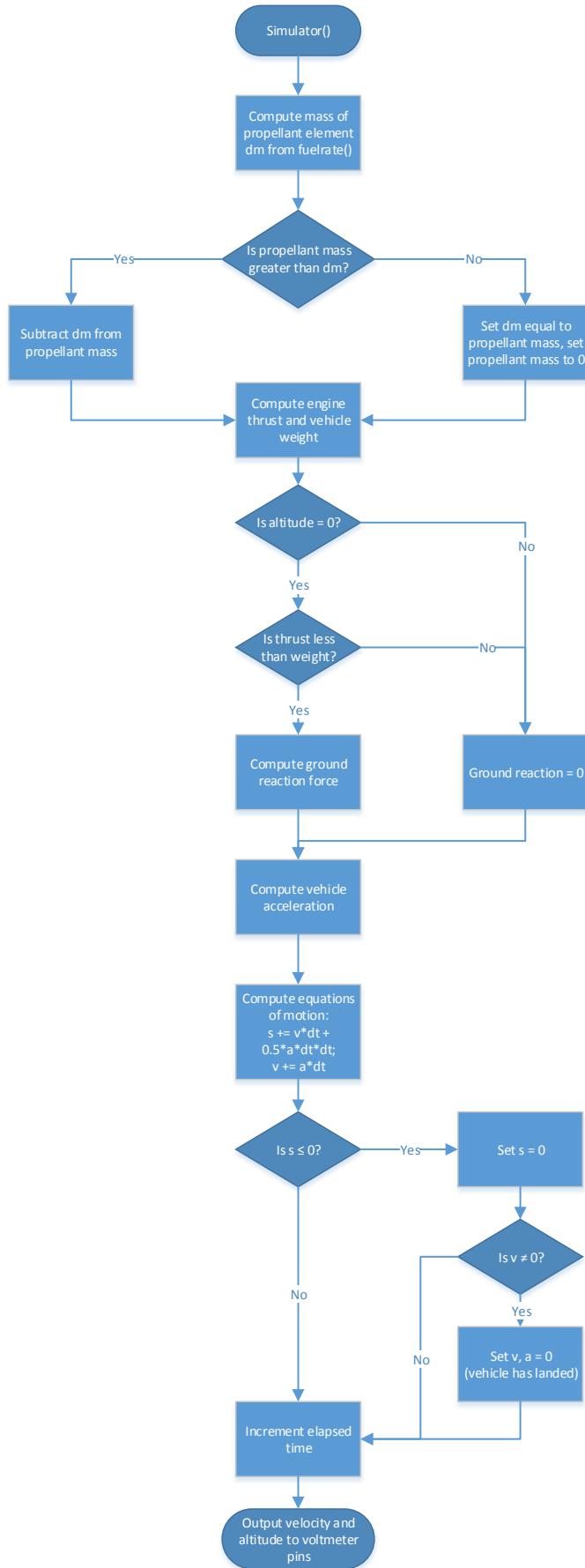


Figure 103: flow chart of functions carried out in simulator()

## 7.3 Electronics

### 7.3.1 Electrical Components

A linear slide potentiometer was selected as the input control for the throttle. This potentiometer provided 100mm of movement and was designed for panel mounting; it also had two markings on the actuator for 2mm holes to be drilled through to support the throttle handle.

Two 5V panel mount voltmeters were sourced to provide output readings for simulator velocity and altitude; 5V meters were required to match the output of the Arduino's PWM output pins. They were tested to confirm that they had sufficient impedance to work with the Arduino; this impedance was also deemed sufficient to smooth the PWM output without additional components. The voltmeters provided screw terminals, to which solder tags were attached to allow electrical connections to be soldered on.

A cable was obtained that could connect the USB port on the Arduino to a USB socket on the exterior of the simulator; the exterior (socket) end was designed to mount to a panel with two screws. This was done to allow the simulator firmware to be updated without having to open the simulator, and to optionally power the simulator from a computer or USB charger should a 9V battery be unavailable. If required, this port could also be used to return simulator data to a computer for further analysis.

A BX0023 Bulgin panel mount battery holder was obtained to provide a battery compartment to the simulator. This holder provides a drawer-type compartment that can be opened with a screwdriver, along with solder tags for electrical connection, and holds one PP3 9V battery to power the Arduino when not connected via the USB port. A 2.1mm 9v jack was also obtained to connect the Arduino to the 9v battery.

A single-pole, single-throw rocker switch was obtained to enable the power from the battery to be switched on and off. This particular switch was designed for 250V applications but was chosen as it was more cost-effective than alternatives; it is designed as a snap-in mounting and has large markings indicating which positions are on and off.

Two push-to-make buttons were sourced from Maplins, one in green and one in red, to provide controls to start and stop/reset the simulator. These mounted in a 16mm circular hole and had solder tags for electrical connections. Like the on-off switch, they were rated for 250V applications and were chosen as being more cost-effective options.

The number of electrical connections running into the Arduino required a board to be constructed to allow multiple components to connect to either 5V or ground. In addition, the push buttons' output pins needed to be connected to ground through resistors ( $10\text{k}\Omega$  resistors were chosen for this). This is because the signals from push buttons are often slow to return to 0V after being pushed when connected directly between 5V and the Arduino's input terminal; it is necessary to force their signal back to 0v by connecting their output pins to ground but to do this without short-circuiting them a resistance is needed to create a measurable potential difference. The board was made from a spare piece of stripboard and a pair of  $10\text{k}\Omega$  resistors that were surplus from other work on the hopper.

### 7.3.2 Housing

A housing was designed to contain the Arduino and mount all the components. This required that the voltmeters, buttons and potentiometer be mounted on top; the battery compartment, USB port and switch were originally to be mounted on the rear.

A housing design was developed that was to be produced from folded mild steel, however for expediency an initial housing was designed to be assembled from laser-cut MDF. This would both enable the components to be test-fitted to catch any issues arising, and to provide an initial housing to allow the simulator to be tested. The housing was a simple box made of flat panels,

with the method used to connect the panels of the instrument box on the hopper itself adopted for the simulator housing.

The initial test fit showed that there was insufficient space in the top panel design for the two voltmeters to fit side by side, and that the holes cut for the push buttons were slightly too small. It was also found that the volume taken by the voltmeters under the panel prevented the battery compartment and other components on the rear panel from fitting in. A second top panel was produced and the front and rear panels swapped round to correct these errors.

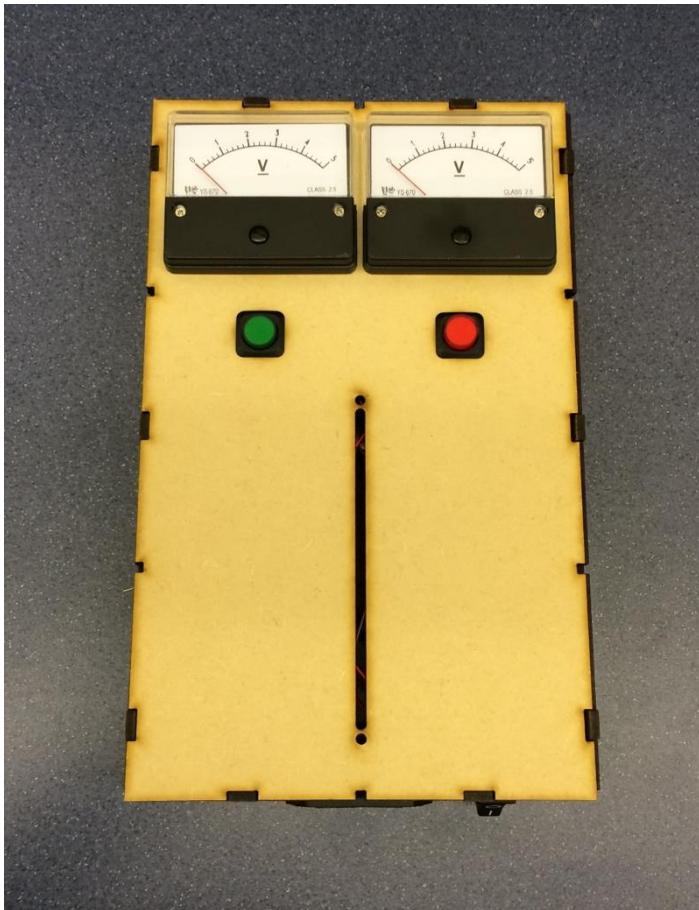


Figure 104: Top of the prototype simulator housing with voltmeters and control buttons in place but awaiting the potentiometer



Figure 105: front panel of the prototype housing showing battery compartment, USB port and on-off switch

## 7.4 Accuracy Analysis and Testing

In order to ensure the accuracy of the simulator's software, a comparison was made of the results obtained from the simulator code to the results of analytical solutions for the hopper's flight. The standard equations of motion could not easily be used for this as the acceleration would be continually changing with weight, but a solution involving the rocket equation allowed the final velocity to be calculated for a fixed thrust by calculating the  $\Delta V$  due to the mass of propellant used (with an initial velocity of  $0 \text{ ms}^{-1}$ ) and subtracting the gravity loss; this was then compared with that calculated by the software. This was done before the software was extended to include the blowdown mode and thus is based on the engine providing constant thrust.

The rocket equation was used to calculate the  $\Delta V_T$  due to engine thrust, assuming constant thrust and immediate take off. At this point the software used assumed values of 220 seconds for  $I_{sp}$ , a total mass of 38 kg, a propellant flow rate of  $0.18 \text{ kgs}^{-1}$  and a flight time of 12 seconds. To do this the change in propellant mass was calculated; since the thrust was fixed, so was the propellant flow rate and this was multiplied by the flight duration to obtain the mass difference required and incorporated into the rocket equation as follows:

$$\begin{aligned}\Delta V_T &= I_{sp} g_0 \ln\left(\frac{m_0}{m_b}\right) \\ \Delta m_p &= \dot{m}t \\ \Rightarrow \Delta V_T &= I_{sp} g_0 \ln\left(\frac{m_0}{m_0 - \dot{m}t}\right).\end{aligned}$$

The gravity loss  $\Delta V_g$  needed to be subtracted from this and was calculated as

$$\Delta V_g = g_0 t$$

and the final velocity of the hopper was obtained from these two equations

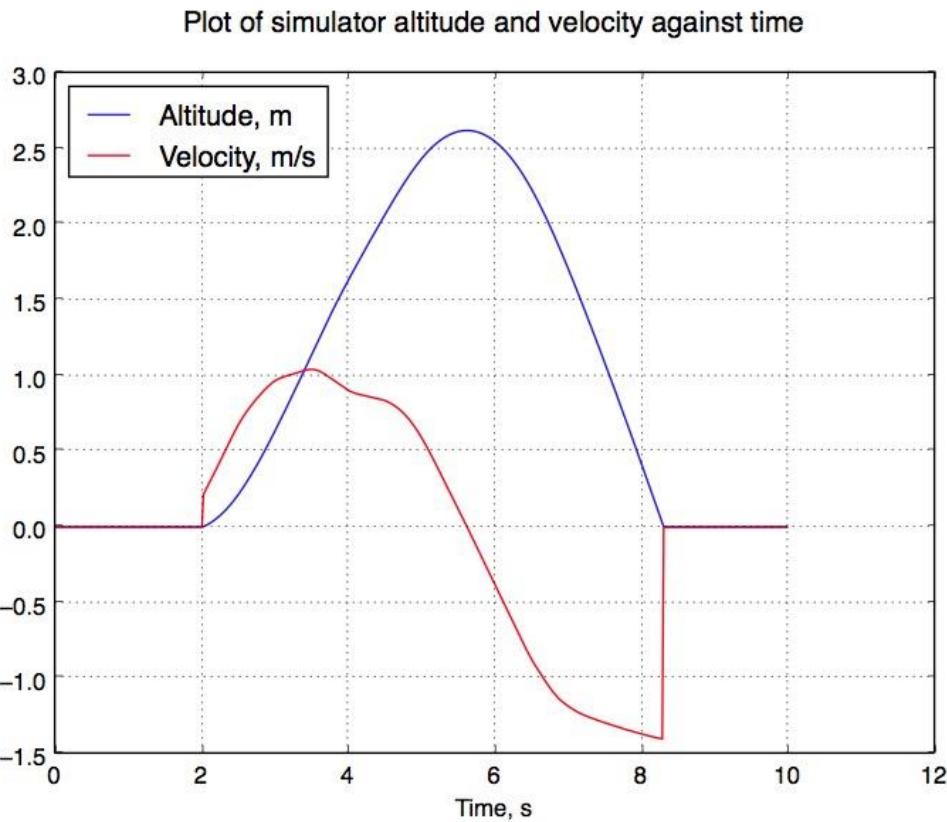
$$\begin{aligned}\Delta V &= I_{sp} g_0 \ln\left(\frac{m_0}{m_0 - \dot{m}t}\right) - g_0 t \\ \Delta V &= 220 \times 9.81 \times \ln\left(\frac{38}{38 - 0.18 \times 12}\right) - 9.81 \times 12\end{aligned}$$

which gave a final velocity of  $8.581 \text{ ms}^{-1}$ . This compared well with the final velocity returned by the simulator software at the time, of  $8.588 \text{ ms}^{-1}$ , a difference of only 7 millimetres per second after twelve seconds, and was deemed satisfactory.

The altitude could not be readily calculated from this but as it was calculated by the software from the same acceleration that was used to calculate the velocity it is assumed that the altitude is also calculated accurately.

## 7.5 Simulator Performance

Practical experience with the simulator showed that the best way to fly it was to start with the throttle at about 75% and bring it up slowly until the altitude started to increase, then pull the throttle back down slowly to bring the altitude back down again, keeping the velocity low at all times. This ensured that the hopper neither ran away nor landed too hard, keeping both altitude and velocity within acceptable limits. Initial attempts at this during development showed that this was achievable with concentration, care and practice. The effect of the blowdown mode was noticeable, slowing the simulator's response to commands to reduce tank pressure and thrust, but this effect was small. With practice and careful flying it should not pose a problem for flight control. A plot of the altitude and velocity computed by the simulator from a practice run is included below.



*Figure 106: A typical altitude and velocity profile obtained from the simulator during testing.*

## 7.6 Summary

The simulator successfully implements a numerical solution to the equations of motion of the hopper, and accurately models the reduction in mass (and corresponding effect on the hopper's flight) due to propellant being expelled from the engine. This enables the changing acceleration to be modelled (as the hopper's weight changes), which is important as the thrust level needed to keep the hopper steady or to land it will reduce during flight. The simulator also implements a solution to the engine's blowdown mode, which will affect the response of the hopper to the pilot's pressure demand, and enables pilots to train themselves to fly the hopper and prepare themselves for these characteristics.

The housing provides mounting points for the necessary controls, including a potentiometer for the throttle, and for voltmeters to display the calculated altitude and velocity of the hopper during the simulation. The inclusion of a USB port on the housing enables the simulator software to be updated without disassembling the simulator and the battery compartment allows the simulator to power itself independently of a computer or other power supply.

## 8 Test Flight Plans

Authors: Marian Daogaru & Alison Dufresne

The Lunar Hopper Project, as a Group Design Project, is aimed to simulate the industry approach in all stages of the project. Following completion of producing Mk. II, the next stage in the project consisted of launch test.

While a successful launch could be done by activating the engine and achieving VTVL, a test plan in order to abide by safety regulations is necessary. In addition, a test plan is required in order to have clear aims of what the test is aiming to achieve, declare if it was a successful test, and the procedure to follow during test.

### 8.1 Objectives

The test represents the final stage of the current Phase of the project. A VTVL test was planned for the project after report deadline, due to time constraints and logistics.

Several objectives were determined for the maiden flight test, based on the overall design.

1. Successful integration of subsystems during field assembly.
  - a. Nominal operation of each subsystem during full system integration.
  - b. The hopper must be able to support its own mass and maintain normal vertical position in the test facility.
  - c. The hopper must be able to achieve communication with control room, while stationed in the launch bay.
2. Successful completion of subsystem tests while stationed in the launch bay.
3. The hopper must be able achieve successful remote engine ignition.
4. Upon launch, the hopper must achieve 1m flight altitude.
5. The hopper must achieve 1s of stable hovering at final flight altitude; optimal 1m.
6. The hopper must perform a successful power descent.
  - a. The success of the power descent must be determined based on descent trajectory, touch-down velocity, stability during descent, and integrity of the vehicle upon touch-down.
  - b. After touch-down, the hopper must remain attached to the ground.
7. The hopper must maintain structural integrity during the entire flight phase.

### 8.2 Constraints

Due to the nature of the project, severe constraints were applied to the project. The overall project includes the use of a rocket engine. The use of rocket engine is a dangerous activity, without proper equipment and facility. Consequently, the first major constraint is due to our propulsion system. A suitable test facility must be used to be able to safely test the hopper take-off capabilities. This facility must be able to resist high temperature exhaust gases, of up to 2500K. In addition, as the hopper is expected to lift-off, a reasonable space is required to allow proper flight operation. Lastly, with a high risk factor involved in case the hopper will malfunction during flight, a safe zone with at least a 50 m radius around the Lunar Hopper, with no structures or traffic, is necessary. Considering the facilities available in the University of Southampton, no suitable local location is available. Consequently, the flight tests shall be done at an external location, identified as the Rocket Propulsion Establishment, Westcott, in Buckinghamshire.

The Lunar Hopper is expected not to exceed a flight altitude of 2m. Due to CAA regulations, and in the interest of safety, a physical constraint shall be applied to the hopper, to limit the flight altitude in case of malfunction. This constraint shall be attached to the launch platform and the hopper.



Figure 107: K2 Rocket Propulsion Establishment. Image courtesy to Wikipedia.

### 8.3 Logistics

The first issue of logistics is transportation of personnel and Lunar Hopper to the test facility. Fortunately the team have two cars between them which will accommodate the team members with the disassembled Lunar Hopper whilst supervisors have agreed to use their own transportation. As mentioned in Section 5.3.5, an assembly manual was created (see Appendix X). This was inspired by a failure due to poor assembly but will also be incredibly useful on the day of test flight. Along with the manual, various spare components (list in Appendix X) will be brought along in case of component failure. A recording of the launch must be made for outreach purposes and therefore the cameras, and their accessories, which were procured in previous phases will be brought along and must be set up. The camera model was chosen due to its small size, low mass and ease of attachment. One camera will be mounted on the Hopper pointed at the team while another will be positioned outside the safe-zone aimed at the Hopper.

Prior to deciding on a launch date, the Lunar Hopper systems must pass flight readiness tests. Pending system validation are PDS and ACS. In order to ensure a safe launch, there are certain flight-readiness checks which must be carried out again after reassembly. The main systems which will need function verification on the day, are comms and PDS (as structural issues will become obvious during assembly). The two tests which much be carried out on site are;

- MECO command transition and acquisition
- PDS Leak test

Once these have been achieved, the set up and assembly will continue until the Hopper is to launch. The launch procedure is as follows;

- Evacuate personnel from safe zone
- Start video capture
- Send ‘start’ command
- Wait for engine pre-heat
- Carefully throttle the engine up
- Once Lunar Hopper risen from the ground cease increasing throttle
- Attempt to keep the Hopper hovering above the ground for a few seconds, should altitude continue to increase, begin lowering throttle
- Lower throttle slowly and gently

Though these logistical have been somewhat concisely described, it is not expected that assembly and pre-flight testing will be a short process. To gauge the expected time a full assembly and testing should be practised. Additional time needs to be factored in which should be roughly 20% of the time taken, including the travel time. Budgetary considerations have been made in case that there are time restrictions which require us to stay overnight near the test facility to cut-out travel time. Health and safety paperwork is to be acquired from the University which will then extend its insurance to the test facility for the flight.

## **9 Conclusions**

The main objective of this phase of the project was to ‘ruggedize’ the Mk II design. This was pursued as the primary objective, ensuring that with or without flight testing, Phase V would be passing down an improved Mk II to Phase VI. This objective has been achieved through extensive structural, PDS and control redesigns carried out and outlined in this report.

The secondary objectives have almost all been achieved, with the exception of pending testing for the PDS. The communications issues have been resolved. The landing mechanism now functions independently of the steel wires.

The Lunar Hopper Mk II has been fully documented, along with the media, test reports and so on. These documents have been combined onto a DVD which will be passed down to the Phase VI. The Lunar Hopper project has been successfully used for outreach and promotion by performing presentations at the open days and generating multimedia for the University to use.

The final secondary objective was to investigate and record the benefits and detriments of integrating an additional oxidiser tank onto the Lunar Hopper. An investigation into whether additional oxidiser tanks were needed was undertaken early in the project, and a feasibility study was generated which highlighted the lack in return for such an endeavour, as the engine run time was currently limited by nozzle melting temperature and current tanks contained sufficient oxidiser for the 15 s run.

### **9.1 Attitude Control System**

The ACS has a new structure, a refinished ACS manifold, and new 3D printed nozzles. The control theory for the ACS that is being used is inherited from the previous team. It will need to be reworked, tested, and verified before flight. The new 3D printed ACTs will also need to be tested to verify that they produce more thrust than the previous ones. If not, they may need to be remanufactured using a different manufacturing technique.

### **9.2 Propellant delivery System**

The propellant delivery system has been redesigned, procured and is awaiting assembly. The new system has been improved in terms of simplicity and weight. The design corrects the issue of the misaligned solenoid valves and has potentially eliminated the leaks. As the inherited system was a contributor to the failure of the Mk II to pass flight-readiness tests, it is important that the assembly and testing is undertaken carefully and thoroughly.

### **9.3 Chassis**

The chassis has been redesigned and built using triangulated struts and radial mounting for the Hopper’s tanks. All of the chassis components are attached to form a rigid structure that can support the Hopper throughout all phases of its flight. This year focused on four main goals for the Hopper chassis – make it rigid, make it light, make it simple, and make it affordable. The final result met all of these goals as well as all of its structural requirements.

### **9.4 Landing System**

The landing system from Phase IV has been upgraded to functional. Improvements have been made in simplicity, mass and number of components. The system is now functional and robust. The 3D printed pieces have been tested thoroughly to validate they can withstand static loading and gentle landing. Resin casting is currently underway for the 3D printed pieces to increase the strength and allow for full system testing of ‘hard landing’ conditions. A design and technical drawing for an aluminium version of the corner piece has been included.

## **9.5 Software & Electronics**

With changes done to the structure and communication, a new modular electronic component system was implemented to enable easy changes between components. The decision to abandon the old PCB for simpler components increased the reliability of the system. The code was improved in response time, by creating a better algorithm, software syntax, and removing the obsolete code. In addition, previously detected software problems, such as the faulty ACS response, were solved, and a new implementation to define the true input from sensors was added.

## **9.6 Communication**

An overall reliability improvement was achieved during the current Phase of the project. Using the knowledge in radio communication from previous projects, a new method of radio communication was created using two ZULU modules to transmit commands from a remote controller to the hopper. Breakout boards were manufactured for the modules to accommodate all necessary extra components required. Overall, the new system is able to achieve the communication range required without any sign of software crashing or hardware malfunction.

## **9.7 Pilot Training Simulator**

The simulator has been designed and assembled. The software provides accurate simulation of the hopper's main flight characteristics, ensuring that pilots will be able to practice flying the hopper before the actual test flight.

# 10 Future Work

## 10.1 Attitude Control System

If it is not completed before this year's flight test, the ACS control theory will need to be reworked, tested, and verified. The ACTs need to produce more thrust, so they need to be tested and potentially reprinted. If a 3D printed nozzle cannot generate an adequate amount of thrust, they may need to be remanufactured using a different manufacturing technique or redesigned entirely.

## 10.2 Propellant Delivery System

PDS system should be in a robust working order upon commencement of Phase VI. A serious upgrade to the system would be to remove the third ACS nitrogen tank in order to reduce unnecessary mass and complexity. This work would have to be done in tandem with the chassis to incorporate this. There's further potential to simplify the PDS by manufacturing it correctly, so that the ACT solenoid valves screw directly into the manifold, tightly, without being misaligned.

## 10.3 Chassis

The chassis is fully functional, but the main struts were originally manufactured at an incorrect angle and have been adjusted to better integrate with the landing mechanism. They could be rebuilt with the correct angle from the beginning. Additionally, more analysis should be done on how the heat transfers from the rocket motor to the chassis and how that affects its material properties. If the top oxidiser mount is weakened significantly, it would be good to look at making it out of a different material, most likely steel.

## 10.4 Landing Mechanism

The landing mechanism is now fully functional but upgrades can always be made. A change to a higher temperature casting resin should take place if the ABS corner pieces are to be reprinted. If preferable, and should the mass allow it, the aluminium corner pieces should be manufactured and added to replace the ABS pieces. Though these pieces are prohibitively heavy for the Phase V design, they are more robust. Removal or re-distribution of the foam pads should be seriously considered to allow the system to function independently or to more effectively dampen the spring oscillations.

## 10.5 Software

The current state of software is functional with the electronics setup available. Future work in developing the software is necessary if the electronics used shall be changed. Additionally, further code changes can be implemented to optimise the overall software operation, by reducing the computing time. Further development of the hopper and controller software can be made by introducing new pre-flight testing functions, such as testing the ACS. Lastly, the hopper could be programmed to have an autopilot function, given additional information regarding the engine.

## 10.6 Electronics

The electronics on the hopper can be upgraded by creating a more versatile power distribution board. Additionally, this board could be included in a more robust PCB design which should incorporate all the sensors attachment points. However, this PCB should be able to allow changes to be made to it, in order for future expansions of the project to be able to profit from this PCB. Lastly, swapping the Arduino for a Raspberry Pi would greatly improve the operating speed of the software, and vastly expand the capabilities of the hopper.

## **10.7 Communication**

While the range obtained by the current comms is suitable for current requirements, future project might require an expanded operating range. As such, improvement can be made by using a different antenna with a positive gain. This would enable the transmitter-receiver pair to communicate over a longer distance. Furthermore, it is in the opinion of the author that by including additional sensors and possible a camera to the hopper, the overall design of the comms must be upgraded.

## **10.8 Pilot Training Simulator**

The design of the simulator allows for the software to be upgraded easily and improvements in the accuracy and detail of the simulation could thus be made. The simulator could also be redesigned to incorporate a graphical display of the simulation, along with LEDs to indicate the simulator's state (e.g. ready, running and complete); this was considered but was not implemented due to time constraints. Finally, the simulator seeks to calculate the main behavioural characteristics of the hopper to allow a pilot to train to fly it, but in a real-moon scenario the hopper would have to take off, hover and land without external input. Thus the hopper behaviour identified could be used to guide the development of an autopilot that could enable the hopper to fly itself.

## 11 References

- [1] “Space Images,” Science Kids, [Online]. Available: <http://www.sciencekids.co.nz/pictures/space/lunarrover.html>. [Accessed 16 January 2016].
- [2] J. Garlick, V. Surendran, P. Symington and M. Vousden, “Lunar Hopper Main Report,” University of Southampton , 2012.
- [3] C. E. S. F. A. L. J. M. M. S. T. M. Z. Justin Diviney, “Lunar Hopper Main Report,” University of Southampton, 2013.
- [4] B. Abbott, Z. Ma, Z. Mirza, A. Mittal, T. Noga, S. Parkes and S. Rylance, “Lunar Hopper Mk.2,” University of Southampton, 2015.
- [5] R. Castelli, R. Moeys, D. Allan, P. Trouton, B. Heslop and B. Tanner Lees, “The Development of a flight Rated 400N Hybrid Rocket,” University of Southampton, Southampton, 2015.
- [6] P. A. Musker, *MA, BSc, MEng, CEng, FRINA, CertEd*, Southampton, 2016.
- [7] “Material Compatibility with Hyrdogen Peroxide,” OzoneLab, [Online]. Available: <http://www.ozoneservices.com/articles/004.htm>. [Accessed 1 Dec 2015].
- [8] K. Cho, “Cleaning Products You Should Never Mix,” goodhousekeeping, [Online]. Available: <http://www.goodhousekeeping.com/home/cleaning/tips/a32773/cleaning-products-never-mix/>. [Accessed 16 May 2016].
- [9] “Zircotec Heat Management,” Zircotec, 2016. [Online]. Available: <http://www.zircotec.com/page/technical-details-for-flexible-ceramic-heat-shields/98>. [Accessed March 2016].
- [10] “Zircotec Heat Management,” Zircotec, 2016. [Online]. Available: [http://www.zircotec.com/rte\\_img\\_large\\_385.jpg](http://www.zircotec.com/rte_img_large_385.jpg). [Accessed March 2016].
- [11] “Aluminum 5052-H32,” ASM Aerospace Specification Metals Inc., 2016. [Online]. Available: <http://asm.matweb.com/search/SpecificMaterial.asp?bassnum=MA5052H32>. [Accessed March 2016].
- [12] “Properties Of Aluminium,” Aluminium Design, [Online]. Available: <http://www.aluminiumdesign.net/why-aluminium/properties-of-aluminium/>. [Accessed March 2016].
- [13] Stock Springs, “Compression Springs (230),” [Online]. Available: [http://www.entexstocksprings.co.uk/index.php?route=product%2Fproduct&product\\_id=1783](http://www.entexstocksprings.co.uk/index.php?route=product%2Fproduct&product_id=1783). [Accessed 25 November 2015].
- [14] Environ Molds, “Polyurethane Resin vs Polyether Resin,” Art Molds, [Online]. Available: <https://www.artmolds.com/polyurethane>. [Accessed 20 March 2016].
- [15] Easycomposites, “Water Clear Polyester Casting Resin,” [Online]. Available: <http://www.easycomposites.co.uk/#!/resin-gel-silicone-adhesive/casting-resin/water-clear-polyester-casting-resin.html>. [Accessed 20 Feburary 2016].
- [16] R. F. S. L. Team, “Smart Radio Telemetry Module,” [Online]. Available:

- ] <http://www.rfsolutions.co.uk/acatalog/DS-ZULUT.pdf>. [Accessed 26 March 2016].
- [17] S. S. L. Team, “CRS05 Angular Rate Sensor,” Silicon Sensing Ltd., 2004. [Online].  
] Available: <http://www.willow.co.uk/CRS05.pdf>. [Accessed 26 March 2016].
- [18] Microbot, “MMA7361L 3-axis  $\pm 1.5g/\pm 6g$  Accelerometer with Voltage Regulator,”  
] Microbot, [Online]. Available: [http://www.microbot.it/documents/mr003-002\\_datasheet.pdf](http://www.microbot.it/documents/mr003-002_datasheet.pdf). [Accessed 26 March 2016].

## 12 Appendix A - Expenses

| Total purchases  |        |        | 1537    |             |   |
|--|--------|--------|---------|-------------|---|
| S  | Price  | No Off | Total   | Item No.    | Link  |
|  | 47.544 |        |         |             |   |
| <b>RS</b>  |        |        |         |             |   |
| C-Grid III Terminal 26-28 AWG,tin, bag   | 4.02   | 1      | 4.02    | 6704938     | <a href="http://uk.rs-online.com/web/p/pcb-connector-con">http://uk.rs-online.com/web/p/pcb-connector-con</a>             |
| 10 piece metric combination spanner set  | 17.196 | 1      | 17.196  | 613044      | <a href="http://uk.rs-online.com/web/p/cable-strippers/06">http://uk.rs-online.com/web/p/cable-strippers/06</a>           |
| Pistol automatic wirestripper,0.2-6sq.mm   | 19.176 | 1      | 19.176  | 1848033     | <a href="http://uk.rs-online.com/web/p/spanner-sets/1848">http://uk.rs-online.com/web/p/spanner-sets/1848</a>             |
| 15PC BRITEGUARD HEX KEY  | 29.076 | 1      | 29.076  | 5423509     | <a href="http://uk.rs-online.com/web/p/hex-keys-sets/5423">http://uk.rs-online.com/web/p/hex-keys-sets/5423</a>           |
| Coloured clip on cable marker,size 2 0-9   | 5.52   | 1      | 5.52    | 4084670     | <a href="http://uk.rs-online.com/web/p/cable-markers/408">http://uk.rs-online.com/web/p/cable-markers/408</a>             |
| RE437-LF, Double Sided DIN 41612 C SMT Eurocard FR4 with 108 x 69 0.35mm Holes, 1.27 x 1.27mm Pitch, 160 x 100 x 1.5mm | 20.64  | 1      | 20.64   | 528-5391    | <a href="http://uk.rs-online.com/web/p/products/5285391">http://uk.rs-online.com/web/p/products/5285391</a>               |
| HARWIN 1.27mm 25 Way 1 Row Straight PCB Socket Through Hole Socket Strip   | 2.184  | 5      | 10.92   | 681-3039    | <a href="http://uk.rs-online.com/web/p/pcb-sockets/68130">http://uk.rs-online.com/web/p/pcb-sockets/68130</a>             |
| HARWIN M52 Series, 1.27mm Pitch 20 Way 1 Row Straight PCB Header, Solder Termination, 1A                               | 1.596  | 5      | 7.98    | 745-7135    | <a href="http://uk.rs-online.com/web/p/pcb-headers/7457135">http://uk.rs-online.com/web/p/pcb-headers/7457135</a>         |
| HARWIN 2.54mm 20 Way 1 Row Straight PCB Socket Through Hole Socket Strip   | 1.6008 | 5      | 8.004   | 681-6820    | <a href="http://uk.rs-online.com/web/p/pcb-sockets/68168">http://uk.rs-online.com/web/p/pcb-sockets/68168</a>             |
| TE Connectivity CFR50 Series Axial Carbon Film Resistor 10kΩ   | 0.0336 | 20     | 0.672   | 132-731     | <a href="http://uk.rs-online.com/web/p/through-hole-fixed">http://uk.rs-online.com/web/p/through-hole-fixed</a>           |
| Alps Slide Potentiometer RSAON Series with a 18.5 x 1.5 mm   | 4.752  | 1      | 4.752   | 263-3270    | <a href="http://uk.rs-online.com/web/p/potentiometers/2633270">http://uk.rs-online.com/web/p/potentiometers/2633270</a>   |
| Murata Radial RCE 1μF Ceramic Multilayer Capacitor, 50 V d   | 0.2688 | 10     | 2.688   | 811-0364    | <a href="http://uk.rs-online.com/web/p/ceramic-multilayer">http://uk.rs-online.com/web/p/ceramic-multilayer</a>           |
| Vishay 1N4001-E3/54, Diode, 50V 1A, 2-Pin DO-204AL   | 0.1458 | 20     | 2.916   | 628-8931    | <a href="http://uk.rs-online.com/web/p/rectifier-schottky-d">http://uk.rs-online.com/web/p/rectifier-schottky-d</a>       |
| ASSMANN WSW AW Series, 2.54mm Pitch 20 Way 1 Row St  | 0.5472 | 5      | 2.736   | 674-2343    | <a href="http://uk.rs-online.com/web/p/pcb-headers/6742343">http://uk.rs-online.com/web/p/pcb-headers/6742343</a>         |
| Toggle switch,SP,on-none-on,solder lug   | 0.846  | 4      | 3.384   | 7347097     | <a href="http://uk.rs-online.com/web/p/toggle-switches/7347097">http://uk.rs-online.com/web/p/toggle-switches/7347097</a> |
| A4 stainless steel full nut,M3   | 4.26   | 1      | 4.26    | 1808563     | <a href="http://uk.rs-online.com/web/p/hex-nuts/01808563">http://uk.rs-online.com/web/p/hex-nuts/01808563</a>             |
| BRASS HEX.THREADSPACER IOM3/25   | 12.192 | 1      | 12.192  | 1058252     | <a href="http://uk.rs-online.com/web/p/threaded-standoff">http://uk.rs-online.com/web/p/threaded-standoff</a>             |
| A4 s/steel skt button head screw,M3x10mm   | 5.772  | 1      | 5.772   | 2320158     | <a href="http://uk.rs-online.com/web/p/socket-screws/2320158">http://uk.rs-online.com/web/p/socket-screws/2320158</a>     |
| Proto shield extension Arduino MEGA Rev3   | 3.468  | 1      | 3.468   | 7097399     | <a href="http://uk.rs-online.com/web/p/processor-microco">http://uk.rs-online.com/web/p/processor-microco</a>             |
| Male Tamiya <-> Female XT60 (3pcs/bag) (UK Warehouse)  | 1.42   | 1      | 1.42    | 602A-6018   | <a href="https://www.hobbyking.com/hobbyking/store/_3">https://www.hobbyking.com/hobbyking/store/_3</a>                   |
| Delivery Hobbyking   | 5.9    | 1      | 5.9     |             |   |
| Solenoid Valve Coil Core 13mm, H37.5mm, W36.5mm, L50mm WP-G1-D   | 36.612 | 1      | 36.612  |             | <a href="http://www.solenoid-valve.world/IP-atex-protectio">http://www.solenoid-valve.world/IP-atex-protectio</a>         |
| Pitch Changer 1.27 mm to 2.54 mm conversion - 20 pin   | 2.3712 | 5      | 11.856  | F127T254P20 | <a href="http://www.proto-advantage.com/store/product_1">http://www.proto-advantage.com/store/product_1</a>               |
| Delivery   | 4.1952 | 1      | 4.1952  |             |   |
| StarTech panel mount USB B to B cable  | 3.39   | 1      | 3.39    | USBPNLBFBM1 | <a href="http://www.amazon.co.uk/StarTech-com-Panel-M">http://www.amazon.co.uk/StarTech-com-Panel-M</a>                   |
| Momentary push switch, green   | 2.59   | 1      | 2.59    | N02AR       | <a href="http://www.maplin.co.uk/p/green-push-to-make-s">http://www.maplin.co.uk/p/green-push-to-make-s</a>               |
| Momentary push switch, red   | 2.69   | 1      | 2.69    | N01AR       | <a href="http://www.maplin.co.uk/p/red-push-to-make-swi">http://www.maplin.co.uk/p/red-push-to-make-swi</a>               |
| DC Plug 2.1mm  | 1.99   | 1      | 1.99    | HH60        | <a href="http://www.maplin.co.uk/p/maplin-21-x-55mm-dc">http://www.maplin.co.uk/p/maplin-21-x-55mm-dc</a>                 |
| 9V PP3 battery   | 2.99   | 1      | 2.99    | L46AL       | <a href="http://www.maplin.co.uk/p/maplin-extra-long-life">http://www.maplin.co.uk/p/maplin-extra-long-life</a>           |
| Structures budget  |        |        | 512.574 |             |   |
| <b>Structures</b>  |        |        |         |             |   |
| Compression Spring (230)   | 3.204  | 4      | 12.816  |             | <a href="http://www.entexstocksprings.co.uk/index.php?ro">http://www.entexstocksprings.co.uk/index.php?ro</a>             |
| Delivery   | 6.6    | 1      | 6.6     |             |   |
| M12 A2 STAINLESS PART THREADED BOLT SCREW + NYLOC  | 17.75  | 1      | 17.75   |             | <a href="http://www.ebay.co.uk/itm/M12-A2-STAINLESS-PA">http://www.ebay.co.uk/itm/M12-A2-STAINLESS-PA</a>                 |
| Screwfix Hose Clips Stainless Steel Size 70-120mm pack of 1  | 11.15  | 2      | 22.3    | 28347       | <a href="http://www.screwfix.com/p/stainless-steel-worm">http://www.screwfix.com/p/stainless-steel-worm</a>               |
| 3M 80610833867 88 Scotch Super Professional Grade Elect  | 9.33   | 1      | 9.33    | 80610833867 | <a href="http://www.amazon.co.uk/gp/product/B00004WG">http://www.amazon.co.uk/gp/product/B00004WG</a>                     |
| Metal blocks - hip section   | 12     | 3      | 36      |             | Metal Supermarket Southampton   |
| M12 A2 STAINLESS PART THREADED BOLT SCREW + NYLOC  | 8.75   | 1      | 8.75    |             | <a href="http://www.ebay.co.uk/itm/M12-A2-STAINLESS-PA">http://www.ebay.co.uk/itm/M12-A2-STAINLESS-PA</a>                 |
| ZircorFlex® III - Triple Layer Flexible Ceramic Heat Shield  | 37.5   | 1      | 37.5    |             | <a href="http://www.zircotecwebstore.com/ZircorFlex%20III">http://www.zircotecwebstore.com/ZircorFlex%20III</a>           |
| Aluminium tube stock, 1/2in OD 16swg   | 19.428 | 1      | 19.428  | 3047973     | <a href="http://uk.rs-online.com/web/p/aluminium-tubes/3">http://uk.rs-online.com/web/p/aluminium-tubes/3</a>             |
| Oxi tank holder  | 100    | 1      | 100     |             |   |
|  |        |        | 0       |             |   |
| <b>3D printing</b>   |        |        | 0       |             |   |
| Foot piece - 3D printed in uni   | 27     | 4      | 108     |             |   |
| Foot piece - 3D printed in 3D HUBS   | 20.7   | 1      | 20.7    |             | <a href="https://www.3dhubs.com/order/2351470">https://www.3dhubs.com/order/2351470</a>                                   |
| Foot piece, spring casing - 3D printed in uni  | 9      | 1      | 9       |             |   |
| Joint 3D   | 27     | 3      | 81      |             |   |
| Casting King SUPACLEAR water clear casting resin (polyester)   | 16.24  | 1      | 16.24   |             |   |

|  |        |    |        |              |   |
|--|--------|----|--------|--------------|---|
| DC power plug 2-way black  | 0.828  | 1  | 0.828  | 1389405      | <a href="http://uk.rs-online.com/web/p/dc-power-plugs/1389405">http://uk.rs-online.com/web/p/dc-power-plugs/1389405</a>   |
| Black drawer type holder for 1xPP3 cell                            | 4.164  | 1  | 4.164  | 593704       | <a href="http://uk.rs-online.com/web/p/battery-holders-mu">http://uk.rs-online.com/web/p/battery-holders-mu</a>   |
| 1P 0-1 high inrush current rocker switch                           | 1.008  | 1  | 1.008  | 2789733      | <a href="http://uk.rs-online.com/web/p/rockier-switches/2789733">http://uk.rs-online.com/web/p/rockier-switches/2789733</a>   |
| 2 way PCB terminal block 5.08mm                                    | 0.3504 | 15 | 5.256  | 7901064      | <a href="http://uk.rs-online.com/web/p/pcb-terminal-block">http://uk.rs-online.com/web/p/pcb-terminal-block</a>   |
| Header 2.54mm vertical 9+9way                                      | 1.776  | 1  | 1.776  | 7671022      | <a href="http://uk.rs-online.com/web/p/pcb-headers/7671022">http://uk.rs-online.com/web/p/pcb-headers/7671022</a>   |
| M20 HEADER, PIN, SIL, VERTICAL, 8 w                                | 0.096  | 10 | 0.96   | 6812997      | <a href="http://uk.rs-online.com/web/p/pcb-headers/6812997">http://uk.rs-online.com/web/p/pcb-headers/6812997</a>   |
| Molex C-GRID III Series, Series Number 90131, 2.54mm               | 0.8304 | 5  | 4.152  | 670-3487     | <a href="http://uk.rs-online.com/web/p/pcb-headers/670-3487">http://uk.rs-online.com/web/p/pcb-headers/670-3487</a>   |
| TE Connectivity AMPMODU Series 10 Way 1 Row Female                 | 0.0744 | 10 | 0.744  | 454-3309     | <a href="http://uk.rs-online.com/web/p/pcb-connector-hous">http://uk.rs-online.com/web/p/pcb-connector-hous</a>   |
| Aluminium tube stock, 1/2in OD 16swg                               | 19.428 | 1  | 19.428 | 3047973      | <a href="http://uk.rs-online.com/web/p/aluminium-tubes/3047973">http://uk.rs-online.com/web/p/aluminium-tubes/3047973</a>   |
| V-Ref Precision 2.5V 10mA 3-Pin TO-92                              | 0.912  | 5  | 4.56   | 6819802      | <a href="http://uk.rs-online.com/web/p/voltage-references">http://uk.rs-online.com/web/p/voltage-references</a>   |
| Toggle Switch SP on off insulated lever                            | 3.54   | 1  | 3.54   | 685057       | <a href="http://uk.rs-online.com/web/p/toggle-switches/685057">http://uk.rs-online.com/web/p/toggle-switches/685057</a>   |
| Crimp terminal,male,tin plated,22-28 awg                           | 2.16   | 1  | 2.16   | 7205763      | <a href="http://uk.rs-online.com/web/p/pcb-connector-con">http://uk.rs-online.com/web/p/pcb-connector-con</a>   |
| M20 CRIMP HOUSING, SIL, 4 w, 2.54mm pack of 10                     | 0.972  | 1  | 0.972  | 6812824      | <a href="http://uk.rs-online.com/web/p/pcb-connector-hou">http://uk.rs-online.com/web/p/pcb-connector-hou</a>   |
| M20 CRIMP HOUSING, SIL, 5 w, 2.54mm pack of 10                     | 1.044  | 1  | 1.044  | 6812828      | <a href="http://uk.rs-online.com/web/p/pcb-connector-hou">http://uk.rs-online.com/web/p/pcb-connector-hou</a>   |
| Low Power Radio Solutions Right Angle 50Ω PCB Mount                | 1.3272 | 5  | 6.636  | 783-0047     | <a href="http://uk.rs-online.com/web/p/sma-connectors/783-0047">http://uk.rs-online.com/web/p/sma-connectors/783-0047</a>   |
| RF Solutions Telemetry Antenna ANT-8WHIP3H-SMA, W                  | 4.02   | 2  | 9.648  | 793-4363     | <a href="http://uk.rs-online.com/web/p/telemetry-antenna">http://uk.rs-online.com/web/p/telemetry-antenna</a>   |
|  |        |    | 0      |              |   |
|  |        |    | 0      |              |   |
| Rf Solution  |        |    |        |              |   |
| Miniature Bead Antenna   | 4.164  | 3  | 12.492 | ANT-BEAD-868 | <a href="http://www.rfsolutions.co.uk/acatalog/info_ANT_BEAD_868.html">http://www.rfsolutions.co.uk/acatalog/info_ANT_BEAD_868.html</a>   |
| ZULU Smart Radio Telemetry Transceiver module                      | 23.1   | 2  | 46.2   | ZULU-T868    | <a href="http://www.rfsolutions.co.uk/acatalog/info_ZULU_T868.html">http://www.rfsolutions.co.uk/acatalog/info_ZULU_T868.html</a>   |
| Delivery   | 10.8   | 2  | 21.6   |              |   |
| Ebay   |        |    | 0      |              |   |
| 8-Channel 5V Relay Shield Opto-couple For Arduino UNO              |        |    |        |              |   |
| MEGA2560 R3 AVR Robot P  | 4.99   | 2  | 9.98   |              | <a href="http://www.ebay.co.uk/itm/8-Channel-5V-Relay-Shield-Opto-couple-for-Arduino-UNO-MEGA2560-R3-AVR-Robot-P-/221111111111111111">http://www.ebay.co.uk/itm/8-Channel-5V-Relay-Shield-Opto-couple-for-Arduino-UNO-MEGA2560-R3-AVR-Robot-P-/221111111111111111</a> |
| Fine Turning Dial Analog Panel Voltmeter Volt Meter DC 0 - 5V 44C2 | 5.82   | 2  | 11.64  |              | <a href="http://www.ebay.co.uk/itm/Fine-Turning-Dial-Analog-Panel-Voltmeter-Volt-Meter-DC-0--5V-44C2-/221111111111111111">http://www.ebay.co.uk/itm/Fine-Turning-Dial-Analog-Panel-Voltmeter-Volt-Meter-DC-0--5V-44C2-/221111111111111111</a>                         |
|  |        |    | 0      |              |   |
| Others   |        |    |        |              |   |
| 9B-30.000MEEJ-B XTAL, 30.000MHZ, 18PF, HC-49S                      | 0.35   | 5  | 1.75   | 1842247      | <a href="http://uk.farnell.com/txr/9b-30-000meej-b/xtal-30-000mhz-18pf-hc-49s-1842247">http://uk.farnell.com/txr/9b-30-000meej-b/xtal-30-000mhz-18pf-hc-49s-1842247</a>   |
| Veroboard Copper Tripad Board 100 x 160mm                          | 6.99   | 1  | 6.99   | A66RL        | <a href="http://www.maplin.co.uk/p/veroboard-copper-tripad-board-100-x-160mm-a66rl">http://www.maplin.co.uk/p/veroboard-copper-tripad-board-100-x-160mm-a66rl</a>   |
| Turnigy 2200mAh 3S 25C Lipo Pack (UK Warehouse)                    | 9.02   | 2  | 18.04  | T2200.35.25  | <a href="https://www.hobbyking.com/hobbyking/store/_/productinfo/turnigy-2200mah-3s-25c-lipo-pack-uk-warehouse">https://www.hobbyking.com/hobbyking/store/_/productinfo/turnigy-2200mah-3s-25c-lipo-pack-uk-warehouse</a>   |

|                                |       |    |        |       |   |
|--------------------------------|-------|----|--------|-------|---|
| Nozzle reprint (pending)       | 7.16  | 1  | 7.16   |       |   |
| Travel                         | 250   | 1  | 250    |       |   |
| Plumbing                       |       |    | 340.32 |       |   |
| Delivery                       | 5     | 1  | 6      |       |   |
| 316S/STEELTUBE 6MMX3M(1.0WALL) | 15.73 | 3  | 56.628 | 13086 | <a href="https://www.bes.co.uk/products/163.asp?#13086">https://www.bes.co.uk/products/163.asp?#13086</a>           |
| 6MM EQUAL TEE STAINLESS STEEL  | 14.54 | 3  | 52.344 | 14020 | <a href="https://www.bes.co.uk/products/163.asp?EPBOE=14020">https://www.bes.co.uk/products/163.asp?EPBOE=14020</a> |
| 6MMX1/8M-STUD COUP.BSPP ST     | 3.84  | 20 | 92.16  | 14043 | <a href="https://www.bes.co.uk/products/163.asp?EPBOE=14043">https://www.bes.co.uk/products/163.asp?EPBOE=14043</a> |
| 6MMX1/4M-STUD COUP.BSPP ST     | 6.5   | 10 | 78     | 14044 | <a href="https://www.bes.co.uk/products/163.asp?EPBOE=14044">https://www.bes.co.uk/products/163.asp?EPBOE=14044</a> |
| 8MMX1/4F-STUD COUP.BSPP ST     | 9.37  | 2  | 22.488 | 14110 | <a href="https://www.bes.co.uk/products/163.asp?EPBOE=14110">https://www.bes.co.uk/products/163.asp?EPBOE=14110</a> |
| 6MM SNG FERRULE 316 S/STEEL    | 1.09  | 25 | 32.7   | 16034 | <a href="https://www.bes.co.uk/products/163.asp?EPBOE=16034">https://www.bes.co.uk/products/163.asp?EPBOE=16034</a> |

## 13 Appendix B – Computational Modelling

Note that this section was written by Tomasz Noga, a member of the previous Lunar Hopper Team.

### Computational Modelling

Author: Tomasz Noga

A MatLab simulator of the Lunar Hopper was prepared to validate the design of the ACS numerically, and to facilitate the optimization process. It also creates an animation to visualise the vehicle's motion. A more elaborate version of the simulator optimizes the control law simultaneously with the boom length. This subsection describes the code in detail and explains how it was used to validate and optimize the design.

#### 8.1 Dynamics modelling

The first few lines of the code define the variables for simulation time, time step, the vehicle's moment of inertia, the thrust from attitude control thrusters, and their moment arm length. Wind strength, air density and the aerodynamic properties of the vehicle (drag coefficient, distance from centre of mass to centre of pressure, and reference surface area) are also defined. Subsequent lines of the code define vectors that describe the vehicles motion rotational and translational positions, velocities, and accelerations over time. Initial values of these quantities are defined along with the values of the control law function. Wind strength is described in a vector form as well. This allows the user to vary the wind strength, and the direction as a function of time, and is hence, useful to simulate a gust. After all the variables and vectors are defined, the main simulation code starts to run. It uses a for-loop with 600 time-steps (a 15 second flight with 25 ms time steps). For every loop iteration, translational and rotational accelerations for a given time step are calculated first. The governing equations for these calculations are based on Newton's laws of motion for 6 degrees of freedom, and the code can be found in Appendix J. The '*trapz*' function is then used to calculate the translational and rotational velocities, and the positions at a given time step. The control law function is used to determine whether an ACS control thruster was open for a given time step. The accelerations resulting from that are used to calculate accelerations in subsequent time steps. The process is repeated for all time steps.

#### 8.2 Animation generation

To visualise the hopper's flight, the simulator creates an animation that shows the vehicle's motion. To achieve that, the geometry had to be defined in a separate script. While the for-loop is executed in the simulation, the geometry is translated and rotated with accordance to motion parameters at given time steps. For each time step, one animation frame is generated. Function *HopperGeometry.m* is a simple script that defines the geometry of the Lunae Hopper. 28 edge points were defined to represent the simplified geometry of the vehicle. This script is opened in *HopperAnimation.m*. During the simulation, for every loop iteration, edge points are translated and rotated using angles and positions calculated by the dynamic model. Hence functions of *getx.m* type are used to feed the edge coordinates to the *fill3* function which plots the geometry at given time steps. Functions of *getx.m* type were written to support in-built *fill3* MatLab functions which were used to prepare the animation. Function *fill3* plots a surface and uses at least three vectors to achieve that; a vector with x-coordinates of edges, a vector with y-coordinates of edges and a vector with z-coordinates of edges. Function *getx.m* takes edge coordinates and returns the corresponding x-coordinates which are fed to *fill3* function. Functions *gety.m* and *getz.m* do the same, as their name suggests, for y and z coordinates. Functions *getx.m*, *gety.m* and *getz.m* work for triangular surfaces, whereas *getx4.m*, *gety4.m* and *getz4.m* work for rectangular geometries (they require coordinates of 4 edges). At every loop iteration, a geometry is plotted using *fill3* function, and hence an animation is created. An example animation can be found on the data disk.

### 8.3 Optimisation

#### 8.3.1 Gains Tuning

This section presents the history of development of a control law to be used for the ACS. The MatLab script used is available on the data disk. Control law was based on the one suggested by supervisor, with following logic

```
Function = Sign( $\theta + a \cdot d(\theta)/dt$ ) ;
If Function = 1, thruster 1 is open
If Function = -1, thruster opposite to thruster 1 is open
```

To finalise the control law, the optimal value of the gain "a" was investigated. This used a MatLab script available on the data disk. The code was tested for a required pitch of 6 degrees in perfect conditions (no wind) and a constant inertia of  $5kgm^2$ . It was found that the minimum gain (to prevent overshooting) is equal to 0.19 and the maximum gain (to ensure a robust response) is 0.7. The resulting figure is shown in 8.1.

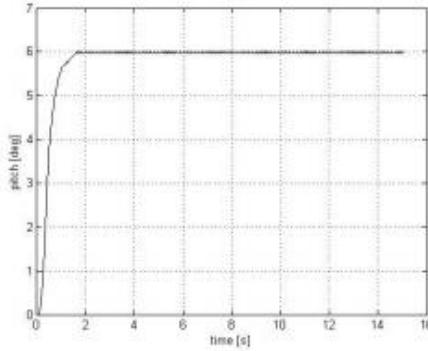


Figure 8.1: Response for gain = 0.35 .

When the pitch required was decreased to 2 degrees, resulting response is shown in figure 8.2

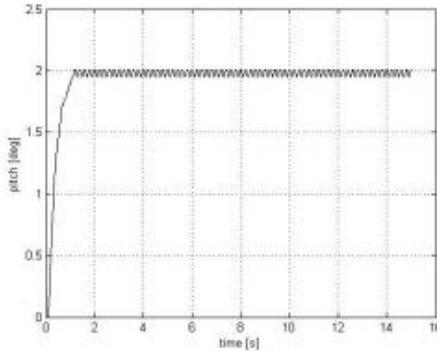


Figure 8.2: Response for gain =0.35 and  $\theta$  required = 2 deg.

When a very high pitch angle (30 degrees) was tried, figure (8.3) was obtained:

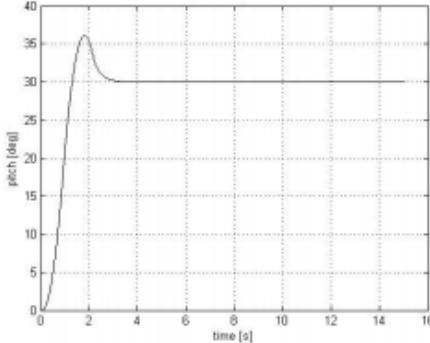


Figure 8.3: Response for gain = 0.35 and  $\theta$  required = 30 deg.

A large overshoot was observed. Increasing gain to 0.3 solved the problem of overshooting.

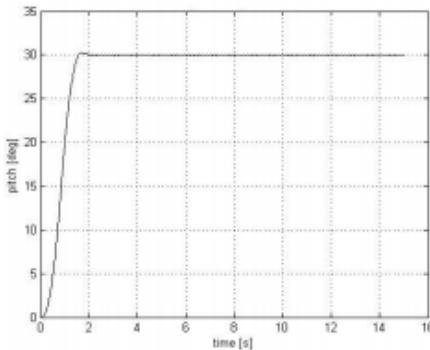


Figure 8.4: Response for gain = 0.3 with  $\theta$  required = 30 deg.

The analysis showed that the bigger the pitch required, the bigger the gain should be to reduce overshooting. However, larger gains result in a longer response time, and hence a trade-off between damping and response time was required. The project was still in its early stages to determine which of the two gets a higher priority over the other, and therefore, a separate script was written that would determine the optimal value of gain.

The effect of the inertia on the response was determined for a gain of 0.35 and angle required of 6 degrees. The response is shown in figure 8.5. As evident, shorter response time and slight increase in oscillations is observed when inertia is decreased.

$$\text{Inertia} = 2 \text{kgm}^2;$$

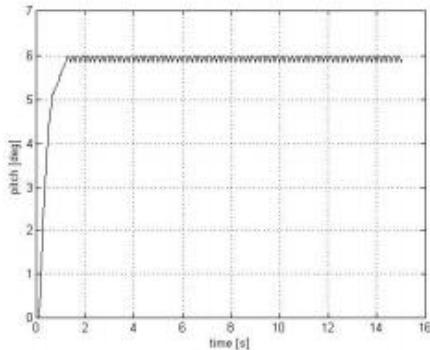


Figure 8.5: Response for an inertia of  $2kgm^2$  and a gain of 0.35 .

Decreasing the gain (figure 8.6) does not solve the problem of oscillations. The oscillations will make rocket thrust vector non-vertical which will cause rocket to lose its altitude. To overcome this problem, filtering was introduced.

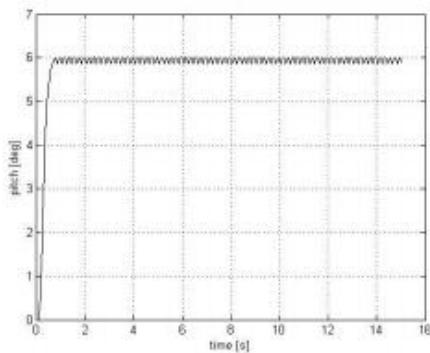


Figure 8.6: Response for an inertia of  $2kgm^2$  and a gain of 0.2 .

### 8.3.1.1 Introducing Filtering

The disadvantage of the proposed control law is that a solenoid valve is open at all times. It was proposed that a filter be added to the control law such that no thruster would fire if the value of the function was smaller than some value, say,  $k$ . Few values of  $k$  were tested, and the findings are presented in figures 8.7 and 8.8.

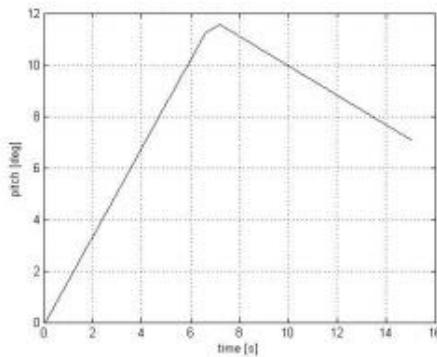


Figure 8.7: Response with  $k = 0.1$ .

With  $k = 0.1$ , the performance of the control law was observed to be below par.  
For  $k = 0.001$ , the performance is relatively good. Oscillations are much smaller.

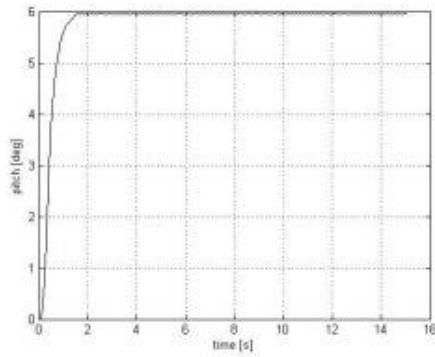


Figure 8.8: Response with  $k = 0.001$ .

Different values of  $k$  were tried and for each simulation, the time duration of an open solenoid valve was measured. Following numbers were obtained:

With no filtering, one thruster is open for all 15 second of simulation

With  $k = 0.001$ , one thruster is open for a little less than 8 seconds.

For  $k = 0.007$ , one thruster is open for less than 4 seconds

As  $k$  increases oscillations grow and as  $k$  decreases system may become unstable. A value of  $k = 0.007$  was tried for a big theta required (30 degrees). The resulting response is shown in figure 8.9.

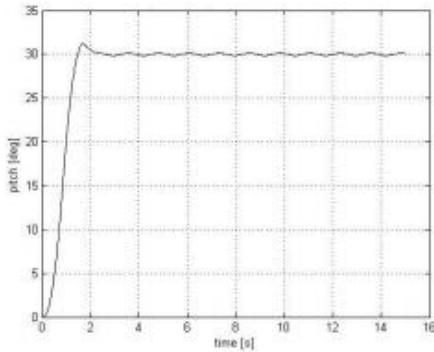


Figure 8.9: Response for  $k = 0.007$  and  $\theta$  required = 30 deg.

The graph above shows that  $k$  can introduce a small overshooting.

Decreasing  $k$  to 0.001 solved the problem of overshooting, and it was observed that the thrusters were open for about 8 seconds. Increasing  $k$  to 0.007 does not decrease firing time significantly (for a theta required of 30 degrees), and it introduces overshooting. Therefore, the proposed values of 'a' and 'k' were 0.3 and 0.001, respectively.

#### 8.3.1.2 Wind Effects

The Matlab model was then upgraded to a more sophisticated script which runs a simulation taking into account a steady wind, blowing at 4m/s. Using a simple aerodynamics model:

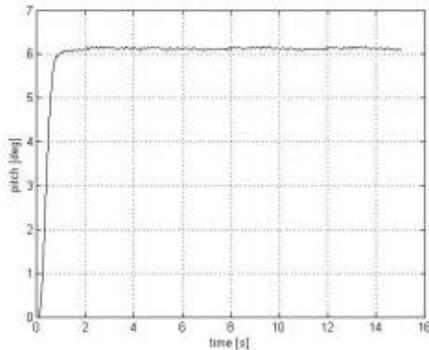


Figure 8.10: Response for a steady wind blowing at 4 m/s with  $k = 0.001$  and gain = 0.3 .

When a wind is introduced, oscillations are bigger and there is a 0.2 degree drift from the required theta. If wind velocity is increased to 5 m/s (figure 8.11)

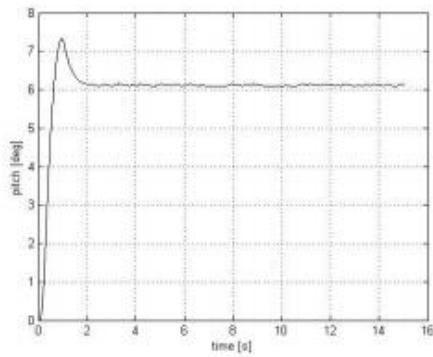


Figure 8.11: Response for wind velocity of 5 m/s with  $k = 0.001$  and gain =0.3 .

At 6 m/s (figure 8.12)

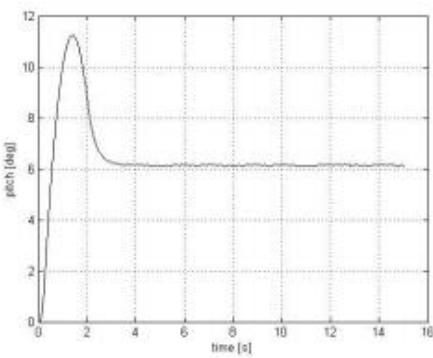


Figure 8.12: Response for wind velocity of 6 m/s with  $k = 0.001$  and gain =0.3 .

At 7 m/s (figure 8.13)

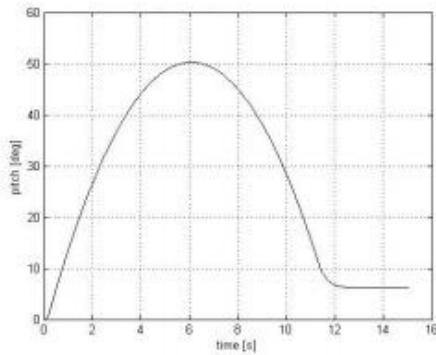


Figure 8.13: Response for wind velocity of 7 m/s with  $k = 0.001$  and gain = 0.3 .

Changing the value of the gain did not help solve the problem in an efficient way because, though a large gain worked for a high velocity gust, it would not work for lower velocity winds/no wind condition after the gust has passed.

Increasing the thrust of ACS (in this example, to 8 N) showed potential to solve the problem.

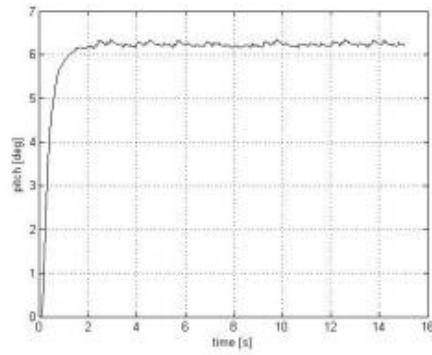


Figure 8.14: Response for  $k = 0.001$  and gain = 0.3 when thrust has been increased to 8 N.

### 8.3.1.3 Gust Response

An additional test was performed for no wind, a  $\theta$  required equal to zero, and a sudden gust with a speed of 7m/s lasting for 2.5 seconds acting on the hopper while mid-flight. The resulting figure was:

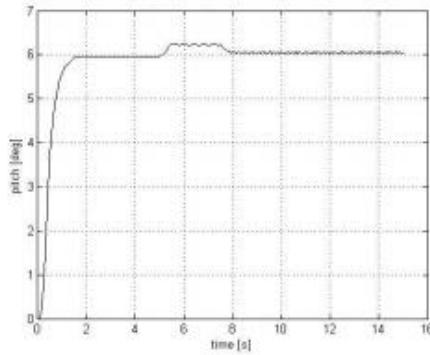


Figure 8.15: Response for a sudden gust of 7 m/s, lasting 2.5 sec. gain =0.3 and k =0.001 .

This shows that although the pitch can change up to 3 degrees, the control law could cope with a sudden gust.

### 8.3.2 Optimiser

To search for optimal "a" and "k" values, a more systematic MatLab code called "Optimizer" was prepared. The optimizer takes values of inertia, thrust, drag coefficient etc. and runs a 1-D flight simulation with different gain and filter ("a" and "k") values. The simulation assumes initial angle of -10 degrees and a required angle of 0 degrees. It records data for the rise time, the nitrogen usage, and the drop in height (resulting from high oscillations) for every "a"- "k" pair. This results in a motion meeting the following requirements:

- maximum oscillation during steady-state is less than 0.5 degrees
- maximum overshooting is less than 1 degree

The script records configurations for best rise time, smallest drop and smallest Nitrogen usage. For example:

```
k_values = 0.0:0.025:0.5;
k_values = k_values*pi/180;
a_values = 0.00:0.05:2.00;
```

"k" varies from 0 to 0.5 with 0.025 steps, and "a" varies from 0 to 2 with 0.05 steps. The program was run using these values and a wind strength of 5 m/s. The best rise time (time it takes to travel from -10 degree to -0.05 degree) was 0.975 seconds with "k" = 0.0044 and "a" = 0.45. Figure 8.16 shows this. The drop (the integral of absolute values of pitch over the time flight) is 6.033.

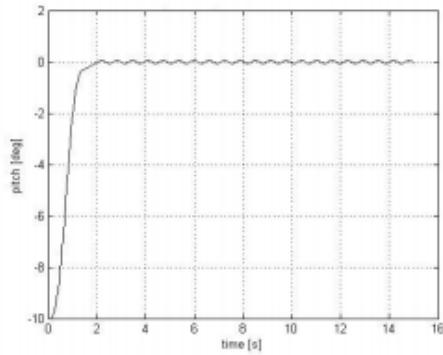


Figure 8.16: Response time for  $k = 0.0044$  and gain  $= 0.45$ .

The best drop scenario sees "k" = 0.0026 and "a" = 0.5. Although the rise time is greater (1.32 seconds), the equivalent drop is smaller (4.755 meters)

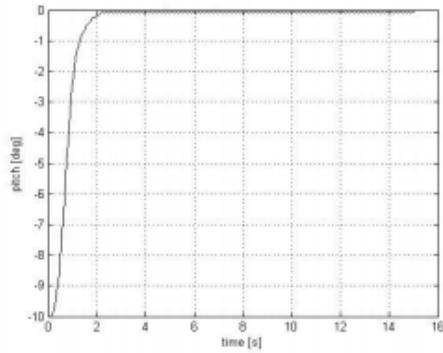


Figure 8.17: Response for  $k = 0.0026$  and gain  $= 0.5$ .

An interesting phenomenon discovered regarding the tuning, was that the assumption of constant firing of two thrusters throughout the entire flight was erroneous. The assumption was used for propulsion systems design which has resulted in three Nitrogen tanks feeding the ACT system. The tuning has shown that theoretically, only one tank is enough to supply enough Nitrogen. However, more testing of ACT hardware is required to confirm that such efficiency can be achieved in real world.

The problem with this optimisation is that many variables assumed at the time of defining it have changed. Also, it does not include the boom length optimisation. This is why it was decided to prepare a more complex script that would model 3-D motion of the hopper and optimising control law for two degrees of freedom and boom length.

### 8.3.3 Boom Length Optimization

One of the important design choices that needed to be made was the length of the boom. A numerical optimisation, based on several design variables and factors, was carried out to arrive at the optimum

length. The longer the boom, the bigger the momentum produced by the ACS. On the other hand, because of the significant mass of the solenoid valves, increasing the boom length means the overall inertia increases as well. Also, the centre of pressure is highly affected by the boom length because of its bluff shape. The structural considerations also play a role, which made this optimization a complex task. Control law optimization also needed to be conducted in parallel to boom-length optimization, simply because boom length will affect the control law. The optimal values of gain and k for pitch and yaw were looked for. Because the mass and inertia of the vehicle change over time, optimal gain and k values also change with time. However, in order to keep the code simple, it was decided to pursue a control law that would be feasible throughout the whole flight. A MatLab script was prepared, which runs a flight simulation for various values of gain, k and a boom length to find the optimal solution. As the mass of the solid fuel being burnt is relatively small, its impact on inertias and mass was neglected in the analysis. As the shift of the centre of mass due to nitrogen and oxidiser decreasing throughout the flight is no more than 1cm, the position of the centre of mass depends only on the boom length. Although, the code runs the same way as the simulator, described at the beginning of the section, it takes changes in inertia and mass into account as well. Centre of mass and inertias were estimated using SolidWorks model. Inertias for full and empty tanks were compared to estimate the effect of Oxygen and Nitrogen usage on centre of mass and inertias.

Effect of rocket work on inertias:

- Pitch inertia (Py) for full oxidiser and nitrogen tanks used in propellant delivery system is  $5.226\text{kgm}^2$
- Yaw inertia (Px) for full oxidiser and nitrogen tanks used in propellant delivery system is  $5.342\text{kgm}^2$

With no oxidiser and nitrogen used to push oxidiser, Py and Px are  $5.16\text{kgm}^2$  and  $5.24\text{kgm}^2$ , respectively.

Because inertia is a linear function of body mass (assuming the distance from centre of gravity has not changed, which is a valid simplification), a linear function was used to model changes of inertia due to the varying mass of the on-board fluids, when in use for the main propulsion system. The pitch and yaw inertias are simply decreased with every time step by the same value,  $0.00011\text{kgm}^2$  for Py, and  $0.00017\text{kgm}^2$  for Px.

Similar analysis was done for the nitrogen tanks. However, the pitch and yaw inertias were decreased only when the ACS's solenoid valve is open. When that occurs, inertias are decreased by  $0.0001\text{kgm}^2$ .

The impact of the boom length on inertia and centre of mass was calculated using data from SolidWorks and is used in the optimisation in form of linear functions. It was found that centre of mass is equal to

$$CoM = 0.125x + 1.2275 \quad (8.1)$$

where x is the boom length. The reference system has the vertical axis aligned with the thrust axis, and the x-y plane is at the hopper's feet.

CP was modelled as suggested in the CFD section (6.2).

$$CoP = 0.2344x + 0.7519 \quad (8.2)$$

Pitch and Yaw inertias are dependent on the boom length in the following way:

$$Py = 8.3333x^2 + 3.1667; \quad (8.3)$$

$$Px = 8.3333x^2 + 3.3067; \quad (8.4)$$

To optimise the control law and the boom length, the script runs the simulation for different values of variables in the respective functions. For each simulation, the script checks whether:

- the mean of pitch and yaw positions is smaller than 0.25 degree
- the maximum of pitch and yaw oscillations is smaller than 2 degrees

These constraints were applied as it was assumed that the horizontal component of the rocket thrust should result in a translation smaller than 1 m, and the altitude decrease due to the smaller vertical component is smaller than 10 cm. Taking average values of the thrust and the rocket mass as 300N and

30kg respectively, it can be calculated that for over 15 seconds of flight, the maximum drift angle is 0.05 degree and the maximum oscillation no more than 2 degrees.

$$s = at^2/2; \quad (8.5)$$

$$a = F/m; \quad (8.6)$$

$$F = Thrust * \sin(a) \quad (8.7)$$

for drift or

$$T = Thrust * (1 - \cos(a)) \quad (8.8)$$

Once the simulation is run, it records the data (control law, boom length and number of time steps when the nitrogen was used) and checks if the requirements have been met for a specific combination. The requirement for the script is to find a high performance control law for different boom lengths (varying from 0.5 m to 1 m). Compared to longer boom lengths, it was found that 0.5 m boom, there are a lot of possible control laws that meet the requirements. As discussed in the ACS testing section (10), the testing revealed that the results from the test rig are different to Matlab simulations due to a number of reasons, the main one being hardware integration in the real world. Therefore, it is ideal to use 0.5 m as the boom length as it can accommodate different control laws and hence, make the ACS more robust and less dependent on a perfect Matlab model.

#### 8.4 Translational Movement

Wind, attitude control thrusters and horizontal components of rocket all contribute to translational motion. The simulator shows that for a steady wind of 5m/s, expected translation is 0.76 metres. If this adds up with expected translation due to horizontal component of rocket (of 1 m, as mentioned in 8.3.3) a large translation occurs. It seems that using a netting for the test is infeasible. The translation for different wind speeds (excluding translation due to rocket horizontal component) is shown in the table 1.

| Wind speed [m/s] | 1    | 2    | 3    | 4    | 5    | 6   | 7    |
|------------------|------|------|------|------|------|-----|------|
| Translation[m]   | 0.05 | 0.18 | 0.43 | 0.76 | 1.18 | 1.7 | 2.31 |

Table 8.1: Translation motion for different wind speeds.

It is recommended that the flight test is carried out in an open environment, and netting should be considered only for no-wind conditions.

Appendix – Mass Budget

| COMPONENT                      | WEIGHT (KG) | QUANTITY | TOTAL (KG) |
|--------------------------------|-------------|----------|------------|
| ELECTRONICS MINUS BATTERIES    | 0.7         | 1        |            |
| HORIZONTAL BAR                 | 0.07        | 3        |            |
| NEW CORNER PIECES              | 0.15        | 3        |            |
| CHASSIS STRUT ATTACHMENT PIECE | 0.11        | 3        |            |
| OXIDISER TANK HOLDERS          | 0.11        | 4        |            |
| ACS -TO-CHASSIS STRUT ADAPTER  | 0.03        | 3        |            |
| LEG BAR                        | 0.26        | 6        |            |
| BOLTS                          | 0.009       | 60       |            |
| PRESSURE TRANSDUCER            | 0.17        | 1        |            |
| NITRO TANK EMPTY               | 1.13        | 5        |            |
| ESTIMATE PDS MASS              | 1.5         | 1        |            |
| ELECTRICAL CABLING             | 0.05        | 1        |            |
| LANDING PADS WITH FOAM         | 0.7         | 3        |            |
| ACT GAS MANIFOLD               | 0.24        | 1        |            |
| MAIN ENGINE MANIFOLD           | 0.23        | 1        |            |
| STAINLESS STEEL PIPE           | 0.07        | 5        |            |
| MECHANICAL RELIEF VALVE        | 0.09        | 1        |            |
| LIPO BATTERY                   | 0.19        | 2        |            |
| ROCKET ENGINE                  | 8.8         | 1        |            |
| OXIDISER                       | 4           | 1        |            |
| OX TANK                        | 1.66        | 2        |            |
| NITROGEN                       | 0.19        | 5        |            |
| SOLENOID VALVES                | 0.593       | 7        |            |
|                                | Total       |          | 40.43      |

## 14 Appendix C– Pressure Drop in Pipes Derivation Phase IV Report [4]

### 5.1.1 Deriving the Pressure Drop in a Pipe

In classic fluid mechanics pressure drops in pipes are caused by:

- Frictional effects
- Vertical pipe change or elevation
- Directional changes in the pipe
- Change in kinetic energy of fluid

To govern the fluid (liquid or gas) pressure drop along a pipe, the Reynolds number must first be deduced. The Reynolds Number is a dimensionless quantity that helps describe and predict a flow pattern in a variety of fluid mechanisms. It is defined by the ratio of inertial forces to viscous forces and therefore quantifies the importance of how much pressure can be lost due to these forces for a given flow condition. It is calculated by

$$Re = \frac{U_e D}{\nu} \quad (5.1)$$

Where  $U_e$  is the flow velocity,  $D$  is the internal diameter of the pipe, and  $\nu$  is the kinematic viscosity which is governed by

$$\nu = \frac{\mu}{\rho} \quad (5.2)$$

$\mu$  is the dynamic viscosity, and is also defined by [31]

$$\mu = \mu_0 \frac{T_0 + C}{T + C} \left( \frac{T}{T_0} \right)^{3/2} \quad (5.3)$$

Where  $\mu$  is the viscosity in  $Pa \cdot s$  at an input temperature  $T$  (in Kelvin) and  $\mu_0$  is the reference viscosity at the reference temperature  $T_0$ .

- $T_0 = 300.55K$ .
- $\mu_0 = 1.781 \times 10^{-5} Pa \cdot s$ .
- $T = 293.15K$ .
- $C = 111$ , Sutherland's constant.

Therefore, inputting these values into equation 5.3

$$\mu = 1.7469 \times 10^{-5} Pa \cdot s.$$

To find the velocity of the flow in the pipe for the ACS, the mass flow rate must be assumed to be equal to the flow rate at the throat of the nozzle, up stream of the paintball tanks. In this case, this is equal to

$$\dot{m}_{throat} = 0.0118 \text{ kg/s}$$

Using the equation for mass flow rate  $\dot{m} = \rho A V$ , and taking the inside area of the pipe for a 4.8mm diameter, the velocity can be acquired

$$V = \frac{\dot{m}_{throat}}{\rho A_{pipe}} = 9.785 \text{ m/s.}$$

Then, using equation 5.1 and substituting  $\nu$  with equation 5.2 the nature of the flow can be described as laminar or turbulent

$$Re = \frac{\rho U_e D}{\mu} = 179107.$$

As the Reynolds number is greater than 2320, we can assume the flow is turbulent. This indicates that the flow has irregular motion and the fluid particles may move in directions transverse to the direction of the main flow. Unlike laminar flow, the velocity distribution of turbulent flow is more uniform across the diameter of the pipe and hence the pressure drop is more dependent on the roughness of the pipe. To determine the pipe friction coefficient in the case of turbulent flow, the Colebrook-White equation must be iteratively solved for the Darcy-Weisbach friction factor [16]  $\lambda$  (pipe friction coefficient)

$$\frac{1}{\sqrt{\lambda}} = -2\log \left[ \frac{2.51}{Re\sqrt{\lambda}} + \frac{k}{D} \times 0.269 \right] \quad (5.4)$$

where  $Re$  is the Reynolds number for the flow in the pipe,  $k$  is the absolute roughness of the pipe (which in this case for a stainless steel pipe is 0.015mm), and  $D$  is the inner diameter of the pipe. Iteratively solving this equation with the above parameters yields

$$\lambda = 0.0272$$

Finally to determine the pressure drop due to friction the following equation is used, where  $L$  is the pipe length, in this case 1 metre.

$$\delta P_f = \lambda \frac{L}{D} \frac{\rho}{2} U_e^2 \quad (5.5)$$

Using previously attained values this can be calculated to give

$$\delta P_f = 0.18 \text{ bar.}$$

So for a 1m length of pipe the loss of pressure due to friction is 0.18 bar. While this is less than 1% of the total pressure, the effects of elevation and directional changes have yet to be taken into consideration and as such will be evaluated in the following calculations to assess the total loss of pressure due to friction, elevation and direction. As the tanks were situated below the ACTs, pressure losses due to elevation can be calculated from the following formula [18]

$$\delta P_e = \rho g \delta H \quad (5.6)$$

where  $\delta H$  is the vertical elevation of the fluid, which is assumed to be equal to the length of the pipe (1m) giving

$$\delta P_e = 0.006535 \text{ bar.}$$

This concludes that the pressure drop due to elevation is negligible, for the case of a 1m increase. Finally the pressure drop due to a change in direction is calculated by [18]

$$\delta P_d = \frac{\rho}{2} U_e^2 \xi \quad (5.7)$$

All values are the same other than  $\xi$ , which is the resistance coefficient of the part, and is determined empirically or by vendor specification. The most dramatic directional change the network will experience will be in a 90° turn, which for a long radius ( $R/D = 1.5$ ) connection has a resistance coefficient  $\xi = 0.45$ . Substituting all respective values into equation 5.7 produces

$$\delta P_d = 0.014 \text{ bar.}$$

Similar to the pressure drop due to elevation, directional changes have relatively little effect on the pressure drop in the network.

As discussed above, the major design trade-off was that of pressure drop versus mass. The table below demonstrates, for a number of different configurations, which solution is best to suit the Mk.2's propellant delivery systems requirements. As the pressure drop for change in direction is independent of the pipes diameter, it is assumed that each configuration has 1 of these components for simplicity.

## 15 Appendix D – Propellant Requirement Calculation from Phase IV [4]

$$m_T = \frac{P_p V_p}{RT_T} \left( \frac{1}{1 - \frac{P_p}{P_T}} \right) \quad (4.18)$$

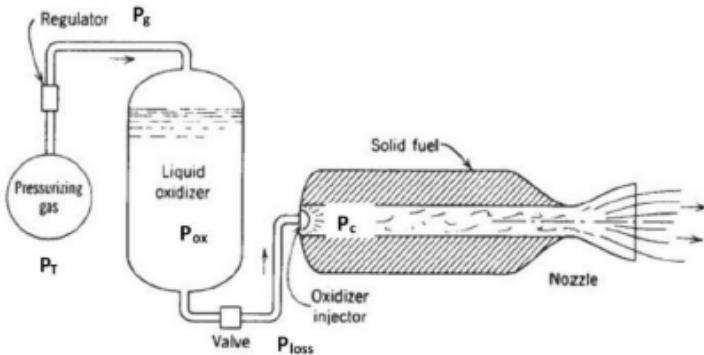


Figure 4.12: Hybrid rocket engine.

Based on the propulsion delivery system of the Mk.1 hopper and making some assumptions based on research, the following constraints are deduced

- $P_p = P_{ox} = 40\text{bar}$ . This parameter is driven from the Hybrid team's calculations, where the pressure inside the oxidiser tank is equal to the combustion chamber pressure, with some additional pressure to make up for any losses in downstream parts. This was to ensure the design chamber

44

pressure is met. The combustion chamber pressure desired was 20 bar, however empirical evidence from the 40N hybrid rocket suggested there was a drop of 15 bar across the catalyst bed, injector plates, and pre-combustion chamber elements. Further drops of 3 and 2 bar were assumed respectively across the injector, and in the valves and pipes. Therefore a total 40 bar was required in the oxidiser tank.

- $V_p = V_{ox}$  By a simple definition of displacement it is assumed that the volume displaced in the tank is the volume of oxidiser required to fuel the full 21 seconds of burn time at the maximum flow rate defined earlier, which was calculated to be approximately 2.76 litres.

Finally the following numerical assumptions are made:

- $\gamma = 1.4$  for diatomic gases, nitrogen in this case.
- $R=297 \text{ J/KgK}$  for nitrogen, derivation earlier
- $T_T=298 \text{ K}$  ambient temperature
- $P_g = P_p = 40\text{bar}$
- $P_T=150 \text{ bar}$ , this parameter has been determined based on previous iterations of the project although may vary subject to the demand of the attitude control thrusters

Introducing the values into equation 4.18 gives

$$m_T = \frac{40 \times 10^5 \times 2.76 \times 10^{-3} \times 1.4}{297 \times 298} \times \left( \frac{1}{1 - \frac{40}{150}} \right) = 0.23813\text{kg.}$$

Therefore the volume of gas this mass will occupy at 150 bar is

$$V = \frac{mRT}{P} = 1.405l$$

To accommodate for the assumptions made at the beginning a safety factor of  $SF=1.25$  is applied to allow for any losses incurred through changes in ambient pressure or as a result of the assumptions previously made. This gives a required volume of 1.75 litres of nitrogen, or 0.298 kg.

#### 4.2.0.5 Oxidiser Budget

To achieve a 15 second flight time, the primary consideration was the amount of oxidiser (the propellant) required to fuel the hybrid rocket over the total burn time. However the nature of hybrid rocket propulsion requires that there is a preheating period, whereby oxidiser is pumped through the rocket, heating the catalyst bed and reducing ignition delay, effects that were critical to the success of a maiden flight.

To allow for both the preheating of the catalyst bed and ignition delay, an additional 6 seconds is added to the burn time, and design optimisation from the Hybrid Rocket Team has stated that at maximum thrust the mass flow rate is 0.16 kg/s whereas during blipping, due to cooler temperature, the mass flow rate would be higher at 0.216 kg/s. Therefore, during the preheating phase, the amount of oxidiser expected to be used can be calculated by:

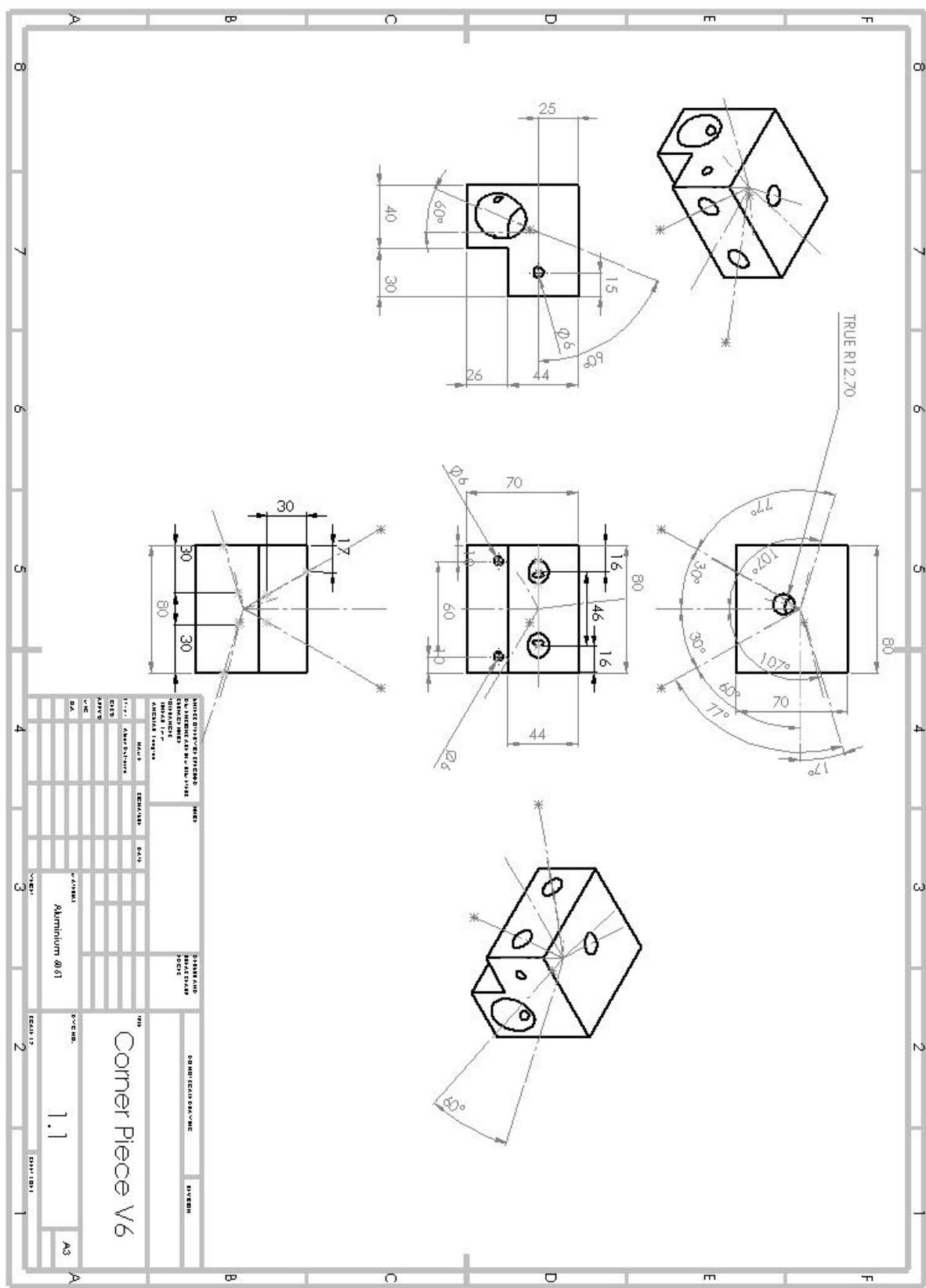
$$V_{ignition} = \frac{\dot{m}t}{\rho} = \frac{0.216 \times 6}{1370} = 0.945 \times 10^{-3} m^3$$

Similarly, the amount of oxidiser required for the flight:

$$V = \frac{\dot{m}t}{\rho} = \frac{0.16 \times 15}{1370} = 1.752 \times 10^{-3} m^3$$

Hence, the total amount of oxidiser required is 2.70l or 3.7kg.

## 16 Appendix E –Technical Drawing for Corner Piece



## 17 Appendix F - Proposal for upgrading the Arduino to a Raspberry Pi

### 17.1 What is Raspberry Pi?

A Raspberry Pi (commonly referred as Pi), is single-board computer, having integrated a processor, video card, audio card. It has the same capabilities as PC unit, requiring only peripherals for programming and visualization. Although having limited computing power, the most current version (Rev 2 Model B) has processor comparable with a general PC from 2007-2009, with 1GB RAM, Quad-Core @ 900 MHz. However, its compact size and low power requirements makes it even better.

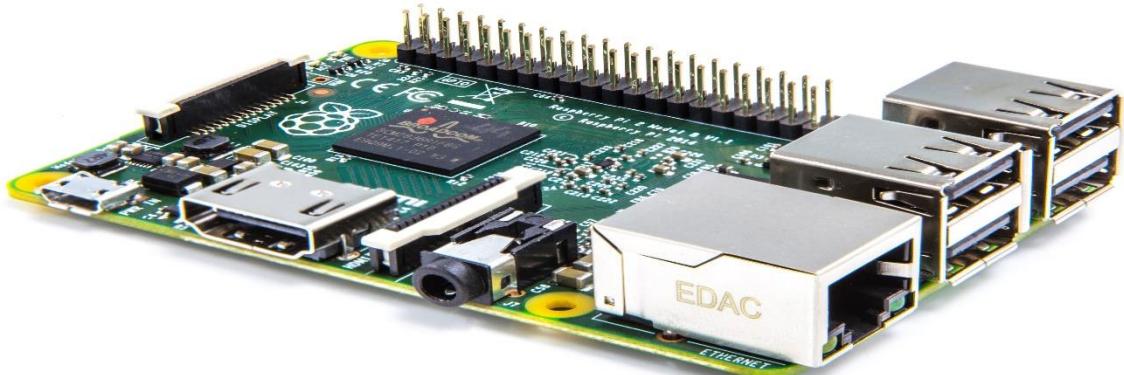


Figure 108: Raspberry Pi 2 Model B. Courtesy to Raspberry Pi Foundation.

### 17.2 Changing the Arduino to a Pi? Is it worth it?

In order to determine the feasibility of changing the electronic of the Hopper, from the current Arduino Mega to a Pi, we must look at the Pros and Cons of each system, both in terms of capabilities and what it means the actual transition.

#### 17.2.1 Pros

| Pi   | Arduino                                       |
|--|---|
| 1 Can do multithreading!   | Already existing code that (partially) works. |
| 2 More computing power.  | Has analog IOs;                               |
| 3 Bigger community;  | Hardware already in place;                    |
| 4 Higher versatility;  | Previous knowledge for the supervisors;       |
| 5 Ability to code it in different languages (can use Python, Java, C, C++);  | Has lower power consumption;                  |
| 6 Potentially more payload available;  | Good for linear processing;                   |
| 7 Can use more devices due to high number of GPIO & I2C bus;   | Good community;                               |
| 8 Code for one device can be easily switched between other platforms: Banana Pi, Odroid (all), any Unix operating board; |   |
| 9 Ease of debugging;   |   |
| 10 Can be operated remotely, without need  |   |

Table 15: Pros of using the Pi and the Arduino

As shown in the table above, the main advantages of changing to a Pi would be the possibility of multithreading, processing multiple actions at the same time. Paired with the option to have a significantly larger number of sensors attached, represents the most important aspect, as they will allow the hopper to operate with an increased precision, while gathering more data. Additionally, a PI is more flexible for programming and debugging: the user is able to execute a

script & debug it at the same time, compared with an Arduino, where the script must first be uploaded, then run. Running the script in Arduino can produce either unexpected bugs that are harder to log and track. Additionally, a script used by a Pi can be swapped easily to another Unix operating board, thus greatly improving the adaptability of the code. On the other hand, a script for Arduino can be only used by Arduino devices, limiting the capabilities of what can be achieved, or by emulating the Arduino IDE in another controller. Finally, one can run & debug a PI remotely, without the need to be physically attached to the Arduino.

However, if the Arduino is kept, the existing code will reduce the time required to program the hopper, while having a lower power consumption and the ability to use analog inputs.

### 17.2.2 Cons

| Pi                                 | Arduino   |
|------------------------------------|---|
| 1 No existing software created;    | No multithreading!  |
| 2 Higher power requirements;       | Debugging is very difficult;  |
| 3 Changes to existing electronics; | Limited capabilities;   |
| 4 Need to purchase more equipment; | Limited versatility in terms of compatibility with sensors and other devices; |
| 5 Limited team knowledge;          | No remote operations;   |
| 6                                  | Just one scripting language available;  |
| 7                                  | Difficult testing;  |

Table 16: Cons of using the Pi and the Arduino

Changing to a Pi will have also some disadvantages. The entire infrastructure, firmware and hardware must be recreated. Moreover, a Pi uses more power compared with the Arduino Mega available. However, given the existing battery that should not be a problem, as the Pi will use a maximum of 800mAh and the battery holds 4500mAh, giving approximately 5.625h (although this figure is more like 2 hours for practical uses). The change will need new sensors and electronics, to be purchased. Lastly, the majority of the team has limited experience with a Pi, however, the learning curve is not too steep.

The main disadvantage of an Arduino is no multithreading capabilities. This greatly limits what can be achieved with the system, and also reduces the versatility of the system. A very important aspect of the system is the need to run multiple processes that have interdependencies. Moreover, the lack of multithreading and the lack of an OS will make debugging significantly more difficult.

Having studied the current code for Arduino, created for the Lunar Hopper MK2, several issues point out that using multithreading is even more important, compared to a typical project. At the moment, the system listens for a command, sends telemetry to the station, acts upon the command, then finally it sends some telemetry back. All this actions are completed in this sequence, one at a time. Although all these methods are executed relatively fast, there is still a 0.05s to 0.1s delay between loops. This is where multi-processing will count. If there is a situation where a new command must be received and executed immediately, or an external factor must be countered, even a 0.05s delay could make the difference. If multiprocessing is to be implemented, all actions can take place simultaneous. This can be translated to a very fast response coming from the controller when either an actions must be completed. Such response is either the hopper shutting down on the spot, a change in thrust, etc. In addition, if the external factor effect is faster than the Arduino loop, this will compensate for it.

While this multithreading will increase the software complexity, it will enable the Hopper to be more robust, and be more flexible.

## 17.3 Conclusion

Given the advantages offered by multithreading that the Pi offers, combined with its expanded capabilities compared with the Arduino, is considered to be a better option for the Lunar Hopper

project. Moreover, it highly expands the capabilities of the project, and it can be delivered at the end.

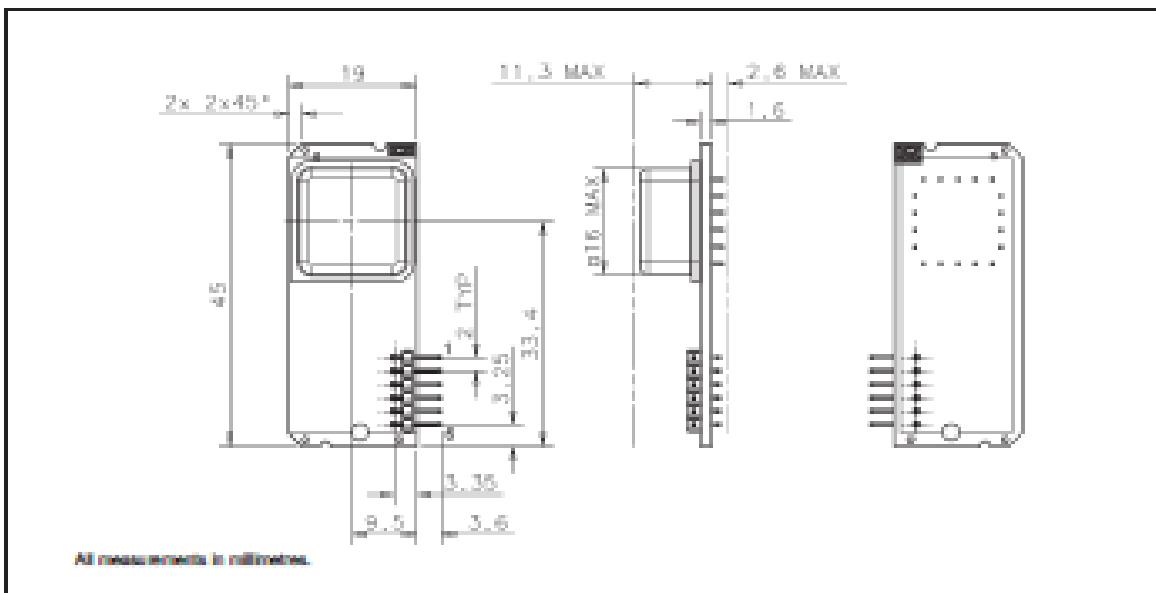
Consequently, the recommendation is to swap the Arduino for a Raspberry Pi.

## 18 Appendix G - Rate Gyro

SILICON  
SENSING.

**CRS05**

Angular Rate Sensor



All measurements in millimetres.

### Specification

| Specification (Typical Data)     | .01                           | .02            | .75            |
|----------------------------------|-------------------------------|----------------|----------------|
| Rate Range                       | <50% $\omega$                 | <200% $\omega$ | <75% $\omega$  |
| Output                           |                               |                |                |
| Analogue voltage (stochastic)    |                               |                |                |
| Scale Factor                     |                               |                |                |
| Nominal                          | 40mV/ $\omega$                | 10mV/ $\omega$ | 21mV/ $\omega$ |
| Variation over temperature range | < ±3%                         |                |                |
| Non Linearity                    | < ±1.0% of full scale         |                |                |
| Bias                             |                               |                |                |
| Setting tolerance                | < ±1.5%                       |                |                |
| Variation over temperature range | < ±3%                         |                |                |
| Repetitive error                 | < ±1%                         |                |                |
| Delt vs. Time                    | < ±0.5%                       |                |                |
| g Sensitivity                    | < ±0.025%g/°C on any axis     |                |                |
| Bandwidth (40° Phase)            | 80Hz                          | 30Hz           | 40Hz           |
| Quiescent Noise                  | < 8mVrms                      | < 4mVrms       | < 6mVrms       |
| Environment                      |                               |                |                |
| Temperature                      | -40°C to +100°C               |                |                |
| Linear acceleration              | < 100g                        |                |                |
| Shock                            | 200g (1ms, 1/2 sine)          |                |                |
| Vibration                        | 2g rms (20Hz to 20Hz, random) |                |                |
| Cross axis sensitivity           | < 5%                          |                |                |
| Mass                             | <11 gram                      |                |                |
| Electrical                       |                               |                |                |
| Voltage (supply)                 | +4.75V to +5.25V              |                |                |
| Current (supply)                 | < 35mA (running)              |                |                |
| Noise and Ripple                 | < 10mVrms (up to 10kHz)       |                |                |
| Ready Time                       | < 0.5s                        |                |                |

#### Pin Connections

| Pin | Function    |
|-----|-------------|
| 1   | Rate Output |
| 2   | +5V         |
| 3   | CBIT        |
| 4   | Ground      |
| 5   | BIT         |
| 6   | NC          |

Catted Road, Southwest  
Plymouth, Devon,  
PL6 6EE, United Kingdom

T: +44 (0)1752 722000  
E: +44 (0)1752 722201  
E: [sales@siliconsensing.com](mailto:sales@siliconsensing.com)  
W: [siliconsensing.com](http://siliconsensing.com)

1-10 Fuso-Cho  
Amagasaki  
Hyogo 660-0821, Japan  
T: +81 0 6443 5260  
E: +81 0 6443 5210  
E: [sales@siliconsensing.com](mailto:sales@siliconsensing.com)  
W: [siliconsensing.com](http://siliconsensing.com)

specification subject to  
change without notice.  
© copyright silicon sensing  
systems limited  
Printed in August 2008 issue

## 19 Appendix H – Test plans electronics

### 19.1 Test 1

#### 19.1.1 Objectives:

1. Measure the input and output voltage during 3 cycles of Gyro 2, when Gyro 1 is present in the system.

#### 19.1.2 Procedure:

1. Equipment: Gyro 1, Gyro 2, multimeter, Arduino Mega, Hopper sim, and Marian's Laptop (ASUS K53S);
2. Plug the sim + Arduino to the laptop;
3. Measure initial input and output voltages;
4. Start the process;
5. Measure input and output currents during calibration;
6. Measure input and output current when the system is active;
7. Repeat for 3 cycles.

#### 19.1.3 Results:

|                         | Output (V) | Input (V) |
|-------------------------|------------|-----------|
| Initial<br>1            | 2.27       | 4.57      |
| 2                       | 2.29       | 4.6       |
| 3                       | 2.27       | 4.59      |
| During calibration<br>1 | 2.32       | 4.67      |
| 2                       | 2.33       | 4.67      |
| 3                       | 2.32       | 4.67      |
| During Operation<br>1   | 2.22       | 4.3       |
| 2                       | 2.22       | 4.3       |
| 3                       | 2.14       | 4.3       |

#### 19.1.4 Conclusions:

1. A bug in code that did not read Y Gyro (1) if X Gyro (2) had a change to be made;
2. Changed to code to add the angles;
3. While the system was stationary, it still picked random vibrations, getting the system to MECO
  - a. This means that if the system would RUN PERFECTLY straight, ACS would fire in one direction, destabilising the system
4. There is still a clear drop in voltage during operation
  - a. Might be the LEDs?

## 19.2 Test Report

### 19.2.1 Objective

- 1) Record the values for input and output voltage of Rate Gyro 2 while the LEDs are swapped for resistors.

### 19.2.2 Item tested description

The main item tested is the rate gyro, in conjunction with the swapped LEDs for resistors.

### 19.2.3 Equipment description

Equipment used is the prototype board that contains the wiring similar to what the actual hopper will look like.

The wiring allows the user to simulate the hopper. The main difference between last test and current version is that the LEDs have been swapped for 560ohm resistors.

Arduino Mega 2560 used as the main controller.

Power Supply provided by Dr. G. Roberts.

2 Rate gyros and 1 accelerometer (inherited from last year).

Any laptop with Arduino IDE.

1 multimeter.

### 19.2.4 Test sequence

- 1) Connect the Arduino mega to the breadboard;
- 2) Connect the accelerometer and 2 rate gyros to the breadboard. Make sure that the gyros are properly connected.
- 3) Connect the Arduino to a laptop and upload the later version of the code the Arduino.
- 4) Disconnect the Arduino from the laptop.
- 5) Switch on the power supply and set it to 11-12V DC. Connect the hooks to the 5.2mm Jack, and plug the jack into the Arduino.
- 6) Measure the initial input and output voltage from the 1 rate gyro. The second one should be similar, but one can check the values.
- 7) Start the software execution by pressing the start button.
- 8) Measure the input and output voltages while the system is calibration. This should take 40 seconds.
- 9) Measure the input and output voltages while the system is executing the ACS maneuvers. Up to 20 seconds after calibration has finished.
- 10) Record the 3 sets of voltages.
- 11) Repeat the experiment 2 more times, 3 in total.

### 19.2.5 Data discussion

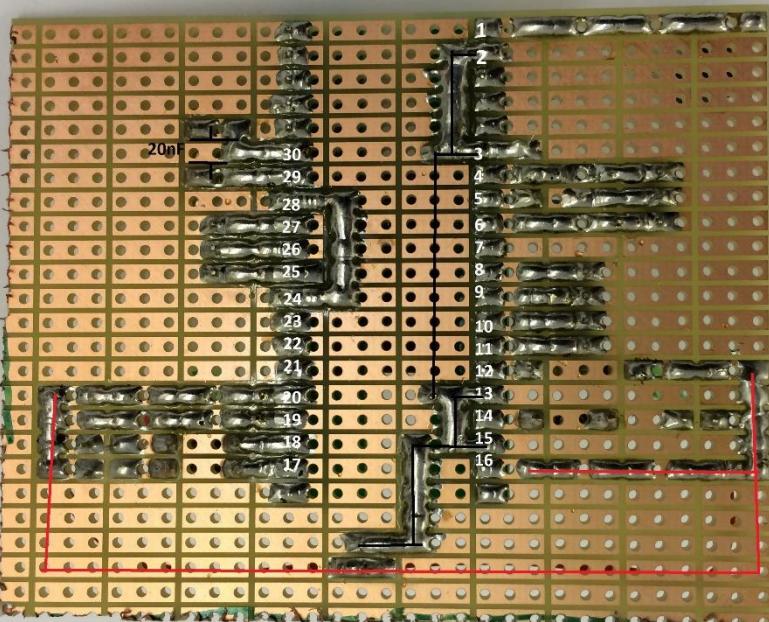
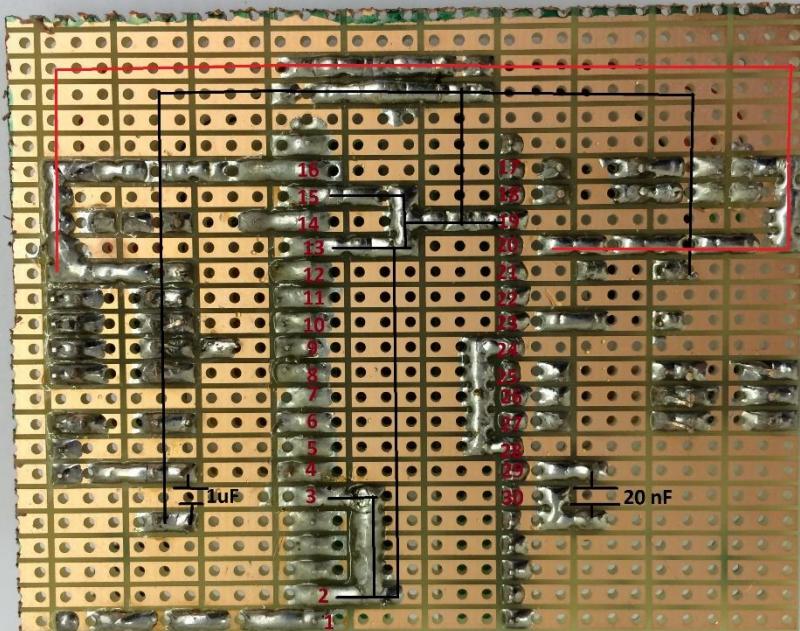
The input voltage from the power source was 11V DC.

|                          | <b>Input (V)</b> | <b>Output (V)</b> |
|--------------------------|------------------|-------------------|
| <i>Initial reading 1</i> | 4.43             | 2.23              |
|                          | 4.73             | 2.36              |
|                          | 4.75             | 2.36              |
| <i>Calibration</i>       | 4.55             | 2.36              |
|                          | 4.76             | 2.37              |
|                          | 4.77             | 2.37              |
| <i>ACS</i>               | 4.58             | 2.28              |
|                          | 4.76             | 2.37              |
|                          | 4.77             | 2.38              |

#### **19.2.6 Conclusion**

1. For the first run, the overall current in all 3 stages of the process has down. However, this might be due to just initial stating of the system, with all electronics cold.
2. Overall, the voltages are consistent. The reading during calibration are a good match of the ACS readings, while the system should “move”.
3. All current provide a similar “no movement” voltage. The system is stable, with no major fluctuations.
4. Main source of fluctuation in previous experiments was coming from LEDs not being active during calibration. However, the current draw from resistors is undetectable with or without the intended flickering imposed for testing.

## 20 Appendix I – ZULU module breakout board detailed schematics.



## 21 Appendix J – Pilot Trainer Simulator Software

```
/* simulator.ino

 * Code for Lunar Hopper Pilot Training Simulator: obtain a
 * throttle setting reading from a potentiometer connected to
 * analog input and compute resulting vehicle flight velocity and
 * altitude.

 *
 * Adam Elkins, 05/04/2016
 */

// Declare variables and constants here
const int throttlePin = 0; // Declare throttle pin
int velocityPin = 11; // Pins for output to voltmeters
int altitudePin = 10;
boolean prnt = true; // true to print to serial monitor

float P = 0; // ox tank pressure
float P_demand; // demand pressure, function of throttle setting
float V_N2 = 0.0002; // initial oxidiser tank N2 volume
float dV_N2 = 0; // change in N2 volume due to expulsion of oxidiser
from tank
float Gamma = 1.4; // ratio of specific heat capacities
float Const = 0; // P*V^Gamma, adiabatic constant
float oxrate = 0; // oxidiser flow rate
float HDPERate = 0; // fuel melt rate
float flowrate = 0; // total propellant flow rate
const float dt = 0.02; // time step in milliseconds
const int maxtime = 10; // length of run
float elapsed_time = 0; //
float Isp = 0; // specific impulse, 0 for monoprop mode
const int Isp_val = 220; // Value of Isp for biprop mode
float monopropTime = 2; // delay before solid propellant combustion
starts
float m_fuel = 4 + 1; // oxidiser plus fuel grain
const int m_structure = 28; // mass of structure
float dm = 0; // propellant increment
const float g = 9.81; // gravity
float s = 0; // displacement, spacecraft altitude
float v = 0; // spacecraft velocity
float Thrust = 0;
float Weight = 0;
float Reaction = 0;
```

```

float a = 0;
int startTime = 0;
int endTime = 0;
int delayTime = 0; // variables to compute correct delay time
float throttle;

float fuelrate() {
    /* obtain demand pressure */
    throttle = analogRead(throttlePin)/1023.0;
    P_demand = 40*throttle;
    if(P_demand > P) {
        P = P_demand;
    }
    Const = P*pow(V_N2, Gamma); // Compute adiabatic constant
    oxrate = 0.16*P/40; // assume linear relationship between pressure
    and ox flow rate
    dV_N2 = oxrate*dt/1370; // find volume of ox used from density and
    dm_ox
    V_N2 += dV_N2; // increase N2 volume as ox expelled from tank
    P = Const*pow(V_N2, -Gamma); // compute new pressure due to
    expansion of N2
    /* Compute HTPB rate: */
    if(elapsed_time < monopropTime) {
        HDPERate = 0; // Monoprop mode
        Isp = 0;
    }
    else {
        HDPERate = (1 + (0.2/maxtime)*(elapsed_time-
        monopropTime))*(0.16/8.5); // compute HDPE rate from proportion of
        propellant used and heating effect
        Isp = Isp_val;
    }
    flowrate = oxrate + HDPERate;
    return flowrate;
}

void setup() {
    //
    if (prnt) {
        Serial.begin(115200);
}

```

```

Serial.println("=====");
Serial.println();
Serial.println();
}

void loop() {
    //
startTime = millis();
simulator();
endTime = millis();

/* to compute dynamic delay:
 * get millis() at start of loop
 * get millis at end of loop and subtract to get difference
 * delayTime = dt - (difference)
 */

delayTime = int(dt*1000) - (endTime - startTime);
//Serial.println(delayTime);
delay(delayTime);
}

void simulator(){
if (elapsed_time < maxtime) {
    /* Compute sum of forces and accelerations: */
    dm = fuelrate()*dt; /* obtain throttle setting */
    if (m_fuel - dm <= 0) { /* ran out of fuel */
        dm = m_fuel;
        m_fuel = 0;
    }
    else {
        m_fuel -= dm; /* reduce vehicle mass as fuel burnt */
    }

    /* Compute forces acting on vehicle, obtain resultant and
    acceleration */
    Thrust = Isp*g*dm/dt; /* Exhaust velocity (Isp*g)*mdot */
    Weight = (m_fuel + m_structure)*g;
    if (s <= 0) { /* Vehicle is on ground, compute ground
    reaction force */
        if (Thrust < Weight) {

```

```

    Reaction = Weight - Thrust;
}
}

else {
    Reaction = 0;
}

/* compute acceleration: sum of forces divided by mass */
a = (Thrust + Reaction - Weight)/(m_fuel + m_structure);
/* Compute altitude s = s0 + u*t + 0.5*a*t^2: */
s += v*dt + 0.5*a*dt*dt;
/* Compute velocity v = u + a*t: */
v += a*dt;

if (s <= 0) { /* Landing or on ground */
    s = 0; /* s < 0 is unphysical */
    if (v != 0) {
        if (prnt) {
            Serial.print("Landing; impact speed = ");
            Serial.println(v); /* print landing speed */
        }
    }
    v = 0; /* vehicle has landed */
    a = 0;
}

elapsed_time += dt; /* Increment time */
/* Print vehicle altitude, velocity and time: */
if (prnt) {
    Serial.print("altitude ");
    Serial.print(s,9);
    Serial.print(" velocity ");
    Serial.print(v,6);
    Serial.print(" acceleration ");
    Serial.print(a);
    Serial.print(" throttle ");
    Serial.print(analogRead(throttlePin)/1023.0);
    Serial.print(" fuel ");
    Serial.print(m_fuel);
    Serial.print(" Pressure ");
    Serial.print(P);
    Serial.print(" Thrust ");
}

```

```

        Serial.print(Thrust);
        Serial.print(" time ");
        Serial.print(elapsed_time, 6);
        Serial.print("\n");
    }

// Output altitude and velocity to analog pins for voltmeters
if(s < 5) {
    analogWrite(altitudePin, s*255/5.0);
}
else {
    if(s >= 5) {
        analogWrite(altitudePin, 255);
    }
}

if(v > -2.5 && v < 2.5) {
    analogWrite(velocityPin, (v + 2.5)*255/5.0);
}
else {
    if(v >= 2) {
        analogWrite(velocityPin, 255);
    }
    if(v <= -2) {
        analogWrite(velocityPin, 0);
    }
}
}
}

```

