

Analysis of Potential Small Satellite Launch Operations at the Denel Overberg Test Range

Aravind Arunakirinathar

Submitted in fulfilment of the academic requirements for the degree of Master of Science in Mechanical Engineering, College of Agriculture, Engineering and Science, University of KwaZulu-Natal.

Durban, South Africa
November 2022

Supervisor: Professor Jean-Francois P. Pitot de la Beaujardiere

Co-Supervisor: Professor Michael J. Brooks

DECLARATION 1 - PLAGIARISM

I, Aravind Arunakirinathar, declare that

1. The research reported in this thesis, except where otherwise indicated, is my original research.
2. This thesis has not been submitted for any degree or examination at any other university.
3. This thesis does not contain other persons' data, pictures, graphs or other information, unless specifically acknowledged as being sourced from other persons.
4. This thesis does not contain other persons' writing, unless specifically acknowledged as being sourced from other researchers. Where other written sources have been quoted, then:
 - a. Their words have been re-written but the general information attributed to them has been referenced
 - b. Where their exact words have been used, then their writing has been placed in italics and inside quotation marks, and referenced.
5. This thesis does not contain text, graphics or tables copied and pasted from the Internet, unless specifically acknowledged, and the source being detailed in the thesis and in the References sections.

Signed:..... Date: 21/03/2023
Mr Aravind Arunakirinathar

As the candidate's supervisor I agree/do not agree to the submission of this thesis.

Signed:..... Date: 23/03/23
Prof Jean-Francois Pitot de la Beaujardiere

As the candidate's co-supervisor I agree/do not agree to the submission of this thesis.

Signed:..... Date: 23/03/2023
Prof Michael J. Brooks

DECLARATION 2 - PUBLICATIONS

Arunakirinathar A., Pitot de la Beaujardiere, J., Brooks, M. J., 2021, ‘Analytical first derivatives of the direct Hermite-Simpson collocation formulation of optimal control problems’, *12th South African Conference on Computational and Applied Mechanics (SACAM2020)*, Cape Town, South Africa, November 29- December 1, 2021.

Signed:..... Date: 21/03/2023
Mr Aravind Arunakirinathar

ACKNOWLEDGEMENTS

I would like to thank the following for their help in the completion of the dissertation:

- I am grateful for my supervisor Prof Jean Pitot, and my co-supervisor, Prof Mike Brooks for always providing motivation, guidance and support whenever I needed. I thank them for the opportunity of presenting parts of my work at the South African Computational and Applied Mechanics 2020 conference.
- The Aerospace Systems Research Group (ASReG) for the opportunity to witness their 2021 sounding rocket launches at the Overberg test range, and for all the advice and guidance they provided, particularly Nino Wunderlin.
- My mum, Santhini, and dad, Arunakirinathar, for their support and guidance, and my sisters Nilambari and Gayathri for always believing in me, and Tasneem Essop for her constant encouragement and support during my Masters journey.
- ASReG, the National Research Foundation and the Department of Science and Innovation for all the funding they provided to support my studies.

ABSTRACT

One of the primary objectives of the South African First Integrated Rocket Engine (SAFFIRE) programme of UKZN's Aerospace System Research Group (ASReG) is to develop the capacity for orbital injection missions to Low Earth Orbits (LEOs) from South Africa. The most likely launch site for these missions is the Denel Overberg Test Range (OTR) near Cape Agulhas in the Western Cape. In order to determine the suitability of OTR as a launch site, it is imperative to gain an understanding of the performance, mechanics and structural loads of a vehicle entering orbit. The goal of this dissertation is to analyse the performance of a variety of modern two-stage launch vehicles as they travel along orbital injection trajectories into LEOs from OTR.

This study considers solutions for the ascent-to-orbit trajectory for various launch vehicles. The primary method was to utilise trajectory optimisation methods and this was achieved by developing an optimal control solver, which makes use of direct Hermite-Simpson collocation methods, and a sequential quadratic programming solver. In order to improve the robustness and speed of the solver, formulae for the first order analytical derivative information of direct Hermite-Simpson collocation were developed. The optimal control solver was then validated using various linear and nonlinear examples from literature.

The optimal control solver was used to analyse the performance of various hypothetical missions conducted by the following established launch vehicles: Rocket Lab's Electron, SpaceX's Falcon 1, SpaceX's Falcon 9, and ASReG's proposed small satellite launch vehicle, CLV. As a baseline comparison, all vehicles were launched from OTR into various LEOs. The payloads, trajectories, control histories and structural loads of these vehicles for injection were investigated. Finally, the effect of perigee altitude, inclination, and eccentricity of orbits on the extracted results was studied.

The payload performance of the launch vehicles considered were relatively similar to that provided by each vehicle's corresponding payload user guide. On all missions, the altitude of the Electron, Falcon 9 and CLV would constantly increase with range, however the Falcon 1 would tend to rise, dip, and then rise once more on missions to orbits with a perigee altitude of 200 km. Such trajectories are referred to as lofted trajectories and are common among vehicles with a low upper stage thrust to weight ratio (Patton and Hopkins, 2006), such as the Falcon 1. The tangent yaw and pitch of the thrust direction was highly linear for all analysed missions. This result allows for a reasonable control law which can be used to determine trajectory solutions using indirect optimal control methods. This study demonstrates the viability of the Denel Overberg Test Range as a competitive base of operation for space launch missions to LEO.

TABLE OF CONTENTS

| | |
|---|------|
| DECLARATION 1 - PLAGIARISM..... | i |
| DECLARATION 2 - PUBLICATIONS | ii |
| ACKNOWLEDGEMENTS | iii |
| ABSTRACT..... | iv |
| LIST OF FIGURES | viii |
| LIST OF TABLES..... | xii |
| NOMENCLATURE..... | xiii |
| Abbreviations..... | xiii |
| Symbols..... | xiv |
| 1. INTRODUCTION | 1 |
| 1.1. Small Satellite Launches | 1 |
| 1.2. Review of Orbits..... | 4 |
| 1.3. International Space Laws and Practices..... | 8 |
| 1.4. South African Launch Capabilities | 9 |
| 1.5. Problem Statement and Research Objectives..... | 10 |
| 1.6. Dissertation Outline | 12 |
| 2. LITERATURE REVIEW..... | 13 |
| 2.1. Optimal Control | 13 |
| 2.2. Nonlinear Programming..... | 17 |
| 2.2.1. Scaling..... | 18 |
| 2.2.2. Gradient Information..... | 19 |

| | |
|---|----|
| 2.2.3. Sparsity..... | 19 |
| 2.2.4 Sequential Quadratic Programming and Interior Point Methods | 20 |
| 2.3. Ascent-to-Orbit Specific Trajectory Optimisation..... | 21 |
| 2.3.1. Gravity Turn..... | 23 |
| 2.3.2. Bi-Linear and Linear Tangent Laws | 25 |
| 2.4. Optimal Control Software | 26 |
| 2.5. NLP Software..... | 27 |
| 3. OPTIMAL CONTROL SOLVER ALGORITHM..... | 29 |
| 3.1. Optimal Control Algorithm Overview | 29 |
| 3.2 Transcription | 30 |
| 3.3. First Order Analytical Derivatives | 32 |
| 3.4. Interpolation of Solution and Error Estimation..... | 36 |
| 4. SOLVER VALIDATION | 38 |
| 4.1. Linear Dynamic System..... | 38 |
| 4.2. Co-Planar Circular Orbit Transfer | 40 |
| 4.2.1. Minimum Time Orbit Transfer | 40 |
| 4.2.2. Maximum Radius Orbit Transfer | 43 |
| 4.3. Delta III Launch Vehicle..... | 45 |
| 5. ASCENT-TO-ORBIT TRAJECTORY MODELLING..... | 48 |
| 5.1. Force Models and Reference Frames..... | 48 |
| 5.2. Ascent Trajectory Phases | 51 |
| 5.2.1. Phase 1: Vertical Ascent | 53 |
| 5.2.2. Phase 2: Initial Pitchover | 53 |

| | |
|---|-----|
| 5.2.3. Phase 3: Gravity Turn | 54 |
| 5.2.4. Phase 4: Stage Separation | 54 |
| 5.2.5. Phase 5: Vacuum Ascent..... | 54 |
| 6. RESULTS | 58 |
| 6.1. Rocket lab’s Electron..... | 60 |
| 6.2. SpaceX’s Falcon 1 | 64 |
| 6.3. SpaceX’s Falcon 9 | 68 |
| 6.4. ASReG’s CLV | 72 |
| 6.5. Launch Vehicle Comparison..... | 76 |
| 7. CONCLUSION | 81 |
| 7.1. Optimal Control Solver | 81 |
| 7.2. Results..... | 83 |
| 7.3. Future Work | 85 |
| REFERENCE LIST | 87 |
| APPENDIX A: ORBITAL ELEMENTS..... | 95 |
| APPENDIX B: COORDINATE TRANSFORMATION MATRICES | 97 |
| APPENDIX C: SIMULATED ALTITUDE VERSUS RANGE PLOTS | 98 |
| APPENDIX D: RASAERO™ INPUTS..... | 111 |
| APPENDIX E: COMPUTER SPECIFICATIONS | 112 |

LIST OF FIGURES

| | |
|--|----|
| Figure 1-1: Bar graph depicting small satellite trends from 2012-2021 (Bryce Space and Technology, 2022) | 3 |
| Figure 1-2: Examples of popular orbits around Earth (Kasonsuwanx, et al., 2017) | 4 |
| Figure 1-3: Polar orbit plane relative to Earth’s equatorial (orbit) plane (Damp, 2017)..... | 6 |
| Figure 1-4: SSO plane relative to equatorial plane and Earth’s poles (ESA, 2020b) | 6 |
| Figure 1-5: Diagram depicting a satellite in SSO passing over sections on Earth at the same local mean solar time (SFSU's Department of Earth and Climate Sciences, 2022)..... | 7 |
| Figure 1-6: Map depicting paths into various orbit inclinations from OTR | 10 |
| Figure 1-7: Diagram showing hypothetical paths from Earth to orbit | 11 |
| Figure 2-1: Typical gravity turn trajectory phases | 23 |
| Figure 4-1: Linear validation example results | 40 |
| Figure 4-2: The orbital transfer trajectory solution of Bryson and Ho (1975)..... | 42 |
| Figure 4-3: Developed solver's orbital transfer trajectory solution for minimum time..... | 42 |
| Figure 4-4: Moyer and Pinkham’s (1964) thrust direction solution..... | 43 |
| Figure 4-5: Developed solver's thrust direction solution for minimum time | 43 |
| Figure 4-6: Developed solver's orbital transfer trajectory solution for maximum orbital radius | 44 |
| Figure 4-7: Developed solver's thrust direction solution for maximum orbital radius | 45 |
| Figure 4-8: Betts’ Delta III altitude vs time solution (Betts, 2010) | 47 |
| Figure 4-9: Developed solver’s Delta III altitude vs time solution..... | 47 |
| Figure 5-1: GEF and ZEN reference frames (Curtis, 2005)..... | 48 |
| Figure 5-2: Thrust pitch and azimuth in ZEN reference frame | 52 |
| Figure 6-1: Electron tangent of thrust pitch vs τ ($e=0$, $i=40^\circ$) | 59 |
| Figure 6-2: Electron tangent of thrust yaw vs τ ($e=0$, $i=40^\circ$)..... | 59 |
| Figure 6-3: Electron first stage altitude vs time | 60 |
| Figure 6-4: Electron payload vs perigee altitude ($e=0$)..... | 61 |

| | |
|---|----|
| Figure 6-5: Electron payload vs perigee altitude ($e=0.005$)..... | 62 |
| Figure 6-6: Electron payload vs perigee altitude ($e=0.01$)..... | 62 |
| Figure 6-7: Electron altitude vs range ($e=0, i=40^\circ$) | 62 |
| Figure 6-8: Average $ r $ of Electron's tangent pitch..... | 63 |
| Figure 6-9: Average $ r $ of Electron's tangent yaw | 63 |
| Figure 6-10: Falcon 1 payload vs perigee altitude ($e=0$) | 65 |
| Figure 6-11: Falcon 1 payload vs perigee altitude ($e=0.005$) | 65 |
| Figure 6-12: Falcon 1 payload vs perigee altitude ($e=0.01$) | 65 |
| Figure 6-13: Falcon 1 altitude vs range ($e=0, i=40^\circ$)..... | 66 |
| Figure 6-14: Average $ r $ of Falcon 1's tangent pitch of the first burn..... | 66 |
| Figure 6-15: Average $ r $ of Falcon 1's tangent pitch of the second burn | 66 |
| Figure 6-16: Average $ r $ of Falcon 1's tangent yaw of the first burn | 67 |
| Figure 6-17: Average $ r $ of Falcon 1's tangent yaw of the first burn second burn..... | 67 |
| Figure 6-18: Falcon 9 payload vs perigee altitude ($e=0$) | 69 |
| Figure 6-19: Falcon 9 payload vs perigee altitude ($e=0.005$) | 69 |
| Figure 6-20: Falcon 9 payload vs perigee altitude ($e=0.01$) | 70 |
| Figure 6-21: Falcon 9 altitude vs range ($e=0, i=40^\circ$) | 70 |
| Figure 6-22: Average $ r $ of Falcon 9's tangent pitch of the first burn..... | 71 |
| Figure 6-23: Average $ r $ of Falcon 9's tangent pitch of the second burn | 71 |
| Figure 6-24: Average $ r $ of Falcon 9's tangent yaw of the first burn | 71 |
| Figure 6-25: Average $ r $ of Falcon 9's tangent yaw of the second burn | 72 |
| Figure 6-26: CLV payload vs perigee altitude ($e=0$) | 73 |
| Figure 6-27: CLV payload vs perigee altitude ($e=0.005$) | 74 |
| Figure 6-28: CLV payload vs perigee altitude ($e=0.01$) | 74 |
| Figure 6-29: CLV altitude vs range ($e=0, i=40^\circ$)..... | 74 |
| Figure 6-30: Average $ r $ of CLV's tangent pitch | 75 |
| Figure 6-31: Average $ r $ of CLV's tangent yaw | 75 |
| Figure 6-32: Pitch rate vs orbital altitude..... | 76 |

| | |
|--|-----|
| Figure 6-33: Pitch rate vs orbital inclination..... | 76 |
| Figure 6-34: Time to orbit vs orbital altitude..... | 77 |
| Figure 6-35: Time to orbit vs orbital inclination..... | 77 |
| Figure 6-36: Time to orbit vs orbital eccentricity | 77 |
| Figure A-1: Elliptic orbit with orbital elements | 95 |
| Figure C-1: Electron altitude vs range (e=0, i=40°)..... | 98 |
| Figure C-2: Electron altitude vs range (e=0, i=64.435°)..... | 98 |
| Figure C-3: Electron altitude vs range (e=0, SSO) | 99 |
| Figure C-4: Electron altitude vs range (e=0.005, i=40°)..... | 99 |
| Figure C-5: Electron altitude vs range (e=0.005, i=64.435°)..... | 99 |
| Figure C-6: Electron altitude vs range (e=0.005, SSO) | 100 |
| Figure C-7: Electron altitude vs range (e=0.01, i=40°)..... | 100 |
| Figure C-8: Electron altitude vs range (e=0.01, i=64.435°)..... | 100 |
| Figure C-9: Electron altitude vs range (e=0.01, SSO) | 101 |
| Figure C-10: Falcon 1 altitude vs range (e=0, i=40°) | 101 |
| Figure C-11: Falcon 1 altitude vs range (e=0, i=64.435°) | 101 |
| Figure C-12: Falcon 1 altitude vs range (e=0, SSO)..... | 102 |
| Figure C-13: Falcon 1 altitude vs range (e=0.005, i=40°) | 102 |
| Figure C-14: Falcon 1 altitude vs range (e=0.005, i=64.435°) | 102 |
| Figure C-15: Falcon 1 altitude vs range (e=0.005, SSO)..... | 103 |
| Figure C-16: Falcon 1 altitude vs range (e=0.01, i=40°) | 103 |
| Figure C-17: Falcon 1 altitude vs range (e=0.01, i=64.435°) | 103 |
| Figure C-18: Falcon 1 altitude vs range (e=0.01, SSO)..... | 104 |
| Figure C-19: Falcon 9 altitude vs range (e=0, i=40°) | 104 |
| Figure C-20: Falcon 9 altitude vs range (e=0, i=64.435°) | 104 |
| Figure C-21: Falcon 9 altitude vs range (e=0, SSO)..... | 105 |
| Figure C-22: Falcon 9 altitude vs range (e=0.005, i=40°) | 105 |
| Figure C-23: Falcon 9 altitude vs range (e=0.005, i=64.435°) | 105 |

| | |
|---|-----|
| Figure C-24: Falcon 9 altitude vs range (e=0.005, SSO)..... | 106 |
| Figure C-25: Falcon 9 altitude vs range (e=0.01, i=40°) | 106 |
| Figure C-26: Falcon 9 altitude vs range (e=0.01, i=64.435°) | 106 |
| Figure C-27: Falcon 9 altitude vs range (e=0.01, SSO)..... | 107 |
| Figure C-28: CLV altitude vs range (e=0, i=40°) | 107 |
| Figure C-29: CLV altitude vs range (e=0, i=64.435°) | 107 |
| Figure C-30: CLV altitude vs range (e=0, i=SSO) | 108 |
| Figure C-31: CLV altitude vs range (e=0.005, i=40°) | 108 |
| Figure C-32: CLV altitude vs range (e=0.005, i=64.435°) | 108 |
| Figure C-33: CLV altitude vs range (e=0.005, i=SSO) | 109 |
| Figure C-34: CLV altitude vs range (e=0.01, i=40°) | 109 |
| Figure C-35: CLV altitude vs range (e=0.01, i=64.435°) | 109 |
| Figure C-36: CLV altitude vs range (e=0.1, i=SSO) | 110 |

LIST OF TABLES

| | |
|---|-----|
| Table 1-1: Classification of smallsats by wet mass (Bryce Space and Technology, 2022) | 1 |
| Table 1-2: Classification of launch vehicles by payload capacity (Wekerle et al., 2017) | 1 |
| Table 1-3: Country of origin of launch vehicle developers (Niederstrasser, 2021) | 4 |
| Table 2-1: Optimal control software overview | 27 |
| Table 4-1: Delta III trajectory phases (Betts, 2010)..... | 46 |
| Table 6-1: Electron simulation parameters | 61 |
| Table 6-2: Electron structural load parameters | 61 |
| Table 6-3: Falcon 1 Simulation Parameters | 64 |
| Table 6-4: Falcon 1 Structural Load Parameters..... | 64 |
| Table 6-5: Falcon 9 Simulation Parameters | 68 |
| Table 6-6: Falcon 9 Structural Load Parameters..... | 69 |
| Table 6-7: CLV Simulation Parameters | 73 |
| Table 6-8: CLV Structural Load Parameters | 73 |
| Table D-1: RASAero™ inputs | 111 |

NOMENCLATURE

Abbreviations

| Acronym | Description |
|----------------|--|
| ASReG | Aerospace Systems Research Group |
| CLV | Commercial Launch Vehicle |
| DAE | Differential Algebraic Equations |
| 3-DOF | Three Degrees of Freedom |
| FT | Full Thrust |
| GEF | Geocentric Equatorial Frame |
| GEO | Geosynchronous Equatorial Orbit |
| IP | Interior Point |
| IT | Intermediate Thrust |
| LEO | Low Earth Orbit |
| NLP | Nonlinear Programming |
| NT | Null Thrust |
| OTR | Denel Overberg Test Range |
| SAFFIRE | South African First Integrated Rocket Engine |
| SQP | Sequential Quadratic Programming |
| SSO | Sun-Synchronous Orbit |
| TPBVP | Two-Point Boundary Value Problem |
| ZEN | Zenith-East-North |

Symbols

| Symbol | Description | Unit |
|--------------------|--|------------------|
| a | Semi-major axis | m |
| a_{axial} | Axial acceleration | g |
| A_e | Nozzle exit area | m ² |
| c | Nonlinear programming constraints | - |
| c | Constant | - |
| C | Cost of optimal control function | - |
| C_D | Drag coefficient | - |
| C_L | lift coefficient | - |
| D | Drag | N |
| e | Eccentricity | - |
| g_o | Standard Earth gravitational acceleration | m/s ² |
| h | Altitude | m |
| \mathcal{H} | Hamiltonian | - |
| i | Inclination | rad |
| J_2 | Second zonal harmonic | - |
| L | Integrand of cost of optimal control problem | - |
| \mathcal{L} | Lagrangian | - |
| L | Lift | N |
| m | Mass | kg |
| \dot{m} | Mass flow rate | kg/s |
| p_a | Atmospheric pressure | Pa |
| q | Dynamic pressure | Pa |
| r | Pearson's correlation coefficient | - |
| r | Radial distance | m |
| R | Planet radius | m |
| R_E | Radius of Earth | m |
| S_p | Aerodynamic reference area | m ² |
| t | Time | s |
| T | Thrust | N |
| T_{vacuum} | Vacuum thrust | N |
| \mathbf{r} | Position vector | m |
| \mathbf{u} | Control vector | - |
| \mathbf{v} | Velocity vector | m/s |
| v_E | East component of velocity | m/s |
| v_N | North component of velocity | m/s |
| v_z | Zenith component of velocity | m/s |
| \mathbf{v}_{rel} | Velocity relative to wind | m/s |
| \mathbf{w} | Wind velocity | m/s |
| \mathbf{x} | State vector | - |

Greek

| Symbol | Description | Unit |
|----------------|---------------------------------------|---------------------------------|
| $\dot{\Omega}$ | Rate of precession of the node Line | rad/s |
| β | Thrust direction angle (2-DOF) | rad |
| ϕ | Latitude | rad |
| θ | Celestial longitude | rad |
| μ | Gravitational parameter | km ³ /m ² |
| ρ | Air density | kg/m ² |
| τ | Scaled time | - |
| ψ | Launch azimuth | rad |
| ϵ | Maximum relative local error estimate | - |
| λ | Adjoint vector | - |
| ω | Angular velocity of Earth | rad/s |

1. INTRODUCTION

This study addresses the potential of South Africa, and in particular, the Denel Overberg Test Range (OTR), as a launch site for orbital missions to space. The introduction provides a brief overview of and insight into small satellite launches, South Africa’s capabilities in respect of orbital launch, and types of orbits. It then details the dissertation problem statement and research objectives, and indicates how the dissertation aims to meet its objectives. Finally, it concludes with a dissertation outline.

1.1. Small Satellite Launches

In order to discuss small satellite (smallsat) launches, it is necessary to discuss the launchers typically used for these missions. In this dissertation, launch vehicles will be classified as either small or large. Tables 1-1 and 1-2 summarise the different small satellite classes (according to the Federal Aviation Administration) and launch vehicle classes (according to Wekerle et al. (2017)).

Table 1-1: Classification of smallsats by wet mass (Bryce Space and Technology, 2022)

| Smallsats | Wet Mass (kg) |
|------------------|----------------------|
| Pico | 0.1-1 |
| Nano | 1.1-10 |
| Micro | 11-200 |
| Mini | 201-600 |

Table 1-2: Classification of launch vehicles by payload capacity (Wekerle et al., 2017)

| Launch Vehicles | Payload Capacity (kg) |
|------------------------|------------------------------|
| Micro | ≤ 500 |
| Small | 501-2000 |
| Medium | 2001-20000 |
| Heavy | >20000 |

Throughout this dissertation, the term “small launch vehicles” will refer to vehicles intended to transport payloads up to 2000 kg to Low Earth Orbits (LEOs), while “large launch vehicles” will refer to all other launch vehicles.

A number of modern heavy launch vehicles, such as SpaceX's Falcon 9 and the European Space Agency's Ariane V, developed from smaller incarnations frequently in the form of small launch vehicles. In the past, launches of small launch vehicles were relatively more expensive per unit payload than their larger counterparts. This was mainly a result of the low launch frequency of small launch vehicles (Niederstrasser, 2018).

The current launch mission classes for small satellites are: dedicated launches, rideshare (cluster launches) and piggy-back launches. Previously, the latter two classes were more commonly used for small satellite launches (Massimiani, 2015).

Dedicated launches allow for satellites to be launched into customer-specified orbits. The primary constraint of this class of launch missions is that they are generally more expensive than rideshare and piggy-back missions on a specific cost basis. This is as fixed costs are for the account of one customer, whereas in cluster launches, fixed costs are shared among customers. However, the costs of this mission class can be reduced if the launch frequency increases (Massimiani, 2015).

In a cluster launch, multiple satellites are placed into an orbit using a single carrier rocket. Piggy-back launches attach a small satellite as a secondary payload to a main payload. These methods are cheaper than dedicated launches, but limit the customer's choice of the payload's final orbit.

Since the 1990s, small satellite capabilities have increased, while mass has generally reduced for a given level of complexity. This has caused an increase in demand for dedicated launch missions for small satellites. This in turn, has generated a desire for cheaper, dedicated small launch vehicles (Niederstrasser, 2021). As such, the commercial prospects of and government contracts for small satellite launch capabilities have increased over the past decade, resulting in the development and/or proposal of numerous small launch vehicles. Ultimately, to ensure such missions are financially feasible, the frequency of such launches has to increase, and the technology developed needs to be low-cost. Reusability of the vehicle is one solution. A common interest among stakeholders of such programmes is developing programmes capable of launching on-demand (Niederstrasser, 2018).

Figure 1-1 depicts the global satellite launch history from 2012 to 2019, illustrating an upward trend in the number of small satellites launched annually. This upward trend is consistent with the increased demand for small satellite launch missions, and private and government financial investment in the industry (Niederstrasser, 2018). As the cost of launch reduces with launch frequency, an upward trend in small satellite launch frequency increases the feasibility of these missions (Bryce Space and Technology, 2022).

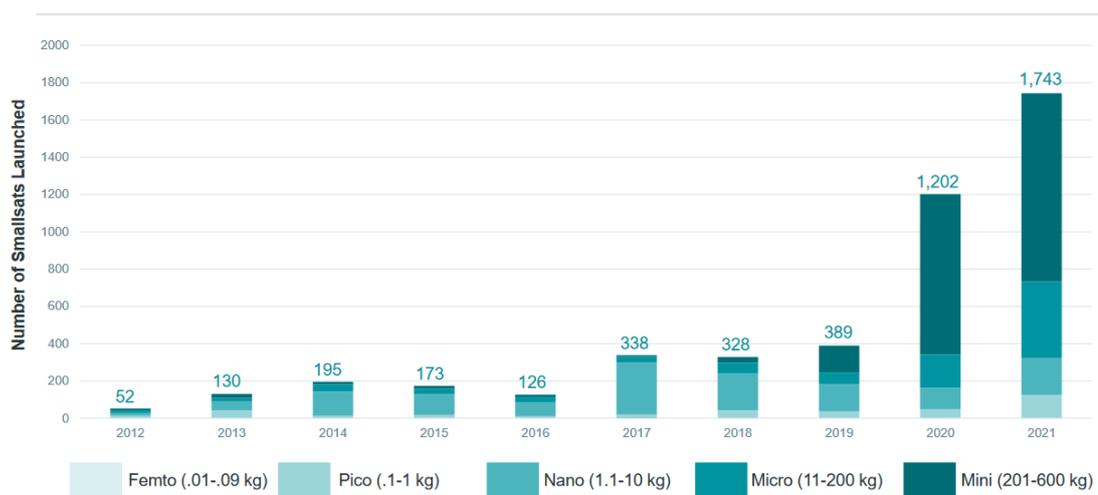


Figure 1-1: Bar graph depicting small satellite trends from 2012-2021 (Bryce Space and Technology, 2022)

The surging small satellite launch vehicle industry has seen diversity in international representation. Currently, the USA dominates, but China and European countries are also vying to be competitive. Europe is mainly represented by Spain, through companies such as Pangea Aerospace and PLD Space, and the UK, by organisations such as Skyrora and Orbex. Table 1-3 depicts the number of small satellite launch vehicle programmes associated with several countries, as well as continental Europe, that were either operational or under development as of 2021 (Niederstrasser, 2021).

Table 1-3: Country of origin of launch vehicle developers (Niederstrasser, 2021)

| Country | Count |
|-------------------------|-------|
| USA | 22 |
| China | 9 |
| India | 4 |
| United Kingdom | 4 |
| Argentina | 2 |
| Germany | 2 |
| Spain | 2 |
| Australia/Singapore | 1 |
| Brazil | 1 |
| Canada | 1 |
| France | 1 |
| Japan | 1 |
| Malaysia | 1 |
| Netherlands/New Zealand | 1 |
| Russia | 1 |
| Singapore | 1 |
| Taiwan | 1 |
| UK/Ukraine | 1 |
| USA/New Zealand | 1 |

1.2. Review of Orbits

Orbits are classified into several categories, typically based on their altitude and inclination. The most common categories are: low Earth orbit (LEO), geosynchronous orbit (GEO), polar and sun-synchronous orbit (SSO). Several of these orbits can be seen in Figure 1-2. Orbits lying within a given category tend to host satellites which have similar functions. However, as technology improves, the capabilities of satellites and thus the type of satellites existing in an orbit category will change.

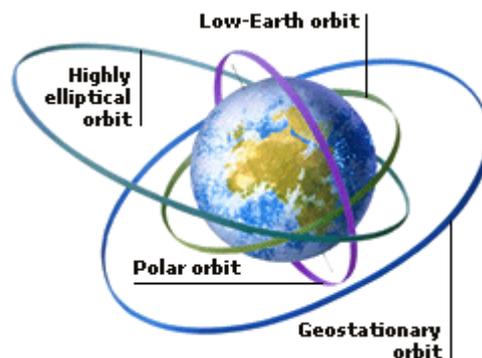


Figure 1-2: Examples of popular orbits around Earth (Kasonsuwanx, et al., 2017)

Currently, 55% of all operational satellites are in LEO, which is at an altitude range between 160 and 2000 kilometres. LEOs are best suited for remote sensing missions such as Earth observation and reconnaissance due to low orbital periods and altitudes (Roberts, 2020).

LEOs have currently become the new target for most communication satellites. These were originally injected into geosynchronous equatorial orbits (GEOs) in order to avoid a high relative velocity between terminals and receivers and to ensure a wide coverage range. However, due to new advances in communication technology and cheaper launch costs, there has been a growth in the interest of developing large LEO constellations which are capable of high-throughput broadband services with low latency. Multiple companies, such as Space Exploration Technologies Corp. (SpaceX), Amazon, OneWeb, and TeleSAT have already announced large LEO constellations each comprising thousands of satellites (Kodheli, et al., 2020). SpaceX, in particular, is aiming to deploy a constellation, referred to as Starlink, of 42000 small satellites with the objective of providing low-cost internet to remote locations (Mann, et al., 2022).

Satellites which are intended for LEOs, are typically injected directly (Coskun, 2014). The injection point is usually chosen as the perigee – the orbit's point of lowest altitude – provided it is high enough to avoid rapid atmospheric decay (Patton and Hopkins, 2006). Another common strategy is for a launch vehicle to enter what is referred to as a transfer orbit. In this instance, a launch vehicle coasts along this orbit and at an appropriate time performs an impulsive manoeuvre in order to enter the desired final orbit (Widnall and Peraire, 2008).

A popular type of LEO is the polar orbit. The polar orbit plane, relative to the equatorial plane can be seen in Figure 1-3. Polar orbits are orbits that have an inclination between sixty and 120 degrees (ESA, 2020a). Satellites in these orbits essentially travel over the poles of a planet.

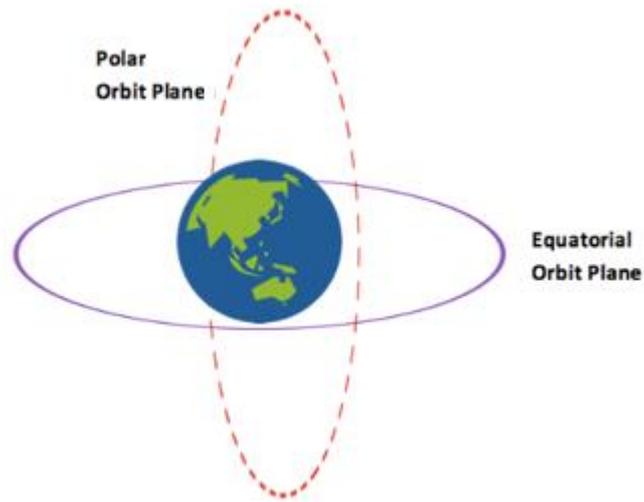


Figure 1-3: Polar orbit plane relative to Earth’s equatorial (orbit) plane (Damp, 2017)

Due to the oblateness of Earth, the orbital parameters defining a particular orbit change with respect to time. Orbits which possess parameters which ensure its orbital plane precession is equivalent to the angular velocity of Earth as it rotates around the sun, are referred to as sun-synchronous orbits (SSOs). The SSO plane, relative to the equatorial plane and the poles of Earth can be seen in Figure 1-4.

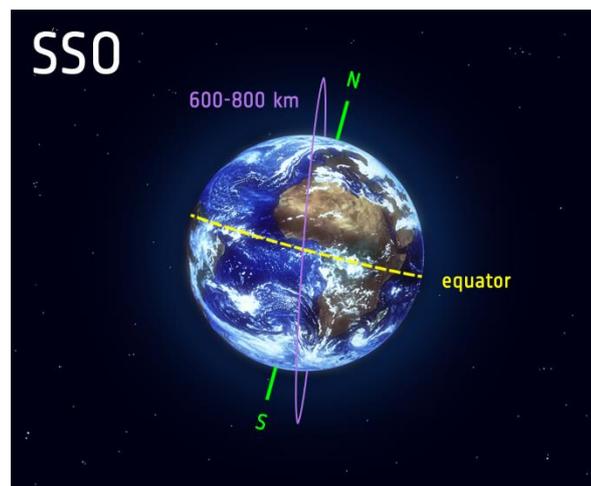


Figure 1-4: SSO plane relative to equatorial plane and Earth’s poles (ESA, 2020b)

These orbits are “very polar” and retrograde, that is, they have inclinations just above ninety degrees. They are also typically LEOs. For Earth, the required average rate of precession of the

node line ($\dot{\Omega}$) in order to achieve a SSO is 1.991×10^{-7} . This can be calculated using Equation 1-1 (Curtis, 2005):

$$\dot{\Omega} = - \left[\frac{3}{2} \frac{\sqrt{\mu} J_2 R^2}{(1-e^2)^2 a^{\frac{7}{2}}} \right] \cos i \quad (1-1)$$

where μ , R and J_2 are respectively the gravitational parameter, radius and second zonal harmonic of the planet being orbited, while e , a and i are the eccentricity, semi-major axis and inclination of the orbit.

Satellites in SSO, have a view of any given section of a planet under essentially the same lighting conditions on each pass. In other words, satellites in SSO will pass any given point of a planet at the same local mean solar time. This characteristic of SSO satellites can be seen in Figure 1-5. These satellites will also always have a constant perspective of the sun (Nazim, 2020), and are frequently used for activities such as Earth observation, reconnaissance, and weather data collection. Some examples of famous sun-synchronous satellites include the NOAA Polar-orbiting Operational Environmental Satellites (NOAA/POES) and the Defense Meteorological Satellite Program (DMSP) satellites – with satellites from both of these series being used for global weather coverage – and the Landsat and the French SPOT series, which are used for high-resolution Earth observation (Curtis, 2005).

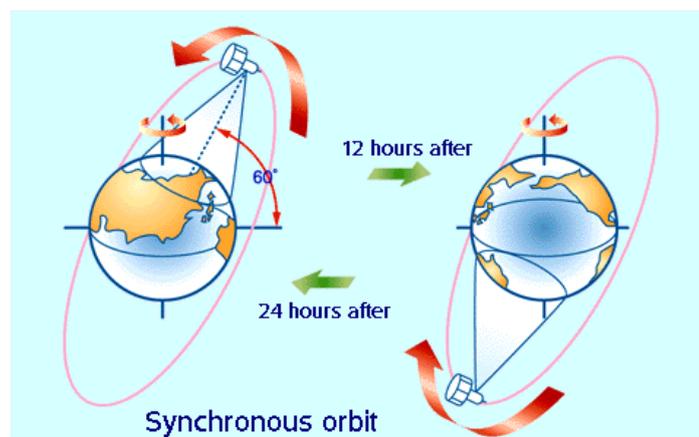


Figure 1-5: Diagram depicting a satellite in SSO passing over sections on Earth at the same local mean solar time (SFSU's Department of Earth and Climate Sciences, 2022)

1.3. International Space Laws and Practices

In order to assess possible orbits which can be launched into, it may be necessary to explore international space laws and practices to ensure that both launch trajectories and orbits which payloads are inserted into are legal and ethical.

The current laws and governance of international space activities were developed, from the 1960s, by the Committee on the Peaceful Uses of Outer Space, a body created by the United Nations. The Committee currently uses five international treaties and sets of principles on space-related activities as the framework for its laws (Swaminathan, 2005). These laws and policies essentially allow responsible access for most orbits, with the exception of highly cluttered orbits, such as the geosynchronous equatorial orbits (GEOs).

The first agreement, is the Outer Space Treaty, which forms the backbone of international space law. It requires that outer space exploration should be pursued for the benefit of mankind, and may be done freely by any state or entity. In addition, states must not place any weaponry in outer space, space exploration is to be conducted for peaceful purposes, astronauts in space should be aided if possible, harmful contamination of space and celestial bodies must be avoided, and owners of space objects are liable for any damage their objects cause (UNOOSA, 2020b).

The remaining four treaties are summarised by UNOOSA (2020a) as follows:

1. Rescue Agreement: rescue of astronauts, the return of astronauts and the return of objects launched into outer space
2. Liability Convention: international liability for damage caused by space objects
3. Registration Convention: convention on registration of objects launched into outer space
4. Moon Agreement: governing the activities of states on the moon and other celestial bodies

Because GEO satellites are fixed at a point in space above the Earth, orbital “real estate” is at a premium, especially in Europe, Japan, and the United States. Closely parked satellites using the same frequency can experience crosstalk and coupling. International regulations require a 2°

spacing for satellites that use the same frequency, and 90° separation for satellites using direct broadcast (Perez, 1998).

Finally, in order to prevent cluttering of space and damage to satellites from debris in the form of defunct satellites and spent stages, most space agencies provide guidelines for deorbiting. As such, some propellant is typically allocated for deorbiting purposes. The propellant quantity required is comparatively small, since the bulk of the required deceleration is effected by atmospheric forces (NASA, 2020).

1.4. South African Launch Capabilities

Launch vehicles have been (and still are) historically launched over the ocean or low population density regions by selecting suitable launch azimuths. This is done to ensure the safety of populations and to prevent damage of expensive infrastructure. The Denel Overberg Test Range (OTR) of South Africa provides launch facilities that can facilitate such launches.

OTR is a flight test facility, currently with a primary focus on weapons testing. OTR regularly provide ground support operations for numerous space missions, such as the 2003 launch of the MARS exploration Rover I. OTR is located near the southern tip of Africa. This location is remote, and allows for southerly and slightly south-easterly launch azimuths, thereby enabling polar and sun-synchronous missions

Figure 1-6 depicts the paths launch vehicles can typically travel along on a direct ascent into orbit. It can be seen that direct ascent is restricted to orbits of inclinations between 34.6° (eastward launch) and 122° (lower bound). Trajectories along any paths outside of this inclination range, or near the boundaries of this range, would travel too close areas with human populations, such as the nearby town of Arniston.

Dog-leg manoeuvres, in which vehicles changes plane over a short period of time, could potentially allow for such launches to occur. Such manoeuvres, however, tend to be expensive and reduce the capabilities of missions (Newcomb and Hampshire, 1970). It can be seen in Figure 1-6 that instantaneous impact points of launch vehicles travelling, by direct ascent, to inclinations

of 40° , 64.435° and SSO (approximately 97° degrees for LEO) are likely to be in the ocean. Thus, these trajectories may not require dog-leg manoeuvres.

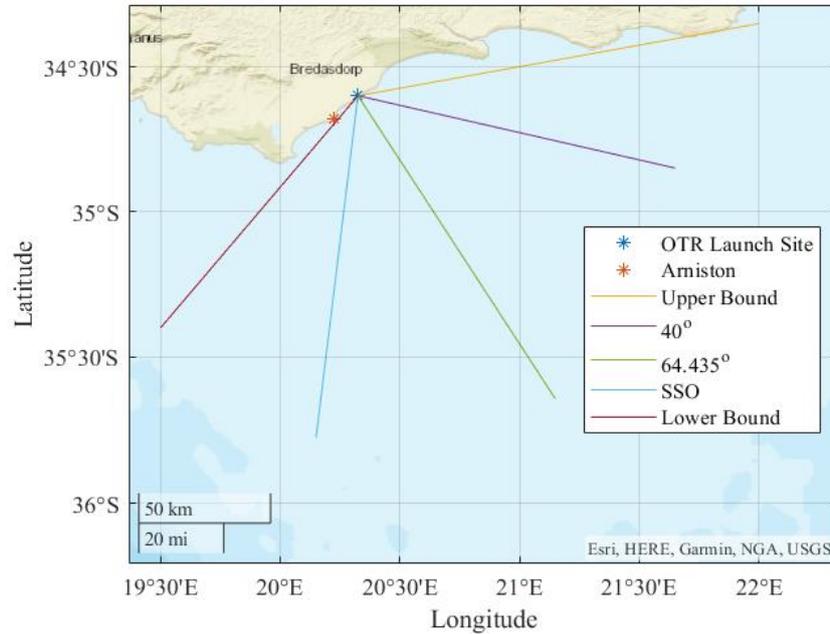


Figure 1-6: Map depicting paths into various orbit inclinations from OTR

1.5. Problem Statement and Research Objectives

The University of KwaZulu-Natal's Aerospace Systems Research Group (ASReG) initiated the South African First Integrated Rocket Engine (SAFFIRE) programme to develop a small, modular liquid propellant rocket engine to propel an indigenously-developed small satellite launch vehicle to LEO. The establishment of an orbital space launch capability in South Africa would present several key economic and strategic benefits to the country, including the generation of foreign revenue, the stimulation of local industry, academia and STEM educational activities, as well as the provision of a platform for space-based defence activities. As such, the SAFFIRE programme has been recognised as being a technology development initiative of national importance (ASReG, 2019).

As one of the primary intentions of the SAFFIRE programme is to enable the launch of a vehicle into LEO, it is imperative to gain an understanding of the mechanics of a vehicle ascending into

a specified orbit. Understanding these mechanics will allow ASReG to determine the capabilities of a small satellite vehicle launched into space. Since such a vehicle would likely be launched from OTR, an understanding must be developed of the characteristics of the ascent to allowable LEOs, and the various orbits that can be attained when launching from the range.

As can be seen in Figure 1-7, many paths can be taken by a launch vehicle into orbit. The need to select a path which reduces mission cost warrants the determination of an optimal and feasible trajectory into the destination orbit. Hence, in order to achieve the main objective of assessing the LEO launch performance capabilities of South Africa OTR, the research objectives are as follows:

1. Review trajectory optimisation techniques, particularly those used for ascent-to-orbit trajectories
2. Implement, develop and validate a strategy to determine optimal trajectories
3. Alter the aforementioned strategy to solve ascent-to-orbit problems
4. Simulate the launches of three existing carrier rockets and ASReG's proposed carrier, CLV, to predefined orbits
5. Analyse the flight performance, dynamic pressure and feasibility of each trajectory scenario for each launch vehicle.

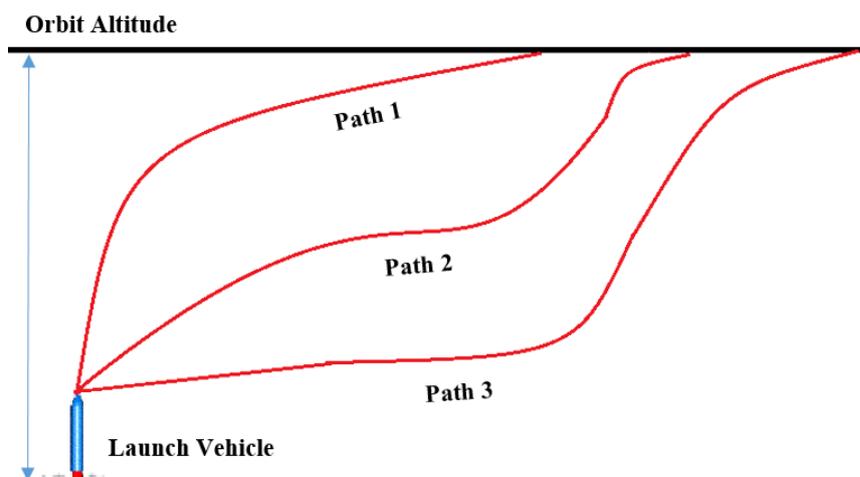


Figure 1-7: Diagram showing hypothetical paths from Earth to orbit

1.6. Dissertation Outline

Chapter 2 provides a review of the field of optimal control. The various methods used to solve optimal control problems are discussed, along with nonlinear programming techniques. Lastly ascent-to-orbit trajectory optimisation literature is reviewed.

Chapter 3 details the direct Hermite Simpson algorithm used by the optimal control solver developed in this work. It details the mathematical description of this optimisation algorithm and the derivation of the first derivative formulae.

Chapter 4 provides the examples used to validate the developed optimal control solver. Three examples are used. The first is a linear problem with terminal constraints. The second is a coplanar orbital transfer problem. The final validation problem is the optimisation of a two stage ascent-to-orbit trajectory.

Chapter 5 first discusses the system dynamics, force models, and reference coordinates used in this work for the ascent-to-orbit problem. The chapter then outlines the algorithm used in order to optimise the ascent-to-orbit trajectories to be analysed.

Chapter 6 provides, for each analysed launch vehicle, the flight performance, dynamic pressure, axial acceleration and properties of the thrust direction (control history).

Chapter 7 provides a summary of the dissertation, and provides potential future considerations with regards to the trajectory analysis and design of small satellite launch missions from OTR.

2. LITERATURE REVIEW

2.1. Optimal Control

The determination of reasonable launch vehicle trajectories which maximise payload capacity whilst conforming to vehicle structural load limits and various other constraints (such as path constraints), necessitates the study of optimal control and trajectory optimisation. Optimal control is the field of study concerned with the determination of the control history or trajectory which optimises a cost or objective function of some dynamic system. The objective function can be one of or a combination of the following: a function of the boundary conditions of the dynamic system or an integral function of the state and control variables (Lawden, 1975).

In the case of an ascent-to-orbit trajectory, the objective function typically revolves around maximising the final mass of the launch vehicle. The control variables are the thrust direction and throttle history. The state variables are the position, velocity and optionally the mass of the launch vehicle (Pontani and Cecchetti, 2013).

Optimal control problems, when solved analytically, require the definition of the Hamiltonian function, from which a system of differential algebraic equations (DAE) is obtained. The DAE system is obtained by optimisation of the Hamiltonian with respect to the control vector, in the region of control space where all constraints are satisfied. The solution to this DAE system provides the optimal control solution. This definition of the optimal control solution is referred to as Pontryagin's principle. The solution of the optimal control problem is typically indicated with a star superscript (Bryson and Ho, 1975).

The Hamiltonian and DAE system of an optimal control problem with a fully defined initial state vector in which the only constraint imposed on the system is its dynamic behaviour, are as follows (Lawden, 1975):

$$\mathcal{H} = L + \lambda^T f(\mathbf{x}, \mathbf{u}) \quad (2-1a)$$

$$\left(\frac{\partial \mathcal{H}}{\partial \mathbf{u}} \right)^T = \mathbf{0} \quad (2-1b)$$

$$\dot{\mathbf{x}} = \left(\frac{\partial \mathcal{H}}{\partial \boldsymbol{\lambda}} \right)^T = f(\mathbf{x}, \mathbf{u}) \quad (2-1c)$$

$$\dot{\boldsymbol{\lambda}} = - \left(\frac{\partial \mathcal{H}}{\partial \mathbf{x}} \right)^T \quad (2-1d)$$

where \mathcal{H} is the Hamiltonian, L is the integrand of the integral objective function, $\boldsymbol{\lambda}$ is the adjoint vector, f is a function outputting a vector describing the dynamics of the state variables, \mathbf{x} is the state vector, and \mathbf{u} is the control vector. It should be noted that a solution to the above DAE system only meets the first-order necessary conditions for optimality of the Hamiltonian.

It is possible to have additional constraints, such as boundary, path and integral constraints. These additional constraints require some sort of special treatment, such as of the Hamiltonian with Lagrange multiplier functions. In-depth description of the formulation of the Hamiltonian, the associated DAE system, and second-order necessary conditions, are given by Lawden (1975) and Bryson and Ho (1975).

The addition of terminal boundary constraints on the state variables imposes terminal boundary conditions on the adjoint variables. The other additional constraints are typically treated by the augmentation. In the event that this DAE system has no closed form solution (as is common with state variables with nonlinear dynamics), numerical methods are required to obtain a solution.

Numerical optimal control methods are classified as either indirect or direct. Indirect methods involve numerically solving the aforementioned DAE system. Direct methods do not involve the Hamiltonian and instead discretise the dynamics describing the state variable, the cost function, and path constraints, using appropriate numerical integration techniques, converting the problem into a nonlinear programming (NLP) problem. The problem is then solved by an appropriate NLP solver. Indirect methods employ an optimise, then discretise approach, whereas direct methods employ a discretise, then optimise approach (Rao, 2010).

The indirect method is typically complex to perform, for the following reasons expressed by Betts (1998):

- 1) The Hamiltonian must be expressed, and from which analytical expressions for the necessary conditions must be obtained, which is complex for systems with nonlinear dynamics
- 2) The initial guess has to be typically relatively close to the solution, as the region of convergence tends to be small. Initialising indirect method solvers is made difficult by the need to guess the abstract adjoint variables.
- 3) In the event that problems have path constraints, it is necessary to guess a sequence of unconstrained and constrained arcs of the trajectory.

Direct methods do not require both the construction of the Hamiltonian and determination of the necessary conditions. The region of convergence is typically relatively large, and there is no need to define or guess the adjoint variables. There is also no need to guess a sequence of unconstrained and constrained arcs of the trajectory. Thus, direct methods do not suffer from the shortcomings of indirect methods and are typically more robust than indirect methods (Betts, 1998). However, there is generally a loss in accuracy in the optimality of the solution obtained using direct methods due to an increase in discretisation errors (Passenberg, 2011). Currently, direct methods are the most popular approach for solving high complexity optimal control problems. This is largely due to the advancements made in NLP (Grüne, 2020).

A solution to the difficulty in initialising the indirect methods solver is to use the solution to the optimal control problem found with direct methods to formulate the initial guess (von Stryk and Bulirsch, 1992).

The popular numerical methods used to solve optimal control problems (with direct or indirect methods) are shooting (single and multiple) and transcription (or collocation) techniques. Both these methods are used to solve two-point boundary value problems (TPBVPs), and use Newton iterative techniques to be solved (Rao, 2010).

Single shooting involves iteratively numerically integrating from the initial boundary to the terminal boundary, until the required boundary conditions are satisfied. The main advantage of

single shooting, is it describes the problem in terms of a relatively small number of variables. However, single shooting lacks robustness, and suffers from several key issues (Betts, 1998).

The first issue of the shooting method, is that it suffers from a high sensitivity to the initial guess and is locally convergent, thus good initial guesses are required (Burden and Faires, 2001). The second issue with this method, is the large computational cost and complexity of determining the derivative information required by the Newton iterative solvers. Typically, finite-difference methods are used to estimate derivative information, this however reduces the accuracy of the derivative information (Betts, 1998).

The robustness of single shooting can be improved by using multiple shooting, a method in which the trajectory is split into several segments, and shooting is performed simultaneously on each segment for every iteration. Lastly, additional boundary conditions constrain the boundaries of adjacent segments to be equivalent. The solver then iterates until boundary conditions are met (Holsapple, et al., 2004).

Collocation discretises the independent variable (typically time) into many intervals, essentially creating a grid or mesh, with each point of the grid representing some finite time. Each grid point is then allocated discrete values of the dependent variables, in the case of optimal control these variables are a state and control vector. The system of differential equations are replaced with defect constraints for each independent variable interval, which relate the dependent variables of adjacent grid points by some suitable quadrature (Topputo and Zhang, 2014).

Collocation provides more robustness than shooting, and allows feasible means of providing analytical gradients to numerical solvers. Collocation's failings lie in the requirement to define and solve for a large number of optimisation variables (Betts, 1998).

The numerical integration techniques typically used for optimal control are Runge-Kutta techniques of certain orders. Popular examples of Runge-Kutta schemes used in optimal control include: Euler, Trapezoidal, Hermite-Simpson, and the classical fourth order Runge-Kutta method (Betts, 2010). Trapezoidal and Hermite-Simpson methods are implicit second and third

order techniques respectively. Euler and the fourth order Runge-Kutta methods are explicit first and fourth order techniques respectively. Another popular Runge-Kutta technique is Gaussian quadrature which has the advantage of not requiring the functional data of grid points in order to perform integration (McLaren, 2018).

2.2. Nonlinear Programming

Due to their consistent use in solving optimal control problems, NLP methods are consistently relevant in the field of optimal control. Betts (1998) stated that the progress in optimal control and NLP methods share a strong resemblance. The class of algorithms which solve optimisation problems with at least some nonlinear constraints is referred to as nonlinear programming (Nocedal and Wright, 1999).

There are three approaches to constrained optimisation problems. The first approach of interest comprises the augmented Lagrangian methods, which requires the definition of the Lagrangian function. In order to create the Lagrangian, additional Lagrange multiplier variables must be solved for. Every constraint of the problem has an associated Lagrange multiplier. An example of such a Lagrangian can be seen in Equation 2-2. This Lagrangian is specifically applicable to an optimisation problem with only equality constraints (Nocedal and Wright, 1999).

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}) - \boldsymbol{\lambda}^T c(\mathbf{x}) \quad (2-2)$$

where \mathcal{L} is the Lagrangian function, \mathbf{x} is the vector containing the decision variables, $\boldsymbol{\lambda}$ is the vector composed of the Lagrange multipliers of the equality constraints, f is the scalar objective function to be optimised and c is the function that outputs the equality constraint vector. It should be noted the requirement for a feasible solution is that $c(\mathbf{x})$ is equivalent to the appropriately sized zero vector. A system of algebraic equations and necessary conditions are obtained by optimising the Lagrangian with respect to the decision variables and the Lagrange multipliers.

A popular method of optimisation which uses the augmented Lagrangian is sequential quadratic programming (SQP). SQP models quadratic “sub-problems” for each iteration. The solution of

these sub-problems provides the required search direction for the next iteration (Nocedal and Wright, 1999).

The other two classes of methods are the penalty methods and barrier (or interior point (IP)) methods. Penalty methods augment the scalar objective function by combining it with some function of the constraint functions and penalty parameters. The new function, called the penalty function, is then optimised, as if it were unconstrained. In the event that the constraints have not been satisfied, the penalty parameters of the constraints are appropriately adjusted. This process is repeated until constraints have been satisfied within a reasonable tolerance (Nocedal and Wright, 1999).

IP methods essentially follow a similar procedure to penalty methods. The fundamental difference being that these methods augment the objective function such that, on a particular iteration, satisfied constraints have insignificant contribution to the augmented objective function (Nocedal and Wright, 1999).

2.2.1. Scaling

One of the most fundamental concerns in the application of NLP methods is the scaling of the problem variables. A problem is said to be poorly scaled if the objective function has relatively high sensitivity to certain decision variables and relatively low sensitivity to other decision variables. Poorly scaled problems tend to lead to poor performance (either slow convergence or failure to converge) (Nocedal and Wright, 1999).

Hence it is necessary to appropriately scale variables. A common practice of scaling variables is to ensure that they lie in similar domains. Nocedal and Wright (1999) provide a topological description of the scaling problem. They state that a feature of poor scaling is that the optimal solution for the decision vector (\mathbf{x}^*) lies in a narrow valley, causing the contours of the objective function in the region of \mathbf{x}^* to approach highly eccentric ellipses. This can lead to poor performance of optimisation algorithms.

2.2.2. Gradient Information

NLP methods typically make use of Newton-based iterative solvers. As such, the first-order derivative (the Gradient) and the second-order derivative (the Hessian Matrix) of the Lagrangian, with respect to the decision vector and the Lagrange multiplier vector, are required in order for solutions to be searched for. NLP solvers typically store derivative information in matrices, which are referred to collectively as derivative matrices. Derivative information can be analytically determined and provided to NLP solvers, however it is possible to use numerical estimates of the derivative information (Betts, 1998). Methods which use analytical derivatives are called Newton methods, and methods which use numerical estimates of derivatives are called Quasi-Newton methods (Nocedal and Wright, 1999).

Finite difference estimates, either forward or central, are commonly used to estimate derivative information. Central difference estimates are more accurate than forward difference estimates. Both methods avoid the task of obtaining analytical derivative information, which can be particularly complex for high dimension nonlinear problems. The main issue with these numerical estimates is their relatively high computational cost per iteration performed (Betts, 1998).

Newton methods are typically described to have quadratic convergence, whereas Quasi-Newton methods have super-linear convergence. Quadratic convergence is relatively more rapid than super-linear convergence. Hence, it is of considerable advantage to provide analytical derivative information to NLP solvers. The disadvantages of Newton methods, other than the difficulty in obtaining this information, lies in the sometimes-expensive and error-prone process of computing the Hessian matrix (Nocedal and Wright, 1999). In the event that non-smooth problems are posed to smooth NLP solvers, analytical derivatives cannot be provided and numerical estimates must be used.

2.2.3. Sparsity

Certain NLP problems involve derivative matrices that are dominated by zeros and referred to as sparse matrices. By exploiting the sparse structure of these matrices, one avoids the unnecessary

computation and storage of elements of matrices that are known to always be zero. This can greatly improve computational performance by reducing the computational and storage requirements of NLP solvers. Sparsity is particularly present in the relatively large derivative matrices of direct collocation optimal control methods. Hence it is typically of interest to attempt to take advantage of this sparsity whilst using direct collocation methods (Rao, 2010).

2.2.4 Sequential Quadratic Programming and Interior Point Methods

Two popular algorithms NLP solvers employ are the SQP and IP methods. Betts (2010), performed a comparison of 99 optimal control problems solved by both these methods. The results of this comparison suggested that IP methods are significantly slower and more computationally expensive than SQP methods.

Three reasons for this include:

- 1) The barrier transformation can be expected to make iterations of IP solvers slower and more nonlinear.
- 2) The number of variables and constraints of the IP formulation of a particular problem is larger than that of the SQP formulation.
- 3) Scaling has the potential to interact with the nonlinearity of the barrier transformation.

It is advantageous to make use of solutions of “nearby” problems to formulate initial guesses, as guesses in the region of the optimal solution lead to faster convergence. SQP methods are capable of exploiting such an approach. IP methods require that iterations (including the initial guess) must be strictly feasible. Typically, computer software will adjust initial guesses so that this condition is met. This adjustment prevents interior point methods from always utilising a good initial guess (Forsgen, 2006).

Direct collocation optimal control solvers typically initialise by solving a problem on a coarse grid. The grid is then updated by adding grid points if the discretisation error estimate is not sufficiently small. This essentially means that these solvers employ a Sequential Nonlinear Programming algorithm to be solved. The solution of the previous grid can be treated as a solution

to a “nearby problem”. Hence it is advantageous to make use of the previous grid’s solution, but this cannot be done with interior point methods (Betts, 2010).

2.3. Ascent-to-Orbit Specific Trajectory Optimisation

It is often of interest to find optimal control solutions to simplified versions of complex problems. This allows for the use of quick-to-compute-and-use control laws, which are often sub-optimal solutions to the complex version of the problem. For example, early flight performance optimisation studies made use of quasi-steady approximations. Once computer technology advanced, more complex models could be used. However these simplifications are still used, mainly due to the increase in complexity of air and space vehicles and the need for sub-optimal automated guidance systems (Calise, 2013).

This study aims to determine feasible ascent-to-orbit trajectories using optimal control methods in order to analyse OTR’s launch performance space. Due to the complexities of these trajectories it is desirable to implement typical and practical methods and constraints for this specific class of trajectory optimisation problems.

The optimisation of the ascent-to-orbit trajectory is extremely complex due to the high dimensionality and nonlinearity of the problem. This is particularly evident during the atmospheric ascent phases due to the additional complexity and constraints imposed on a vehicle caused by aerodynamic loads (Suresh and Sivan, 2015).

The characteristics of the atmospheric phases and launch vehicles are often represented using non-smooth modelling, usually in the form of tabular data. As typical numerical optimisation and numerical integration algorithms assumes smoothness of its input functions, to at least a second order, the non-smooth representation of aerodynamic forces and vehicle characteristics is inconsistent with the numerical methods employed. This can lead to poor computational performance, a reduction in robustness of the solver, and errors in the gradient estimates used by the Newton-method based solvers (Betts, 1998).

For example, Jansch, et al. (2013), solved the solution to a trajectory optimisation of a two stage ascent-to-orbit problem, and found that the convergence rate of the solver was linear, instead of the typical superlinear rates of the methods they employed. The authors attributed this to non-differentiable tabular data used to model the engine.

One solution to the aforementioned issue is to represent tabular data using smooth cubic B-spline interpolation. Another solution is to split the atmospheric ascent into phases, and represent tabular data for each phase using smooth cubic B-spline interpolation. The splines must also be designed such that unrealistic values are not represented by the splines (such as negative air density). This process is achievable but must be done for each individual vehicle and varying atmospheric conditions. The multiple phase approach increases the complexity of the problem to be solved by an optimal control solver (Betts, 2010).

During the atmospheric ascent phase, launch vehicles experience high aerodynamic loads that could lead to structural failure of the vehicle. This subjugates the optimal control problem to additional nonlinear path constraints during the atmospheric phase (Suresh and Sivan, 2015).

These aerodynamic loads are dependent on the properties of the local atmosphere and wind. A key parameter used to measure these loads is the dynamic pressure, or q , which can be seen represented by Equation 2-3 (The Advisory Group for Aerospace and Research and NATO, 1970).

$$q = \frac{1}{2}\rho v_{rel}^2 \quad (2-3)$$

where ρ is air density and v_{rel} is the magnitude of the relative velocity to the atmosphere. Dynamic pressure typically follows a parabolic trajectory with respect to altitude or time, reaching a maximum value (referred to as max q) at an altitude that is typically between 7 and 15km (Suresh and Sivan, 2015).

2.3.1. Gravity Turn

Suresh and Sivan (2015) state that the primary criterion of the atmospheric ascent is to reduce and restrict the aerodynamic loads experienced by the launch vehicle, referring to such a strategy as a load relief trajectory. They suggest the use of a gravity turn as a solution. A vehicle which follows a gravity turn follows a zero angle of attack. In other words, the longitudinal axis of the launch vehicle is aligned with its velocity relative to the air. In the presence of wind, a gravity turn is referred to as wind-biasing as the launch vehicle must tilt, or “bias” towards the wind. Alternatively, as in the simulations performed by Coşkun (2014), one could restrict the angle of attack to be relatively small. This strategy does not appear to be followed in actual practice, and requires additional path constraints to ensure the experienced aerodynamic bending loads are reasonable.

The gravity turn optimises two characteristics of the trajectory of the launch vehicle. It both minimises the aerodynamic bending loads experienced by the launch vehicle (The Advisory Group for Aerospace and Research and NATO, 1970) and maximises the energy gained per unit weight propellant (Stengel, 2016). Not following a load relief trajectory may restrict launch availability to particular wind profiles (Suresh and Sivan, 2015). A schematic of the typical trajectory phases a launch vehicle follows when a gravity turn is implemented can be seen in Figure 2-1 (Suresh and Sivan, 2015).

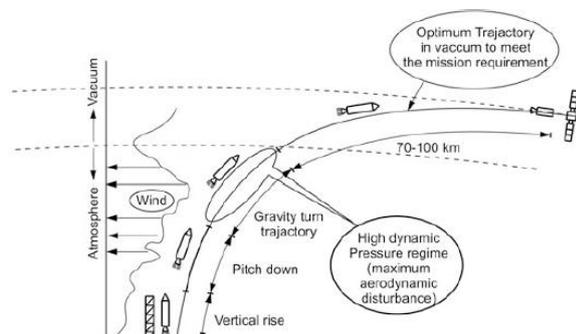


Figure 2-1: Typical gravity turn trajectory phases

Most launch vehicles must perform a vertical lift-off for a duration sufficient enough to safely clear launch pad structures. Performing a gravity turn immediately after a vertical lift-off will be

meaningless and wasteful as the initial relative velocity of the vehicle will be primarily vertical, implying that the vehicle will continue in an essentially vertical path if the gravity turn is initiated directly after a vertical lift-off. Hence it is necessary for a launch vehicle to pitch over towards the required orbit as soon as the vertical rise is complete (Suresh and Sivan, 2015). An example of a pitchover control law is a linear pitch control law, in which the pitch varies linearly with time (Federici, et al., 2019).

Examples of launch vehicles which have followed gravity turn and wind-biasing schemes include the Vega (Federici, et al., 2019), the Saturn V (Haeussermann, 1970) and the Falcon 9 (SpaceX, 2009). Another load relief strategy, implemented by Coskun (2014), is to restrict the absolute value of the angle of attack of launch vehicles to be less than 5 degrees during the atmospheric ascent phase.

One final load relief strategy is to design the thrust history. In the cases of liquid rocket engines, this would require designing, or optimising the throttle profile. Throttling is essentially linear control, and as is typical with linear controls, the throttle level is a control parameter that has the potential to cause singular arcs of the optimised ascent to orbit trajectory (Lawden, 1975). During singular arcs, the throttle level parameter will take on intermediate values between the maximum and minimum throttle. These arcs, called intermediate thrust (IT) arcs, adds further complexity to the optimal control problem, and create extreme computational difficulties when they are attempted to be solved numerically (Betts, 2010).

It has been proven by Kopp and Moyer (1965) and Robbins (1965) that these arcs are not optimal under vacuum conditions, implying that vacuum ascent will only include arcs of the trajectory with either full thrust (FT), or null thrust (NT). Such a control scheme is referred to as bang-bang control (Bryson and Ho, 1975). One treatment of bang-bang control in numerical analysis is to predefine the bang-bang scheme. For example, in the numerical analysis of a two-stage-to-orbit trajectory optimisation problem, Pesch (2014) predefined the bang-bang scheme as FT for the first stage, and FT,NT,FT for the second stage.

2.3.2. Bi-Linear and Linear Tangent Laws

The vacuum phase of the ascent-to-orbit trajectory is itself a highly complicated and nonlinear phase to be optimised. Optimal control laws for simplified versions of the vacuum phase allow for the circumvention of these complexities, and have been shown to be valuable in practice. These control laws are the bi-linear tangent law and the linear tangent law. The bi-linear tangent law is the optimal thrust-direction programming of a trajectory which must have its end conditions extremalised or final time minimised under the following assumptions: vacuum conditions; planar motion; and constant gravitational and thrust acceleration. The linear tangent law is the specific case of the bi-linear tangent law with an unrestricted terminal horizontal coordinate (Bryson and Ho, 1975). The bi-linear and linear tangent laws, as stated by Bryson and Ho (1975), are as follows:

$$\text{Bi-Linear tangent law:} \quad \tan\beta = \frac{c_4 - c_2 t}{c_3 - c_1 t} \quad (2-4)$$

$$\text{Linear tangent law:} \quad \tan\beta = c_1 - c_2 t \quad (2-5)$$

where t is time, β is the thrust angle measured from the horizontal, and c_k is used to represent constants. Note that the constants in the bi-linear tangent law and the linear tangent law are not necessarily the same.

Examples of studies which have applied these laws in their guidance algorithms for the vacuum ascent phase to orbit, can be found in the trajectory optimisation studies conducted by Marrdonny and Mobed (2009) and Federici, et al. (2019). Closed loop augmentations of the Linear Tangent Law were successfully implemented for the vacuum stages of the Space Transportation System (the Space Shuttle) (Betts, 1998) and the Saturn V launcher (Haeussermann, 1970) to achieve orbital injection.

On the practicality of the linear tangent law, Haeussermann (1970) states that the collective experience gained from numerous orbital injection missions has shown that during vacuum flight, the optimal thrust angle closely resembles a linear function of time. In addition to this, the size of

the angle between the longitudinal axis of the vehicle and the thrust direction (thrust angle) is always relatively small. These results are highly consistent with the linear tangent law. Haeussermann (1970) suggested linearly approximating the linear tangent law by representing the thrust angle as a linear function of time. Such an approximation is reasonable due to the relatively small magnitude of the thrust angle throughout the vacuum phase.

2.4. Optimal Control Software

There is currently a variety of optimal control software programs in existence, the majority of which use direct methods. These programmes, such as GPOPS II and ICLOCS2, typically interface with externally sourced NLP software, such as IPOPT or SNOPT. Some well-known indirect method solvers include BNDSCO and PROMIS, which both use multiple shooting methods (Rao, 2010).

Popular optimal control software generally has some means of estimating derivative information using sparse finite-differences. Depending on the solver to determine estimates of derivative information makes for easy-to-use software, however, as discussed before, it reduces the robustness of the solver.

One of the most powerful optimal control software programs, SOCS, is capable of employing both direct and indirect methods. SOCS makes use of its own built-in NLP solvers when using direct methods. In addition to using both classes of numerical optimal control methods, it also has substantial versatility in its quadrature method (Betts, 2010).

A software program worth noting is ASTOS. ASTOS is a space mission analysis tool that was originally created for trajectory optimisation. ASTOS incorporates various optimal control software (such as PROMIS) and other tools in order to achieve this. It currently possesses features that facilitate various analyses associated with space travel, such as design optimisation and mission safety analyses (Astos Solutions, 2020).

The optimal control software programs that have been discussed above do not make use of heuristic methods, as there has been a focus on popular techniques. An example of a heuristic

method is the particle swarm optimisation method which was implemented in an ascent trajectory optimisation problem by Pontani (2013). Table 2-1 provides a summary of all investigated optimal control software programmes, detailing their specific characteristics.

Table 2-1: Optimal control software overview

| Software | Open-Source | Direct or Indirect | TPBVP Method | NLP |
|-----------------|--------------------|---------------------------|--------------------------|--------------------------------------|
| ASTOS | No | Both | Collocation and Shooting | Various Externally Licensed Software |
| PROPT | No | Direct | Collocation | KNITRO,SNOPT,CONOPT,CPELX |
| DIRCOL | No | Direct | Collocation | SNOPT, NPSOL |
| GPOPS II | No | Direct | Collocation | SNOPT,IPOPT |
| ICLOCS2 | Yes | Direct | Collocation and Shooting | IPOPT, WORHP, fmincon (IP) |
| SOCS | No | Both | Collocation and Shooting | Built-in |
| MUSCO D II | No | Direct | Multiple Shooting | QPOPT |
| RIOTS | No | Direct | Collocation | CFSQP, NPSOL |
| BNDSCO | No | Indirect | Multiple Shooting | Not Applicable |
| PROMIS | No | Indirect | Multiple Shooting | Not Applicable |

2.5. NLP Software

As stated earlier, direct method optimal control solvers typically make use of externally-sourced NLP solvers. As the direct method is the most popular method, it is necessary to review available NLP software.

Much like optimal control software, a variety of powerful NLP tools have been developed. NLP programs used by direct method optimal control solvers are typically solvers dedicated to sparse NLP problems. This is done in order to take advantage of the sparse nature of the derivative matrices found in direct methods (Rao, 2010).

The three NLP programs that will be specifically discussed here are the SNOPT, IPOPT, and fmincon (MATLAB) programs. The NLP software used by specific optimal control solvers can be found in Table 2-1.

SNOPT is commercial software that makes use of sparse SQP methods. SNOPT is capable of using finite difference methods to estimate first and second derivative information. The associated user guide states that SNOPT, as with typical NLP solvers, ideally requires nonlinear functions to be smooth. However it is acceptable for functions to be non-smooth, as long as the functions are relatively smooth in the vicinity of the local optimal solution (Gill, 2008). Supplying non-smooth functions causes computational difficulties for the solver, and is inconsistent with NLP solvers meant for smooth problems. Nevertheless, NLP solvers such as SNOPT, have shown capability in solving non-smooth problems.

For example, in an ascent-to-orbit trajectory optimisation conducted with SNOPT and GPOPS II by Coskun (2014), non-smooth representations of atmospheric tabular data were used. The solver converged slowly, with one of the main factors contributing to this likely being the use of non-smooth functions. Nonetheless, feasible solutions were still obtained.

IPOPT is open-source software that uses sparse IP methods. It requires the provision of at the very least first derivative information. In the event only first derivative information is provided, the second derivative information is approximated using sparse finite differences (Kawajir, et al., 2015). Notable optimal control software which use IPOPT include GPOPS II and ICLOCS2.

The fmincon program forms part of the optimisation suite of Matlab. It is capable of using IP methods and dense SQP methods, but it only allows for sparsity exploitation and provision of the analytical Hessian when using IP methods (Mathworks Inc., 2020). ICLOCS2 is capable of interfacing with the IP method of fmincon, but claims fmincon is slower and has a higher chance of failure than other NLP software (Nie, et al., 2018).

3. OPTIMAL CONTROL SOLVER ALGORITHM

This study aims to develop a solver that uses direct optimal control methods to determine feasible ascent-to-orbit trajectories from OTR. Since direct methods make use of NLPs, it is necessary to select an available NLP solver to use. NLP solvers which make use of SQP methods are preferable due to their ability to handle nonlinear constraints and to make use of warm starts.

3.1. Optimal Control Algorithm Overview

The current chapter details the optimal control solver developed as part of this study. As available solvers which use SQP methods were unavailable, it was decided to develop a solver. The solver was primarily designed with the intention of determining optimal multi-phase ascent-to-orbit trajectories. The optimal control solver makes use of a direct Hermite-Simpson Collocation approach and can be applied to a general optimal control problem. Analytical first derivative information was obtained for direct Hermite-Simpson collocation methods. This section will detail the analytical derivative formulae.

Due to the numerous ascent-to-orbit trajectories that were simulated here, a direct collocation approach was used as this allowed for faster simulations compared to indirect methods and direct shooting. Hermite-Simpson quadrature was chosen as the quadrature rule used to discretise the differential equations, as it has an acceptable degree of error and complexity for direct methods.

Optimisation of ascent-to-orbit trajectories typically requires having prescribed trajectory phases of unknown duration. In the work presented by Rosmann et al. (2020), time-optimal control using direct Hermite-Simpson collocation for various control splines was presented. Quadratic control splines resulted in undesirable oscillations appearing in the solution. In addition to this, quadratic splines require the definition an additional variable for each collocation constraint to represent the midpoint of the quadratic control spline. Linear control splines produced solutions with no such oscillations and do not require the definition of additional control variables, hence such linear control splines were used here.

The solver makes use of warm starts, which are achieved by first obtaining a solution for a coarse time grid from which the coarse solution undergoes an error estimate test. Should this fail thereafter, the solution will then be used to formulate the initial guess for the same problem with a finer time grid. This process is repeated until a solution with an acceptable error estimate is formed.

Interior point methods increase the non-linearity of NLP problems and cannot take advantage of warm starts due to the requirement that the initial guess must be strictly feasible. SQP algorithms face no such issues, hence the SQP approach was adopted as the NLP method used by the present solver. In particular, MATLAB's SQP solver was chosen as the NLP solver as it was readily available. The decision variables were manually scaled as the author saw fit.

The use by MATLAB's SQP of dense finite difference estimates for the determination of derivative information, as compared to sparse finite difference estimates, led to slowness and unreliability when solving ascent-to-orbit problems. Hence formulae were developed to determine the analytical first derivative information for direct Hermite-Simpson Collocation.

3.2 Transcription

This section details the transcription of the optimal control problem into an NLP problem using Hermite-Simpson Collocation. The formulation of the optimal control problem presented has terminal boundary constraints, however, additional constraints can be imposed on the problem if necessary. Equation 3-1 is the optimal control formulation of a dynamic system with terminal constraints (end conditions).

$$\min_{\mathbf{u}} \phi(\mathbf{x}_f, t_f) + \int_{t_0}^{t_f} L(\mathbf{x}(t), \mathbf{u}(t)) dt \quad (3-1a)$$

$$\text{s. t. } \psi(\mathbf{x}_f, t_f) = 0 \text{ (end conditions)} \quad (3-1b)$$

$$\dot{\mathbf{x}} = g(\mathbf{x}(t), \mathbf{u}(t)) \quad (3-1c)$$

where x_f is the terminal state vector, t_f is the unknown final time of the phase of the trajectory. $\mathbf{x} \in \mathbb{R}^{m \times 1}$, and $\mathbf{u} \in \mathbb{R}^{n \times 1}$ are the state and control vectors, respectively. The function, $g: \mathbb{R}^{(m+n) \times 1} \rightarrow \mathbb{R}^{m \times 1}$, outputs the first time derivative of the state vector. The function, $\phi: \mathbb{R}^{(m+1) \times 1} \rightarrow \mathbb{R}^{m \times 1}$, outputs the component of the cost function which is dependent on terminal values of the system. The function, $L: \mathbb{R}^{(m+n) \times 1} \rightarrow \mathbb{R}^{m \times 1}$, outputs the integrand of the integral component of the cost function. Finally, the function, $\psi: \mathbb{R}^{(m+n) \times 1} \rightarrow \mathbb{R}^{m \times 1}$, outputs the component of terminal boundary constraints, which is dependent on terminal values of the system.

In the event that the final time is unknown, the following augmentation to the problem may be performed in order to allow for numerical solutions to be obtained.

Let $t = t_f \tau; \tau \in [0,1]$

$$\therefore \frac{dt}{d\tau} = t_f \quad (3-2)$$

Let there be a function, $f: \mathbb{R}^{(m+n+1) \times 1} \rightarrow \mathbb{R}^{m \times 1}$, which outputs the first derivative of the state vector with respect to the new independent variable, τ .

$$\begin{aligned} f(\mathbf{x}(\tau), \mathbf{u}(\tau), t_f) &= \frac{d\mathbf{x}}{d\tau} = \frac{d\mathbf{x}}{dt} \frac{dt}{d\tau} \\ \therefore f(\mathbf{x}(\tau), \mathbf{u}(\tau), t_f) &= t_f g(\mathbf{x}(\tau), \mathbf{u}(\tau)) \end{aligned} \quad (3-3)$$

The transcription of the above optimal control formulation into an NLP problem, using collocation, is achieved by discretising the independent variable of the system in order to create what is typically referred to as a time grid or a mesh. Each grid point is then allocated a time, control vector, and state vector. Collocation, using some suitable quadrature, is then used to convert any differential equations or integrals of the original problem into an algebraic form. Thus converting the problem into an NLP problem. Before the transcribed form of the optimal control problem is presented, some variables and equations must be defined.

Let $\mathbf{c}_k \in \mathbb{R}^{m \times 1}$ represent the collocation constraint vector of the $(k + 1)$ st grid point. Let there be N grid points. At a particular grid-point k :

$$\mathbf{x}_k \in \mathbb{R}^{m \times 1}; \mathbf{u}_k \in \mathbb{R}^{n \times 1}$$

$$\mathbf{z}_k = \begin{bmatrix} \mathbf{x}_k \\ \mathbf{u}_k \end{bmatrix} \in \mathbb{R}^{(m+n) \times 1}$$

$$\text{let } \mathbf{f}_k = \mathbf{f}(\mathbf{z}_k, t_f)$$

$$\bar{\mathbf{z}}_k = I_{m+n} \mathbf{z}_k + I_{m+n} \mathbf{z}_{k+1} + \frac{h_k}{8} \begin{bmatrix} \mathbf{f}_k - \mathbf{f}_{k+1} \\ \mathbf{0}_{n \times 1} \end{bmatrix}$$

The transcribed form of the optimal control problem, with terminal constraints to the state variables, can then be presented as follows:

$$\min_{\mathbf{u}_k, \forall k} \phi(\mathbf{x}_f, t_f) + \sum_{k=1}^{N-1} \frac{h_k}{6} \left(L(\mathbf{x}_k, \mathbf{u}_k, t_f) + 4L(\bar{\mathbf{x}}_k, \bar{\mathbf{u}}_k, t_f) + L(\mathbf{x}_{k+1}, \mathbf{u}_{k+1}, t_f) \right) \quad (3-4a)$$

$$\text{s. t. } \psi(\mathbf{x}_f, t_f) = 0 \quad (\text{end conditions}) \quad (3-4b)$$

$$\mathbf{c}_k = I_m \mathbf{x}_k - I_m \mathbf{x}_{k+1} - \frac{h_k}{6} (\mathbf{f}_k + 4\bar{\mathbf{f}}_k + \mathbf{f}_{k+1}) = \mathbf{0}, \forall k \quad (3-4c)$$

Here, \mathbf{c}_k is referred to as a defect or collocation constraint. In the event that there are multiple trajectory phases, the following constraints must be imposed:

$$\mathbf{x}_f^i = \mathbf{x}_o^{i+1} \quad (3-5)$$

The superscript i indicates the phase, and \mathbf{x}_o^i indicates the initial state vector of phase i .

3.3. First Order Analytical Derivatives

This section presents the analytical first derivative information of the defect constraints and the integral component of the cost function of the direct Hermite-Simpson collocation method.

For a particular phase of the trajectory:

$$\text{let } \bar{\mathbf{f}}_k = \mathbf{f}(\bar{\mathbf{z}}_k, t_f) \text{ let } J_k = J_{f_k}(\mathbf{z}_k); \bar{J}_k = J_{\bar{f}_k}(\bar{\mathbf{z}}_k);$$

$$\begin{aligned}
\therefore \begin{bmatrix} \frac{\partial \mathbf{c}_k}{\partial \mathbf{z}_k} \\ \frac{\partial \mathbf{c}_k}{\partial \mathbf{z}_{k+1}} \end{bmatrix}^T &= \begin{bmatrix} I_m \\ 0_{n \times m} \\ I_m \\ 0_{n \times m} \end{bmatrix} - \frac{h_k}{6} \begin{bmatrix} \left(\frac{\partial \mathbf{f}_k}{\partial \mathbf{z}_k} \right)^T \\ \left(\frac{\partial \mathbf{f}_{k+1}}{\partial \mathbf{z}_{k+1}} \right)^T \end{bmatrix} - \frac{2h_k}{3} \left(\frac{\partial \bar{\mathbf{z}}_k}{\partial \mathbf{p}_k} \right)^T \left(\frac{\partial \bar{\mathbf{f}}_k}{\partial \bar{\mathbf{z}}_k} \right)^T \\
&= \begin{bmatrix} I_m \\ 0_{n \times m} \\ I_m \\ 0_{n \times m} \end{bmatrix} - \frac{h_k}{6} \begin{bmatrix} J_k^T \\ J_{k+1}^T \end{bmatrix} - \frac{2h_k}{3} \left(\begin{bmatrix} I_{m+n} \\ I_{m+n} \end{bmatrix} + \begin{bmatrix} J_k^T & 0_{n \times (m+n)} \\ J_{k+1}^T & 0_{n \times (m+n)} \end{bmatrix} \right) \bar{J}_k^T \quad (3-6)
\end{aligned}$$

$$\begin{aligned}
\left(\frac{\partial \mathbf{c}_k}{\partial t_f} \right)^T &= \frac{-h_k}{6} \left(\left(\frac{\partial (\mathbf{f}_k + 4\bar{\mathbf{f}}_k + \mathbf{f}_{k+1})}{\partial t_f} \right)^T + 4 \left(\frac{\partial \bar{\mathbf{z}}_k}{\partial t_f} \right)^T \left(\frac{\partial \bar{\mathbf{f}}_k}{\partial \bar{\mathbf{z}}_k} \right)^T \right) \\
&= \frac{-h_k}{6} \left(\left(\frac{\partial (\mathbf{f}_k + 4\bar{\mathbf{f}}_k + \mathbf{f}_{k+1})}{\partial t_f} \right)^T + \frac{h_k}{2} \begin{bmatrix} \left(\frac{\partial (\mathbf{f}_k - \mathbf{f}_{k+1})}{\partial t_f} \right)^T \\ 0_{n \times 1} \end{bmatrix} \bar{J}_k^T \right)
\end{aligned}$$

If the system is autonomous (that is, not explicitly dependent on the independent variable):

$$\left(\frac{\partial \mathbf{c}_k}{\partial t_f} \right)^T = \frac{-h_k}{6} \left(\left(\frac{\mathbf{f}_k + 4\bar{\mathbf{f}}_k + \mathbf{f}_{k+1}}{t_f} \right)^T + \frac{h_k}{2} \begin{bmatrix} \mathbf{f}_k - \mathbf{f}_{k+1} \\ t_f \\ 0_{n \times 1} \end{bmatrix}^T \bar{J}_k^T \right) \quad (3-7)$$

Let $\mathbf{y}_1 = [\mathbf{z}_1^T, \mathbf{z}_2^T, \dots, \mathbf{z}_N^T, t_f]^T \in \mathbb{R}^{(N(m+n)+1) \times 1}$ and $\mathbf{c} = [\mathbf{c}_1^T, \mathbf{c}_2^T, \dots, \mathbf{c}_{N-1}^T]^T \in \mathbb{R}^{(N-1)m \times 1}$

Then the Jacobian matrix for phase 1, $J_c(\mathbf{y}_1)$, can be represented as follows as follows:

$$(J_c(\mathbf{y}_1))^T = \left(\frac{\partial \mathbf{c}}{\partial \mathbf{y}_1} \right)^T$$

$$= \begin{bmatrix} \left(\frac{\partial \mathbf{c}_1}{\partial \mathbf{z}_1}\right)^T & \mathbf{0}_{(m+n) \times m} & \mathbf{0}_{(m+n) \times m} & \cdots & \mathbf{0}_{(m+n) \times m} \\ \left(\frac{\partial \mathbf{c}_1}{\partial \mathbf{z}_2}\right)^T & \left(\frac{\partial \mathbf{c}_2}{\partial \mathbf{z}_2}\right)^T & \mathbf{0}_{(m+n) \times m} & \cdots & \mathbf{0}_{(m+n) \times m} \\ \mathbf{0}_{(m+n) \times m} & \left(\frac{\partial \mathbf{c}_2}{\partial \mathbf{z}_3}\right)^T & \left(\frac{\partial \mathbf{c}_3}{\partial \mathbf{z}_3}\right)^T & \cdots & \mathbf{0}_{(m+n) \times m} \\ \mathbf{0}_{(m+n) \times m} & \mathbf{0}_{(m+n) \times m} & \left(\frac{\partial \mathbf{c}_3}{\partial \mathbf{z}_4}\right)^T & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \left(\frac{\partial \mathbf{c}_{N-1}}{\partial \mathbf{z}_{N-1}}\right)^T \\ \mathbf{0}_{(m+n) \times m} & \mathbf{0}_{(m+n) \times m} & \mathbf{0}_{(m+n) \times m} & \cdots & \left(\frac{\partial \mathbf{c}_{N-1}}{\partial \mathbf{z}_N}\right)^T \\ & & \sum_{k=1}^{N-1} \left(\frac{\partial \mathbf{c}_k}{\partial t_f}\right)^T & & \end{bmatrix} \quad (3-8)$$

This matrix is the transpose of Jacobian of the nonlinear constraints.

The gradient of the integral component of the cost function for a particular phase, $\nabla_{\mathbf{y}_1} C$, is found as follows:

$$\text{Let } L_k = L(\mathbf{z}_k, t_f); \text{ let } \bar{L}_k = L(\bar{\mathbf{z}}_k, t_f)$$

$$\text{Let } G_k = \left(\frac{\partial L_k}{\partial \mathbf{z}_k}\right)^T; \bar{G}_k = \left(\frac{\partial \bar{L}_k}{\partial \bar{\mathbf{z}}_k}\right)^T$$

$$\text{Let } C_k = \frac{h_k}{6} \left(L(\mathbf{x}_k, \mathbf{u}_k, t_f) + 4L(\bar{\mathbf{x}}_k, \bar{\mathbf{u}}_k, t_f) + L(\mathbf{x}_{k+1}, \mathbf{u}_{k+1}, t_f) \right)$$

$$\begin{aligned}
\nabla_{\begin{bmatrix} \mathbf{z}_k \\ \mathbf{z}_{k+1} \end{bmatrix}} C_k &= \frac{h_k}{6} \left(\begin{bmatrix} G_k \\ G_{k+1} \end{bmatrix} + 4 \left(\frac{\partial \bar{\mathbf{z}}_k}{\partial \mathbf{p}_k} \right)^T \left(\frac{\partial \bar{L}_k}{\partial \bar{\mathbf{z}}_k} \right)^T \right) \\
&= \frac{h_k}{6} \left(\begin{bmatrix} G_k \\ G_{k+1} \end{bmatrix} + 4 \left(\begin{bmatrix} I_{m+n} \\ I_{m+n} \end{bmatrix} + \begin{bmatrix} J_k^T & \mathbf{0}_{n \times (m+n)} \\ J_{k+1}^T & \mathbf{0}_{n \times (m+n)} \end{bmatrix} \right) \bar{G}_k \right)
\end{aligned}$$

$$\text{let } \mathbf{H}_k = \begin{cases} \begin{bmatrix} \nabla_{\begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \end{bmatrix}} C_1 \\ \mathbf{0}_{N-2m+1} \end{bmatrix} & , \text{ if } k = 1 \\ \begin{bmatrix} \mathbf{0}_{(m+n)(k-1)} \\ \nabla_{\begin{bmatrix} \mathbf{z}_k \\ \mathbf{z}_{k+1} \end{bmatrix}} C_k \\ \mathbf{0}_{(m+n)(N-k-1)+1} \end{bmatrix} & , \text{ if } 1 < k \leq N-1 \end{cases}$$

$$\begin{aligned}\frac{\partial C}{\partial t_f} &= \sum_{k=1}^{N-1} \frac{h_k}{6} \left(\frac{\partial(L_k + 4\bar{L}_k + L_{k+1})}{\partial t_f} + \frac{h_k}{2} \left(\frac{\partial \bar{\mathbf{z}}_k}{\partial t_f} \right)^T \left(\frac{\partial \bar{L}_k}{\partial \bar{\mathbf{z}}_k} \right)^T \right) \\ &= \sum_{k=1}^{N-1} \frac{h_k}{6} \left(\frac{\partial(L_k + 4\bar{L}_k + L_{k+1})}{\partial t_f} + \frac{h_k}{2} \left[\begin{array}{c} \left(\frac{\partial(\mathbf{f}_k - \mathbf{f}_{k+1})}{\partial t_f} \right)^T \\ 0_{n \times 1} \end{array} \right]^T \bar{G}_k \right)\end{aligned}$$

If the system is autonomous, then $\frac{\partial C}{\partial t_f}$ and $\nabla_{\mathbf{y}_1} C$ are:

$$\begin{aligned}\frac{\partial C}{\partial t_f} &= \sum_{k=1}^{N-1} \frac{h_k}{6} \left(\frac{(L_k + 4\bar{L}_k + L_{k+1})}{t_f} + \frac{h_k}{2} \left[\begin{array}{c} (\mathbf{f}_k - \mathbf{f}_{k+1}) \\ t_f \\ 0_{n \times 1} \end{array} \right]^T \bar{G}_k \right) \\ \nabla_{\mathbf{y}_1} C &= \sum_{k=1}^{N-1} \mathbf{H}_k + \left[\begin{array}{c} \mathbf{0}_{(m+n)N} \\ \frac{\partial C}{\partial t_f} \end{array} \right] \quad (3-9)\end{aligned}$$

In the event that there are multiple trajectory phases, the derivative information is as follows:

Let t_f^i represent the final time of phase i , \mathbf{y}_k represent a vector of all the unknown variables of phase k , \mathbf{c}^i represent a vector of the collocation constraints of phase i , and P be the number of phases.

$$\text{Let } \mathbf{y} = [\mathbf{y}_1^T, \mathbf{y}_2^T, \dots, \mathbf{y}_P^T]^T$$

$$\text{Let } \mathbf{cc} = \begin{bmatrix} \mathbf{c}^1 \\ \mathbf{c}^2 \\ \vdots \\ \mathbf{c}^P \end{bmatrix}$$

Then the Jacobian matrix for phase for all phases, $J_{cc}(\mathbf{y})$, can be represented as follows:

:

$$(J_{cc}(\mathbf{y}))^T = \begin{bmatrix} (J_{y_1}(\mathbf{c}^1))^T & 0 & 0 & 0 \\ 0 & (J_{y_2}(\mathbf{c}^2))^T & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & (J_{y_P}(\mathbf{c}^P))^T \end{bmatrix} \quad (3-10)$$

and gradient of the integral component of the cost function for multiple phase trajectories, $\nabla_{\mathbf{y}}C$, is:

$$\nabla_{\mathbf{y}}C = \begin{bmatrix} \nabla_{y_1}C \\ \nabla_{y_2}C \\ \vdots \\ \nabla_{y_P}C \end{bmatrix} \quad (3-11)$$

3.4. Interpolation of Solution and Error Estimation

In order to use coarse grids to represent the solution, and to use the solutions of coarse grids to initialise the solver for a finer grid, it was necessary to use interpolation consistent with Hermite-Simpson in order to represent the solution.

The following interpolant function of the state and control variables for a particular grid interval was employed:

First, let:

$$p = t - t_k$$

then:

$$\begin{aligned} \tilde{\mathbf{x}}(t) = \mathbf{x}_k + \left(\frac{p}{h_k}\right) \mathbf{f}_k + \frac{1}{2} \left(\frac{p}{h_k}\right)^2 (-3\mathbf{f}_k + 4\bar{\mathbf{f}}_k - \mathbf{f}_{k+1}) \\ + \frac{1}{3} \left(\frac{p}{h_k}\right)^3 (2\mathbf{f}_k - 4\bar{\mathbf{f}}_k + 2\mathbf{f}_{k+1}) \end{aligned} \quad (3-12)$$

and:

$$\frac{d\tilde{\mathbf{x}}(t)}{dt} = \mathbf{f}_k + \left(\frac{p}{h_k}\right) (-3\mathbf{f}_k + 4\bar{\mathbf{f}}_k - \mathbf{f}_{k+1}) + \left(\frac{p}{h_k}\right)^2 (2\mathbf{f}_k - 4\bar{\mathbf{f}}_k + 2\mathbf{f}_{k+1}) \quad (3-13)$$

$$\tilde{\mathbf{u}}(t) = \mathbf{u}_k + \frac{p}{h_k}(\mathbf{u}_{k+1} - \mathbf{u}_k) \quad (3-14)$$

The relative discretisation error for each mesh or grid interval is calculated as follows (Betts, 2010):

$$\boldsymbol{\eta}_k = \int_{t_k}^{t_{k+1}} \left| \frac{d\tilde{\mathbf{x}}(s)}{ds} - \mathbf{f}(\tilde{\mathbf{x}}(s), \tilde{\mathbf{u}}(s), s) \right| ds$$

$$w_j = \max_k \left[\left| \frac{d\tilde{x}_{j,k}}{dt} \right|, |\tilde{x}_{j,k}| \right]$$

$$\epsilon = \max_{j,k} \frac{\eta_{j,k}}{w_j + 1} \quad (3-15)$$

where j indicates the j th state variable, k indicates the k th grid interval, and ϵ is the maximum relative local error. The integral which is equivalent to $\boldsymbol{\eta}_k$ is solved with Romberg quadrature (Betts, 2010).

4. SOLVER VALIDATION

This section aims to present a validation of the developed optimal control solver. This has been done by comparing solutions obtained using the developed solver against existing solutions of known optimal control problems. This section details the examples used for validation and compares the existing solutions to those obtained by the developed solver. The specifications of the computer used for the validation problems can be found in Appendix E.

Three optimal control problems were used to validate the optimal control code, with first order derivatives being determined for all problems. The error of the solution was primarily measured by comparing the values of the cost or performance index against an established solution. The WebPlotDigitizer was used to extract data from the numerical solutions found in literature.

The first problem is a problem that was presented by Lawden (1975). It entails the optimisation of a linear dynamic system with terminal boundary constraints. It was chosen as it has an exact analytical solution to which the obtained numerical solution could be compared. The second problem relates to a co-planar circular orbital transfer, which, in this case, was treated as two different problems, with the optimisation objectives for each problem being to minimise the time taken to transfer between orbits and to maximise the radius of the destination orbit respectively. These problems, at the time this dissertation was written, have no closed-form solution. The obtained numerical solutions were therefore compared to the solutions obtained by Moyer and Pinkham (1964), and Bryson and Ho (1975). The third problem considered, which was originally presented by Benson (2010), aims to maximise the final mass of a two-stage-to-orbit launch vehicle; in particular, the Delta III launcher. The obtained numerical solution was compared to the solution obtained by Betts (2010).

4.1. Linear Dynamic System

The first problem (from Lawden (1975)) was chosen as it has an analytical solution, and has terminal boundary constraints. The problem is as follows:

Consider the dynamic system subject to the following state equations:

$$\dot{x} = y - x; \dot{y} = u - y \quad (4-1)$$

where x and y are the state variables, and u is the control variable. The system starts at a time of 0 and ends at a time of 1. The initial values for both state variables is 0. The final value of x is 1.

Optimise the following cost function, C :

$$C = \int_0^1 u^2 dt \quad (4-2)$$

The Hamiltonian of the problem is as follows:

$$H = u^2 + \lambda_x(y - x) + \lambda_y(u - y)$$

$$\therefore \frac{\partial H}{\partial u} = 2u + \lambda_y = 0$$

$$\therefore \dot{\lambda}_x = -\frac{\partial H}{\partial x} = -\lambda_x$$

$$\therefore \dot{\lambda}_y = -\frac{\partial H}{\partial y} = \lambda_y - \lambda_x$$

Substituting λ_y with $-2u$, leads to the following system of ordinary differential equations:

$$\dot{x} = y - x; \dot{y} = u - y; \dot{\lambda}_x = -\lambda_x; \dot{u} = u + \frac{\lambda_x}{2} \quad (4-3)$$

As the final value of x is defined and the final value of y is defined, λ_y at the final time must be 0. Therefore the value of u at the final time must be 0. This system of equations has been solved for using the Laplace transform, providing the following solution for the optimal control and cost:

$$u(t) = \frac{4e^{t+1}(1-t)}{e^2-5}; C = \frac{4e^2}{e^2-5} = 12.3715 \quad (4-4)$$

The problem was solved using the developed solver, with re-meshing, in 0.35 s. Without the provision of analytical first derivative information, the solver required 4.79 s to obtain the same solution. The relative discretisation error of the presented solution is 3.1285×10^{-9} . The maximum difference between the analytical and numerical solutions of the control variable is 4.2969×10^{-4} . The cost found by the solver is 12.3715, implying a 0 % objective function error. Figure 4-1 depicts the time history of the numerical solution of the state and control variables, the discretisation error, and the analytically obtained control solution. It should be noted that only first order conditions were used to determine the analytical control solution.

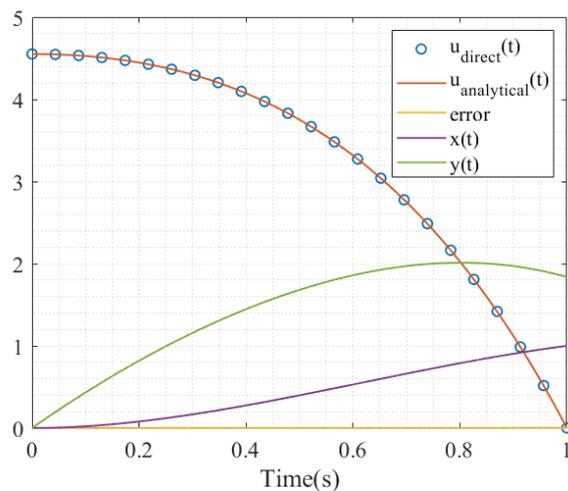


Figure 4-1: Linear validation example results

4.2. Co-Planar Circular Orbit Transfer

4.2.1. Minimum Time Orbit Transfer

The next two validation problems regard co-planar circular orbital transfer. The system dynamics were augmented from the example provided by Moyer and Pinkham (1964), such that the problems were autonomous.

The first validation problem is to determine the minimum time to transfer an object from the average circular orbit of the Earth around the Sun, to the average circular orbit of Mars around the Sun. The unit system was normalised by Moyer and Pinkham such that the time unit is equivalent to 58.18 days. The distance unit used is AU (astronomical unit). The mass unit used is

equivalent to 46.58 slugs, or 14.594 kg. The following assumptions were applied (Moyer and Pinkham, 1964)

- Constant thrust
- Gravity and thrust are the only forces acting on the rocket
- Earth and Mars have circular, co-planar orbits
- Only the gravitational attraction between the sun and rocket was considered

The dynamic equations used to describe the system are as follows:

$$\dot{r} = v_r \quad (4-5a)$$

$$\dot{\theta} = \frac{v_\theta}{r} \quad (4-5b)$$

$$\dot{v}_r = \frac{v_\theta^2}{r} - \frac{\mu}{r^2} + \frac{T u_1}{m} \quad (4-5c)$$

$$\dot{v}_\theta = -\frac{v_\theta v_r}{r} + \frac{T u_2}{m} \quad (4-5d)$$

$$\dot{m} = -|\dot{m}| \quad (4-5e)$$

$$s. t. u_1^2 + u_2^2 = 1 \quad (4-5f)$$

where r and θ are the polar coordinates of the vehicle, with the sun as the origin; v_r is the radial component of the velocity; v_θ is the circumferential component of the velocity; T is the thrust magnitude, u_1 and u_2 are the components of the thrust direction vector; and m is the mass.

The normalised boundary constraints are:

$$r(0) = 1; r(t_f) = 1.525; m(0) = 1 \quad (4-6a)$$

$$v_r(0) = 0; v_r(t_f) = 0 \quad (4-6b)$$

$$v_\theta(0) - \sqrt{\frac{\mu}{r(0)}} = 0; v_\theta(t_f) - \sqrt{\frac{\mu}{r(t_f)}} = 0 \quad (4-6c)$$

The following constants were used:

$$T = 0.8098; |\dot{m}| = 0.07487; \mu = 1$$

The following solution (Figure 4-2), obtained from Bryson and Ho (1975), is a representation of the optimal trajectory, and the corresponding optimal control history.

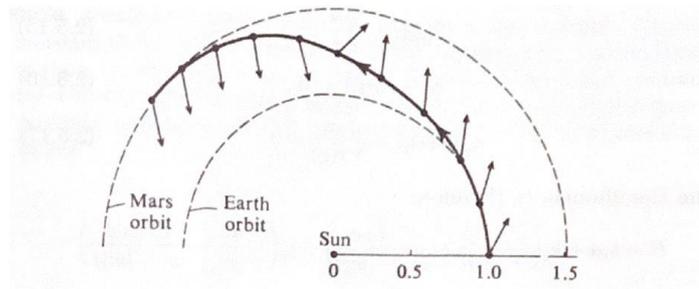


Figure 4-2: The orbital transfer trajectory solution of Bryson and Ho (1975)

The problem was solved in 1.34 seconds using the developed optimal control solver with re-meshing. Without the provision of analytical first derivative information, the solver required 14.82 seconds to obtain the same solution. The relative discretisation error of the presented solution is 7.659×10^{-7} . The minimum time found by Moyer and Pinkham (1964) was 193.2 days, or 3.3207 time units. The minimum time found by the developed optimal control solver was 3.3206 time units. The variables were unscaled. The following solution (Figure 4-3) was obtained using the developed optimal control solver:

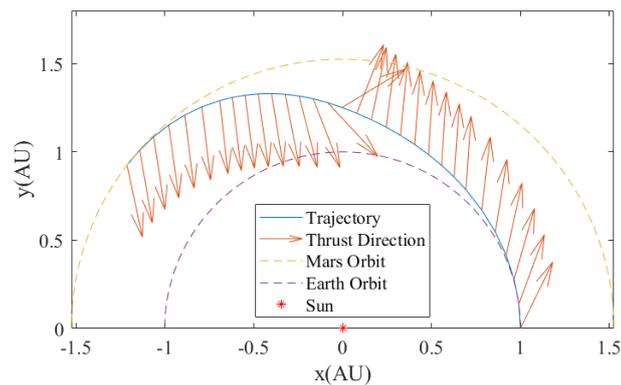


Figure 4-3: Developed solver's orbital transfer trajectory solution for minimum time

As stated in Bryson and Ho (1975), the optimal radial thrust component is outward for roughly the first half of the trajectory, and inward for the rest. The results obtained are consistent with this observation.

Figures 4-4 and 4-5 compare the thrust direction history determined by Moyer and Pinkham (1964), to the history determined by the developed optimal control solver. Figure 4-4 shows that the thrust direction history solved for is consistent with the thrust direction history found by Moyer and Pinkham (note that θ_k indicates the k th iteration of the solver and the optimal solution is represented by θ^*).

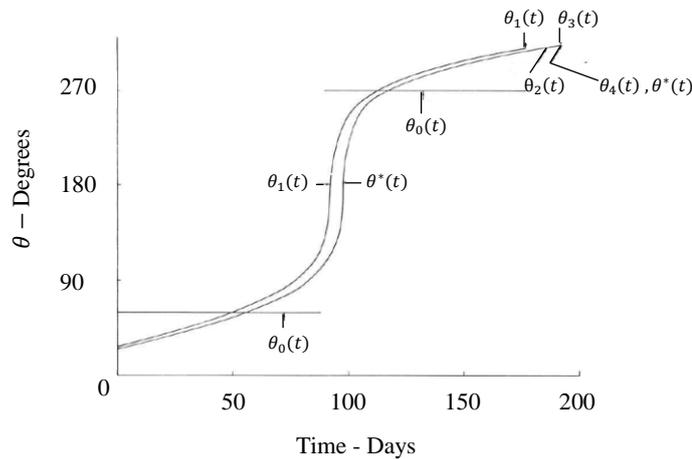


Figure 4-4: Moyer and Pinkham's (1964) thrust direction solution

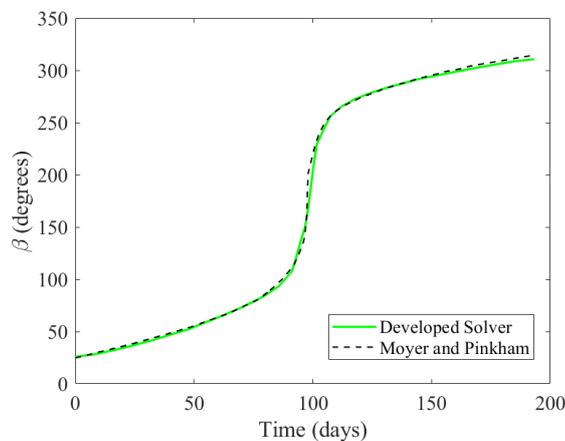


Figure 4-5: Developed solver's thrust direction solution for minimum time

4.2.2. Maximum Radius Orbit Transfer

Bryson and Ho (1975) presented a similar problem to the minimum time orbital transfer, the difference being that the final time is fixed and the objective is to maximise the radius of the orbit being transferred to. As stated before, Bryson and Ho (1975) provided an example of a numerical solution to this problem by using the solution obtained by Moyer and Pinkham (1964) for the

minimum time orbital transfer problem. The time unit used is the same as the minimum time problem.

The terminal boundary constraints become non-linear constraints as the final radius is unknown. Hence it was necessary (as analytical gradients were used) to determine the gradient of these constraints and provide them to the solver.

The problem was solved in 2.41 s using the developed optimal control solver with re-meshing. Without the provision of analytical first derivative information, the solver required 22.34 seconds to obtain the same solution. The relative discretisation error of the presented solution is 6.567×10^{-7} . The maximum radius found by Bryson and Ho (1975) was 1.525 AU. The maximum radius of the destination orbit was found to be 1.525 AU by the developed solver, this implies a 0% objective function error relative to the solution found in Bryson and Ho (1975). The variables were unscaled.

The trajectory and control history solutions reflected in Figures 4-6 and 4-7 were found using the developed optimal control solver. These solutions are consistent with the solutions presented for the minimum time problem. Figure 4-6 matches the trajectory obtained by Bryson and Ho in Figure 4-2.

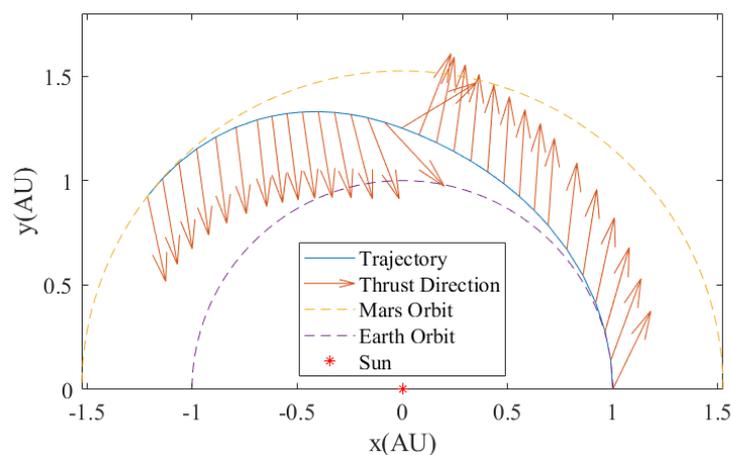


Figure 4-6: Developed solver's orbital transfer trajectory solution for maximum orbital radius

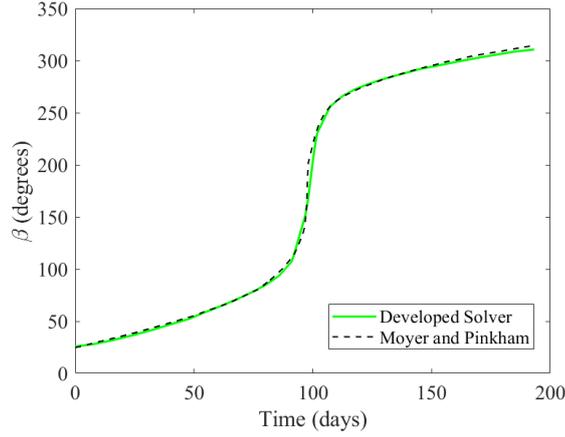


Figure 4-7: Developed solver's thrust direction solution for maximum orbital radius

4.3. Delta III Launch Vehicle

This problem, presented by Benson (2010), is a simplified version of the ascent-to-orbit trajectory of the Delta III launch vehicle. The objective of this problem is to maximise the final mass of the vehicle whose motion is limited to three degrees of freedom. Only drag, thrust and gravity are considered. The Earth is assumed to be spherical and is rotating. The drag coefficient is assumed to be constant. The purpose of considering this problem was to validate that problems with trajectory phases can be solved for by the developed solver. The following equations represent the system dynamics and constraints for a particular phase of flight:

$$\dot{\mathbf{r}} = \mathbf{v} ; \|\mathbf{r}\| \geq R_E \quad (4-7a)$$

$$\dot{\mathbf{v}} = -\frac{\mu}{\|\mathbf{r}^3\|} + \frac{T}{m}\mathbf{u} + \frac{1}{m}\mathbf{D} \quad (4-7b)$$

$$\dot{\mathbf{x}} = g(\mathbf{x}(t), \mathbf{u}(t)) \quad (4-7c)$$

$$\dot{m} = -k \quad (4-7d)$$

$$\|\mathbf{u}\|^2 = 1 \quad (4-7e)$$

Drag (\mathbf{D}) is modelled as follows:

$$\mathbf{D} = -\frac{1}{2}\rho 2\pi\|\mathbf{v}_{rel}\|\mathbf{v}_{rel} \quad (4-8a)$$

$$\mathbf{v}_{rel} = \mathbf{v} - \boldsymbol{\omega} \times \mathbf{r} \quad (4-8b)$$

$$h = \|\mathbf{r}\| - R_E \quad (4-8c)$$

$$\rho = 1.225e^{-\frac{h}{7200}} \quad (4-8d)$$

where \mathbf{r} is the position, \mathbf{v} is the velocity, \mathbf{u} is the control vector, and m is the mass of the vehicle. Air density is represented by ρ .

The magnitude of $\boldsymbol{\omega}$ is $7.29211585 \times 10^{-5}$ rad/s. The direction of $\boldsymbol{\omega}$ is coincident with spin axis of Earth, towards North. The magnitude of R_E is 6378145 m. Table 4-1 indicates the constraints associated with each of the four phases of the Delta III launch vehicle trajectory.

Table 4-1: Delta III trajectory phases (Betts, 2010)

| Phase | Final time (s) | Thrust (kN) | Mass flow rate (kg/s) | Initial mass (kg) |
|-------|----------------|-------------|-----------------------|-------------------|
| 1 | 75.2 | 4854.1 | 1723.27 | 301454 |
| 2 | 150.4 | 2968.6 | 1044.68 | 158184.032 |
| 3 | 261 | 1083.1 | 366.09 | 70451.064 |
| 4 | Unknown | 110.094 | 24.03 | 2480 |

The orbit into which the vehicle is injected is defined as follows. The apogee is 24361140 m with an eccentricity of 0.7308 and the inclination is 28.5 degrees. The argument of perigee is 130.5 degrees, and the right ascension of the ascending node is 269.8 degrees.

The problem was augmented by using the injection point taken from Coskun (2013) found using GPOPS II and SNOPT. The injection point thus became a known variable. The true anomaly of the injection point was set to be 6.7834 degrees. The solution presented in Figure 4-8 is the altitude vs time solution obtained in Betts (2010).

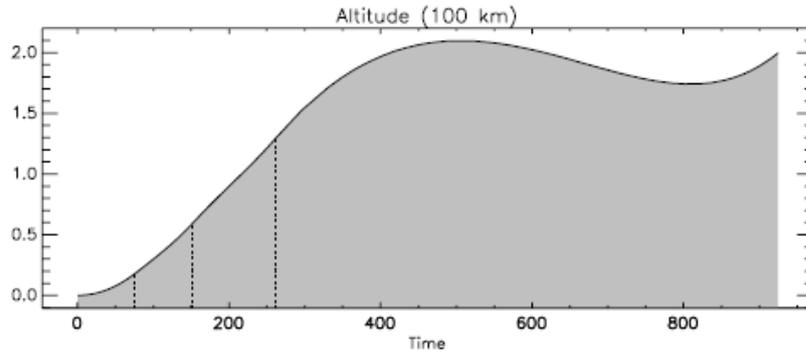


Figure 4-8: Betts' Delta III altitude vs time solution (Betts, 2010)

The problem was solved in 152.26 seconds using the developed optimal control solver, with re-meshing. Without the provision of analytical first derivative information, the solver required 1712.22 seconds to obtain the same solution. The relative discretisation error of the obtained solution is 3.6315×10^{-5} . The maximum time to orbit and final mass (of the launch vehicle and payload) found by Betts (2010) were 924.139 seconds and 7529.712 kg. The maximum time to orbit and final mass found by the solver were 924.142 seconds and 7529.742 kg; this implies a 3.984×10^{-4} objective function error relative to the solution found by Betts (2010).

Figure 4-9, is the altitude vs time solution found using the developed optimal control solver. It can be seen to be similar to the solution obtained by Betts (2010). It is worth noting that the vehicle follows a lofted trajectory, that is, a trajectory in which a vehicle ascends to a peak altitude, following which it descends to a locally minimum altitude, and then ascends once more.

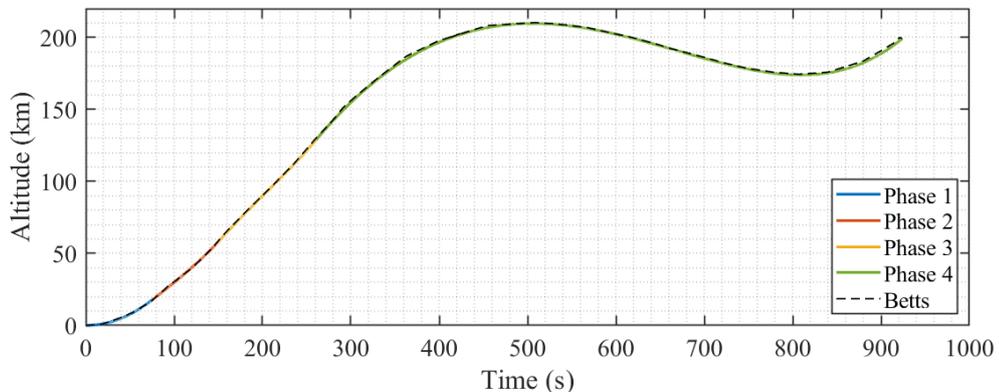


Figure 4-9: Developed solver's Delta III altitude vs time solution

5. ASCENT-TO-ORBIT TRAJECTORY MODELLING

5.1. Force Models and Reference Frames

This section details the reference frames and dynamic equations used to model the three Degrees of Freedom (3-DOF), two-stage ascent-to-orbit trajectories that were analysed in this study. It is necessary to define the frames of reference in which the position and velocity of the vehicle will be represented. All used reference frames are defined by an orthogonal set of axes. First, the geocentric equatorial frame (GEF) will be defined. The origin of this axis is located at the center of Earth, the x-axis of this reference frame points in the direction of the vernal equinox, and the z-axis is the spin axis of Earth. The second coordinate system used is a zenith-east-north (ZEN) topocentric horizon coordinate system. The location of the origin is chosen. The x-axis points in the zenith direction, the y-axis points east, and the z-axis points north.

The launch vehicles considered in the study were modelled as point masses with three degrees of freedom, represented by the origin of the ZEN reference frame. In order to define the ZEN reference frame, it is necessary to define r , the distance of the vehicle away from the centre of the Earth, ϕ , the latitude of the vehicle, and θ , the celestial longitude of the vehicle. The velocity of the vehicle will be represented by three different orthogonal components using the ZEN frame. v_z , v_E and v_N represent the zenith, east and north velocity components respectively. Figure 5-1 depicts the GEF and ZEN coordinate systems.

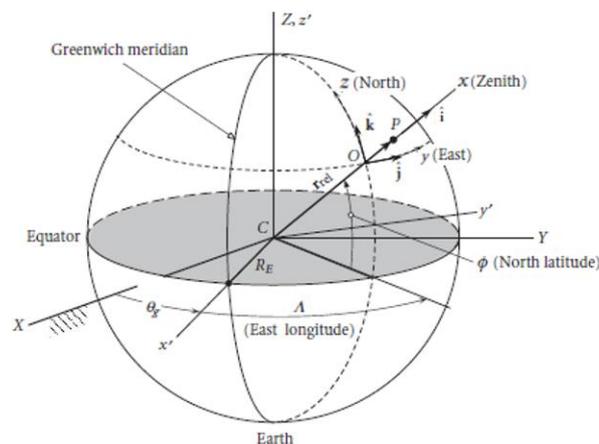


Figure 5-1: GEF and ZEN reference frames (Curtis, 2005)

Note that:

$$\phi \in [-90^\circ, 90^\circ]; \theta \in [0^\circ, 360^\circ]$$

Before the state vector differential equations are defined, the assumptions made to model the dynamics of the launch vehicle will first be stated. These are as follows:

- The Earth is a spherical rotating body
- Earth has a uniform density
- The radius of the spherical Earth is the mean radius, 6371km
- Only Newtonian Mechanics apply
- The GEF is an inertial reference frame
- Cross wind force is negligible
- The orientation of the vehicle is determined by the thrust of the vehicle
- Wind velocity is solely dependent on altitude
- Solar radiation pressure is negligible
- Earth only rotates around the Z-axis shown in Figure 5-1
- Throttle setting is constant for the duration of any given phase
- The momentum thrust is constant for the duration of a given phase
- The mass flow rate of propellants is constant for the duration of any given phase
- The 1976 US atmosphere model is applied
- All assumptions applied by the RASAero™ software program in computing aerodynamic data for the considered launch vehicles

The variables representing position and velocity were converted into standard state variables, x_i (where i indicates the i th state variable), as follows:

$$\text{Let } x_1 = r; x_2 = \theta; x_3 = \delta; x_4 = v_z; x_5 = v_E; x_6 = v_N$$

The dynamic equations are thus as follows:

$$\dot{\mathbf{x}} = \begin{bmatrix} x_4 \\ \frac{x_5}{x_1} \sec x_3 \\ \frac{x_6}{x_1} \\ \frac{1}{x_1} (x_5^2 + x_6^2) - \frac{\mu}{x_1^2} + \frac{T}{x_7} u_1 \\ \frac{x_5}{x_1} (x_6 \tan x_3 - x_4) + \frac{T}{x_7} u_2 \\ \frac{-1}{x_1} (x_5^2 \tan x_3 + x_4 x_6) + \frac{T}{x_7} u_3 \\ -k \end{bmatrix} + \begin{bmatrix} 0_{3 \times 1} \\ \frac{\mathbf{D} + \mathbf{L}}{x_7} \\ 0 \end{bmatrix} \quad (5-1)$$

Here T represents the thrust magnitude, \mathbf{D} and \mathbf{L} represent the drag and lift force vectors, and u_1, u_2 and u_3 represent the different components of the thrust direction. Under vacuum conditions, the drag and lift forces are zero.

Thrust magnitude is modelled as follows:

$$T = T_{vacuum} - p_a A_e \quad (5-2)$$

where T_{vacuum} is the vacuum thrust, p_a is the atmospheric pressure at a given altitude, and A_e is the nozzle exit area. If the vehicle is throttled, the vacuum thrust magnitude must be multiplied by the throttle parameter in Equation 5-2 above.

Wind velocity profiles were obtained from the European Centre for Medium-Range Weather Forecasts (ECMWF) datasets. The wind is the 2021 annual average wind localised at the OTR launch site. Drag and lift coefficients were obtained from RASAeroTM using approximations of the launch vehicles geometry. The information used to approximate their geometry was obtained from each launch vehicle's respective payload user's guides. All aerodynamic data were then interpolated using cubic interpolants. The results were stored in tables from which they were extracted using table look-up procedures. The following equations represent the aerodynamic forces:

$$D = \frac{1}{2} C_D S_p \rho_{air} v_{rel}^2 \quad (5-3)$$

$$L = \frac{1}{2} C_L S_p \rho_{air} v_{rel}^2 \quad (5-4)$$

$$v_{rel} = \|\mathbf{v} - \mathbf{w}\| \quad (5-5)$$

where C_D and C_L are the drag and lift coefficient respectively, S_p is the aerodynamic reference area, ρ_{air} is the air density at the position of the launch vehicle, \mathbf{v} is the velocity vector, \mathbf{w} is the wind velocity, and v_{rel} is the magnitude relative velocity of the vehicle with respect to the atmosphere.

5.2. Ascent Trajectory Phases

This section describes the methodology used in determining the optimal two stage ascent-to-orbit trajectories of the analysed missions. Lower stages operate in high atmospheric pressure whereas upper stages operate under vacuum conditions (Coskun, 2014). Hence, the second stage was assumed to have achieved sufficient altitude to experience vacuum conditions.

The first stage trajectory was designed to follow a gravity turn, whereas the second stage had no restriction on thrust direction. The first stage trajectory was solved using MATLAB's "ode45" function. This function is an initial value numerical integrator that makes use of an adaptive fifth order Runge-Kutta scheme and error estimation. The second stage trajectory was solved for using the developed optimal control solver, by using the first stage outputs as inputs. The first stage trajectory was not solved for using the optimal control solver for the following reasons: the first stage trajectory was mostly predefined by the gravity turn; the complications of optimising atmospheric ascent trajectories associated with tabular data; and MATLAB's SQP solver's poor performance for nonlinear optimal control problems. In order to compensate for this and to optimise the first stage, different pitch rates were considered for the initial pitchover manoeuvre.

Structural constraints were imposed for the vehicles that were analysed. The maximum allowable dynamic pressure was assumed to be 55 kPa, as this is deemed to be an acceptable constraint value for most launch vehicles (Coskun, 2014).

In addition to this, axial acceleration constraints were imposed on each vehicle. The maximum axial acceleration of Electron was 7.5g; of Falcon 1 was 6.4g and of Falcon 9 was 6.5g. These

values were obtained from the payload user guides of each respective vehicle. Due to the similarity in size of the CLV and Electron, it was assumed that they shared the same axial acceleration constraint of 7.5 g. The axial acceleration formula is as follows (Federici, et al., 2019):

$$a_{axial} = \frac{\mathbf{u} \cdot \dot{\mathbf{v}}}{g_o} \quad (5-6)$$

where a_{axial} is the axial acceleration in g, \mathbf{u} is the thrust direction vector, $\dot{\mathbf{v}}$ is the acceleration vector and g_o (9.81 m.s⁻²) is the standard Earth gravitational acceleration.

In a 3-DOF study, it is necessary to assume that the thrust direction represents the orientation of the vehicle. During the initial pitchover, the thrust direction will be defined by pitch and local azimuth. These two angles can be seen in Figure 5-2 (obtained from Federici, et al. (2019)). The thrust pitch is represented by θ_T and the thrust azimuth is represented by ψ_T .

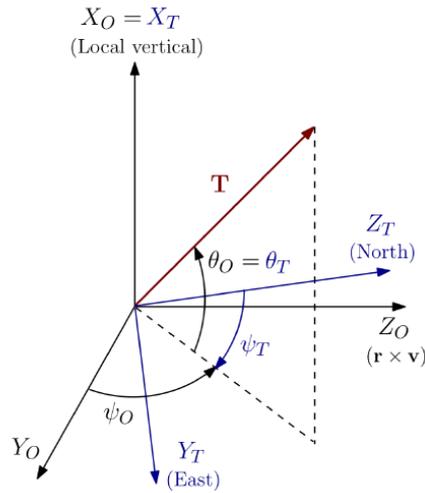


Figure 5-2: Thrust pitch and azimuth in ZEN reference frame

Trajectories to 36 distinct LEOs were analysed for each respective vehicle, with orbit altitudes of 200, 300, 400 and 500 km being considered. The inclinations of these orbits were set at 40° and 64.435°. In addition SSOs at each of these altitudes were considered. At an inclination of 64.435 degrees, the argument of perigee of the orbit does not change over time. The eccentricities of the orbits were set as 0, 0.005 and 0.01 (an eccentricity of 0 implies a circular orbit). The vehicles

were injected into the perigee point of each respective orbit. The perigee point does not exist for a circular orbit, hence the vehicles were set to be injected anywhere along the desired circular orbits. It is typical in such studies not to set the argument of perigee. The true anomaly can be controlled by selecting the time of launch, hence it was not set.

There are five trajectory phases. The phases, in order are the vertical ascent, the initial pitchover, the gravity turn, stage separation, and the vacuum ascent. The parameters for the vertical ascent and initial pitchover are obtained and modified from Coskun (2014) and Federici, et al. (2019). The trajectory phases are detailed as follows:

5.2.1. Phase 1: Vertical Ascent

The vehicle ascends from the launch infrastructure with the thrust aimed in the direction of the zenith axis. To achieve this, a pitch angle of 90° was used. Vertical ascent periods of 15, 20 and 25 seconds were attempted.

5.2.2. Phase 2: Initial Pitchover

The vehicle begins to pitchover so that a meaningful gravity turn can be achieved. A linear pitch control law was used for this purpose, in which the pitch angle decreased linearly with respect to time according to a chosen pitch rate. The initial pitch angle of this phase is 90 degrees. Pitch rates of 1, 1.1, 1.2, 1.3, 1.4 and 1.5 °/s were attempted. The azimuth of the thrust direction was calculated with the formula in Equation 5-3. This formula determines the optimal azimuth the vehicle must travel along in order to achieve its desired inclination. The launch azimuth is represented by ψ , i represents the orbit inclination, and δ represents the launch latitude. This formula must be adjusted accordingly to be used for northward launches.

$$\psi = \pi - \arcsin\left(\frac{\cos i}{\cos \delta}\right) \quad (5-7)$$

End times of 30, 35 and 40 seconds were attempted for this phase.

5.2.3. Phase 3: Gravity Turn

Once the initial pitchover was completed, the vehicles followed a gravity turn, or wind-biasing trajectory. This was achieved by aligning the thrust direction vector with the velocity of the vehicle relative to the wind. This phase lasts for the remainder of the first stage burn time.

5.2.4. Phase 4: Stage Separation

In order to compensate for stage separation, a time period of 5 seconds is allocated during which the vehicle is not powered. Stage separation is assumed to occur during this phase.

5.2.5. Phase 5: Vacuum Ascent

The outputs of Phase 4 are used to initialise the second stage of the trajectory. The second stage was defined as a vacuum ascent optimal control problem. The problem is presented as follows:

$$\max_{\mathbf{u}} m_f \quad (5-8a)$$

$$s. t. \psi(\mathbf{x}_f, t_f) = 0 \text{ (end conditions)} \quad (5-8b)$$

$$\dot{\mathbf{x}} = \mathbf{g}(\mathbf{x}, \mathbf{u}) \quad (5-8c)$$

$$\mathbf{x} \in \mathbb{R}^{m \times 1}; \mathbf{u} \in \mathbb{R}^{n \times 1} \quad (5-8d)$$

The size of the dimensions m and n are equivalent to 7 and 3 respectively. The subscript o indicates initial conditions of the trajectory; the subscript f indicates end conditions of the trajectory; and the function, $\mathbf{g}: \mathbb{R}^{(m+n) \times 1} \rightarrow \mathbb{R}^{m \times 1}$, outputs the first time derivative of the state vector.

Under vacuum conditions, it has been shown that intermediate throttle (that is, throttle between its minimum and maximum) is sub-optimal (Kopp and Moyer, 1965). Therefore, the optimal solution to the throttle history is a sequence of trajectory arcs or phases of full throttle and null throttle (if the vehicle is capable of restarting its engine). It is assumed that the first arc is an FT arc. The strategy used to find the optimal sequence of arcs is to attempt different sequences of arcs and compare the solution of each combination which was tried. The combinations considered

in this study were: [FT]; [FT NT]; [FT NT FT]. Hence, in the discretised version, the throttle parameter was not directly solved for, and was thus not considered as a part of the control vector.

Let the total number of phases be known as P, and the subscript p denote the phase of a particular variable. Let the time period of each phase be t_p , *Let there be a function, $f: \mathbb{R}^{(m+n+1) \times 1} \rightarrow \mathbb{R}^{m \times 1}$* , which outputs the first derivative of the state vector with respect to the independent variable τ , where $\tau \in [0,1]$.

On this basis:

$$f(\mathbf{x}, \mathbf{u}, t_p) = \frac{dx}{d\tau} = \frac{dt}{d\tau} \dot{\mathbf{x}} = t_p \begin{bmatrix} x_4 \\ \frac{x_5}{x_1} \sec x_3 \\ \frac{x_6}{x_1} \\ \frac{1}{x_1} (x_5^2 + x_6^2) - \frac{\mu}{x_1^2} + \frac{T}{x_7} u_1 \\ \frac{x_5}{x_1} (x_6 \tan x_3 - x_4) + \frac{T}{x_7} u_2 \\ \frac{-1}{x_1} (x_5^2 \tan x_3 + x_4 x_6) + \frac{T}{x_7} u_3 \\ -a \end{bmatrix} \quad (5-9)$$

In order to apply the direct numerical approach, Hermite-Simpson collocation is used to represent the differential constraints. The variables were scaled manually.

The NLP version of the problem is redefined as follows:

$$\max_{\mathbf{x}, \mathbf{u}} m_f \quad (5-10a)$$

$$s. t. \psi(\mathbf{x}_f^p, t_f^p) = 0 \text{ (end conditions)} \quad (5-10b)$$

$$\phi(\mathbf{x}_o^1, t_o^1) = 0 \text{ (initial conditions)} \quad (5-10c)$$

For $p=1, 2, \dots, P$; For $k=1, 2, \dots, N_p$

Let:

$$\mathbf{x}_k^p \in \mathbb{R}^{7 \times 1}; \mathbf{u}_k^p \in \mathbb{R}^{3 \times 1}$$

$$\mathbf{z}_k^p = \begin{bmatrix} \mathbf{x}_k^p \\ \mathbf{u}_k^p \end{bmatrix} \in \mathbb{R}^{(7+3) \times 1}$$

$$\mathbf{f}_k^p = \mathbf{f}_p(\mathbf{z}_k^p, t_p); \bar{\mathbf{f}}_k^p = \mathbf{f}(\bar{\mathbf{z}}_k, t_p)$$

Then:

$$\bar{\mathbf{z}}_k^p = I_{m+n} \mathbf{z}_k^p + I_{m+n} \mathbf{z}_{k+1}^p + \frac{h_k}{8} \begin{bmatrix} \mathbf{f}_k^p - \mathbf{f}_{k+1}^p \\ 0_{n \times 1} \end{bmatrix} \quad (5-10d)$$

$$\mathbf{c}_k^p = I_m \mathbf{x}_k^p - I_m \mathbf{x}_{k+1}^p - \frac{h_k}{6} (\mathbf{f}_k^p + 4\bar{\mathbf{f}}_k^p + \mathbf{f}_{k+1}^p) \quad (5-10e)$$

$$\mathbf{x}_f^p = \mathbf{x}_o^{p+1} \quad (5-10f)$$

The Jacobian necessary to use first order analytical derivatives is then defined as follows:

$$J_{f_p}(\mathbf{x}, \mathbf{u}) = [J_{f_p}(\mathbf{x}) \ J_{f_p}(\mathbf{u})]$$

$$\text{where } J_{f_p}(\mathbf{x}) = \frac{\partial f_p}{\partial \mathbf{x}}$$

$$= t_p \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ -\frac{x_5}{x_1^2} \sec x_3 & 0 & \frac{x_5}{x_1} \tan x_3 \sec x_3 & 0 & 0 & \frac{1}{x_1} \sec x_3 & 0 \\ \frac{-x_6}{x_1^2} & 0 & 0 & 0 & 0 & \frac{1}{x_1} & 0 \\ \frac{-1}{x_1^2} (x_5^2 + x_6^2) + \frac{2\mu}{x_1^3} & 0 & 0 & 0 & \frac{2x_5}{x_1} & \frac{2x_6}{x_1} & -\frac{T}{x_7^2} u_1 \\ \frac{x_5}{x_1} (x_6 \tan(x_3) - x_4) & 0 & \frac{-x_5 x_6 \sec^2 x_3}{x_1} & -\frac{x_5}{x_1} & \frac{x_6 \tan(x_3) - x_4}{x_1} & \frac{x_5 \tan x_3}{x_1} & -\frac{T}{x_7^2} u_2 \\ \frac{1}{x_1^2} (x_5^2 \tan(x_3) + x_4 x_6) & 0 & \frac{-x_5^2 \sec^2 x_3}{x_1} & -\frac{x_6}{x_1} & \frac{-2x_5 \tan x_3}{x_1} & \frac{-x_4}{x_1} & -\frac{T}{x_7^2} u_3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (5-11)$$

and:

$$J_{f_p}(\mathbf{u}) = \frac{\partial f_p}{\partial \mathbf{u}} = \frac{T t_p}{x_7} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \quad (5-12)$$

On this basis, each mission problem was then solved by the optimal control solver. Only solutions with an error estimate less than 10^{-7} were accepted.

6. RESULTS

This chapter details the trajectory performance, structural loading and control history data predicted for the example launch vehicles considered in this study, as per the model described above. A subsection is dedicated to each launch vehicle, containing a description of the vehicle and the parameters used to conduct the associated trajectory analysis, in addition to payload performance charts and an altitude versus range sample plot. Stage separation in all the altitude versus range plots is indicated by a black dot. All the altitude versus range plots can be found in Appendix C. Appendix D contains the RASAero™ inputs used to obtain the aerodynamic coefficients of the launch vehicles analysed in this work.

Additionally, the linearity of the tangent pitch and tangent yaw of the thrust direction vector is calculated relative to time. A final section includes and compares the time to orbit and the pitch rate during the initial pitchover of all the analysed launch vehicles. Launch vehicle parameters were obtained from the following sources: Space Launch Report (2021), each vehicle's respective payload user's guide, the SpaceX website (SpaceX, 2021) and a website created by Brugge (2021). All results were computed using a HP 15-da0031ni notebook, the specifications of which can be found in Appendix E.

As stated in Chapter 2, the linear tangent law for the thrust direction can provide near optimal if not optimal solutions to the orbital injection problem. Utilising this law would allow for shooting methods to be used in future work, which improves the accuracy of solutions. Examples of tangent pitch and yaw plots of Rocket Lab's Electron can be found in Figures 6-1 and 6-2. It is evident that these plots follow linear trends relative to τ (the ratio between time and the total trajectory time).

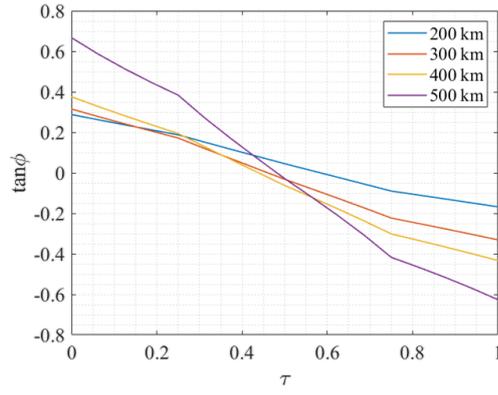


Figure 6-1: Electron tangent of thrust pitch vs τ ($e=0$, $i=40^\circ$)

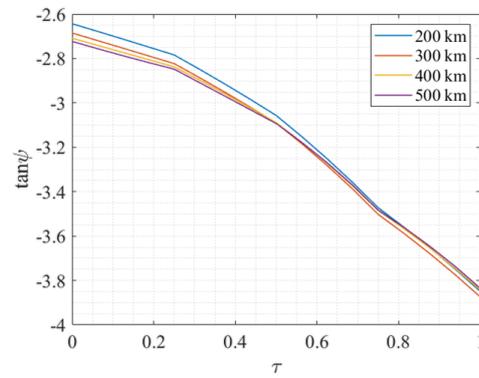


Figure 6-2: Electron tangent of thrust yaw vs τ ($e=0$, $i=40^\circ$)

The absolute value of the Pearson or linear correlation coefficient will be used to measure the linearity of the tangent of the control data. This was determined using the following formula (Essa, 2013):

$$|r| = \left| \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \sum(y_i - \bar{y})^2}} \right| \quad (6-1)$$

where r is the Pearson correlation coefficient, x_i is the i th dependent variable, y_i is the i th independent variable, \bar{x} is the mean of the dependent variable and \bar{y} is the mean of the independent variable. The closer this value is to 1, the greater the linearity of the data. The linearity was found to barely vary as the eccentricity of orbit was changed, hence an average of the linearity coefficient across the different eccentricities was used for missions to orbits of the same altitude and inclination.

In order to show that the first stage trajectory phases are reasonable, the first stage altitude and time of the “running out of fingers” launch mission, conducted by Rocket Lab’s Electron was compared to the simulated results. This mission entailed first entering a low eccentricity polar transfer orbit with a perigee 186.01 km of an inclination of 97.01°. Thus it was compared to the simulated first stage launch data to an SSO orbit with a perigee of 200 km and eccentricity of 0.01. Figure 6-3 depicts both the flight data and the simulated data. The actual data was obtained from a webcast posted by Rocket Lab (Rocket Lab, 2019). Obtaining data from a video source can lead to some error. There is a good agreement between the two sets of data. The actual flight path is consistently at a lower altitude than the simulated data, these discrepancies can be explained due to the difference in destination orbit, launch site inclination, and wind conditions.

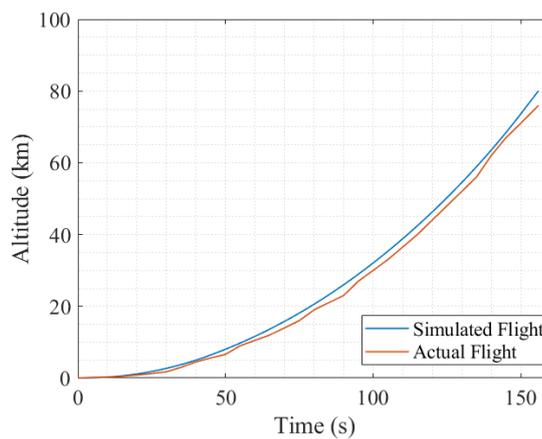


Figure 6-3: Electron first stage altitude vs time

6.1. Rocket lab’s Electron

The Electron launch vehicle is a relatively light launch vehicle that is designed for the injection of light payloads into LEOs. The vehicle is manufactured by Rocket Lab and is powered by Rocket Lab’s Rutherford engines. Table 6-1 details the input parameters used to generate the results data associated with the Electron launch vehicle. Table 6-2 details the relevant information regarding the predicted structural loads experienced by the Electron launch vehicle during the analysed missions.

Table 6-1: Electron simulation parameters

| Parameter Description | Parameter value |
|--------------------------|-----------------|
| m_1 (kg) | 12500 |
| \dot{m}_1 (kg/s) | 61.26 |
| Thrust ₁ (kN) | 225 |
| A_e (m ²) | 0.375 |
| m_2 (kg) | 2300 |
| \dot{m}_2 (kg/s) | 7.67 |
| Thrust ₂ (kN) | 26.3 |

Table 6-2: Electron structural load parameters

| Structural Load Parameter | Structural Load Parameter Value |
|------------------------------|---------------------------------|
| Maximum Q (kPa) | 44.44 |
| Altitude range of max Q (km) | 8.64-10.44 |
| First stage max a_x (g) | 4.43 |
| Second stage max a_x (g) | 7.27 |

Figures 6-4 to 6-6 detail the payload performance of the Electron launch vehicle for the analysed missions.

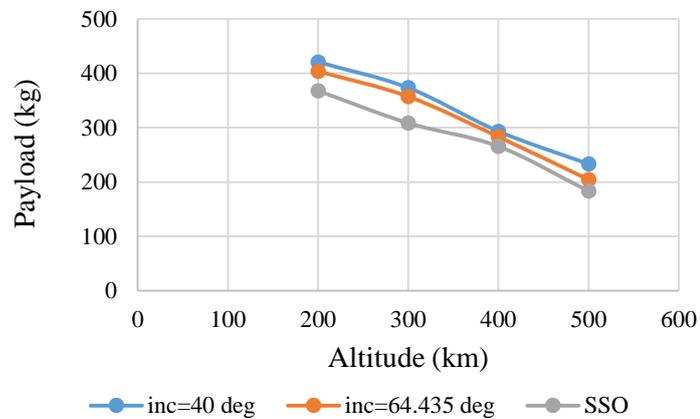


Figure 6-4: Electron payload vs perigee altitude (e=0)

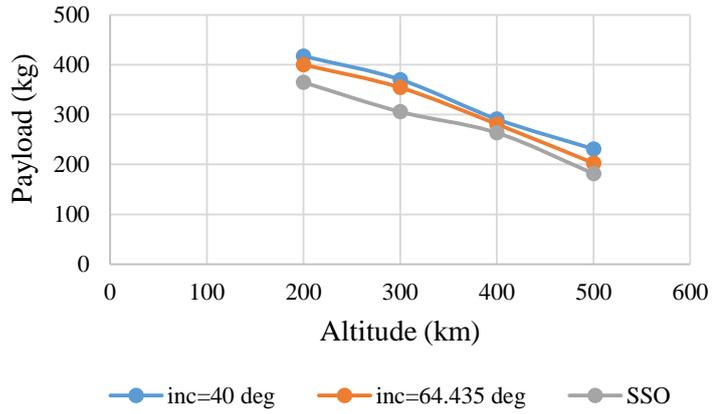


Figure 6-5: Electron payload vs perigee altitude (e=0.005)

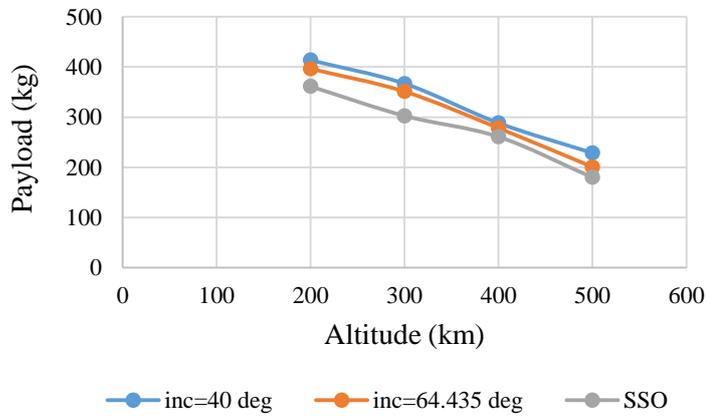


Figure 6-6: Electron payload vs perigee altitude (e=0.01)

Figure 6-7 details the altitude versus range plots of the Electron launch vehicle for the analysed missions to orbits of an inclination of 40° and eccentricity of 0.

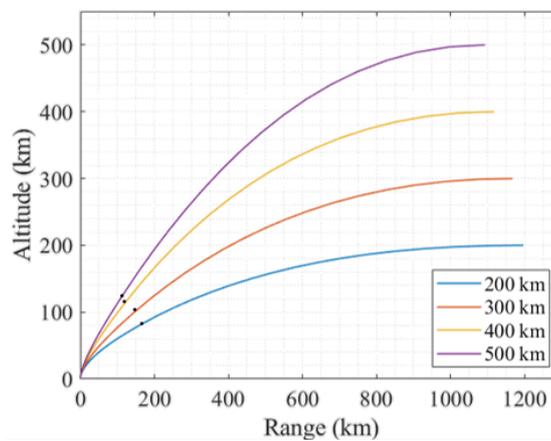


Figure 6-7: Electron altitude vs range (e=0, i=40°)

Figures 6-8 and 6-9 depict the average absolute value of the correlation coefficient of the tangent of the yaw and pitch angle of the second stage thrust direction of the Electron.

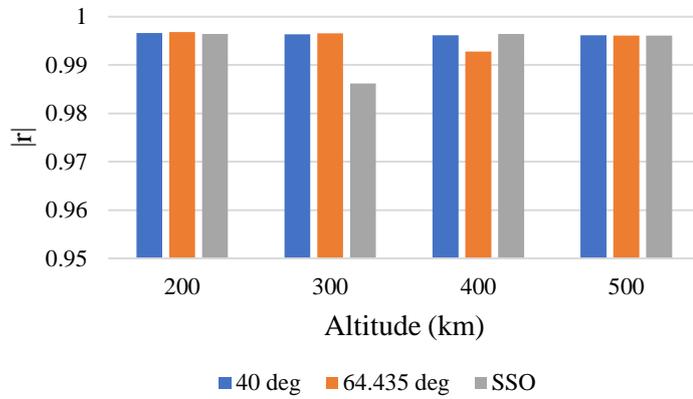


Figure 6-8: Average $|r|$ of Electron's tangent pitch

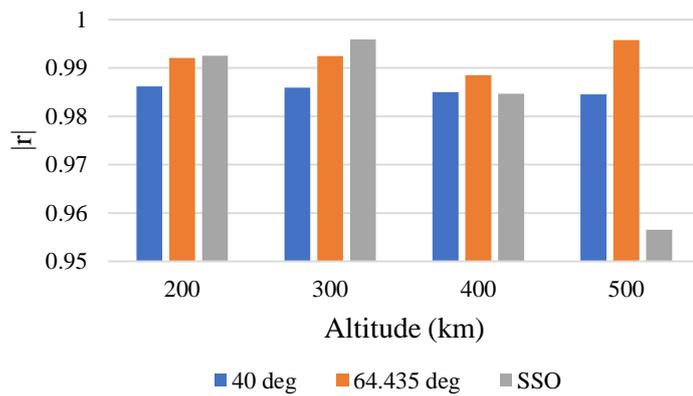


Figure 6-9: Average $|r|$ of Electron's tangent yaw

The simulated results of Rocket Lab's Electron predict that the vehicle is capable of inserting into orbit payloads of 228.70 to 420.85 kg at an inclination of 40°, 200.89 to 403.70 kg at an inclination of 64.435°, and 180.00 to 367.84 kg at SSO, for all eccentricities and altitudes. Due to availability of data, the payload to the circular orbit with a 400 km perigee and 40° inclination was chosen as a point of comparison. The simulated maximum payload to this orbit is 293.42 kg whereas Rocket Lab's (2019) payload user's guide claims a maximum payload of 270 kg. These values are relatively similar. The small difference can be explained to be a result of different launch inclination, wind conditions, and vehicle modelling. Figure 6-6 and Figures C-1 to C-9 show that

the altitude of the launch vehicle is predicted to strictly increase with respect to range. The maximum range achieved in all scenarios was shown to be approximately 1000 km. The average absolute values of the Pearson’s correlation coefficient in Figures 6-8 and 6-9 were found to range between 0.95 and 1. This implies a strong correlation between the simulated control history and the linear tangent law.

6.2. SpaceX’s Falcon 1

The Falcon 1 is another relatively light launch vehicle with a low thrust-to-weight ratio that was designed for the injection of payloads into LEOs. Now discontinued, it was manufactured by SpaceX and made use of a Merlin Engine variant. Table 6-3 details the input parameters used to generate the Falcon 1’s results data. Table 6-4 details the relevant information regarding the structural loads experienced by the Falcon 1 launch vehicle during the analysed missions. Figures 6-10 to 6-12 detail the payload performance of the Falcon 1 launch vehicle.

Table 6-3: Falcon 1 Simulation Parameters

| Parameter Description | Parameter value |
|------------------------------|------------------------|
| m_1 (kg) | 27200 |
| \dot{m}_1 (kg/s) | 126.16 |
| Thrust ₁ (kN) | 417.2 |
| A_e (m ²) | 0.6257 |
| m_2 (kg) | 4540 |
| \dot{m}_2 (kg/s) | 9.6 |
| Thrust ₂ (kN) | 31.75 |

Table 6-4: Falcon 1 Structural Load Parameters

| Structural Load Parameter | Structural Load Parameter Value |
|----------------------------------|--|
| Maximum Q (kPa) | 33.47 |
| Altitude range of max Q (km) | 9.04-11.38 |
| First stage max a_x (g) | 3.23 |
| Second stage max a_x (g) | 5.32 |

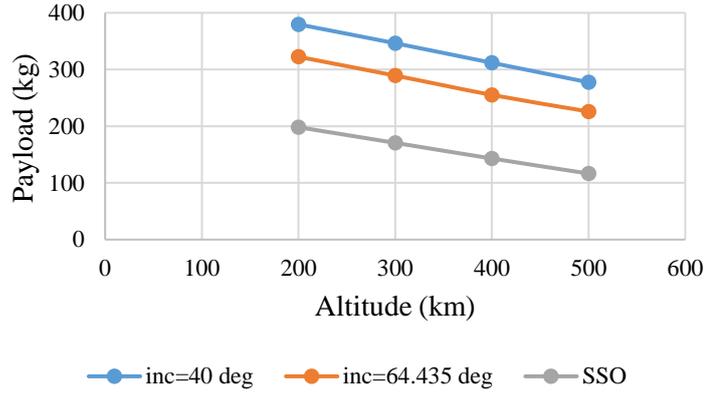


Figure 6-10: Falcon 1 payload vs perigee altitude (e=0)

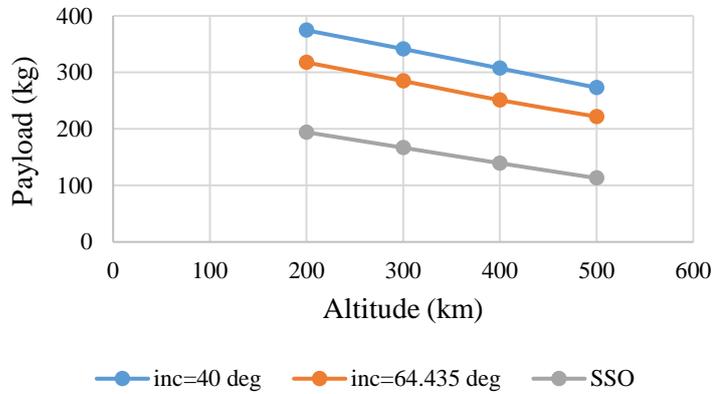


Figure 6-11: Falcon 1 payload vs perigee altitude (e=0.005)

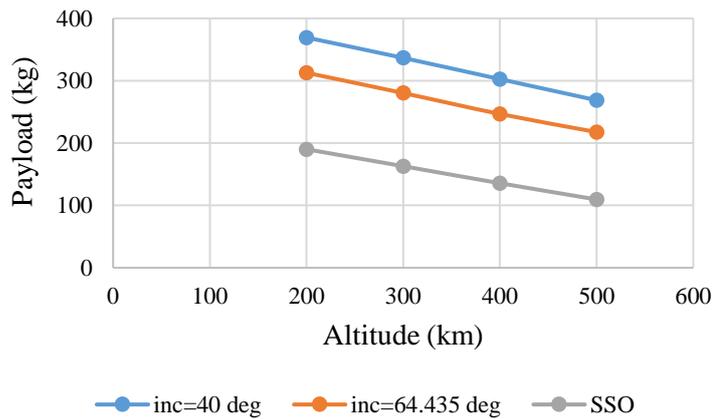


Figure 6-12: Falcon 1 payload vs perigee altitude (e=0.01)

Figure 6-13 details the altitude versus range plot of the Falcon 1 launch vehicle for the analysed missions to orbits of an inclination of 40° and eccentricity of 0.

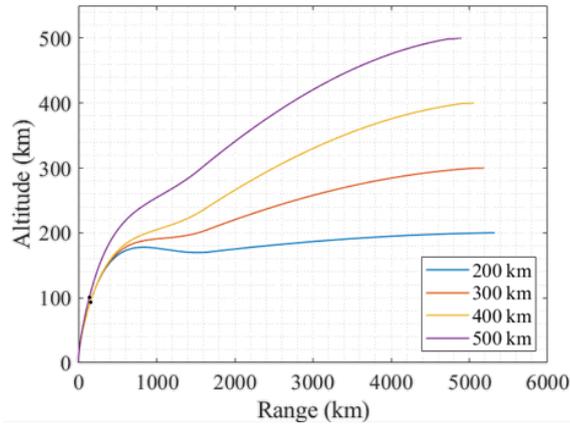


Figure 6-13: Falcon 1 altitude vs range ($e=0, i=40^\circ$)

Figures 6-14 to 6-17 depict the average absolute value of the correlation coefficient of the tangent of the yaw and pitch angle of the second stage thrust direction of the Falcon 1. The Falcon 1 second stage flight made use of two burns.

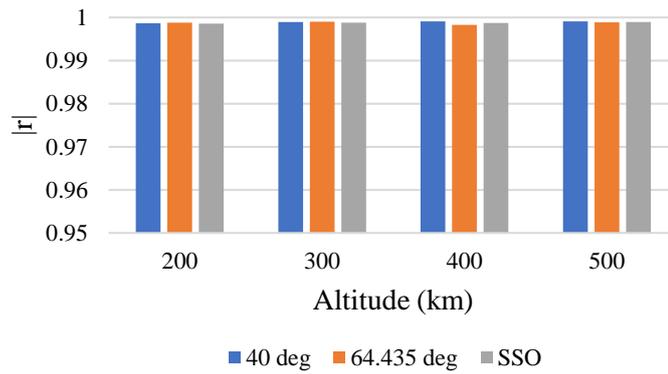


Figure 6-14: Average $|r|$ of Falcon 1's tangent pitch of the first burn

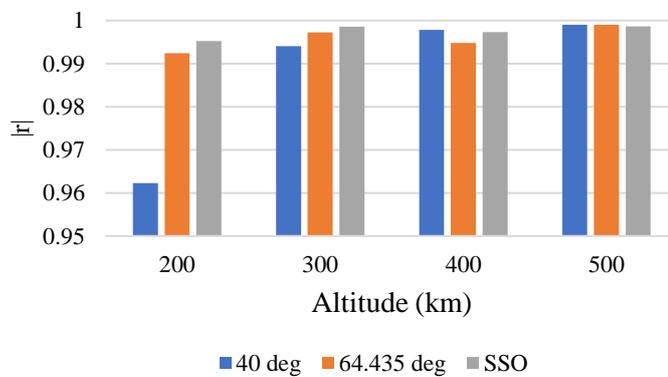


Figure 6-15: Average $|r|$ of Falcon 1's tangent pitch of the second burn

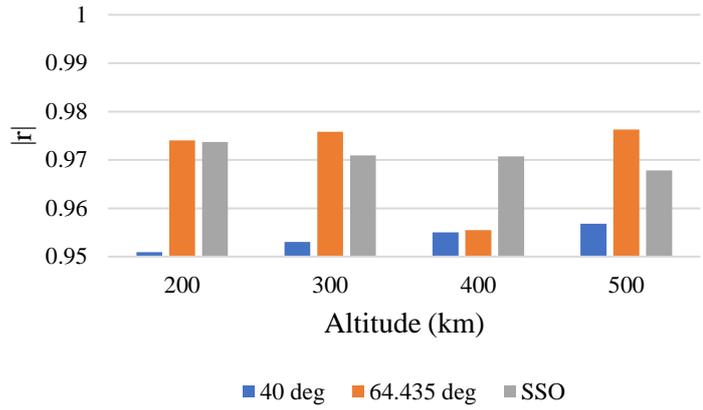


Figure 6-16: Average |r| of Falcon 1's tangent yaw of the first burn

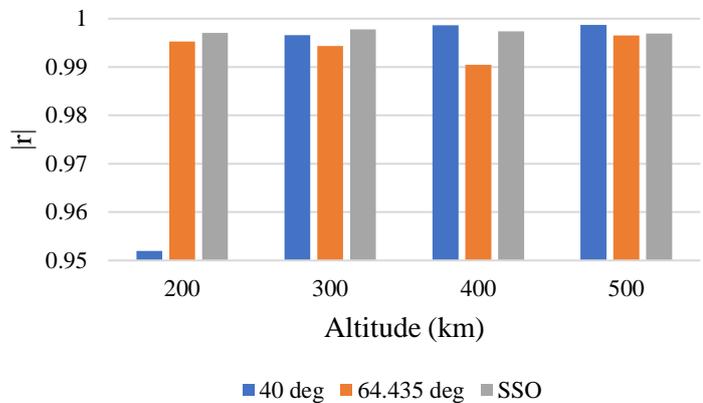


Figure 6-17: Average |r| of Falcon 1's tangent yaw of the first burn second burn

The simulated results of SpaceX's Falcon 1 predict that the vehicle is capable of inserting into orbit payloads of 268.71 to 379.58 kg at an inclination of 40°, 217.66 to 322.38 kg at an inclination of 64.435°, and 109.54 to 198.79 kg at SSO, for all eccentricities and altitudes. Due to availability of data, the payload to the circular orbit with a 300 km perigee and 40° inclination was chosen as a point of comparison. The simulated maximum payload to this orbit is 346.35 kg whereas SpaceX's (2008) payload user's guide claims a maximum payload of 358.33 kg. These values are relatively similar. The small difference can be explained to be a result of different launch inclination, wind conditions, and vehicle modelling. Figure 6-13 and Figures C-10 to C-18 show that all the Falcon 1 trajectories had an inflection point. The altitude of the launch vehicle was strictly increasing with respect to range for trajectories to orbits of perigee 300-500 km. The

altitudes to orbits of 200 km reached a local maximum, following which the altitude decreased to a local minimum. After this local minimum was reached the launch vehicle ascended into the desired orbit with an increasing altitude. The maximum range achieved in all scenarios was approximately 5000 km. The traits of the Falcon 1 launch vehicle’s trajectory is possibly due to its low upper stage thrust to weight ratio and its second stage having an intermediate coast phase. The average absolute value of the Pearson’s correlation coefficient in Figures 6-15 to 6-17 were found to range between 0.95 and 1. This implies a strong correlation with the simulated control history and the linear tangent law.

6.3. SpaceX’s Falcon 9

The Falcon 9 launch vehicle is a relatively heavy launch vehicle with a high thrust-to-weight ratio. It is capable of performing a variety of missions, and is capable of injecting payloads into altitudes greater than 2000 km. The vehicle is manufactured by SpaceX and is powered by a Merlin Engine variant.

Table 6-5 detail the input parameters used to generate the results data of the Falcon 9 launch vehicle.

Table 6-5: Falcon 9 Simulation Parameters

| Parameter Description | Parameter value |
|------------------------------|------------------------|
| m_1 (kg) | 561900 |
| \dot{m}_1 (kg/s) | 2769.57 |
| Thrust ₁ (kN) | 8460 |
| A_e (m ²) | 0.375 |
| m_2 (kg) | 116000 |
| \dot{m}_2 (kg/s) | 281 |
| Thrust ₂ (kN) | 981 |

Table 6-6 details the relevant information regarding the structural loads experienced by the Falcon 9 launch vehicle during the analysed missions.

Table 6-6: Falcon 9 Structural Load Parameters

| Structural Load Parameter | Structural Load Parameter Value |
|------------------------------|---------------------------------|
| Maximum Q (kPa) | 39.04 |
| Altitude range of max Q (km) | 11.24-11.53 |
| First stage max a_x (g) | 3.57 |
| Second stage max a_x (g) | 4.42 |

Figures 6-18 to 6-20 detail the payload performance of the Falcon 9 launch vehicle for the analysed missions.

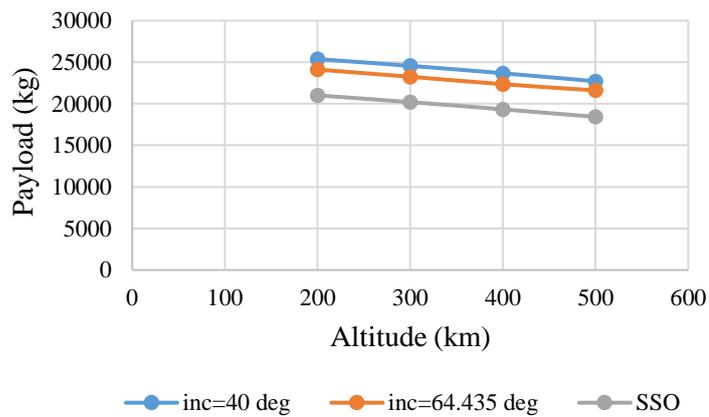


Figure 6-18: Falcon 9 payload vs perigee altitude (e=0)

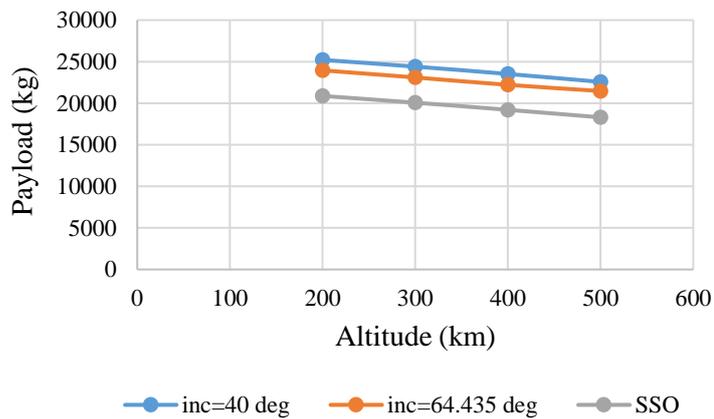


Figure 6-19: Falcon 9 payload vs perigee altitude (e=0.005)

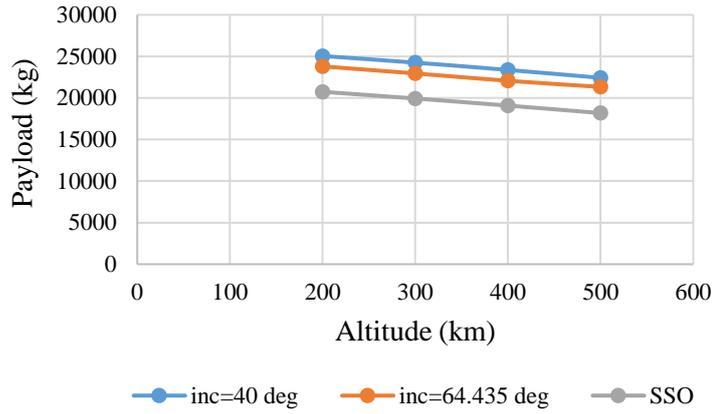


Figure 6-20: Falcon 9 payload vs perigee altitude (e=0.01)

Figure 6-21 details the altitude versus range plot of the Falcon 9 launch vehicle for the analysed missions to orbits of an inclination of 40° and eccentricity of 0.

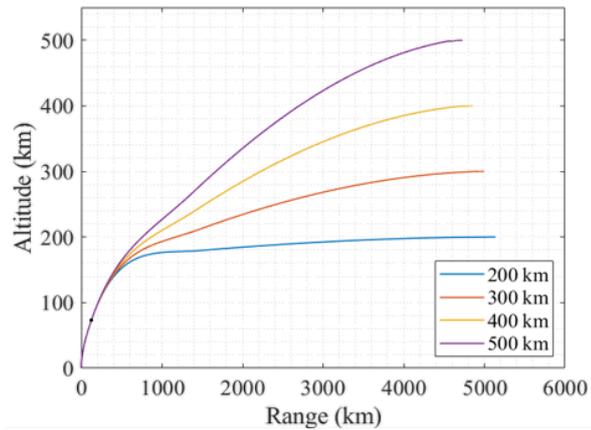


Figure 6-21: Falcon 9 altitude vs range (e=0, i=40°)

Figures 6-22 to 6-25 depict the average absolute value of the correlation coefficient of the tangent of the yaw and pitch angle of the second stage thrust direction of the Falcon 9. The Falcon 9 second stage flight made use of two burns.

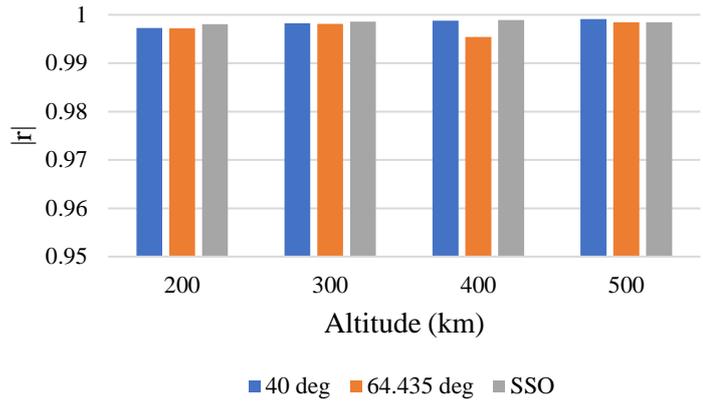


Figure 6-22: Average $|r|$ of Falcon 9's tangent pitch of the first burn

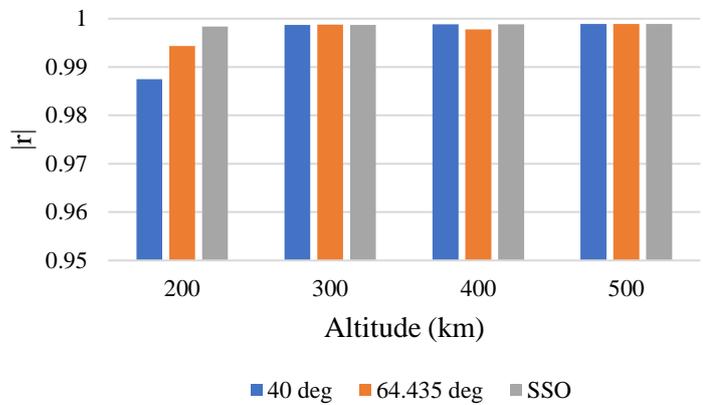


Figure 6-23: Average $|r|$ of Falcon 9's tangent pitch of the second burn

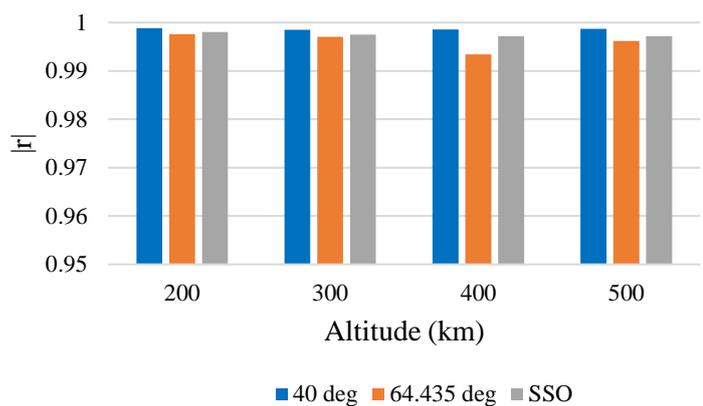


Figure 6-24: Average $|r|$ of Falcon 9's tangent yaw of the first burn

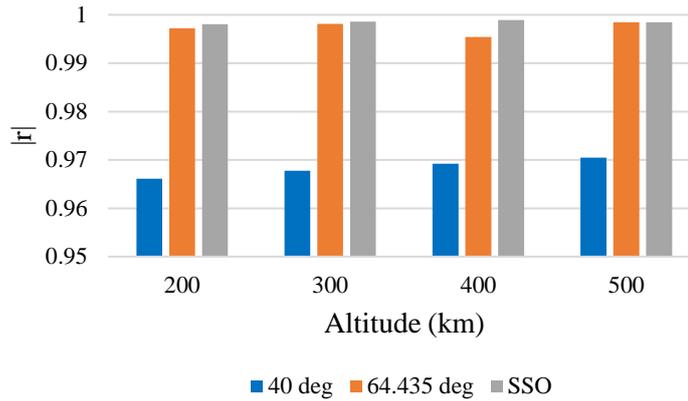


Figure 6-25: Average $|r|$ of Falcon 9's tangent yaw of the second burn

The simulated results of SpaceX's Falcon 9 predict that the vehicle is capable of inserting into orbit payloads of 22421.08 to 25379.46 kg at an inclination of 40°, 21328.05 to 24112.39 kg at an inclination of 64.435°, and 18180.46 to 21016.64 kg at SSO, for all eccentricities and altitudes. According to SpaceX (2022), the Falcon 9 is capable of transporting payloads of 22800 kg into LEO. The obtained payload data is relatively close to this value. Figures 6-21 and Figures C-19 to C-27 show that the altitude of the launch vehicle is predicted to strictly increase with respect to range, and that all trajectories exhibit an inflection point. The maximum range achieved in all scenarios was approximately 5000km. The average absolute values of the Pearson's correlation coefficient in Figures 6-22 to 6-25 were found to range between 0.95 and 1. This implies a strong correlation with the simulated control history and the linear tangent law.

6.4. ASReG's CLV

The CLV launch vehicle has been conceptualised by ASReG as a light launch vehicle designed for the injection of small satellites into LEOs from South African soil, and in particular, from the Denel Overberg Test Range in the country's Western Cape province. The vehicle is to be propelled by ASReG's SAFFIRE engines, which are currently under development. This concept intends to utilize resources, technologies and manufacturing processes available in South Africa.

The CLV vehicle is a two-stage liquid-propellant rocket which makes use of liquid oxygen as the oxidiser and Jet A-1 kerosene (commercial jet fuel) as the fuel. The booster stage is powered by

a cluster of nine ablatively cooled SAFFIRE engines, while a single engine powers the second stage. The engines use electrically-driven propellant pumps and stainless steel propellant tanks. Table 6-7 details the input parameters used to generate the results data of the CLV launch vehicle. Table 6-8 details the relevant information regarding the structural loads experienced by the CLV launch vehicle during the analysed missions.

Table 6-7: CLV Simulation Parameters

| Parameter Description | Parameter value |
|--------------------------|-----------------|
| m_1 (kg) | 19491.245 |
| \dot{m}_1 (kg/s) | 90 |
| Thrust ₁ (kN) | 304.695 |
| A_e (m ²) | 0.5547 |
| m_2 (kg) | 2684.312 |
| \dot{m}_2 (kg/s) | 10 |
| Thrust ₂ (kN) | 33.855 |

Table 6-8: CLV Structural Load Parameters

| Structural Load Parameter | Structural Load Parameter Value |
|------------------------------|---------------------------------|
| Maximum Q (kPa) | 39.53 |
| Altitude range of max Q (km) | 10.32-11.28 |
| First stage max a_x (g) | 3.36 |
| Second stage max a_x (g) | 7.35 |

Figures 6-26 to 6-28 detail the payload performance of the CLV launch vehicle for the analysed missions

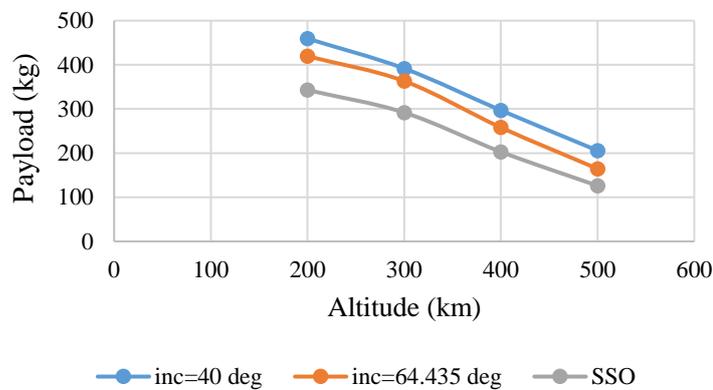


Figure 6-26: CLV payload vs perigee altitude (e=0)

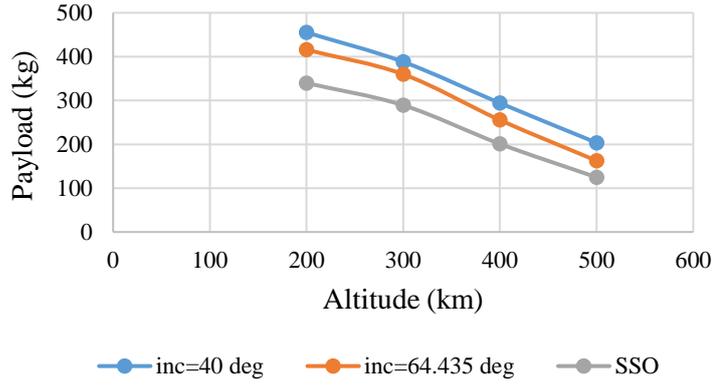


Figure 6-27: CLV payload vs perigee altitude (e=0.005)

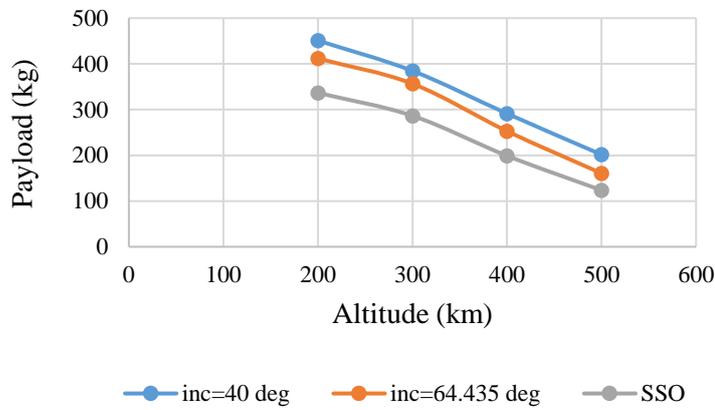


Figure 6-28: CLV payload vs perigee altitude (e=0.01)

Figure 6-29 details the altitude versus range plot of the Falcon 1 launch vehicle for the analysed missions to orbits of an inclination of 40° and eccentricity of 0.

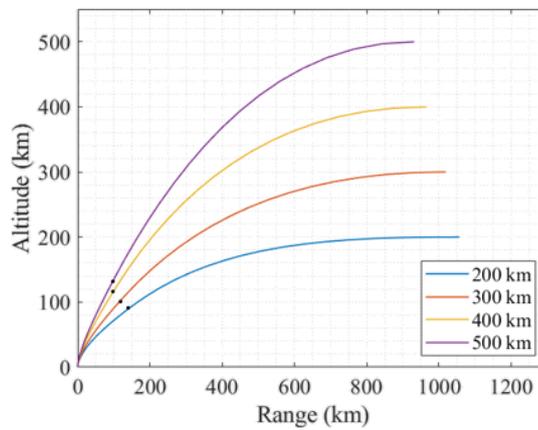


Figure 6-29: CLV altitude vs range (e=0, i=40°)

Figures 6-30 and 6-31 depict the average absolute value of the correlation coefficient of the tangent of the yaw and pitch angle of the second stage thrust direction of the CLV.

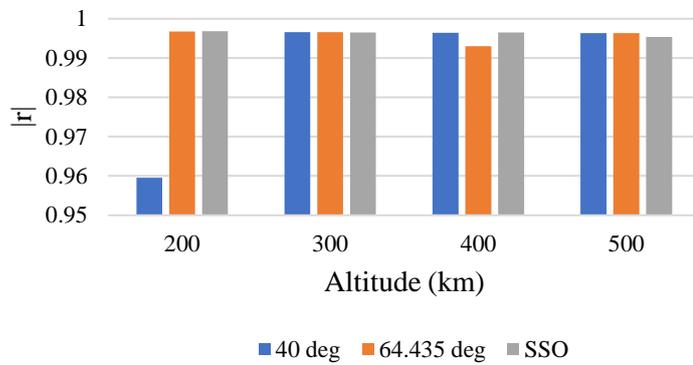


Figure 6-30: Average $|r|$ of CLV's tangent pitch

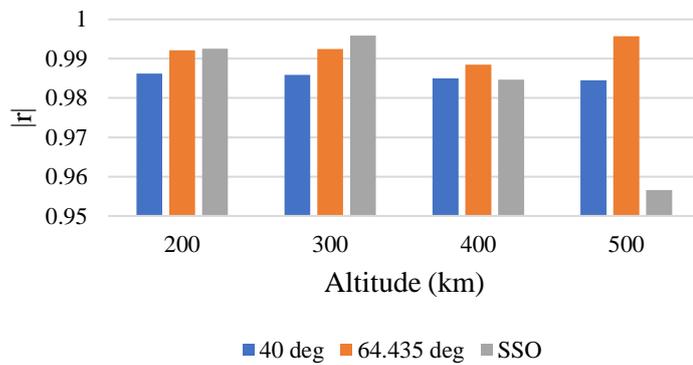


Figure 6-31: Average $|r|$ of CLV's tangent yaw

The simulated results of ASReG's CLV predict that the vehicle is capable of inserting into orbit payloads of 201.48 to 459.47 kg at an inclination of 40°, 160.59 to 419.57 kg at an inclination of 64.435°, and 123.55 to 342.92 kg at SSO, for all eccentricities and altitudes. Figure 6-29 and Figures C-28 to C-36 show that the altitude of the launch vehicle is predicted to strictly increase with respect to range. The maximum range achieved in all scenarios was shown to be approximately 1000km. The average absolute values of the Pearson's correlation coefficient in Figures 6-30 and 6-31 were found to range between 0.95 and 1. This implies a strong correlation with the simulated control history and the linear tangent law.

6.5. Launch Vehicle Comparison

This section serves to compare the trajectories and performance of the analysed launch vehicles. Figures 6-32 and 6-33 compare the pitch rate during the initial pitchover manoeuvre of all the analysed launch vehicles.

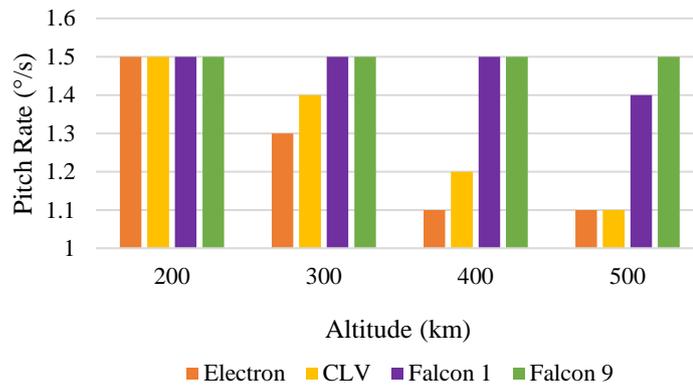


Figure 6-32: Pitch rate vs orbital altitude

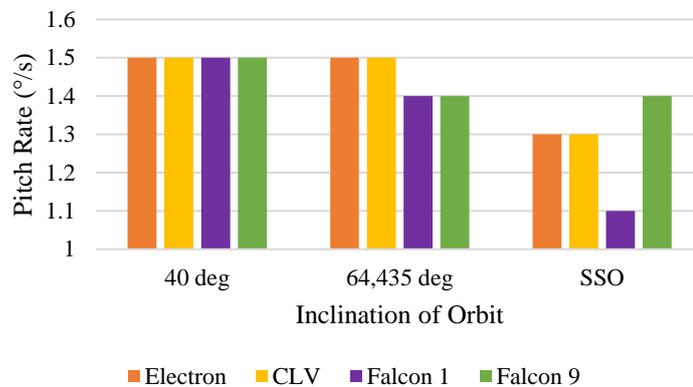


Figure 6-33: Pitch rate vs orbital inclination

The initial pitchover pitch rate did not vary with the studied eccentricities so only inclination and perigee were used as independent variables. The value decreased as orbital altitude and inclination increased. This is as the vehicles need to pitch less in order to obtain a reasonable terminal first stage altitude and velocity as altitude and inclination increases. If the vehicles were to pitch more, their terminal altitudes would be sub-optimal and too low. Relative to Rocket lab's Electron and ASReG's CLV, SpaceX's Falcon 1 and Falcon 9 pitch less as the orbital perigee increases and

pitch more as inclination increases. This may be due to both of the SpaceX launch vehicles making use of an intermediate coast phase, whereas the Electron and CLV do not.

Figures 6-34 to 6-36 compare the time to orbit of all the analysed launch vehicles.

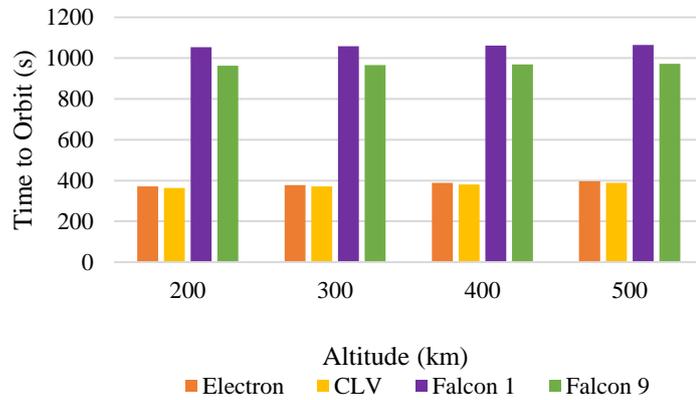


Figure 6-34: Time to orbit vs orbital altitude

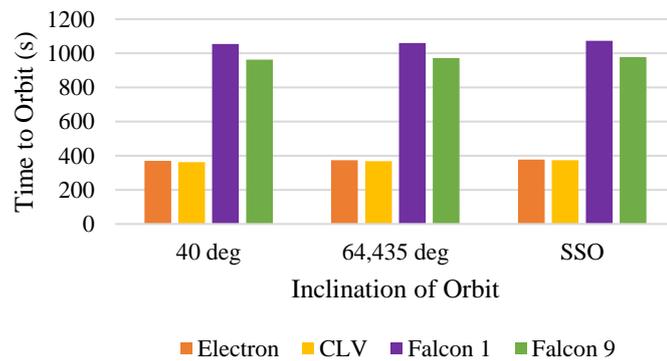


Figure 6-35: Time to orbit vs orbital inclination

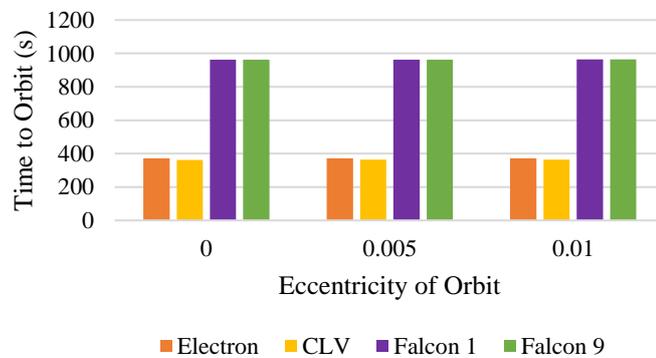


Figure 6-36: Time to orbit vs orbital eccentricity

Increasing orbital perigee, inclination and eccentricity increases the time to orbit. All the vehicles have different engine mass flow rates so time to orbit is not necessarily a fair means to compare performance. Unlike Rocket lab's Electron and ASReG's CLV, SpaceX's Falcon 1 and Falcon 9 coasted during the 2nd stage trajectory, hence their time to orbit is not necessarily proportional to performance and they cannot be compared to the Electron and the CLV on this basis. However both the Falcon 1 and Falcon 9 have coast phases of approximately 500 s. The Falcon 1 required significantly more time to reach the same orbits relative to the Falcon 9, yet (as can be seen in Figures 6-34 to 6-36) the Falcon 9 was capable of transporting payloads almost fifty times larger than the Falcon 1. This seems reasonable as both the Falcon 9 stages have a significantly greater thrust to weight ratio than the Falcon 1 stages. The CLV and electron needed fairly similar amounts of time to orbit, and have relatively similar performances, both being capable of inserting payloads in the range of 100-400 kg into the studied orbits.

The payload graphs in subsections 6.1.-6.4. all possess a Pearson's correlation coefficient less than -0.99, therefore they all depict a virtually linear decrease in the payload capacity as orbit altitude is increased, for all launch vehicles. Additionally, an increase in inclination and eccentricity individually resulted in a decrease in payload to orbit. These trends are expected and typical characteristics of payload to orbit trends of launch vehicles.

Both the CLV and Electron found no improvements in performance when allowing for a coast phase during the second stage trajectory. In the context of the CLV vehicle, this may suggest that the complexities associated with providing its second stage with an engine restart capability could be avoided without penalty. Both the Falcon 1 and Falcon 9 are capable of restarting their engines. Making use of this and allowing for a coast phase during their second stage trajectory improved their individual performance, thus justifying their restart capability.

The altitude versus range plots also show the effects of the altitude, inclination and eccentricity of the orbit on the final range of the launch vehicles' trajectories at the moment of orbital injection. A decrease in orbit altitude, an increase in inclination and an increase in eccentricity each

individually resulted in an increase in the range travelled by the studied launch vehicles. The results in question indicate that the most significant factor on range is the altitude of the orbit.

It should be noted, however, that the eccentricities studied here were small, and that the chosen injection point was the perigee point. For the case of larger eccentricities and/or non-perigee injection, an increase in eccentricity may result in a greater effect on range.

Both the CLV and Electron altitude versus range plots showed a monotonic increase for all ascent-to-orbit trajectories. There were no inflection points in these plots. This suggests that trajectories to LEO travelled by small launch vehicles with relatively high thrust to weight ratio follow this trend.

The Falcon 1 and Falcon 9 altitude versus range plots shared similar trajectory shapes to each other, in that in almost all cases, these plots each featured an inflection point. The exceptions to this are the Falcon 1 trajectories to orbits of 200 km. The altitude along exceptions decreased after reaching a local maximum, after which a local minimum was reached. The local minimum was less than 20 km lower than the local maximum. Following the local minimum, the vehicle ascended to the desired orbit.

The highest maximum dynamic pressure of 44.44 kPa was predicted for the Electron launch vehicle, whereas the lowest maximum dynamic pressure of 33.47 kPa was predicted for the Falcon 1 launch vehicle. The largest altitude of maximum dynamic pressure of 11.53 km was predicted for the Falcon 9 launch vehicle, while the smallest altitude of maximum dynamic pressure of 8.64 km was predicted for the Electron launch vehicle. Thus the studied launch vehicles (launched from OTR) experienced maximum dynamic pressure between 8.64 km and 11.53 km. This range lies within the general altitude range reflected in literature, between 7 km and 15 km.

CLV was predicted to experience a maximum dynamic pressure of 39.53 kPa. It experienced maximum dynamic pressure at altitudes between 10.32 km and 11.28 km. The maximum dynamic pressure experienced is on the higher end of the range of the predicted maximum dynamic

pressure values of the analysed launch vehicles. The altitudes at which maximum dynamic pressure was experienced is within the altitude range of the analysed launch vehicles and the altitude range obtained from literature.

7. CONCLUSION

7.1. Optimal Control Solver

The developed optimal control solver made effective use of Hermite-Simpson direct collocation methods to solve the second stage of the analysed ascent-to-orbit trajectories. Initially, all derivative information necessary for the solver to function was estimated using MATLAB's dense finite-difference estimation methods. Estimating this derivative information with MATLAB's dense methods led to slow convergence, and in the case of the ascent-to-orbit problem, it would on occasion fail to converge to a solution. Hence provision of first-order analytical derivative information was attractive in the analysis of ascent-to-orbit trajectories due the improvements in robustness and solver speed it provides.

The developed formula for first-order analytical derivative information, once implemented, greatly improved solver time and robustness, and was relatively simple to implement. The analytical derivatives were tested by MATLAB's derivative test, which compares numerical estimates of the derivative information to the analytical derivatives. The analytical derivatives passed this test, further validating the developed formulae. The improved robustness of the solver allowed for initial guesses which were formed by a linear interpolation between the boundary points of the trajectory.

The examples solved in Chapter 4 validated the solver. Overall, the validation examples were solved in a reasonable time, with a good degree of accuracy. The first validation problem was used as a test problem so as to show that the solver can solve a simple and linear optimal control problem with initial and terminal boundary constraints. The developed solver computed a highly accurate solution in a relatively short time period, proving the solver is capable of solving such problems. The orbital transfer problem proved the solver can work effectively for the case of nonlinear 2-DOF flight under vacuum conditions. The Delta III validation example required 152.26 seconds in order to obtain a solution with an error estimate of 3.6315×10^{-5} and an

objective function accuracy of 3.984×10^{-4} %. The accuracy of this solution is reasonable, however the solution time is relatively long.

One factor contributing to this observation was that the problem in question is more complex than the other problems considered in this work. Furthermore, the solver makes use of manual scaling, and it is possible there may have been a more optimal scale for the decision variables. Having said this, there is no set rule to determine the optimal scale for variables in an optimisation problem. A second factor was that the method employed by the solver to generate new grids is not optimal. Grid generation using mixed-integer programming would potentially have allowed for coarser grid representations which output lower error estimates.

It is most likely, however, that the primary contributor to the slow solution time was the NLP solver used by the optimal control solver. As mentioned in the literature review, MATLAB's SQP algorithm estimates second derivative information using dense finite differencing methods. It has been consistently shown in the field of optimal control that taking advantage of the sparsity of the derivative information, and determining sparse finite difference estimates, greatly improves the robustness and speed of optimal control solvers which make use of NLP techniques. MATLAB's SQP also requires all first and second derivative information to be supplied or none at all. This greatly restricts users from employing analytical derivative formulae for extremely complex problems such as aero-assisted re-entry trajectories.

The solver performance could have been greatly improved if it made use of an NLP solver that both allows for the use of sparse finite difference estimates of the second derivative information, and does not require all derivative information of a particular order to be supplied. An NLP solver that meets both of these requirements, such as SNOPT, would allow for efficient trajectory analysis in which complex phases such as atmospheric flight are optimised. Nevertheless analytical first derivatives greatly improved the performance of the solver, and where possible, it is advisable to take advantage of them.

7.2. Results

The work detailed in Chapter 6 served to meet the final two research objectives: simulating the flight to predefined orbits of three existing carrier rockets as well as ASReG's proposed carrier, CLV, and then in each instance, analysing the associated flight performance and structural loads.

Payload is typically the main criteria by which the performance of a launch vehicle's trajectory to a particular orbit is valued. Of the orbits considered, the least propellant-expensive orbit to ascend to was found to be the circular 200 km orbit with an inclination of 40 degrees. The most propellant-expensive orbit to attain was determined to be the 500 km SSO orbit with an eccentricity of 0.01. The Rocket Lab Electron vehicle was capable of transporting a payload of 420.85 kg to the least expensive orbit and a payload of 180.00 kg to the most expensive orbit. The Falcon 1 was capable of transporting a payload of 379.58 kg to the least expensive orbit and a payload of 109.54 kg to the most expensive orbit. The Falcon 9 was capable of transporting a payload of 25379.46 kg to the least expensive orbit and a payload of 18180.45 kg to the most expensive orbit. The CLV was capable of transporting a payload of 459.47 kg to the least expensive orbit and a payload of 123.45 kg to the most expensive orbit.

The maximum dynamic pressure and the altitude at which the dynamic pressure was predicted to be experienced were relatively similar for all launch vehicles. The studied launch vehicles (launched from OTR) experienced maximum dynamic pressure between 33.47 kPa and 44.44 kPa. The studied launch vehicles experienced maximum dynamic pressure at altitudes between the general altitude range reflected in literature, between 7 km and 15 km.

It should be noted, that as this was a 3-DOF study, rotational loads were not directly considered. During an ideal gravity turn, a launch vehicle will experience no rotation loads due to aerodynamic forces, however it is still an important factor during the initial pitchover manoeuvre.

One method to reduce rotational loads would be to plan flight missions such that flight paths would be in a similar direction to the current wind. According to Denel (2022), summer winds at OTR are typically from the east and south-east, and winter winds at OTR are typically from the

north-west and south-west. Thus to reduce rotational loads, eastern launches could be done in summer to take advantage of the south-east winds, and polar launches could be done in winter to take advantage of north-west winds. However it could prove impractical to only allow seasonal launches, so this method is one that should be used sparingly.

The bar graphs in Chapter 6 depict the Pearson's correlation coefficient of the tangent function of the yaw and pitch angles of the thrust direction for the various trajectories studies. These graphs show that these tangent functions behave highly linearly relative to time, with the lowest Pearson correlation coefficient value being 0.951. This is fairly consistent with the linear tangent steering law and still held true for multiple burn trajectories. The accuracy of this law could be improved by splitting the trajectory into phases, each with its own distinct steering law. The data further supports the usage of a parametric linear tangent steering law in order to obtain solutions to ascent-to-orbit problems using indirect methods.

As the analysed modern launch vehicles were capable of transferring competitive payloads to orbit, it can be claimed that OTR can be used as a competitive launch site into various LEO. Additionally, the studied trajectories exist solely over the ocean, hence the instantaneous impact points of vehicles launched from OTR would be in the ocean, instead of any populated regions.

A launch vehicle, such as the CLV might be launched as follows. A customer would request a payload to be taken to a particular orbit within a particular time period. Wind estimates within this time period would then be obtained, allowing for an optimal trajectory to this orbit to be determined. If a trajectory can be determined, then the mission is possible, and a specific launch window can be chosen. Prior to launch, a multidimensional statistical analysis would be performed to determine a safety footprint, outside of which the launch vehicle has an extremely low probability of impacting as a result of a vehicle anomaly occurring. Thereafter, maritime and air traffic notifications would be issued on the basis of this safety footprint to communicate the marine and airspace exclusion zones that would be in force during the launch window. The vehicle would subsequently be prepared by engineers and technicians and then launched.

7.3. Future Work

The performance of the optimal control solver developed here was greatly hindered by the inability of MATLAB's SQP solver to take advantage of sparsity of the second derivative information. Making use of an SQP solver which employs sparse methods, such as SNOPT, would significantly improve the robustness and speed of the optimal control solver. Improving the performance of the optimal control solver will allow reliable addition of more complexities, such as additional degrees of freedom, to the ascent-to-orbit problem. Additionally, utilising a different optimal control solver, such as SNOPT, would remove the restriction of providing either all or none of the analytical first order derivative information, as most sparse NLP solvers accept either none, some or all of the analytical first derivative information.

In this work the initial pitchover manoeuvre was implemented using a linear pitch control law with discrete parameters. There are other variations of pitch control laws used for the initial pitchover. Attempting other pitch control laws and pitch parameters could possibly improve the performance of the analysed launch vehicles. The calculated launch azimuth is not necessarily the optimal launch azimuth, rather the optimal launch azimuth is relatively close to this value. Hence defining the ascent-to-orbit problem such that the launch azimuth is an optimization variable, with possible values being within a close proximity to the calculated "optimal" launch azimuth, could possibly improve the achievable performance of the launch vehicles studied here.

A significant limiting factor of a launch vehicle's performance is its structural capacity, which limits the aerodynamic loading and maximum dynamic pressure that it can experience during flight. The structural limitations of a vehicle are of the greatest concern during the atmospheric ascent phase, where air density is still significant. The dynamic pressure constraint value used in this work is a generic value obtained from Coskun (2014), but the true dynamic pressure constraint for each individual launch vehicle is based on its structural capacity, and is potentially higher than this value. A more lenient structural constraint could potentially increase the achievable performance of the launch vehicles studied here as this would allow for a greater variation in the vertical rise and initial pitchover parameters. Additionally, future studies related

to the CLV launch vehicle should evaluate the impact of wind parameters on launch vehicle structural integrity.

In the future, the developed solver could be utilised to analyse payload injection into all manner of orbits, in addition to re-entry trajectories and orbital transfers. In the case of orbital injection trajectories, the solver can be set such that the injection point is a free variable, which should potentially improve the optimal trajectory solution. The solver could be adjusted to include six degrees of freedom, thus allowing for rotational load constraints to be imposed. The solutions of the solver, the linear tangent steering law and the first order derivative information could be utilised in order to use shooting methods. These methods, particularly multiple shooting, are better suited for final mission design as they improve solution accuracy.

REFERENCE LIST

- ASReG, 2019. *SAFFIRE Brochure*. [Online]
Available at: aerospace.ukzn.ac.za/Libraries/Documents/SAFFIRE_Brochure.sflb.ashx
[Accessed 19 January 2020].
- Astos Solutions, 2020. *ASTOS*. [Online]
Available at: <https://www.astos.de/products/astos>
[Accessed 02 April 2020].
- Betts, J. T., 1998. A Survey of Numerical Methods for Trajectory Optimisation. *Journal of Guidance, Control, and Dynamics*, 21(2), pp. 193-207.
- Betts, J. T., 2010. *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*. 2nd ed. Philadelphia: SIAM.
- Brugge, N., 2021. *Rocket Lab Ltd.*. [Online]
Available at: http://www.b14643.de/Spacerockets_3/Rocket-Lab/Description/Frame.htm
[Accessed 5 March 2021].
- Bryce Space and Technology, 2022. *Smallsats by the Numbers*. [Online]
Available at: <https://brycetech.com/reports/report-documents>
[Accessed 22 May 2022].
- Bryson, A. E. & Ho, Y.-C., 1975. *Applied Optimal Control*. Washington D.C.: Hemisphere Publishing Corporation.
- Burden, R. L. & Faires, J. D., 2001. *Numerical Analysis*. 7th ed. s.l.:Wadsworth Group.
- Calise, A. J., 2013. Single Perturbations in Flight Mechanics. In: A. Miele & A. Salvetti, eds. *Applied Mathematics in Aerospace Science and Engineering*. New York City: Springer Science and Business Media, pp. 115-132.

COŞKUN, E. C., 2014. *Multistage Launch Vehicle Design with Thrust Profile and Trajectory Optimisation*. Ankara: The Graduate School of Natural and Applied Sciences of Middle East Technical University .

Curtis, H. D., 2005. *Orbital Mechanics for Engineering Students*. Oxford: Elsevier Butterworth-Heinemann.

Damp, L., 2017. *Safe and simple launch access to polar and Sun synchronous orbits*. [Online] Available at: <https://www.openforum.com.au/safe-simple-launch-access-to-polar-and-sun-synchronous-orbits/>
[Accessed 12 May 2022].

Denel, 2022. *Location & Test Arena*. [Online] Available at: <http://www.denelotr.co.za/products--services/location--test-arena>
[Accessed 28 02 2022].

ESA, 2020a. *Types of orbits*. [Online] Available at: https://www.esa.int/Enabling_Support/Space_Transportation/Types_of_orbits#SSO
[Accessed 11 April 2020].

ESA, 2020b. *Polar and Sun-synchronous orbit*. [Online] Available at: https://www.esa.int/ESA_Multimedia/Images/2020/03/Polar_and_Sun-synchronous_orbit
[Accessed May 12 2022].

Essa, P., 2013. *Correlation Coefficient: Simple Definition, Formula, Easy Steps*. [Online] Available at: <https://www.statisticshowto.com/probability-and-statistics/correlation-coefficient-formula/>
[Accessed 12 May 2022].

- Federici, L. et al., 2019. Integrated Optimisation of Ascent Trajectory and SRM design of Multistage Launch Vehicles. In: *29th AIAA Space Flight Mechanics Meeting*. Ka'anapali: s.n., p. 3953.
- Forsgen, A., 2006. On Warm Starts for Interior Methods. In: F. Ceragioli, et al. eds. *System Modeling and Optimisation*. Boston: Springer-Verlag, pp. 51-66.
- Gill, P. E., 2008. *User's Guide for SNOPT Version 7: Software for Large-Scale Nonlinear Programming*. San Diego: Stanford University.
- Grüne, L., 2020. Numerical Methods for Nonlinear Optimal Control Problems. In: B. J. & S. T., eds. *Encyclopedia of Systems and Control*. s.l.:Springer International Publishing.
- Haeussermann, W., 1970. *Description and Performance of the Saturn Launch Vehicle's Navigation, Guidance, and Control System*. Washington D.C.: NASA.
- Hall, N., 2015. *Liquid Rocket Engine*. [Online]
Available at: <https://www.grc.nasa.gov/www/k-12/airplane/lrockth.html>
[Accessed 25 March 2020].
- Holsapple, R., Iyer, R. & Doman, D., 2004. New, Fast Numerical Method for Solving Two-Point Boundary-Value Problems. *Journal of Guidance Control and Dynamics*, 27(2), pp. 301-304.
- HP, 2022. *HP Notebook - 15-da0031ni Product Specifications*. [Online]
Available at: <https://support.hp.com/za-en/document/c06204008>
[Accessed 14 January 2022].
- Jansch, C., Schnepfer, K. & Well, K. H., 2013. Multi-Phase Trajectory Optimisation Methods with Applications to Hypersonic Vehicles. In: A. Miele & A. Salvetti, eds. *Applied Mathematics in Aerospace Science and Engineering*. New York City: Springer Science and Business Media, pp. 133-164.
- Kasonsuwanx, K., Wannawichian, S. & Kird, T., 2017. Observation of GEO Satellite Above Thailand's Sky. *Journal of Physics Conference Series: : Conf. Ser. 901 012009*.

Kawajir, Y., Laird, C., Vigerske, S. & Wächter, A., 2015. *Introduction to IPOPT: A tutorial for downloading, installing, and using IPOPT*. [Online]

Available at: <https://projects.coin-or.org>stable>Ipopt>doc>

[Accessed 31 March 2020].

Kodheli, O. et al., 2020. Satellite Communications in the New Space Era. *IEEE COMMUNICATIONS SURVEYS & TUTORIALS*.

Kopp, R. E. & Moyer, H. G., 1965. Necessary Conditions for Singular Extremals. *AIAA*, 3(8), pp. 1439-1444.

Lawden, D. F., 1975. *Analytical Methods of Optimisation*. Edinburgh: Scottish Academic Press Ltd..

Leitmann, G., 1981. *The Calculus of Variations and Optimal Control*. New York City: Plenum Press.

Lu, P., Sun, H. & Tsai, B., 2003. Closed-loop Endo-Atmospheric Ascent Guidance. *Control and Dynamics*, 26(2), pp. 283-294.

Mann, A., Pultarova, T. & Howell, E., 2022. *SpaceX Starlink internet: Costs, collision risks and how it works*. [Online]

Available at: <https://www.space.com/spacex-starlink-satellites.html>

[Accessed 29 August 2022].

Marrdonny, M. & Mobed, M., 2009. A guidance algorithm for launch to equatorial orbit. *Aircraft Engineering and Aerospace Technology: An International Journal*, 81(2), pp. 137-148.

Massimiani, M., 2015. *IOD/IOV LAUNCHERS ANALYSIS*. [Online]

Available at:

<https://ec.europa.eu/research/participants/documents/downloadPublic?documentIds=080166e5a1c9afe2&appId=PPGMS>

[Accessed 21 March 2021].

Mathworks Inc., 2020. *Choosing the Algorithm*. [Online]

Available at: https://www.mathworks.com/help/optim/ug/choosing-the-algorithm.html?s_tid=mwa_osa_a

[Accessed 01 April 2020].

NASA, 2020. *Debris Mitigation*. [Online]

Available at: <https://orbitaldebris.jsc.nasa.gov/mitigation/>

[Accessed 23 March 2020].

Nazim, M., 2020. *Sun-synchronous Satellites*. [Online]

Available at: https://patnauniversity.ac.in/e-content/social_sciences/geography/MAGeog78.pdf

[Accessed 12 April 2020].

Newcomb, J. F. & Hampshire, W. . F., 1970. *PARAMETRIC DESIGN OF INTERPLANETARY AND ORBITAL TRAJECTORIES WITH EXAMPLES FOR ARS 1973, 1975, AND 1977 OPPORTUNITIES*, Hampton: NASA.

Niederstrasser, C., 2018. *Small Launch Vehicles – A 2018 State of the Industry Survey*. Logan, 32nd AIAA/USU Conference on Small Satellites.

Niederstrasser, C., 2021. *Small Launchers in a Pandemic World - 2021 Edition of the Annual Industry Survey*. Logan, 35th AIAA/USU Conference on Small Satellites.

Nie, Y., Faqir, O. & Kerrigan, E., 2018. *NLPSolvers*. [Online]

Available at: <http://www.ee.ic.ac.uk/ICLOCS/Downloads.html#NLPSolvers>

[Accessed 01 April 2020].

NOAA, 1976. *U.S. STANDARD ATMOSPHERE, 1976*. Washington: NASA.

Nocedal, J. & Wright, S. J., 1999. *Numerical Optimisation*. New York City: Springer.

Patton, J. A. & Hopkins, J. B., 2006. *Atlas V for commercial passenger transportation*. San Jose, AIAA Space 2006 Conference and Exposition.

- Perez, R., 1998. *Wireless Communications Design Handbook*. 1st ed. s.l.:Elsevier.
- Pesch, H. J., 2014. Optimal Trajectories in the Aerospace Field. In: A. Miele & A. Salvetti, eds. *Applied Mathematics in Aerospace Science and Engineering*. New York City: Plenum Press, pp. 165-220.
- Pontani, M. & Cecchetti, G., 2013. Ascent Trajectories of Multistage Launch Vehicles: Numerical Optimisation with Second-Order Conditions Verification. *International Scholarly Research Notices*, Volume 2013.
- Pontani, M., 2013. Ascent Trajectories of Multistage Launch Vehicles: Numerical Optimisation with Second-Order Conditions Variations. *International Scholarly Research Notices*, Volume 2013.
- Rao, A. V., 2010. A Survey of Numerical Methods for Optimal Control. *Advances in the Astronautical Sciences*.
- Robbins, H. M., 1965. Optimality of Intermediate-Thrust Arcs of Rocket Trajectories. *AIAA*, 3(6), pp. 1094-1098.
- Roberts, T. G., 2020. *Popular Orbits 101*. [Online]
Available at: <https://aerospace.csis.org/aerospace101/popular-orbits-101/>
[Accessed 10 April 2020].
- Rocket lab, 2020. *Launch: Payload User's Guide*. s.l.:Rocket lab.
- Rosmann, C., Makarow, A. & Bertram, T., 2020. Time-optimal control with direct collocation and variable discretization. *arXiv preprint arXiv:2005.12136*.
- San Francisco State University's Department of Earth and Climate Sciences, 2022. *Polar Orbiting Environmental Satellites (POES)*. [Online]
Available at:
http://tornado.sfsu.edu/geosciences/classes/m415_715/Monteverdi/Satellite/PolarOrbiter/Polar

Orbits.html

[Accessed 12 May 2022].

Space Launch Report, 2014. *SpaceX Falcon Data Sheet*. [Online]

Available at: <https://www.spacelaunchreport.com/falcon.html>

[Accessed 10 May 2021].

Space Launch Report, 2021. *Rocket Lab Electron*. [Online]

Available at: <https://www.spacelaunchreport.com/electron.html>

[Accessed 10 May 2021].

Space Launch Report, 2021. *SpaceX Falcon 9 v1.2*. [Online]

Available at: <https://www.spacelaunchreport.com/falcon9ft.html>

[Accessed 10 May 2021].

SpaceX, 2008. *Falcon 1 Launch Vehicle Payload User's Guide*. California: SpaceX.

SpaceX, 2009. *Falcon 9 Launch Vehicle Payload User's Guide*. California: Space Exploration Technologies Corporation.

SpaceX, 2020. *Falcon User's Guide*. California: SpaceX.

SpaceX, 2021. *Falcon 9*. [Online]

Available at: <https://www.spacex.com/vehicles/falcon-9/>

[Accessed 10 May 2021].

Stengel, R., 2016. *Spacecraft System Design*. [Online]

Available at: <http://www.princeton.edu/~stengel/MAE342.html>

[Accessed 20 May 2020].

Suresh, B. N. & Sivan, K., 2015. *Integrated Design for Space Transportation System*. New Dehli: Springer.

Sutton, G. P. & Biblarz, O., 2001. *Rocket propulsion elements*. 6th ed. New York City: John Wiley & Sons.

Swaminathan, S., 2005. *The Applicability of Space Law Principles to Basic Space Science: an Update*. [Online]

Available at: <http://adsabs.harvard.edu> › pdf

[Accessed 22 March 2020].

The Advisory Group for Aerospace and Research and NATO, 1970. *Wind Effects on Launch Vehicles*. 1st ed. Slough: Technivision Services.

Topputo, F. & Zhang, C., 2014. Survey of Direct Transcription for Low-Thrust Space Trajectory Optimisation with Applications. *Abstract and Applied Analysis*, Volume 2014.

von Stryk, O. & Bulirsch, R., 1992. Combining Direct and Indirect Methods in Optimal Control. *Annals of Operation Research*, Volume 26, pp. 357-373.

Wekerle, T., Filho, J., Costa, L. & Trabasso, L., 2017. Status and Trends of Smallsats and Their Launch Vehicles — An Up-to-date Review. *Journal of Aerospace Technology and Management*, 9(3), p. 269.

Widnall, S. & Peraire, J., 2008. *Lecture L17 -Orbit Transfers and Interplanetary Trajectories*. [Online]

Available at: <https://ocw.mit.edu/courses/aeronautics-and-astronautics/16-07-dynamics-fall-2009/lecture-notes/>

[Accessed 2020 April 2020].

Zhang, L., 2006. *A fixed-point framework for launch vehicle ascent*. Ames: Iowa State University.

APPENDIX A: ORBITAL ELEMENTS

The standard for fully defining a particular orbit requires the provision of the following six parameters (or orbital elements): specific angular momentum (\mathbf{h}), inclination (i), right ascension of the ascending node (Ω), eccentricity (e), argument of perigee (ω) and true anomaly (θ). The argument of perigee and true anomaly are both undefined for circular orbits. Circular orbits possess an eccentricity of zero. In addition to this, the minimum and maximum altitude an orbiting body can achieve along a particular orbit may also be defined. These two altitudes are referred to as the perigee and apogee respectively.

Figure A-1 (Curtis, 2005) depicts an elliptical orbit with all six orbital elements. The origin is the centre of mass of the body which is orbited around. The X-axis is in the direction of the Vernal Equinox. The Z-axis is in the direction of the vector from the origin to the North Pole.

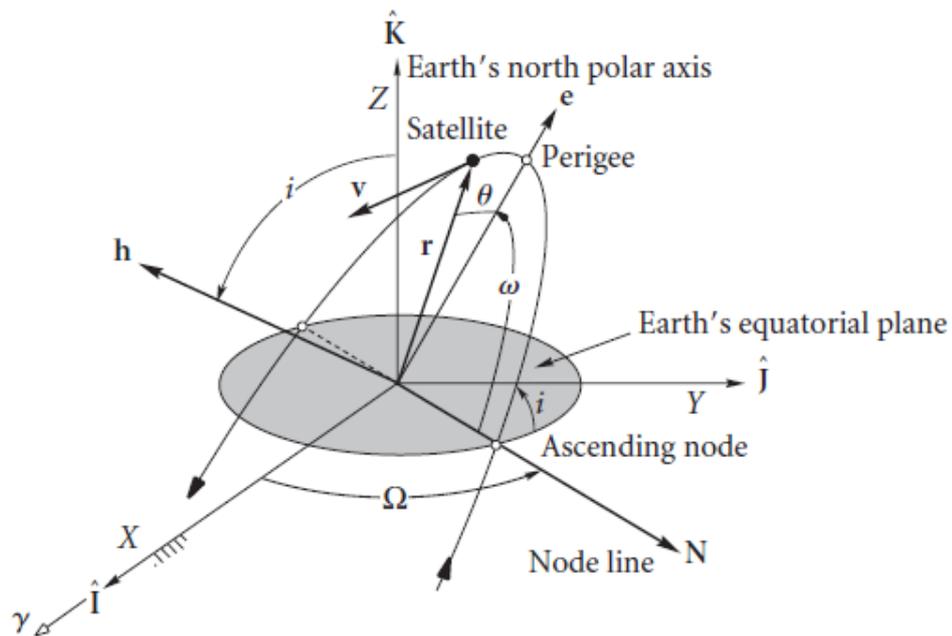


Figure A-1: Elliptic orbit with orbital elements

The specific angular momentum of an orbit is the angular momentum per unit mass which a body with possess within the orbit. This orbital element will remain constant, regardless of the position of an orbiting body along a particular orbit. The specific angular momentum can be defined by the cross product of the position and velocity of an orbiting body at any point along an orbit.

The inclination of an orbit is the dihedral angle between the orbital plane and the equatorial plane. It can also be measured as the smallest angle between the specific angular momentum vector and the positive Z-axis in Figure A-1. This angle ranges between $0^\circ - 180^\circ$.

The eccentricity of an orbit is the ratio of the distance between the two foci to the length of the major axis. The higher the eccentricity of an orbit, the less “circular” an orbit becomes, increasing the difference in magnitude between the apogee and perigee altitudes of an orbit.

In order to define the right ascension of the ascending node and the argument of perigee, the node line (depicted by “N” in Figure A-1) must be defined. This line is formed by joining the centre of the mass which is orbited around (or origin), to the point on the equatorial plane at which the orbiting body travels in a northern direction (this point is referred to as the ascending node).

The angle measured from the positive X-axis to the Node line in Figure A-1 is the right ascension of the ascending node. The angle between the node line and the line from the origin to the perigee, is the argument of perigee. Lastly, true anomaly is defined as the angle between the position of the orbiting body, and a line from the origin to the perigee. All three of these angles range between $0^\circ - 360^\circ$.

APPENDIX B: COORDINATE TRANSFORMATION MATRICES

This appendix details the relevant transformation matrices for the work covered in this dissertation. Transformation matrices, in the context of this work, are used to transform vectors from one system of coordinates to another. If a transformation matrix transforms a vector from coordinate system 1 to coordinate system 2, and both coordinate systems are defined using orthogonal axes, then the transpose of this transformation matrix will transform a vector from coordinate system 2 to coordinate system 1.

The transformation matrix from GEF to ZEN (R_1) is as follows:

$$R_1 = \begin{bmatrix} \cos \phi \cos \theta & \cos \phi \sin \theta & \sin \phi \\ -\sin \theta & \cos \theta & 0 \\ -\sin \phi \cos \theta & -\sin \phi \sin \theta & \cos \phi \end{bmatrix} \quad (\text{B-1})$$

The perifocal frame is referred to as the “natural frame” for an orbiting body within an orbit. It is centred at the focus of the orbit. The orbital plane is its x-y plane, and its x-axis is directed towards the periapse.

The transformation matrix from GEF to the perifocal frame (R_2) is as follows:

$$R_2 = \begin{bmatrix} \cos \Omega \cos \omega - \sin \Omega \sin \omega \cos i & \sin \Omega \cos \omega + \cos \Omega \sin \omega \cos i & \sin i \sin \omega \\ -\cos \Omega \sin \omega - \sin \Omega \cos \omega \cos i & -\sin \Omega \sin \omega + \cos \Omega \cos \omega \cos i & \sin i \cos \omega \\ \sin \Omega \sin i & -\cos \Omega \sin i & \cos i \end{bmatrix} \quad (\text{B-2})$$

APPENDIX C: SIMULATED ALTITUDE VERSUS RANGE PLOTS

This appendix contains all the altitude versus range plots of the simulated launch trajectories in Chapter 6.

Figures C-1 to C-18 detail the altitude versus range plots of the Electron launch vehicle for the analysed missions.

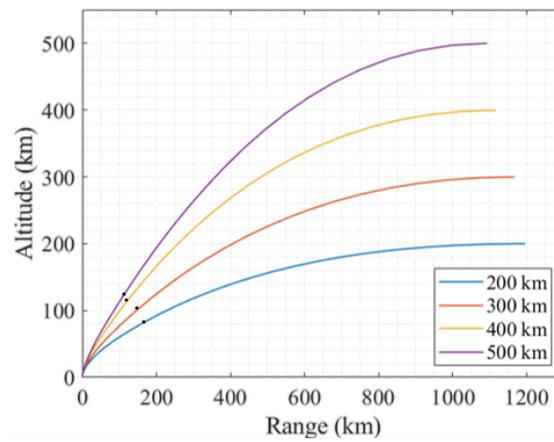


Figure C-1: Electron altitude vs range ($e=0$, $i=40^\circ$)

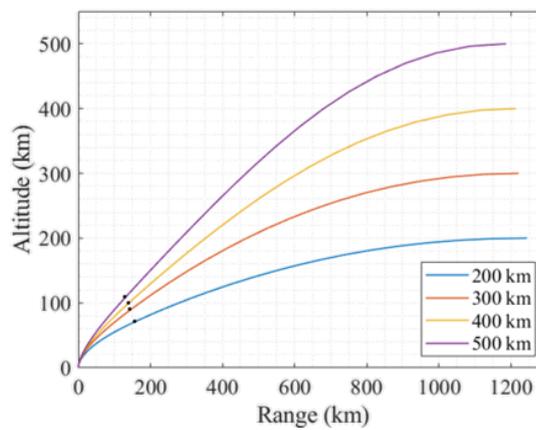


Figure C-2: Electron altitude vs range ($e=0$, $i=64.435^\circ$)

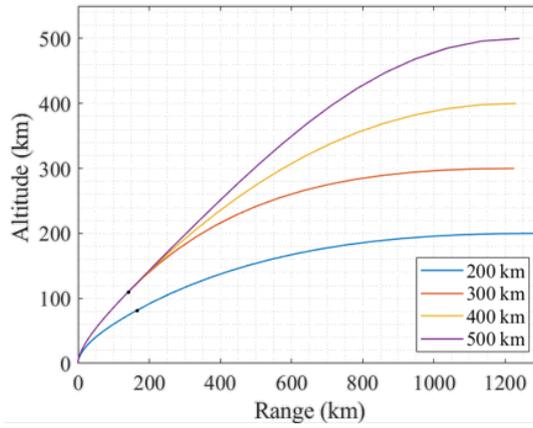


Figure C-3: Electron altitude vs range ($e=0$, SSO)

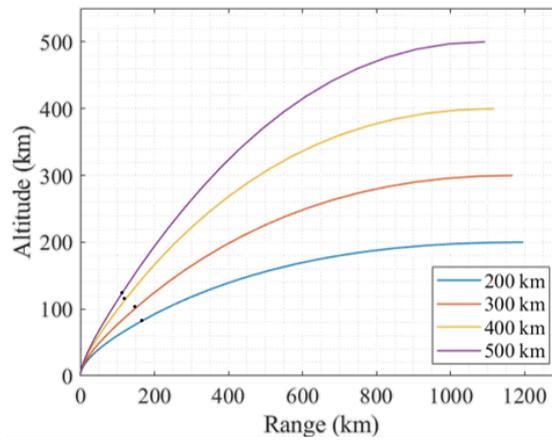


Figure C-4: Electron altitude vs range ($e=0.005$, $i=40^\circ$)

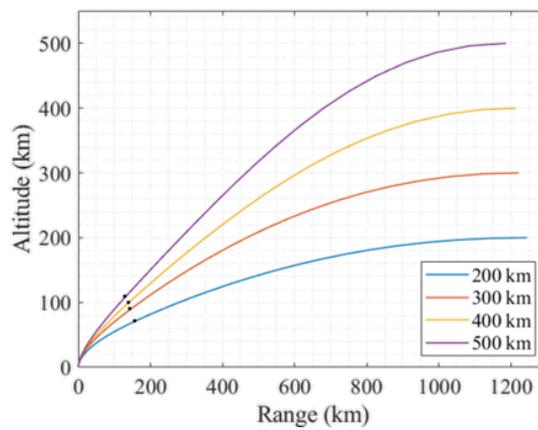


Figure C-5: Electron altitude vs range ($e=0.005$, $i=64.435^\circ$)

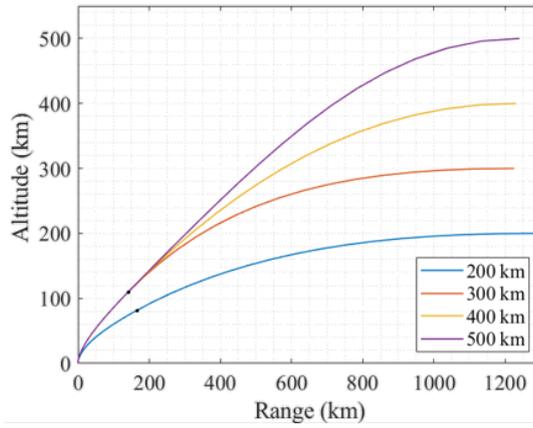


Figure C-6: Electron altitude vs range ($e=0.005$, SSO)

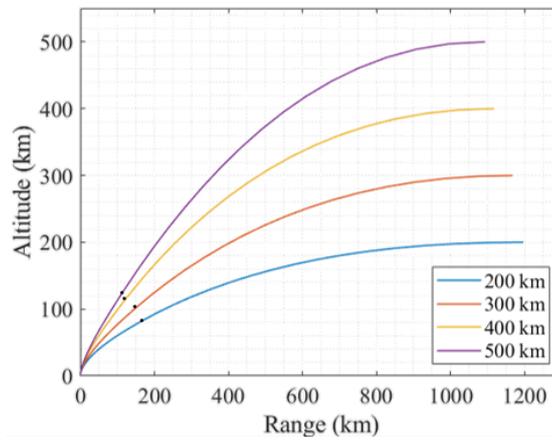


Figure C-7: Electron altitude vs range ($e=0.01$, $i=40^\circ$)

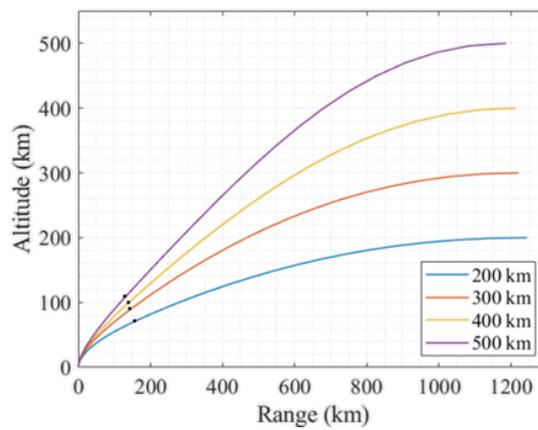


Figure C-8: Electron altitude vs range ($e=0.01$, $i=64.435^\circ$)

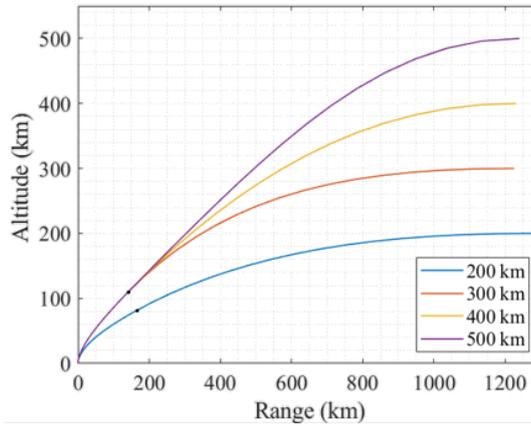


Figure C-9: Electron altitude vs range ($e=0.01$, SSO)

Figures C-10 to C-18 detail the altitude versus range plots of the Falcon 1 launch vehicle for the analysed missions.

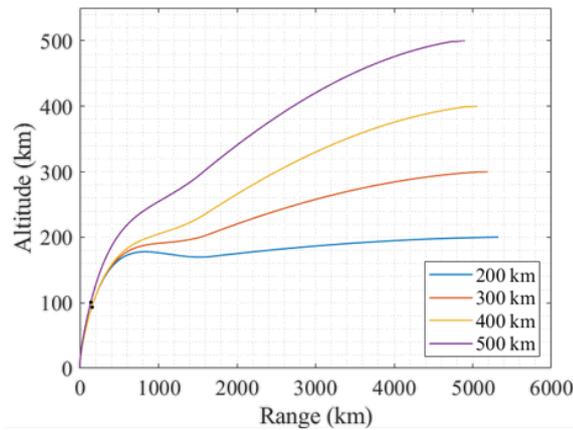


Figure C-10: Falcon 1 altitude vs range ($e=0$, $i=40^\circ$)

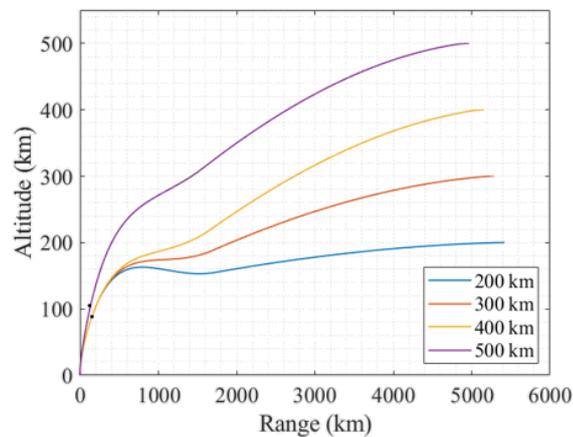


Figure C-11: Falcon 1 altitude vs range ($e=0$, $i=64.435^\circ$)

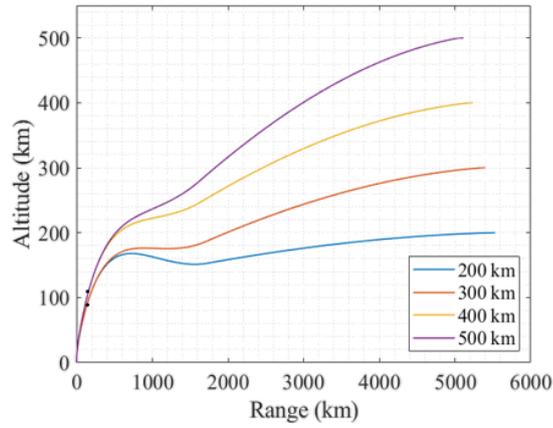


Figure C-12: Falcon 1 altitude vs range ($e=0$, SSO)

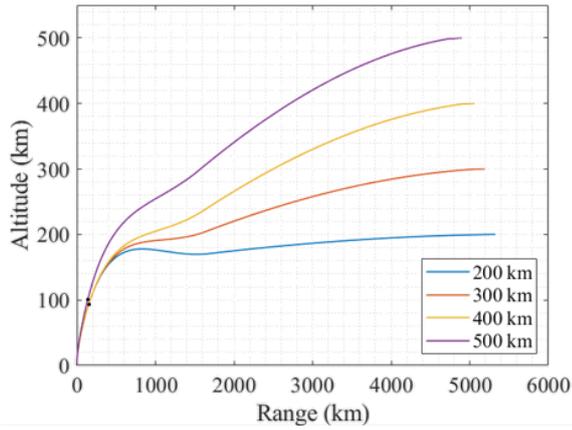


Figure C-13: Falcon 1 altitude vs range ($e=0.005$, $i=40^\circ$)

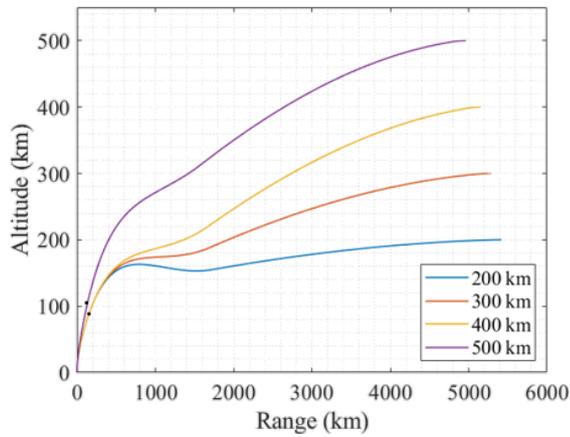


Figure C-14: Falcon 1 altitude vs range ($e=0.005$, $i=64.435^\circ$)

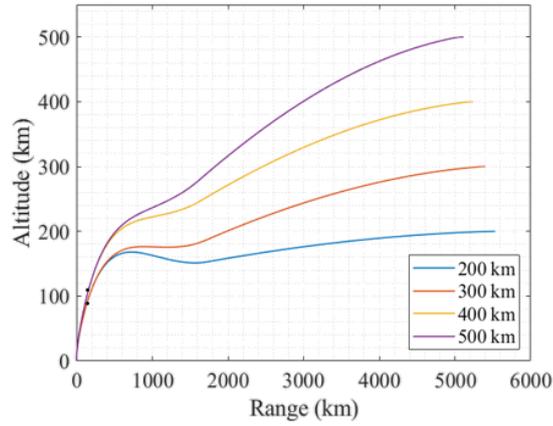


Figure C-15: Falcon 1 altitude vs range ($e=0.005$, SSO)

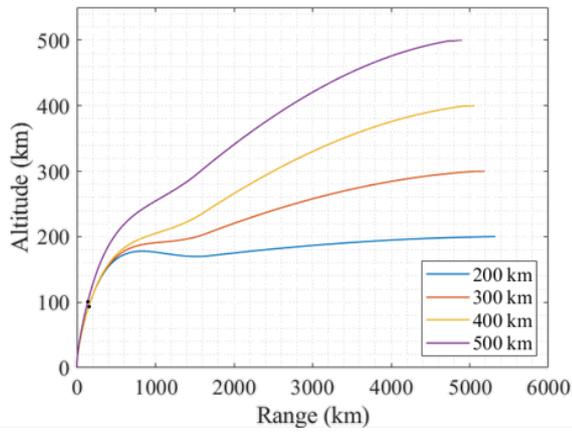


Figure C-16: Falcon 1 altitude vs range ($e=0.01$, $i=40^\circ$)

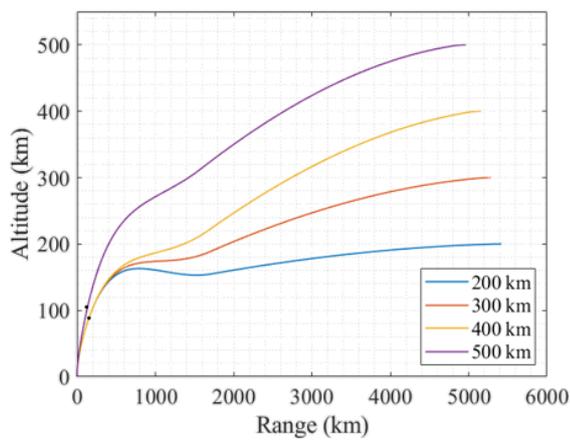


Figure C-17: Falcon 1 altitude vs range ($e=0.01$, $i=64.435^\circ$)

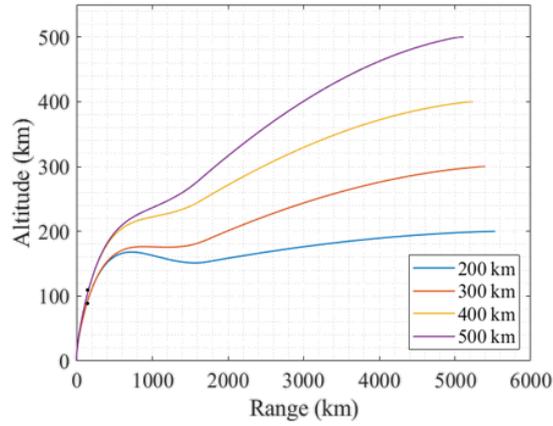


Figure C-18: Falcon 1 altitude vs range ($e=0.01$, SSO)

Figures C-19 to C-27 detail the altitude versus range plots of the Falcon 9 launch vehicle for the analysed missions.

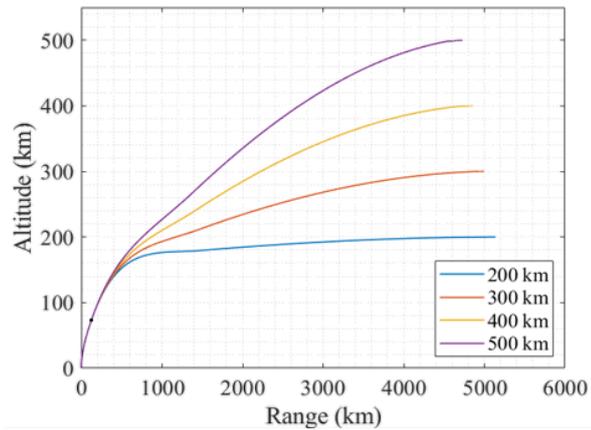


Figure C-19: Falcon 9 altitude vs range ($e=0$, $i=40^\circ$)

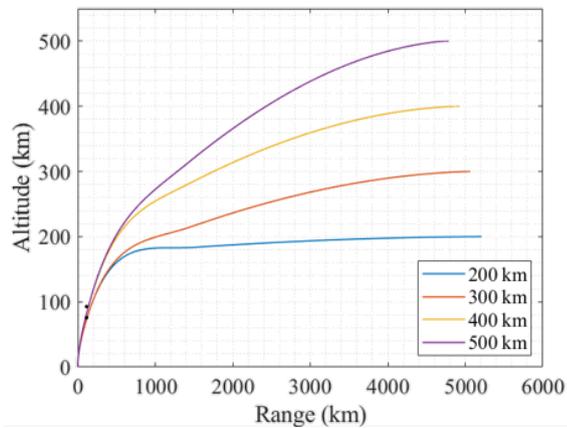


Figure C-20: Falcon 9 altitude vs range ($e=0$, $i=64.435^\circ$)

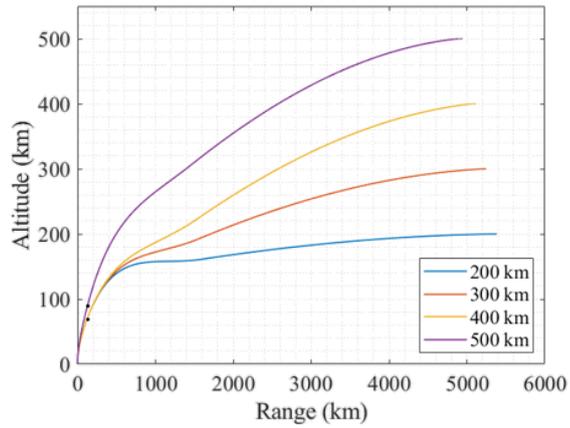


Figure C-21: Falcon 9 altitude vs range ($e=0$, SSO)

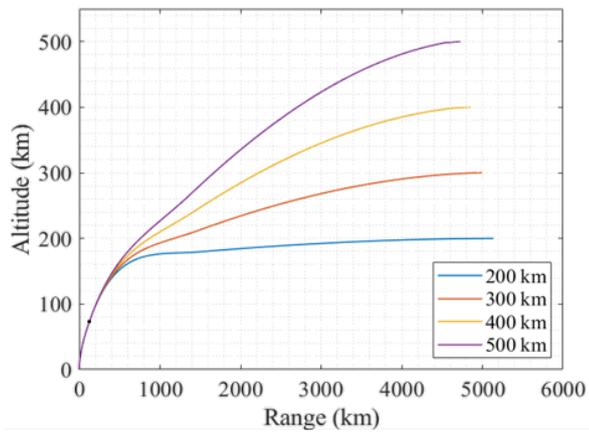


Figure C-22: Falcon 9 altitude vs range ($e=0.005$, $i=40^\circ$)

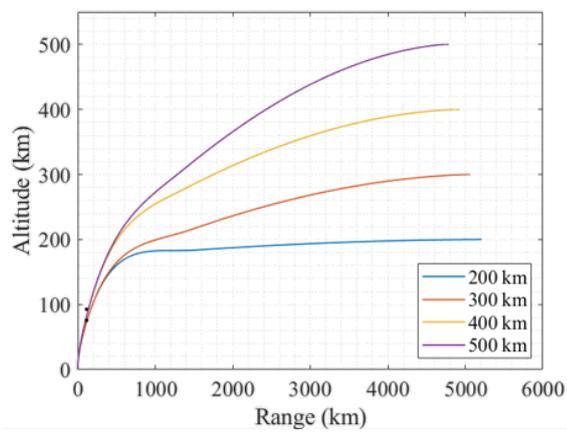


Figure C-23: Falcon 9 altitude vs range ($e=0.005$, $i=64.435^\circ$)

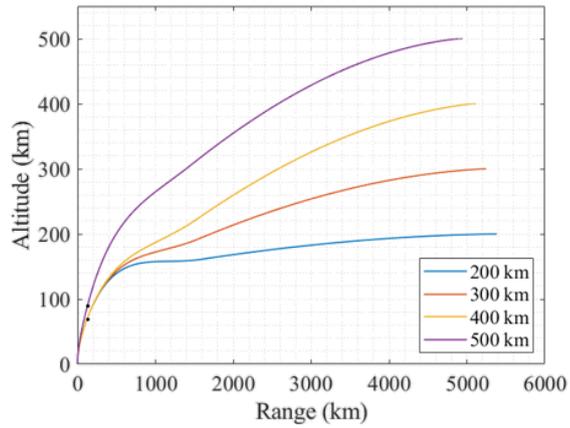


Figure C-24: Falcon 9 altitude vs range ($e=0.005$, SSO)

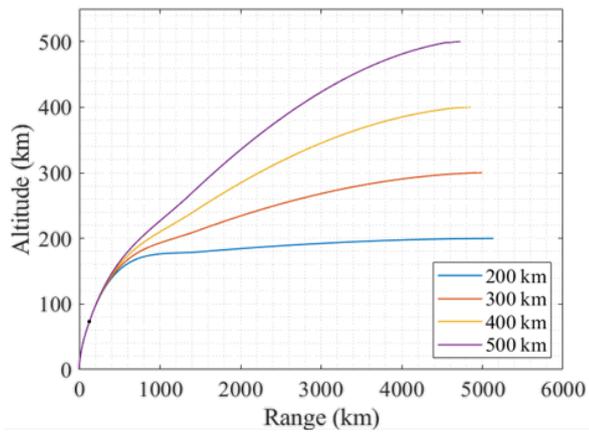


Figure C-25: Falcon 9 altitude vs range ($e=0.01$, $i=40^\circ$)

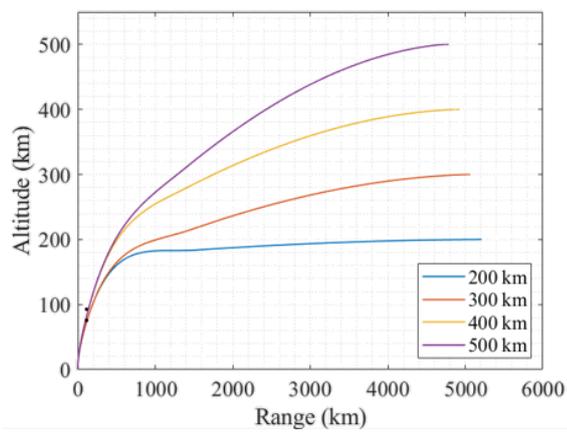


Figure C-26: Falcon 9 altitude vs range ($e=0.01$, $i=64.435^\circ$)

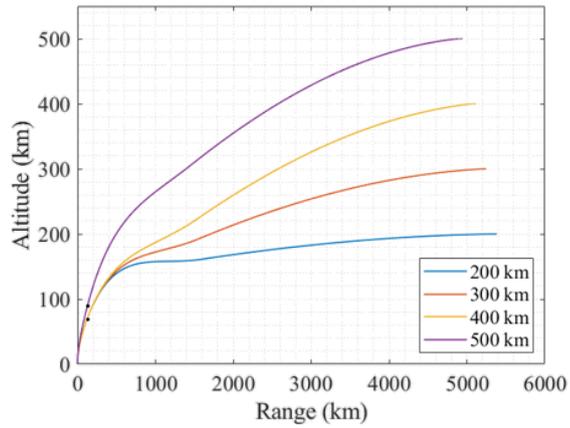


Figure C-27: Falcon 9 altitude vs range ($e=0.01$, SSO)

Figures C-28 to C-35 detail the altitude versus range plots of the CLV for the analysed missions.

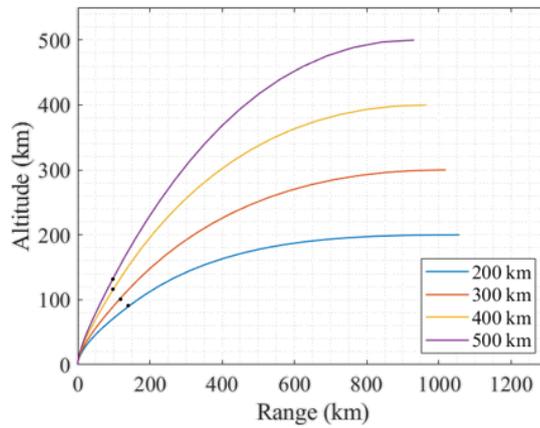


Figure C-28: CLV altitude vs range ($e=0$, $i=40^\circ$)

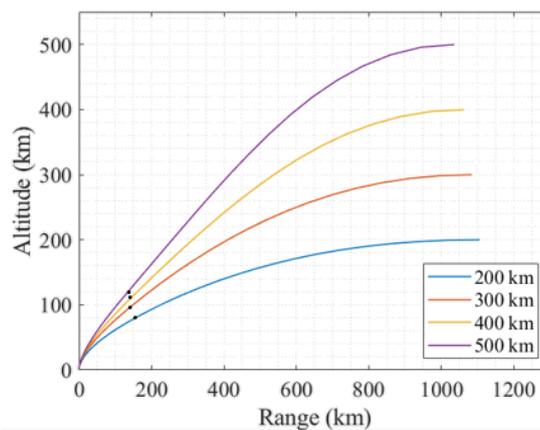


Figure C-29: CLV altitude vs range ($e=0$, $i=64.435^\circ$)

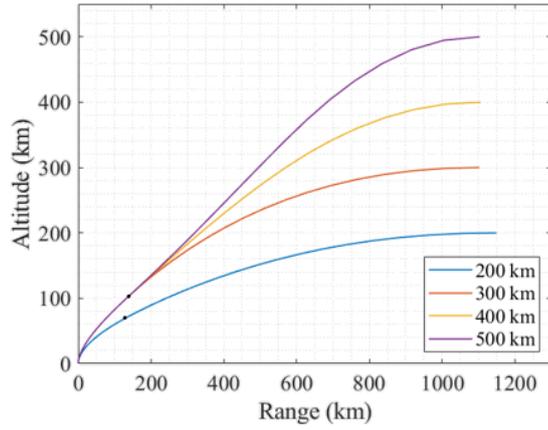


Figure C-30: CLV altitude vs range ($e=0$, $i=SSO$)

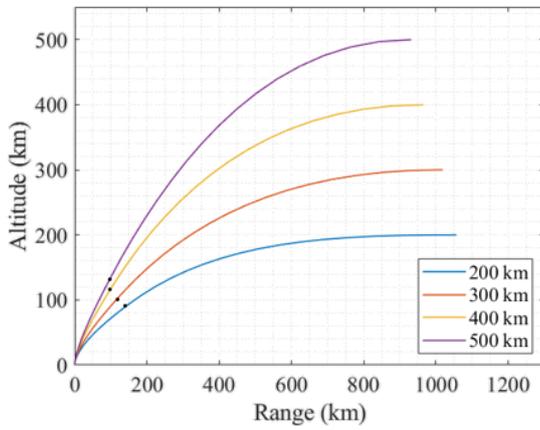


Figure C-31: CLV altitude vs range ($e=0.005$, $i=40^\circ$)

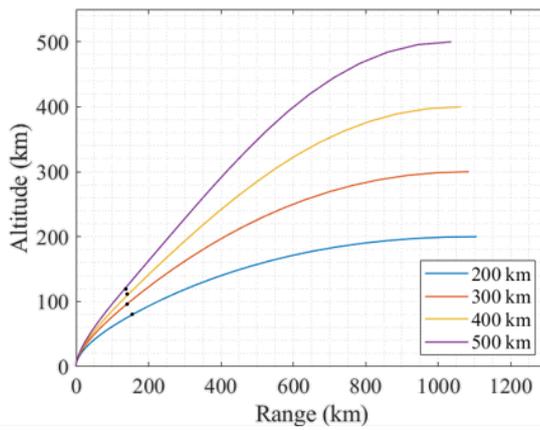


Figure C-32: CLV altitude vs range ($e=0.005$, $i=64.435^\circ$)

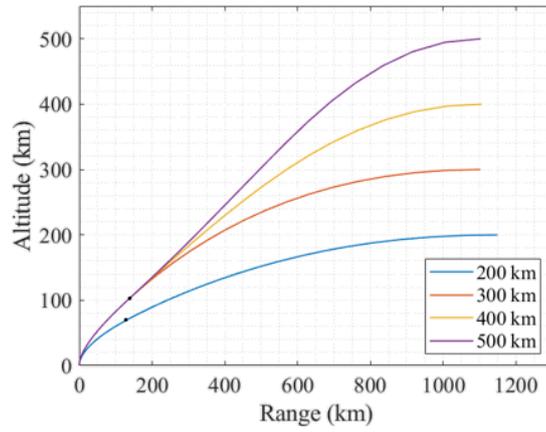


Figure C-33: CLV altitude vs range ($e=0.005$, $i=SSO$)

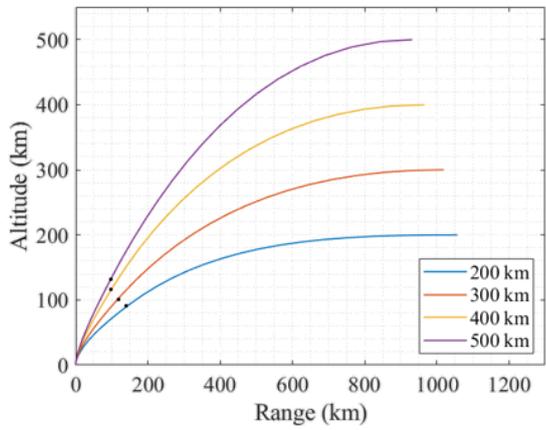


Figure C-34: CLV altitude vs range ($e=0.01$, $i=40^\circ$)

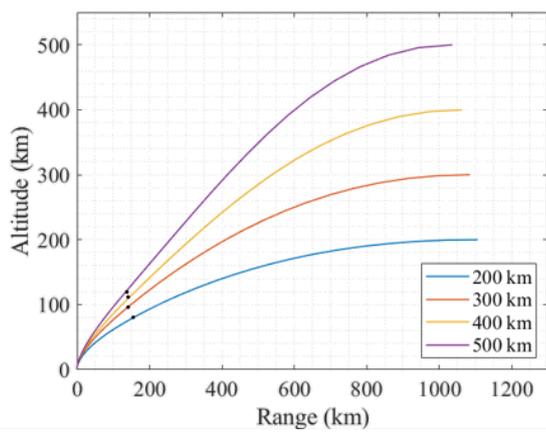


Figure C-35: CLV altitude vs range ($e=0.01$, $i=64.435^\circ$)

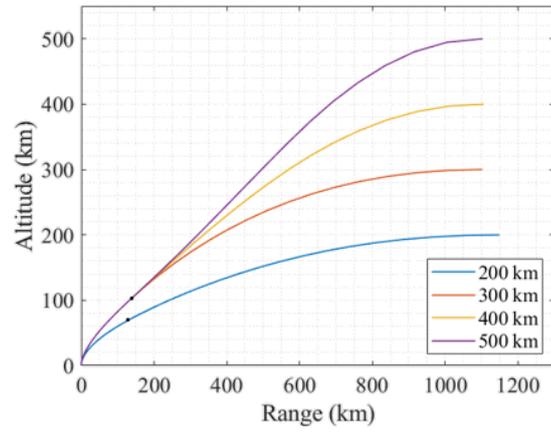


Figure C-36: CLV altitude vs range ($e=0.1$, $i=SSO$)

APPENDIX D: RASAERO™ INPUTS

Table D-1 contains the RASAero™ inputs used to obtain the aerodynamic coefficients of the launch vehicles analysed in this work. The dimensions are provided in inches as RASAero™ by default uses inches, and payload user's guides tend to use inches to describe dimensions of launch vehicles.

Table D-1: RASAero™ inputs

| Launch vehicle | Nose type | Nose tip radius (inches) | Nose diameter (inches) | Nose length (inches) | First body tube length (inches) | Transition length (inches) | Transition rear diameter (inches) | Second body tube length (inches) |
|-----------------------|------------------|---------------------------------|-------------------------------|-----------------------------|--|-----------------------------------|--|---|
| Electron | parabolic | 3.0 | 47.4 | 66.0 | 604.0 | 0 | 0 | 0 |
| Falcon 1 | von Karman ogive | 7.0 | 54.0 | 79.0 | 49.0 | 12.0 | 66.0 | 700.0 |
| Falcon 9 | von Karman ogive | 26.5 | 204.0 | 231.0 | 261.0 | 54.0 | 144.0 | 2202.0 |
| CLV | von Karman ogive | 0.0 | 51.2 | 98.4 | 675.5 | 0 | 0 | 0 |

APPENDIX E: COMPUTER SPECIFICATIONS

All simulations were conducted using a HP 15-da0031ni notebook. The operating system used was Windows 10. The specifications of this notebook, obtained from the official HP website (HP, 2022), are as follows:

| | |
|--------------------------|---|
| Product name: | HP Notebook - 15-da0031ni |
| Microprocessor: | Intel® Core™ i7-8550U (1.8 GHz base frequency, up to 4 GHz with Intel® Turbo Boost Technology, 8 MB cache, 4 cores) |
| Memory, standard: | 8 GB DDR4-2400 SDRAM (1 x 8 GB) |
| Video graphics: | NVIDIA® GeForce® MX130 (2 GB DDR3 dedicated) |
| Hard drive: | 1 TB 5400 rpm SATA |