

Copyright  
by  
Martin James Brennan  
2015

The Dissertation Committee for Martin James Brennan  
certifies that this is the approved version of the following dissertation:

**Preliminary Interplanetary Trajectory Design Tools using  
Ballistic and Powered Gravity Assists**

Committee:

---

Wallace Fowler, Supervisor

---

Ryan Russell

---

Srinivas Bettadpur

---

E. Glenn Lightsey

---

Carrie Olsen

**Preliminary Interplanetary Trajectory Design Tools using  
Ballistic and Powered Gravity Assists**

**by**

**Martin James Brennan, B.S.; B.S.; M.S.E.**

**DISSERTATION**

Presented to the Faculty of the Graduate School of  
The University of Texas at Austin  
in Partial Fulfillment  
of the Requirements  
for the Degree of

**DOCTOR OF PHILOSOPHY**

THE UNIVERSITY OF TEXAS AT AUSTIN

August 2015

To Holly for your encouragement, commitment, and love.

## Acknowledgments

My research and academic career was enabled by the unconditional support of my family, friends, and teachers. I give my deepest thanks to all the people in my life who provided their advice, understanding, and love that allowed me to continue along such a difficult path:

My parents and family, who continue to push me to do my best and pursue my dreams.

The community of brilliant UT graduate students, who are always willing to share their time and experience. I strive to live up to their example and to pay forward the same assistance that I've received.

Dr. Russell for his honesty, experience, and sharing his wealth of knowledge on trajectory design and orbital mechanics

Dr. Wallace Fowler for his endless insight and cheer. His dedication to students and education is an inspiration.

# **Preliminary Interplanetary Trajectory Design Tools using Ballistic and Powered Gravity Assists**

Publication No. \_\_\_\_\_

Martin James Brennan, P.h.D.  
The University of Texas at Austin, 2015

Supervisor: Wallace Fowler

Preliminary interplanetary trajectory designs frequently use simplified two-body orbital mechanics and linked conics methodology to model the complex trajectories in multi-body systems. Incorporating gravity assists provides highly efficient interplanetary trajectories, enabling otherwise infeasible spacecraft missions. Future missions may employ powered gravity assists, using a propulsive maneuver during the flyby, improving the overall trajectory performance.

This dissertation provides a complete description and analysis of a new interplanetary trajectory design tool known as TRACT (TRAjectory Configuration Tool). TRACT is capable of modeling complex interplanetary trajectories, including multiple ballistic and/or powered gravity assists, deep space maneuvers, parking orbits, and other common maneuvers. TRACT utilizes an adaptable architecture of modular boundary value problem (BVP) algorithms for all trajectory segments. A bi-level optimization scheme is employed to reduce the number of optimization variables, simplifying the user provided trajectory information. The standardized

optimization parameter set allows for easy use of TRACT with a variety of optimization algorithms and mission constraints.

The dissertation also details new research in powered gravity assists. A review of literature on optimal powered gravity assists is presented, where many optimal solutions found are infeasible for realistic spacecraft missions. The need was identified for a mission feasible optimal powered gravity assist algorithm using only a single impulsive maneuver. The solution space was analyzed and a complete characterization was developed for solution types of the optimal single-impulse powered gravity assist. Using newfound solution space characteristics, an efficient and reliable optimal single-impulse powered gravity assist BVP algorithm was formulated. The mission constraints were strictly enforced, such as maintaining the closest approach above a minimum radius and below a maximum radius.

An extension of the optimal powered gravity assist research is the development of a gravity assist BVP algorithm that utilizes an asymptote  $\Delta V$  correction maneuver to produce ballistic gravity assist trajectory solutions. The efficient algorithm is tested with real interplanetary mission trajectory parameters and successfully converges upon ballistic gravity assists with improved performance compared to traditional methods. A hybrid approach is also presented, using the asymptote maneuver algorithm together with traditional gravity assist constraints to reach ballistic trajectory solutions more reliably, while improving computational performance.

# Table of Contents

|  |             |
|--|-------------|
| <b>Acknowledgments</b>   | <b>v</b>    |
| <b>Abstract</b>  | <b>vi</b>   |
| <b>List of Tables</b>  | <b>xii</b>  |
| <b>List of Figures</b>   | <b>xiii</b> |
| <b>Chapter 1. Introduction</b>   | <b>1</b>    |
| 1.1 Preliminary interplanetary trajectory problem definition . . . . .       | 1           |
| 1.2 Motivation . . . . .   | 3           |
| 1.3 Contributions . . . . .  | 5           |
| 1.4 Dissertation organization . . . . .                                      | 6           |
| <b>Chapter 2. Multi-body Trajectory Design &amp; Optimization Background</b> | <b>8</b>    |
| 2.1 Trajectory design & optimization strategies . . . . .                    | 8           |
| 2.1.1 Multi-body systems . . . . .   | 9           |
| 2.1.2 Two-body systems . . . . .   | 10          |
| 2.2 Patched/linked conics . . . . .  | 15          |
| 2.3 Boundary value problem methods . . . . .                                 | 20          |
| 2.3.1 Lambert targeting . . . . .  | 21          |
| 2.3.2 Hyperbolic terminal velocity problem . . . . .                         | 26          |
| 2.4 Planetary encounters . . . . .   | 32          |
| 2.4.1 Parking Orbits . . . . .   | 32          |
| 2.4.2 Ballistic Flybys . . . . .   | 36          |
| <b>Chapter 3. TRACT: TRAjectory Configuration Tool</b>                       | <b>44</b>   |
| 3.1 TRACT summary . . . . .  | 46          |
| 3.1.1 Assumptions . . . . .  | 47          |
| 3.1.2 TRACT architecture . . . . .   | 49          |

|   |           |
|---|-----------|
| 3.1.2.1 Mission driver interface . . . . .  | 51        |
| 3.1.2.2 Trajectory configuration function . . . . .                                       | 52        |
| 3.1.3 TRACT capabilities . . . . .  | 55        |
| 3.1.3.1 Planetary state ephemerides . . . . .   | 55        |
| 3.1.3.2 Optimization schemes . . . . .  | 56        |
| 3.1.3.3 Trajectory constraints . . . . .  | 57        |
| 3.1.3.4 Departure/arrival maneuvers . . . . .   | 58        |
| 3.1.3.5 Gravity assist maneuvers . . . . .  | 60        |
| 3.1.3.6 Deep space maneuvers . . . . .  | 61        |
| 3.1.3.7 MATLAB trajectory visualization output . . . . .                                  | 62        |
| 3.1.3.8 Excel data output . . . . .   | 70        |
| 3.2 Bi-level optimization . . . . .   | 73        |
| 3.2.1 Bi-level problem definition . . . . .   | 73        |
| 3.2.2 Linked BVPs architecture . . . . .  | 76        |
| 3.2.3 Optimization performance . . . . .  | 77        |
| 3.3 Primary level optimization procedure . . . . .  | 80        |
| 3.4 Secondary level BVPs . . . . .  | 83        |
| 3.4.1 Departure/arrival transfer with parking orbit . . . . .                             | 83        |
| 3.4.2 Departure/arrival transfer with launch/entry conditions . . . . .                   | 86        |
| 3.4.3 Gravity assists . . . . .   | 87        |
| 3.5 Chapter conclusions . . . . .   | 88        |
| <b>Chapter 4. Powered Gravity Assist Boundary Value Problem</b>                           | <b>90</b> |
| 4.1 Motivation for powered gravity assist . . . . .                                       | 90        |
| 4.2 Background on types of flyby BVPs . . . . .   | 95        |
| 4.2.1 Typical powered gravity assist strategy (impulse at periapse) . .                   | 95        |
| 4.2.2 Literature review: optimal multiple impulse flyby problem . .                       | 100       |
| 4.3 Optimal single impulse flyby BVP . . . . .  | 111       |
| 4.3.1 Assumptions . . . . .   | 111       |
| 4.3.2 Optimal powered gravity assist $\Delta V$ maneuver solution neighborhoods . . . . . | 112       |
| 4.3.3 Analytic solution for unconstrained periapse powered gravity assist . . . . .       | 117       |
| 4.3.4 Unconstrained periapse powered gravity assist solution analysis                     | 119       |

|                   |   |            |
|-------------------|---|------------|
| 4.3.4.1           | Introductory solution space dynamics . . . . .  | 120        |
| 4.3.4.2           | Strict $r_{min}$ constraint handling for transfer maneuvers<br>near periapse . . . . .                  | 128        |
| 4.3.4.3           | Linked conics $r_{SOI}$ constraint handling . . . . .   | 130        |
| 4.3.5             | Constrained periapse powered gravity assist $\Delta V$ function and<br>derivative development . . . . . | 132        |
| 4.3.5.1           | Development of the $\Delta V$ function and derivative . . . . .   | 133        |
| 4.3.6             | Constrained periapse powered gravity assist solution analysis .   | 144        |
| 4.3.6.1           | Introductory solution space dynamics . . . . .  | 145        |
| 4.3.6.2           | Strict $r_{min}$ constraint handling for periapse constrained<br>powered flybys . . . . .               | 149        |
| 4.3.6.3           | Optimal transfer maneuvers near SOI boundary . . . . .  | 154        |
| 4.3.6.4           | Multimodal solution space dynamics . . . . .  | 158        |
| 4.3.6.5           | General solution space dynamics . . . . .   | 167        |
| 4.3.7             | Optimal periapse constrained powered gravity assist solution<br>algorithm . . . . .                     | 177        |
| 4.3.7.1           | Implementation of optimal periapse constrained pow-<br>ered gravity assist method . . . . .             | 177        |
| 4.3.7.2           | Performance analysis of optimal periapse constrained<br>powered gravity assist algorithm . . . . .      | 183        |
| <b>Chapter 5.</b> | <b>Flyby Constraints Utilizing Penalty <math>\Delta V</math> at Asymptote</b>                           | <b>198</b> |
| 5.1               | Typical ballistic flyby constraint handling . . . . .   | 198        |
| 5.2               | Asymptote corrected ballistic gravity assist algorithm development .                                    | 199        |
| 5.3               | Asymptote corrected ballistic gravity assist algorithm analysis . . .                                   | 207        |
| 5.4               | Spacecraft mission trajectory examples with ballistic gravity assists .                                 | 210        |
| 5.4.1             | Juno mission trajectory with accurate initial parameter values .  | 210        |
| 5.4.2             | Juno mission trajectory with rough initial parameter values . .   | 217        |
| 5.4.3             | Galileo mission trajectory with accurate initial parameter values                                       | 220        |
| 5.4.4             | Galileo mission trajectory with rough initial parameter values .  | 224        |
| 5.5               | Extension of asymptote $\Delta V$ for other constraint handling . . . . .                               | 229        |
| 5.6               | Chapter conclusions . . . . .   | 234        |

|   |            |
|---|------------|
| <b>Chapter 6. Interplanetary Spacecraft Mission Case Studies</b>                              | <b>235</b> |
| 6.1 Recent Interplanetary Spacecraft Missions . . . . .                                       | 235        |
| 6.1.1 New Horizons mission example . . . . .  | 235        |
| 6.1.2 MESSENGER mission example . . . . .   | 242        |
| 6.1.3 Cassini mission example . . . . .   | 252        |
| 6.2 Student designed interplanetary missions . . . . .  | 256        |
| 6.2.1 Orbiter mission to Jupiter’s moon Europa . . . . .                                      | 256        |
| 6.2.2 Flyby mission to dwarf planet Eris . . . . .  | 259        |
| <b>Chapter 7. Conclusion</b>  | <b>262</b> |
| 7.1 Conclusions . . . . .   | 262        |
| 7.2 Contributions . . . . .   | 265        |
| 7.3 Recommendations for Further Study . . . . .   | 266        |
| 7.3.1 Powered gravity assist algorithms . . . . .   | 266        |
| 7.3.2 TRACT modifications and improvements . . . . .  | 267        |
| <b>Appendices</b>   | <b>268</b> |
| <b>Appendix A. Strict <math>r_{min}</math> constraint handling analysis for Type F flybys</b> | <b>269</b> |
| <b>Appendix B. Strict <math>r_{SOI}</math> constraint handling analysis for Type F flybys</b> | <b>275</b> |
| <b>Appendix C. Hyperbolic terminal velocity BVP hodograph analysis</b>                        | <b>282</b> |
| <b>Appendix D. GTOC 7 Contribution</b>  | <b>286</b> |
| <b>Appendix E. TRACT optimization schemes</b>   | <b>289</b> |
| <b>Appendix F. TRACT constraints</b>  | <b>293</b> |
| F.1 Linear constraints: . . . . .   | 293        |
| F.2 Nonlinear constraints . . . . .   | 293        |
| <b>Appendix G. TRACT Mission Driver options</b>   | <b>295</b> |
| <b>Bibliography</b>   | <b>302</b> |
| <b>Vita</b>   | <b>312</b> |

## List of Tables

|     |   |     |
|-----|---|-----|
| 2.1 | Characteristics of Keplerian Orbits [1] . . . . .   | 10  |
| 2.2 | Relationships of planets' sphere of influence with distances from Sun and planets . . . . .   | 18  |
| 4.1 | Optimal powered flyby BVP solutions for the case of $\Phi_r < \Phi_p$ and $V_\infty^- > V_\infty^+$ [2, 3, 4] . . . . .             | 107 |
| 4.2 | Optimal powered flyby BVP solutions for the case of $\Phi_r > \Phi_p$ and $V_\infty^- < V_\infty^+$ [3, 4] . . . . .                | 109 |
| 5.1 | Juno mission trajectory simulation example cases using unconstrained fminunc quasi-Newton algorithm . . . . .                       | 213 |
| 5.2 | Juno mission trajectory simulation example cases using gradient-free fminsearch simplex algorithm . . . . .                         | 214 |
| 5.3 | Juno mission trajectory simulation example cases using constrained fmincon interior-point algorithm . . . . .                       | 215 |
| 5.4 | Juno mission trajectory simulation example cases using constrained fmincon SQP algorithm . . . . .                                  | 216 |
| 5.5 | Rough initialization Juno mission trajectory simulation example cases using fmincon SQP optimization scheme . . . . .               | 219 |
| 5.6 | Galileo mission trajectory simulation example cases using fmincon SQP algorithm . . . . .   | 224 |
| 5.7 | Galileo mission trajectory simulation rough initial guess cases using fmincon interior-point algorithm . . . . .                    | 229 |
| 6.1 | New Horizons mission trajectory simulation cases with best results  | 237 |
| 6.2 | New Horizons mission trajectory simulation cases for reduced flight time using powered gravity assist . . . . .                     | 240 |
| 6.3 | MESSENGER mission trajectory simulation stages . . . . .  | 247 |
| 6.4 | Cassini mission trajectory simulation cases using fmincon SQP algorithm and the asymptote correction gravity assist model . . . . . | 254 |

## List of Figures

|      |  |    |
|------|--|----|
| 2.1  | Elliptical orbit with defining parameters labeled . . . . .  | 11 |
| 2.2  | Hyperbolic orbit diagram with defining parameters labeled . . . . .  | 13 |
| 2.3  | Patched conic trajectory from inner to outer planet (not to scale) . .   | 16 |
| 2.4  | Transfer types for Lambert's problem [Source: Vallado [1]] . . . . .   | 23 |
| 2.5  | Varying time of flight for Lambert trajectory [Source: Vallado [1]] .  | 24 |
| 2.6  | Multiple revolution Lambert solution pair for prograde direction with long and short period trajectories identified by the larger and smaller orbit respectively . . . . .             | 25 |
| 2.7  | Hyperbolic terminal velocity problem diagram for departing trajectory . . . . .  | 27 |
| 2.8  | Hyperbolic terminal velocity problem diagram for arriving trajectory . . . . .   | 29 |
| 2.9  | Short and long way hyperbolic terminal velocity solutions for outgoing transfer . . . . .  | 30 |
| 2.10 | Short and long way hyperbolic terminal velocity solutions for incoming transfer . . . . .  | 30 |
| 2.11 | Hyperbolic departure trajectories from periape to $\vec{V}_\infty$ asymptote defined by asymptote offset $\Delta$ and departure plane angle $\zeta$ [Source: Curtis [5]] . . . . .     | 33 |
| 2.12 | Example departure trajectory from a parking orbit about a planet [Source: Curtis [5]] . . . . .  | 34 |
| 2.13 | Hyperbolic arrival trajectories from $\vec{V}_\infty$ asymptote to periape, as defined by the asymptote offset $\Delta$ and arrival plane angle $\zeta$ [Source: Curtis [5]] . . . . . | 35 |
| 2.14 | Example arrival trajectory into a parking orbit about a planet [Source: Curtis [5]] . . . . .  | 36 |
| 2.15 | Increasing velocity gravity assist on trailing side of planet . . . . .  | 38 |
| 2.16 | Decreasing velocity gravity assist on leading side of planet . . . . .   | 39 |
| 2.17 | Special case gravity assist of $V_{sc} < V_p$ with leading and trailing configuration . . . . .  | 41 |
| 2.18 | Special case gravity assist of $V_{sc} > V_p$ with leading and trailing configuration . . . . .  | 41 |

|      |  |    |
|------|--|----|
| 2.19 | Varying periapsis radius changes the gravitational turning angle of the gravity assist [Source: Curtis [5]] . . . . .  | 42 |
| 2.20 | Flyby potential trajectories for a fixed $\Delta$ offset distance and a revolution of hyperbolic transfer planes by plane angle $\zeta$ . . . . .  | 43 |
| 3.1  | Juno mission trajectory to Jupiter using DSMs and an Earth flyby [6]   | 47 |
| 3.2  | Juno trajectory configuration sequence example and nomenclature (relative scale slightly adjusted for clarity) . . . . .   | 48 |
| 3.3  | Trajectory design tool flowchart . . . . .   | 50 |
| 3.4  | Trajectory optimization flowchart depicting the performance and constraint functions interacting with the various heliocentric and planetocentric boundary value problems through the trajectory generator loop . . . . .  | 55 |
| 3.5  | Juno mission heliocentric trajectory with Earth and Jupiter orbits depicting the launch (cyan), DSM (black), Earth gravity assist and Jupiter orbit insertion (red), and Jupiter's alignment at launch (magenta) . . . . .   | 64 |
| 3.6  | Juno mission heliocentric trajectory (top) with Earth and Jupiter orbits, depicting each planetary encounter by its SOI disk oriented normal to the planet's pole vector and showing the encounter angular momentum vector. Close up of Earth gravity assist displayed below . . . . .   | 65 |
| 3.7  | Juno mission launch trajectory from Earth, depicting the SOI and the various orientation vectors (top). A close up view of the hyperbolic launch trajectory is shown along with the launch "parking orbit" to reference the desired orientation of the launch plane and periapse altitude. . . . .                                       | 67 |
| 3.8  | Juno mission Earth gravity assist, showing the incoming and exiting hyperbolic asymptotes as radial vectors from the planet (top). The negligible periapse $\Delta V$ maneuver direction vector (green arrow) is also depicted. A close up view of the 800 km close encounter above the 3D spherical Earth model is shown below. . . . . | 68 |
| 3.9  | Juno mission arrival into the highly eccentric polar parking orbit about Jupiter depicting the SOI and the various orientation vectors (top). A close up view of the orbit insertion at periapse is shown about the 3D spherical Jupiter model. . . . .  | 69 |
| 3.10 | Juno mission general trajectory data Excel spreadsheet example . .   | 70 |
| 3.11 | Juno mission departure and arrival trajectory data Excel spreadsheet example . . . . .   | 72 |
| 3.12 | Juno mission flyby trajectory data Excel spreadsheet example . . .   | 72 |
| 3.13 | Juno mission DSM trajectory data Excel spreadsheet example . . .   | 73 |

|      |   |     |
|------|---|-----|
| 4.1  | Flyby velocity vector diagrams representing the different types of gravity assists . . . . .  | 91  |
| 4.2  | Powered flyby velocity vector diagrams showing the improved performance and change in original $V_{\infty_o}^+$ to the final $V_{\infty}^+$ due to the interior $\Delta V$ maneuver . . . . .   | 92  |
| 4.3  | Hyperbolic boundary value problem setup for powered flyby with incoming hyperbola on left and outgoing hyperbola on right . . . . .   | 93  |
| 4.4  | General hyperbolic boundary value problem diagram for powered flyby . . . . .   | 94  |
| 4.5  | Periapse restricted powered flyby before (left) and after (right) determining common periapse . . . . .   | 96  |
| 4.6  | Four-impulse bi-parabolic transfer between hyperbolic asymptotes [Source: Gobetz [7]] . . . . .   | 102 |
| 4.7  | Diagram of resulting optimal solution cases under varying flyby conditions with a minimum radius constraint [Source: Gobetz [7]] .  | 103 |
| 4.8  | The maximum natural turning angle $\Phi_p$ powered flyby reference trajectory used to categorize all flybys . . . . .   | 105 |
| 4.9  | The two classes of powered gravity assist trajectories are split into the unconstrained periapse flyby (left) and constrained periapse flyby (right) . . . . .  | 106 |
| 4.10 | Single impulse powered flyby example solutions for unconstrained periapse (type F) cases of $\Phi_r < \Phi_p$ with $V_{\infty}^- < V_{\infty}^+$ (left) and $V_{\infty}^- > V_{\infty}^+$ (right) . . . . .   | 114 |
| 4.11 | Single impulse powered flyby example solutions for $r_{min}$ constrained periapse cases of $\Phi_r > \Phi_p$ with $V_{\infty}^- < V_{\infty}^+$ (type RF: left) and $V_{\infty}^- > V_{\infty}^+$ (type FR: right) . . . . .  | 115 |
| 4.12 | Type F powered gravity assist formulation of transfer maneuver location for analytic BVP solution [Source: Walton [2, 3, 4]] . . . . .  | 118 |
| 4.13 | Solution space of unconstrained periapse case with minimum periapse $\Delta V$ and overall minimum $\Delta V$ (at positive $v_{\Delta V}$ ) solutions shown (overhead view) . . . . .   | 122 |
| 4.14 | Solution space of unconstrained periapse case with minimum periapse $\Delta V$ and overall minimum $\Delta V$ solutions shown (3D view) . . . . .   | 122 |
| 4.15 | Solution neighborhood of unconstrained periapse case containing minimum periapse $\Delta V$ and overall minimum $\Delta V$ solutions. Only half of the potential solution space is displayed here, focusing on the positive true anomalies where the optimal maneuver location occurs . . . . . | 123 |

|      |  |     |
|------|--|-----|
| 4.16 | Optimal minimum impulse $\Delta V$ solution compared with collinear periapse impulse cases (left) and optimal case impulse angle $\Lambda_{\Delta V}$ (right) for varying outgoing $V_\infty^+$ magnitudes, showing the convergence of both solution types to coincide when $V_\infty^- = V_\infty^+$ for Type F flybys . . . . .                                  | 125 |
| 4.17 | Optimal solution periapse radius $r_p$ (left) and $\Delta V$ maneuver radius $r_{\Delta V}$ (right) for varying outgoing $V_\infty^+$ magnitudes, showing the convergence of both solution types to coincide when $V_\infty^- = V_\infty^+$ for Type F flybys . . . . .  | 125 |
| 4.18 | Unconstrained periapse powered gravity assist solution space depicting locus of minimum $\Delta V$ maneuvers (cyan) and locus of collinear/tangential $\Delta V$ maneuvers (red), as well as the global minimum $\Delta V$ location (yellow) about Earth: above view (top left), zoomed at optimal $\Delta V$ maneuver (top right), and 3D view (bottom) . . . . . | 127 |
| 4.19 | Example potential outgoing transfer maneuver locations and trajectories (blue) for $V_\infty^- < V_\infty^+$ and $\Phi_r > \Phi_p$ of a constrained periapse flyby configuration, showing the $\Delta V$ maneuver direction and relative scale (red) . . . . .   | 134 |
| 4.20 | Lambert problem with defined radial and chordal skewed velocity components . . . . .   | 136 |
| 4.21 | Constrained powered gravity assist of $\Phi_r > \Phi_p$ with $V_\infty^- < V_\infty^+$ (type RF: left) and $V_\infty^- > V_\infty^+$ (type FR: right) incorporating outgoing and incoming hyperbolic terminal velocity transfers respectively. . . . .   | 139 |
| 4.22 | Outgoing hyperbolic terminal velocity problem diagram of velocity components as related to flight path angle $\gamma$ at the transfer maneuver location, where the flyby trajectory view is isolated just on the outgoing hyperbolic transfer leg . . . . .  | 139 |
| 4.23 | Example constrained periapse powered gravity assist with minimum $\Delta V$ maneuver near periapse . . . . .   | 146 |
| 4.24 | Similar example constrained periapse powered gravity assist with minimum $\Delta V$ maneuver location receding from periapse . . . . .   | 147 |
| 4.25 | Candidate flyby types for constrained powered gravity assist analysis with periapse restriction either applied to the outgoing leg (left) or applied to the incoming leg (right) . . . . .   | 148 |
| 4.26 | Constrained gravity assist infeasible maneuver location of maneuver that breaches $r_{min}$ constraint . . . . .   | 149 |
| 4.27 | Constrained gravity assist depicting the minimum feasible maneuver location near periapse of each flyby solution set . . . . .   | 150 |
| 4.28 | Example constrained periapse powered gravity assist reduced solution set with feasible maneuver locations (blue) and infeasible maneuver locations (red) shown . . . . .   | 151 |

|      |   |     |
|------|---|-----|
| 4.29 | Example constrained periapse powered gravity assist with minimum $\Delta V$ maneuver near sphere of influence $v_{SOI}$   | 154 |
| 4.30 | Example of candidate constrained periapse powered gravity assist transfer solutions with minimum $\Delta V$ maneuver transfer location near sphere of influence $v_{SOI}$   | 155 |
| 4.31 | Example constrained periapse powered gravity assist with minimum $\Delta V$ maneuver on the boundary at sphere of influence $v_{SOI}$   | 156 |
| 4.32 | Example of candidate constrained periapse powered gravity assist transfer solutions with minimum $\Delta V$ maneuver transfer location on the boundary at sphere of influence $v_{SOI}$   | 157 |
| 4.33 | Example constrained periapse powered gravity assist multimodal solution space depicting both a maximum and minimum $\Delta V$ maneuver locations near periapse and sphere of influence respectively   | 160 |
| 4.34 | Family of Type RF-2 constrained periapse powered gravity assists of $V_\infty^- = 1.1$ km/s case (top) 1.5 km/s case (bottom) with optimal maneuver location locus for each exiting $V_\infty^+$ shown along black line near periapse and outward (overhead views on left and 3D view on right)   | 161 |
| 4.35 | Family of Type RF-T constrained periapse powered gravity assists with optimal maneuver location locus shown at transition condition at $V_\infty^- = 1.635$ km/s, where flyby turning angle is nearly that of the constrained leg (overhead views on left and 3D view on right)   | 163 |
| 4.36 | Family of Type RF-1 constrained periapse powered gravity assists of $V_\infty^- = 1.7$ km/s cases (top) and 1.9 km/s cases (bottom) with optimal maneuver location locus shown from beyond SOI and inward as $V_\infty^+$ increases (overhead views on left and 3D view on right)   | 164 |
| 4.37 | Each family of constrained periapse powered gravity assist's locus of optimal solutions asymptotically stagnates at $\Delta V$ maneuver location for increasingly extreme flyby configurations, clockwise from bottom: Type RF-1, Type RF-2, and Type RF-T  | 165 |
| 4.38 | Example Type RF reduced solution spaces with SOI derivative $\frac{d\Delta V}{dv}$ depicted vertically along $\Delta V$ direction with relative scale, displaying all interior and all boundary solutions (top left and right), as well as a transitioning case from boundary to interior solutions (center) with an additional focused perspective (bottom). | 169 |
| 4.39 | Examples of increasing infeasible regions for Earth Type RF flybys with varying flyby configurations (top left to right: increasing $V_\infty^-$ , top right to bottom left: increasing $\Phi_r$ , and bottom left to right: increasing $r_{min}$ )   | 171 |

|  |     |
|--|-----|
| 4.40 Examples of increasing infeasible regions for Jupiter Type RF flybys with varying flyby configurations (top left to right: increasing $\Phi_r$ , top right to bottom left: increasing $V_\infty^-$ , and bottom left to right: increasing $r_{min}$ ) . . . . . | 172 |
| 4.41 Constrained powered gravity assist solution space with minimum $\Delta V$ maneuver locus (cyan) and collinear $\Delta V$ locus (red) for Earth encounters . . . . .   | 174 |
| 4.42 Constrained powered gravity assist solution space with minimum $\Delta V$ maneuver locus (cyan) and collinear $\Delta V$ locus (red) for Jupiter encounters . . . . .   | 176 |
| 4.43 Constrained periapse powered gravity assist pseudo-algorithm . . . . .  | 182 |
| 4.44 Earth Type RF gravity assist optimal $\Delta V$ magnitude and maneuver location for typical lower $V_\infty$ range . . . . .  | 185 |
| 4.45 Earth Type RF gravity assist $\Delta V$ -span of the solution space and # of optimization iterations for typical lower $V_\infty$ range . . . . .   | 185 |
| 4.46 Earth Type RF gravity assist 1 <sup>st</sup> iteration $\Delta V$ accuracy for typical lower $V_\infty$ range . . . . .   | 187 |
| 4.47 Earth Type RF gravity assist optimal maneuver data for typical higher $V_\infty$ range . . . . .  | 189 |
| 4.48 Jupiter Type RF gravity assist optimal maneuver data for typical lower $V_\infty$ range similar to Voyager mission . . . . .  | 192 |
| 4.49 Jupiter Type RF gravity assist optimal maneuver data for typical higher $V_\infty$ range similar to Ulysses mission . . . . .   | 194 |
| 4.50 Jupiter Type RF gravity assist optimal maneuver data for very high $V_\infty$ range similar to New Horizons mission . . . . .   | 195 |
| <br>   |     |
| 5.1 Gravity assist example of collinear $\Delta V$ maneuver at asymptote . . . . .   | 201 |
| 5.2 Gravity assist example of non-tangential $\Delta V_\infty$ maneuvers at asymptote: $\Phi_{max} < \Phi_r$ (left) and $\Phi_{min} > \Phi_r$ (right) . . . . .  | 205 |
| 5.3 Varying asymptote $\Delta V$ maneuver locations and $\Delta V$ directions (collinear or non-tangential) for decreasing $\Phi_r$ assuming $V_\infty^- < V_\infty^+$ . . . . .   | 208 |
| 5.4 Juno mission trajectory: Launch from Earth, DSM near aphelion, Earth gravity assist, and capture into parking orbit at Jupiter . . . . .   | 211 |
| 5.5 Juno mission trajectory using rough initialized guess parameters: Launch from Earth, DSM near aphelion, Earth gravity assist, and capture into parking orbit at Jupiter . . . . .  | 219 |
| 5.6 Galileo mission trajectory: Launch from Earth, Venus-Earth-Earth gravity assist, and capture into parking orbit at Jupiter . . . . .   | 221 |
| 5.7 Galileo mission inner planet trajectory: Launch from Earth, Venus-Earth-Earth gravity assist . . . . .   | 222 |

|      |  |     |
|------|--|-----|
| 5.8  | Galileo mission trajectory with rough initial guess parameters: September launch from Earth, Venus-Earth-Earth gravity assist, and capture into parking orbit at Jupiter (top), close-up inner planet trajectory view (bottom) . . . . .   | 226 |
| 5.9  | Galileo mission trajectory with rough initial guess parameters: November launch from Earth, Venus-Earth-Earth gravity assist, and capture into parking orbit at Jupiter (top), close-up inner planet trajectory view (bottom) . . . . .  | 227 |
| 5.10 | Periapse impulse powered gravity assist with $\Delta V$ constraint applied with a penalty $\Delta V$ maneuver, showing infeasible trajectory (dashes) and feasible trajectory (solid) . . . . .  | 232 |
| 5.11 | Optimal impulse powered gravity assist with maximum $\Delta V$ constraint applied at $r_{\Delta V}$ and a $\Delta V_\infty$ correction maneuver applied at the asymptote, showing infeasible trajectory that requires to large of $\Delta V$ at $r_{\Delta V}$ (dashes) and feasible trajectory using an acceptable $\Delta V$ magnitude at $r_{\Delta V}$ (solid) . . . . . | 233 |
| 6.1  | New Horizons mission trajectory: Launch from Earth, Jupiter gravity assist, and Pluto flyby [Source: Guo and Farquhar [8]] . . . . .   | 235 |
| 6.2  | New Horizons mission heliocentric velocity plot throughout interplanetary trajectory [Source: Guo and Farquhar [8]] . . . . .  | 236 |
| 6.3  | New Horizons mission trajectory to Kuiper belt post Pluto system flyby [Source: Guo and Farquhar [8]] . . . . .  | 236 |
| 6.4  | New Horizons mission TRACT simulation trajectory . . . . .   | 239 |
| 6.5  | New Horizons mission trajectory comparison of 8.55 year flight time case (blue) with 9.55 year baseline trajectory (red) . . . . .   | 241 |
| 6.6  | MESSENGER mission trajectory first section: Launch from Earth, Earth flyby after 1 year, DSM 1 near perihelion, and first Venus flyby [Source: McAdams et al. [9]] . . . . .   | 243 |
| 6.7  | MESSENGER mission trajectory second section: After first Venus flyby, second Venus flyby after full Venus orbit (synchronous), DSM 2 performed just before completing a full solar orbit, followed by first Mercury flyby [Source: McAdams et al. [9]] . . . . .   | 244 |
| 6.8  | MESSENGER mission trajectory last section: After first Mercury flyby, DSM 3 occurs, then Mercury flyby 2 after about 1.5 solar orbits. DSM 4 is performed about 2.5 solar orbits before Mercury flyby 3, followed by DSM 5 and a 4.5 solar orbit waiting period before MOI. [Source: McAdams et al. [9]] . . . . .   | 245 |
| 6.9  | MESSENGER mission trajectory simulation first section, similar to the actual trajectory except the shift in the first leg: Launch from Earth, Earth flyby after 1 year, DSM 1 near perihelion, and first Venus flyby . . . . .   | 249 |

|      |  |     |
|------|--|-----|
| 6.10 | MESSENGER mission trajectory simulation second section, very similar to the actual trajectory: After first Venus flyby, second Venus flyby after full Venus orbit (synchronous), DSM 2 performed just before completing a full solar orbit, followed by first Mercury flyby  | 250 |
| 6.11 | MESSENGER mission trajectory simulation last section, very similar to the actual trajectory: After first Mercury flyby, DSM 3 occurs, then Mercury flyby 2 after about 1.5 solar orbits. DSM 4 is performed about 2.5 solar orbits before Mercury flyby 3, followed by DSM 5 and a 4.5 solar orbit waiting period before MOI   | 251 |
| 6.12 | Cassini mission trajectory: Launch from Earth, followed by a Venus flyby, then a DSM maneuver at aphelion, a second Venus flyby, with an Earth flyby shortly after, and a Jupiter flyby before reaching Saturn for insertion into a highly eccentric capture orbit (top), close-up inner planet trajectory view (bottom).  | 253 |
| 6.13 | Cassini trajectory simulation depicting an 11.7 year flight time minimum $\Delta V$ trajectory solution with 290 m/s lower total $\Delta V$ compared to actual mission, starting from actual mission initial guess values.   | 255 |
| 6.14 | Student design team mission trajectory to Jupiter similar to Juno mission trajectory: Earth launch (1) followed by $V_\infty$ -leveraging DSM (2) and Earth gravity assist (3) to transfer to Jupiter for insertion into capture orbit (4)   | 257 |
| 6.15 | Example orbital lowering from capture orbit about Jupiter using Ganymede gravity assists shown above, followed by a DSM (5) and orbit insertion about Europa (6) shown below   | 258 |
| 6.16 | Student design team mission to dwarf planet Eris (overhead view above and out-of-plane view below): Launch from Earth followed by a Jupiter and Neptune gravity assist combination before reaching Eris in only 18 years   | 260 |
| A.1  | Optimal feasible flyby solution $\Delta V$ is shown to increase and slowly diverge away from periapse maneuver solutions as $V_\infty^+$ increases away from $V_\infty^-$ magnitude for unconstrained $r_p$ flyby cases, then the optimal feasible maneuver $\Delta V$ converges back to periapse maneuver solution line as flyby $r_p$ is restricted by $r_{min}$ (left). The associated effect on impulse angle $\Lambda_{\Delta V}$ is shown where unconstrained flyby maneuvers were growing away from velocity alignment (collinear) as $V_\infty^+$ increases away from $V_\infty^-$ , then the impulse angle $\Lambda_{\Delta V}$ quickly drops back to collinear aligned maneuvers for $r_p$ constrained flybys approaching the maximum turning angle $\Phi_p$ conditions (right). | 271 |

|     |   |     |
|-----|---|-----|
| A.2 | Zoomed in view of optimal feasible flyby solution $\Delta V$ converges to periapse maneuver solution line as flyby $r_p$ is restricted by $r_{min}$ (left). The associated effect on impulse angle $\Lambda_{\Delta V}$ is shown to drastically change its trend of increasing away from collinear aligned maneuvers to quickly dropping back towards collinear alignment for $r_p$ constrained flyby cases as the flyby configuration approaches the maximum turning angle $\Phi_p$ conditions (right) . . . . . | 272 |
| A.3 | Optimal flyby maneuver periapse radius is shown transitioning from an unconstrained periapse flyby to a periapse constrained flyby (left) and the associated optimal $\Delta V$ maneuver radius transitioning to coincide with the periapse maneuver location for constrained flyby cases (right) for $\Phi_p$ turning angle case . . . . .   | 273 |
| A.4 | Optimal maneuver periapse radius meeting $r_{min}$ constraint (left) and the associated optimal $\Delta V$ maneuver radius transitioning to periapse for constrained flyby cases (right): zoomed-in view . . . . .  | 274 |
| B.1 | Optimal feasible solution $\Delta V$ is shown to be restricted from reducing to zero as $V_\infty^+$ approaches $V_\infty^-$ for $r_{SOI}$ and $r_{max}$ upper bound constrained cases (left). The associated effects on the impulse angle $\Lambda_{\Delta V}$ is shown, as $V_\infty^+$ approaches $V_\infty^-$ the unconstrained case impulse was becoming more aligned with the velocity, but abruptly begins increasing away from collinear conditions for constrained cases (right) . . . . .               | 277 |
| B.2 | Optimal feasible solution periapse is shown to increase as $V_\infty^+$ approaches $V_\infty^-$ , until restricted at $r_{SOI}$ (left). The $\Delta V$ maneuver radius has slow near-linear increase as $V_\infty^+$ approaches $V_\infty^-$ for unconstrained $r_p$ flyby cases, then it shows a nonlinear increase for constrained $r_p$ flybys until the maneuver radius is restricted at $r_{max}$ (right) . . . . .  | 278 |
| B.3 | Optimal feasible solution space example for $r_{SOI}$ and $r_{max}$ upper bound constrained cases, displaying the concave solution space characteristic of these special types of flyby configurations. The optimal maneuver location is always found at $r_{max}$ boundary or at periapse, where $r_p = r_{SOI}$ . . . . .   | 278 |
| B.4 | Optimal feasible solution $\Delta V$ for $r_{SOI}$ and $r_{max}$ upper bound constrained cases, showing the change in the $\Delta V$ solution line dynamics as the maneuver location switches from the $r_{max}$ boundary to periapse at the lower $V_\infty^+$ region presented (left). The associated effects on the impulse angle $\Lambda_{\Delta V}$ is displayed with imperceptible changes during the maneuver location switch (right) . . . . .   | 280 |

|     |  |     |
|-----|--|-----|
| B.5 | Optimal solution periapse radius is shown to be maintained at $r_{SOI}$ for all flyby configurations presented (left). The optimal $\Delta V$ maneuver location radius is clearly shown to transition from the $r_{max}$ boundary to periapse at $r_{SOI}$ as $V_\infty^+$ approaches $V_\infty^-$ for these extreme flyby conditions (right).                       | 280 |
| B.6 | Optimal maneuver location true anomaly $v$ is clearly shown to transition from the $r_{max}$ boundary to periapse as $V_\infty^+$ approaches $V_\infty^-$ for these extreme flyby conditions (left). The associated concave solution space example shows the periapse boundary with lower $\Delta V$ magnitudes and therefore the optimal maneuver location (right). | 281 |
| C.1 | General hyperbolic terminal velocity vector hodograph diagram [Source: Battin [10]]  | 283 |
| C.2 | Hyperbolic terminal velocity vector hodograph diagram for transfer to $V_\infty$ asymptote   | 284 |
| D.1 | GTOC 7 mothership trajectory formulated and optimized with TRACT   |     |
|     | 288  |     |

# **Chapter 1**

## **Introduction**

### **1.1 Preliminary interplanetary trajectory problem definition**

The advancement of modern spacecraft technology and launch vehicle capability has enabled more ambitious interplanetary spacecraft missions. Achieving the science and mission goals with realistic spacecraft constraints at increasingly challenging destinations necessitates efficient and complex trajectory designs. The trajectories for these complex interplanetary missions must be designed in an effective and adaptable manner that supports the ever changing mission goals. To better explore the vastly different types of potential interplanetary trajectories for each preliminary mission design, mission designers require efficient, robust, and adaptable preliminary interplanetary trajectory design tools and algorithms.

A preliminary interplanetary trajectory can be defined as a spacecraft trajectory, approximated by simpler dynamics compared to reality, that travels through interplanetary space potentially encountering planets and/or small bodies along its path to a final destination orbit about a planet, small body, or the Sun. The simpler dynamics are typically selected as two-body orbital mechanics, which assumes a spacecraft of negligible mass to orbit about a central massive gravitational body, such as the Sun or planets. Changes in the trajectory due to spacecraft propulsion are modeled as impulsive perturbations in velocity  $\Delta V$ . Each required  $\Delta V$  maneuver

can be related to the propellant required to complete the propulsive maneuver by the well known rocket equations. Common spacecraft trajectory maneuvers included in preliminary interplanetary trajectories are departure/arrival transfers into parking orbits, departure on launch vehicle hyperbolic transfers, arrival to planetary entry conditions, ballistic and powered gravity assists, as well as deep space maneuvers.

Interplanetary spacecraft missions frequently utilize planetary gravity assists in their trajectory design. A gravity assist can be defined as a hyperbolic flyby of a celestial body, where the local gravitational perturbations due to the planet or small body aid in providing energy for and shaping the overall interplanetary trajectory. Many interplanetary trajectories benefit from including multiple consecutive gravity assists, which greatly increases the complexity of the trajectory design process. The use of planetary gravity assists is an effective way to reduce the required launch vehicle energy supplied to the spacecraft, which also can reduce the mission cost and/or increase the spacecraft mass and capability. Planetary gravity assists usually aid in reducing the amount of propellant required onboard the spacecraft, as well as potentially decreasing the flight time required to get to the destination.

A gravity assist can be ballistic (no thrust event during the flyby) or powered (with one or more thrust events during the flyby). As will be explored in the following chapters, a powered gravity assist can enhance the effectiveness of a planetary gravity assist, further reducing the overall trajectory  $\Delta V$ , therefore reducing the amount of required spacecraft propellant. An optimal single-impulse powered gravity assist trajectory applies the spacecraft  $\Delta V$  impulse at an optimal maneuver location during the flyby that minimizes the required  $\Delta V$  maneuver. Unfortunately, a powered gravity assist has not been performed in an actual flown mission. There

are multiple reasons that current and past missions have avoided implementing a powered gravity assist, stemming from technological and operational difficulties. The sensitive dynamics during a flyby require that an engine burn maneuver be performed during the flyby with very high accuracy requirements, beyond that of current navigation technology. Also, including such a precise maneuver during a gravity assist requires complex mission planning and/or automated spacecraft guidance strategies, which incurs higher risk to the mission. Considering these and other reasons attributed to not utilizing powered gravity assists, improved spacecraft navigation technology and onboard automation will enable the use of powered gravity assists in future missions and broaden the horizon of achievable spacecraft destinations [11]. Anticipating these improved spacecraft capabilities, the current research provides the tools to aid in designing and planning such future missions.

## 1.2 Motivation

As an assistant instructor for the spacecraft mission design course series at the University of Texas at Austin, the author noticed a need for an interplanetary trajectory design tool and associated algorithms that the students could quickly learn to use and produce realistic interplanetary trajectories. The difficulty in generating feasible preliminary trajectories was a major stumbling block in the mission designs of the undergraduate student teams. Many student teams relied almost totally on the literature for their baseline trajectories, and often did not progress very far beyond these trajectories in their final designs. Some teams attempted to use existing trajectory design tools, but failed because these tools required extensive time learning the software. Thus the author developed TRACT (TRAjectory Configuration Tool)

as a preliminary interplanetary trajectory design tool, that can incorporate common mission maneuvers (parking orbit transfers, rendezvous, etc.) as well as advanced maneuvers (ballistic and powered gravity assists, etc.), while maintaining a short user learning curve. This was achieved by reducing the amount of user provided mission information to a minimum and employing two-body boundary value problem (BVP) algorithms to automate much of the trajectory evaluation.

While developing TRACT, the author found the literature and research lacking in characterizing optimal powered gravity assists, particularly needing a complete and rigorous characterization of optimal single-impulse powered gravity assists. Most optimal minimum  $\Delta V$  powered flyby literature presented multiple-impulse globally optimal powered gravity assist solutions as academic exercises. Many of these multiple-impulse powered gravity assist problems resulted in solutions that aren't translatable to feasible mission trajectories (requiring extremely long flight times about the planet to execute the gravity assist or flybys that penetrate the central planet). Consequently, an in-depth study of mission feasible single-impulse optimal powered gravity assist problem was conducted, characterizing its features and dynamics for an organized understanding of the problem and its solutions. This characterization lead to developing a new BVP algorithm for optimal single-impulse powered gravity assist solutions, as well as to alternative methods and algorithms for obtaining ballistic gravity assist trajectory solutions.

## 1.3 Contributions

The major contributions of this research to the areas of preliminary trajectory design and fundamental astrodynamics are listed below.

1. An interplanetary trajectory design tool (TRACT) has been designed, developed, and documented, using a new bi-level optimization formulation of two-body trajectory boundary value problems that provides efficient and robust trajectory optimization solutions with a reduced set of required user inputs. TRACT offers enhanced accessibility and a short user learning curve, as well as providing an adaptable open source architecture in MATLAB for specialized user modifications.
2. A complete and rigorous characterization of the optimal single-impulse powered gravity assist problem is presented, organizing the transfer maneuver solution types into a convenient categorization scheme and identifying key solution space features for each category.
3. A new single-impulse powered gravity assist boundary value problem is formulated. The problem is then solved with an efficient and robust cubic interpolation optimization algorithm. The solution algorithm features strict enforcement of minimum and maximum flyby radius constraints.
4. The powered assist algorithms have been integrated into the TRACT design tool.
5. The TRACT tool has been tested and verified by using it to re-create preliminary mission design trajectories for the Galileo, Cassini, Juno, New Horizons,

and MESSENGER missions. The resulting preliminary mission design trajectories successfully converge to trajectories that closely match the actual trajectory timelines, maneuvers, and propulsion requirements.

Additionally, the research presents an alternative ballistic gravity assist constraint methodology and boundary value problem solution algorithm. The algorithm is also incorporated into TRACT and analyzed with mission trajectories, showing potential improvements in optimization performance.

## 1.4 Dissertation organization

The research is divided into seven chapters. The introductory chapter provides the interplanetary trajectory design problem definition, the general motivations of the research, and identifies the major contributions of the work presented. Chapter 2 provides necessary background information on preliminary interplanetary trajectory optimization strategies using linked conics, two-body orbital mechanics, and common BVPs, such as Lambert targeting and the hyperbolic terminal velocity problem. Chapter 3 presents TRACT, detailing the tool's architecture, capabilities, and bi-level optimization scheme utilizing a linked BVP methodology. Chapter 4 presents the powered gravity assist BVP beginning with a review of past literature and common algorithms for analysis of powered gravity assists. The chapter then provides a rigorous formulation of the optimal single-impulse powered gravity assist problem, presents a thorough characterization of its solution space, and then presents an efficient and robust algorithm for the solution of the BVP. Chapter 5 develops an asymptote  $\Delta V$  correction maneuver technique to accommodate ballistic

gravity assist trajectories using a penalty maneuver rather than applying nonlinear optimization constraints. The chapter analyzes the resulting algorithm for its performance and effectiveness at producing ballistic gravity assist trajectory solutions. Chapter 6 presents interplanetary mission trajectory case studies with examples of mission trajectories from past student design teams using TRACT. The chapter also provides detailed trajectory optimization performance analysis of real interplanetary mission trajectories using TRACT, as well as demonstrating the effectiveness of optimal powered gravity assists, achieving ballistic gravity assists with the asymptote penalty maneuver, and other TRACT capabilities. Finally, Chapter 7 concludes the dissertation with a summary of the work presented and outlining recommendations for further study. Appendix A and B at the end of the dissertation provides extended discussions on powered flyby constraint handling. Appendix C discusses additional hyperbolic terminal velocity hodograph analysis for powered gravity assists. Appendix D details the contributions made with TRACT for the 7<sup>th</sup> international Global Trajectory Optimization Competition. Finally, the Appendix E - G present additional details for TRACT: optimization schemes, constraints, and user input options.

## **Chapter 2**

# **Multi-body Trajectory Design & Optimization Background**

### **2.1 Trajectory design & optimization strategies**

The current trajectory design process for an interplanetary mission begins by defining the design space (the total set of variables and boundary conditions) and outlining a method of exploring that space (a mission scenario). Mission parameters and constraints help to identify requirements for the trajectory design space. There are many strategies and tools available that may be used to design and explore within these boundaries. The strategies and tools are typically dictated by the maturity level of the mission and/or trajectory design. Mature designs have more information and data precision which correlates to needing a higher fidelity strategy such as utilizing complex multi-body dynamical system tools. Less mature mission designs can have drastically varying objectives and lack information to accommodate such sophisticated tools. There may be many different mission scenarios that should be explored during the preliminary mission design process. These preliminary mission designs must rely on simpler two-body orbital relations and more adaptable trajectory tools.

The sections in this chapter develop the theoretical background and algorithms necessary to fully appreciate the work in later chapters. The two-body or-

bital relationships and algorithms presented in this chapter are well known in astrodynamics and are presented as an introduction and segway to the original work found in later chapters. Many of the diagrams and figures presented are created by the author, but some are excerpts from other sources, such as Curtis [5] or Vallado [1]. These referenced figures and diagrams are noted by the source citation within the caption.

### 2.1.1 Multi-body systems

A true interplanetary trajectory encounters multiple gravitational forces at all times throughout a mission as well as other perturbations, such as solar radiation pressure, atmospheric drag, magnetic forces, etc. This complex set of forces requires numerical integration and detailed knowledge of the mission environment to determine a complete trajectory solution . These trajectories are typically broken into separate discontinuous propagated segments, where a continuous trajectory that satisfies all the mission constraints is generated iteratively through differential correction procedures [12, 13]. There are many powerful software toolkits available commercially or native to companies/institutions which provide these capabilities. Some well known industry examples are Copernicus, AGI's Systems Tool Kit (STK), JPL's MALTO and VARITOP, etc. High fidelity mission trajectory tools are very computationally intensive and are extremely sensitive to the initial guess trajectory values provided. The focus of the current research is to provide a more robust, efficient, and accurate way to produce initial trajectories that can be used as starting points for high fidelity mission trajectory programs.

Table 2.1: Characteristics of Keplerian Orbits [1]

| Quantity                       | Circle                      | Ellipse                  | Parabola          | Hyperbola         |
|--------------------------------|-----------------------------|--------------------------|-------------------|-------------------|
| Semimajor Axis, $a$            | $a = r$                     | $r_p \leq a \leq r_a$    | $a = \infty$      | $a < 0$           |
| Eccentricity, $e$              | $e = 0$                     | $0 < e < 1$              | $e = 1$           | $e > 1$           |
| Semiparameter, $p$             | $p = r$                     | $r_p < p < 2r_p$         | $p = 2r_p$        | $p > 2r_p$        |
| Period, $P$                    | $P = 2\pi\sqrt{a^3/\mu}$    | $P = 2\pi\sqrt{a^3/\mu}$ | $P = \infty$      | $P = \infty$      |
| Specific Energy, $\varepsilon$ | $\varepsilon = -\mu/2a < 0$ | $\varepsilon < 0$        | $\varepsilon = 0$ | $\varepsilon > 0$ |

### 2.1.2 Two-body systems

The two-body systems considered in this dissertation assumes an environment with a primary central gravitational point mass body (Sun or planet) and a much smaller secondary body of negligible mass, such as a spacecraft. Two-body dynamics offer a wealth of analytic relationships and closed form solutions. These dynamics provide a robust and efficient method for approximating multi-body trajectories and when applied in an appropriate manner, offer acceptable accuracy for preliminary trajectory design. Since the following chapters frequently reference two-body relations in derivations and analysis, key equations are outlined below along with specific syntax and nomenclature.

Two-body dynamical motion is described by the relatively simple equation of motion, Equation 2.1, which determines all of the relationships outlined in Table 2.1 and throughout the chapter.

$$\ddot{\vec{r}} = -\frac{\mu}{r^3} \vec{r} \quad (2.1)$$

The conic orbits about a celestial body can be described analytically in the two-body problem by elliptic and hyperbolic orbital locations solved as a scalar

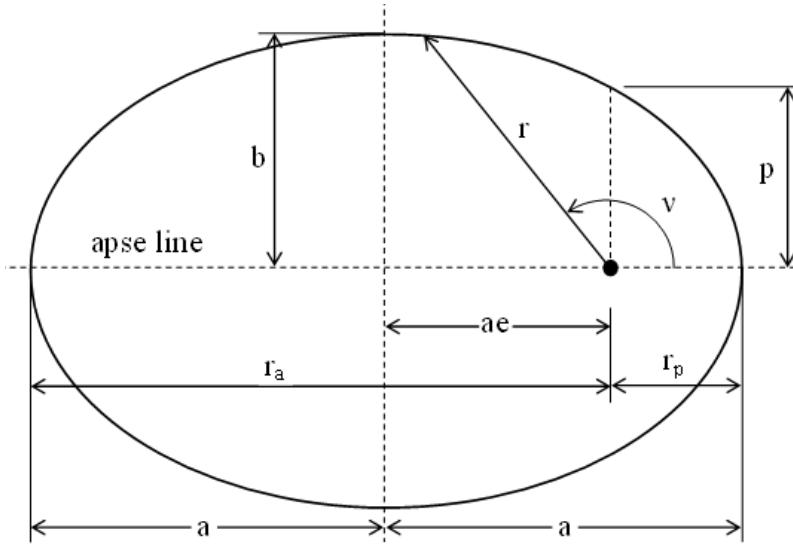


Figure 2.1: Elliptical orbit with defining parameters labeled

radius. The semimajor axis and eccentricity are used with the true anomaly of the orbit to define the radius of the specific location in the orbit, Equation 2.2.

$$r = \frac{a(1 - e^2)}{1 + e \cos(v)}. \quad (2.2)$$

In most trajectory designs, elliptical orbits account for a majority of the interplanetary flight, including parking orbits about planets and most segments of the heliocentric trajectory. The geometry and parameters that will be used to define an elliptic orbit are outlined in Figure 2.1. The total energy of an orbit is conserved, equaling the sum of the potential and kinetic energy of the spacecraft at any point in the orbit, where  $V$  and  $\mu$  are the velocity at the point  $r$  and gravitational parameter of the primary body, respectively [1].

$$\epsilon = \frac{V^2}{2} - \frac{\mu}{r} = -\frac{\mu}{2a} \quad (2.3)$$

When combined with the specific energy, Equation 2.3 leads to the vis-viva equation for the spacecraft's velocity magnitude,  $V$ . As the trajectory of a hyperbolic orbit increases to a sufficiently large radius that the potential energy term is negligible, the velocity of the orbit approaches an asymptotic velocity magnitude. This is seen by the  $2\mu/r$  term in the vis-viva equation being driven to zero as  $r$  approaches infinity, leaving the asymptotic excess velocity  $V_\infty$  of the hyperbolic orbit, Equation 2.5. Note that the standard sign convention for semimajor axis of hyperbolic orbits is negative, which maintains a real valued velocity from these equations.

$$V = \sqrt{\frac{2\mu}{r} - \frac{\mu}{a}} \quad (2.4)$$

$$V_\infty = \sqrt{-\frac{\mu}{a}} \quad (2.5)$$

Specific points of interest for an elliptical orbit are the periapsis  $r_p$  and apoapsis  $r_a$ . The periapsis is the location in the orbit for the closest approach and highest velocity with respect to the gravitational body. The apoapsis is the location in the orbit for the farthest point and lowest velocity. An elliptical orbit is a closed orbit with negative specific energy, where (in a two-body system) a spacecraft will remain in orbit about the gravitational body for all time. A hyperbolic orbit is an example of an open orbit with positive specific energy in which a spacecraft is not bound to the celestial body.

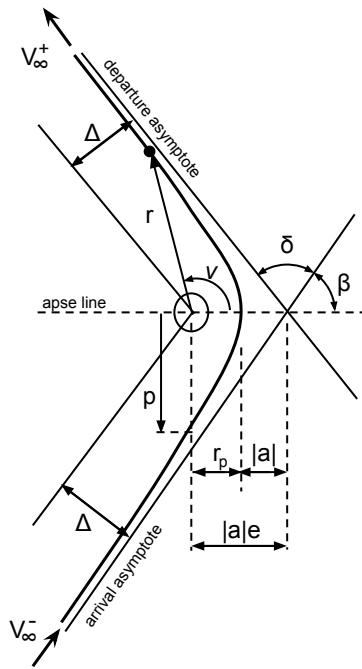


Figure 2.2: Hyperbolic orbit diagram with defining parameters labeled

Hyperbolic orbits appear in trajectory designs to describe a spacecraft's departure, arrival, or flyby of a celestial body, involving only brief periods of time compared to the entire interplanetary trajectory's duration. This high energy orbit is illustrated in Figure 2.2 with its defining parameters. All vector parameters are identified throughout the dissertation by a vector arrow over the parameter. In many mission planning and trajectory design investigations, a method of measuring the hyperbolic orbital energy is the parameter  $C_3$ , which has units  $km^2/s^2$ . The equations relating  $C_3$ ,  $V_\infty$ , and  $\epsilon$  are shown below.

$$C_3 = V_\infty^2 = 2\epsilon \quad (2.6)$$

Because a hyperbolic orbit is an open orbit and continues its flight to an “infinite”

distance from the gravitational body, only a periapsis exists in a two body model. Although the periapsis is the location of the highest velocity in the hyperbolic orbit, there is no specific point for the lowest velocity. Yet, as an object's distance from the gravitational body grows, the velocity of a hyperbolic orbit slows to a near constant value and direction, which is approximated as,  $V_\infty$ , (known as hyperbolic excess velocity) [14].

The directions of both approaching and exiting  $V_\infty$  are depicted in Figure 2.2 by an asymptote, translated a distance  $\Delta$ , known as the aiming radius or miss distance, from a line drawn radially out from the body. The hyperbolic eccentricity can be determined by the periapse radius  $r_p$  for a given  $V_\infty$ .

$$\Delta = \frac{\mu}{V_\infty^2} \sqrt{(e^2 - 1)} \quad (2.7)$$

$$e = 1 + \frac{r_p V_\infty^2}{\mu} \quad (2.8)$$

The turning angle,  $\delta$ , is a commonly referenced parameter that describes how much an encounter with a celestial body has changed the  $V_\infty$  direction in planet centered coordinates. The location of the periapsis can be defined with respect to either asymptote line in celestial body fixed coordinates by the angle  $\beta$ , herein referenced as the gravitational bend angle.

$$\delta = 2 \sin^{-1} \left( \frac{1}{e} \right) \quad (2.9)$$

$$\beta = \cos^{-1} \left( \frac{1}{e} \right) \quad (2.10)$$

In the case of an unpowered gravity assist, a non-thrusting ballistic planetary flyby, the incoming and exiting  $V_\infty$  magnitudes are equal. Parameter values associated with the arrival segment of the gravity assist trajectory are identified by a minus ( $-$ ) and those associated with the departure segment is identified by a plus ( $+$ ). The use of the plus minus symbol ( $\pm$ ) as a superscript or subscript references both the arrival and departure segment parameters, such as  $V_\infty^\pm$ . It can also indicate either segment according to the appropriate case,  $V_\infty^-$  for certain cases and  $V_\infty^+$  for others.

$$V_\infty^- = V_\infty^+ \quad (2.11)$$

## 2.2 Patched/linked conics

The patched conic and linked conic methods utilize analytical two-body equations to simplify the trajectory calculation process. Since a preliminary trajectory design usually involves minimal mission information, the patched conic or linked conic procedure is preferred for early mission planning. Both conic approaches assume only the dominant gravitational field acts on the spacecraft at each stage of the flight and all other forces are negligible. The trajectory is divided into heliocentric and planetocentric (planet centered) stages, as shown in Figure 2.3. The radial distance surrounding a planet where its gravity becomes dominant over the Sun's is defined as the sphere of influence, SOI, of that planet. A common equation for defining the SOI radius for a planet is Equation 2.12.

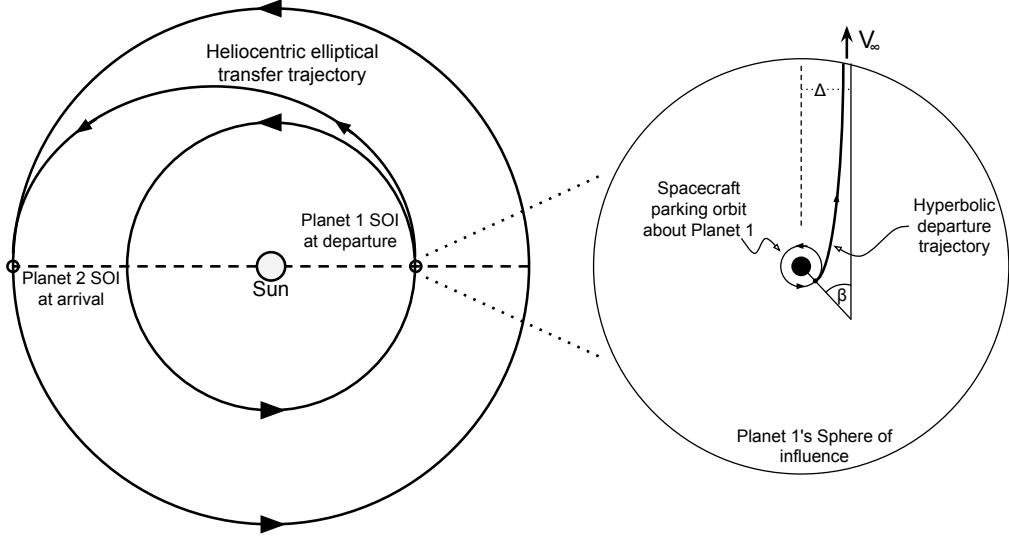


Figure 2.3: Patched conic trajectory from inner to outer planet (not to scale)

$$r_{SOI} = R_p \left( \frac{m_p}{m_{sun}} \right)^{\frac{2}{5}} \quad (2.12)$$

The difference between the conic methods is that linked conic method is a simplified version of patched conics. Patched conics switches the central gravitational body from the Sun to the planet once the trajectory has entered into the planet's SOI, truncating the heliocentric conic. Therefore the asymptotic excess velocity  $\vec{V}_\infty = \vec{V}_{sc} - \vec{V}_P$  for the hyperbolic planetocentric trajectories are defined by the planet's velocity  $\vec{V}_P$  and the spacecraft heliocentric velocity  $\vec{V}_{sc}$  at the moment of breaching  $r_{SOI}$ . This method usually requires an iterative procedure to determine the conditions when the  $r_{SOI}$  transition occurs. Linked conics assumes the SOI for the planets is small enough to be neglected and the heliocentric trajectory is a complete conic from an initial planet's position to the final planet's position. This simplification removes the need for any iteration since the  $\vec{V}_\infty$  for each planetary

encounter can be determined directly from the planet and spacecraft heliocentric velocities at the intersection points. Any maneuvers performed during the planetary encounter are assumed to occur in a relatively short time frame, such that it can be neglected and considered instantaneous compared to the heliocentric trajectory. The instantaneous planetary encounter approximation is supported by most impulsive interplanetary trajectories having hyperbolic planetary encounters with relatively fast planetocentric transits. The instantaneous encounter allows the planet's heliocentric state to be assumed fixed during the encounter. Therefore a gravity assist would relate both  $\vec{V}_\infty^-$  and  $\vec{V}_\infty^+$  with the same planetary velocity  $\vec{V}_P$ . These assumptions have led the linked conic method to also be known as zeroth order patched conics or zero sphere of influence patched conics. From this point on linked conics will be the preferred method used throughout this research, as this approximation allows a convenient and sufficiently accurate way to divide a complex single level optimization problem into a much simpler bi-level optimization procedure.

The linked conic method provides a close approximation to the true trajectory, because the SOI of each planet is very small compared to the scale of the entire heliocentric trajectory, spending the majority of the trajectory under the sun's gravitational influence [5]. Consequently, the heliocentric conic legs of the trajectory are determined while treating the planets' SOI as points in space, having no effect on the heliocentric trajectory except at the flyby point itself. Although, when considering each trajectory in planetocentric coordinates, the SOI is a large enough radial distance from the planet it is considered to be at infinity. Using values from the *Explanatory Supplement to the Astronomical Almanac* (Seidelmann 1992) [15], each planet's SOI and the various relationships as described above are found in Table

Table 2.2: Relationships of planets' sphere of influence with distances from Sun and planets

| Planet  | $r_{planet}$<br>(km) | $R_{planet}$<br>(AU) | $R_{SOI}$<br>(AU) | $R_{SOI}$<br>( $r_{sun}$ ) | $\frac{R_{planet}}{R_{SOI}}$ | $\frac{R_{SOI}}{R_{planet}}$<br>(%) | $\frac{R_{SOI}}{r_{planet}}$ |
|---------|----------------------|----------------------|-------------------|----------------------------|------------------------------|-------------------------------------|------------------------------|
| Mercury | 2440                 | 0.39                 | 0.0008            | 0.16                       | 515                          | 0.19                                | 46                           |
| Venus   | 6052                 | 0.72                 | 0.0041            | 0.89                       | 176                          | 0.57                                | 102                          |
| Earth   | 6378                 | 1.00                 | 0.0062            | 1.33                       | 162                          | 0.62                                | 145                          |
| Mars    | 3396                 | 1.52                 | 0.0039            | 0.83                       | 395                          | 0.25                                | 170                          |
| Jupiter | 71492                | 5.20                 | 0.3222            | 69.26                      | 16                           | 6.19                                | 674                          |
| Saturn  | 60268                | 9.55                 | 0.3653            | 78.52                      | 26                           | 3.82                                | 907                          |
| Uranus  | 25559                | 19.22                | 0.3462            | 74.41                      | 56                           | 1.8                                 | 2026                         |
| Neptune | 24764                | 30.11                | 0.5808            | 124.83                     | 52                           | 1.9                                 | 3509                         |

2.2. The planetary radius and average heliocentric orbital distance of each planet ( $r_{planet}$  and  $R_{planet}$ ) are shown to compare the radius of each planet's SOI ( $R_{SOI}$ ) in terms of astronomical units (AU) and in radius of the sun ( $r_{sun}$ ). Various ratios are also displayed to compare how  $R_{SOI}$  measures up to  $R_{planet}$  and  $r_{planet}$ .

When encountering the largest planet Jupiter, the linked conics assumptions may not agree with reality as well as it does with the smaller planets. Jupiter's gravitational field has significant influence on a trajectory as a spacecraft approaches the vicinity of the planet. The large distance of gravitational influence alters a trajectory by increasing the spacecraft's velocity toward Jupiter. The increase in heliocentric velocity is not accounted for in linked conics. This results in the actual trajectory leg to Jupiter having a shorter time of flight than what is estimated by linked conics. The effect on a flight ending at Jupiter is not negligible, but the overall effect on a Jupiter flyby is partially canceled out, where the incoming leg would have a shorter time of flight and the outgoing leg would have a longer time of flight. For these reasons linked conics is primarily used for preliminary mission planning tools.

Each individual heliocentric and planetocentric segment of the trajectory is a continuous path between maneuvers/encounters. The linked conic method assumes the location of the  $\vec{V}_\infty$  asymptote is free to be adjusted with respect to the encountered planet, as defined by hyperbolic parameters such as the periapse radius  $r_p$ , the asymptote offset distance  $\Delta$ , etc.

The key component of the linked conic method that makes it so appealing is that it allows a complex trajectory problem to be broken up into smaller trajectory problems solved independently (to a certain extent) of one another. The heliocentric trajectory segments of an interplanetary mission connect between each planetary encounter location and/or deep space maneuver (DSM). Implementing the linked conics method, the trajectories of the heliocentric legs are decoupled from each other and can be solved independently. Linked conics also allows planetocentric trajectories to be decoupled from other planetocentric trajectories, but each is still dependent on the  $\vec{V}_\infty$  information provided by the heliocentric trajectory solutions. Therefore the heliocentric trajectories must be solved first, then each planetocentric trajectory can be solved independently of the others. This decoupling makes linked conics a powerful technique, especially for computationally intensive trajectory problems, where parallel computing can be implemented easily for the all the smaller independent trajectory problems.

The planetary states (heliocentric position and velocity) are found using the encounter dates and ephemeris data. Knowing each planet's position as a function of time, the heliocentric trajectory legs are determined using a two-body method such as Lambert targeting or a propagation method. Once the heliocentric trajectories are computed, each planetocentric hyperbolic encounter is defined by the in-

coming and/or exiting excess velocity,  $\vec{V}_\infty^-$  and  $\vec{V}_\infty^+$ . Despite the goal of the planetary encounter whether it is a flyby, transfer to or from a parking orbit, etc., a hyperbolic transfer occurs between the  $\vec{V}_\infty$  asymptote and the desired objective. This hyperbolic trajectory between a  $\vec{V}_\infty$  asymptote direction a planetocentric  $\Delta V$  maneuver location  $\vec{r}_{\Delta V}$  defines a hyperbolic boundary value problem, which can be solved analytically. If the objective is to transfer with a parking orbit about a planet, then the orbital state at the maneuver location is the final information required to complete the encounter. If a gravity assist is desired, then the hyperbolic boundary value problem is solved for the incoming leg and outgoing leg of the flyby trajectory, connecting the planetocentric flyby transfer trajectory. Each of these tools and methods are described in further detail in the next sections.

## 2.3 Boundary value problem methods

Boundary value problems (BVP) in orbital mechanics are typically defined as trajectory problems where orbital trajectory parameters are known for an initial state and a final state, but the trajectory required to transition between these states is unknown. Some orbital BVP have only a single unique trajectory solution that connects the initial and final states and satisfies all the conditions. Other orbit BVPs can have multiple trajectories that complete all the conditions, therefore requiring an additional metric to determine a single unique solution, such as minimizing/maximizing certain orbital parameters or a specified performance index. BVPs are essential to efficiently solving two-body trajectories. Occasionally BVPs are formulated to have analytical solutions, otherwise they generally have efficient numerical/iterative solutions. BVP solutions also provide useful insight

to difficult problems and complex dynamical environments. For preliminary interplanetary mission design, the trajectory can be broken down into multiple different BVPs that are connected through linked conics. The following subsections detail the most useful BVPs to preliminary trajectory design: Lambert targeting for heliocentric trajectories and hyperbolic terminal velocity problem for planetocentric trajectories.

### 2.3.1 Lambert targeting

Lambert targeting is one of the most common tools used for designing realistic baseline trajectories in mission design. The Lambert targeting problem focuses on the following orbital BVP: given an initial and final location with respect to a central gravitational body and a desired time of flight (TOF) for a spacecraft to complete the orbit segment, find the required velocities at the initial and final positions in the two-body system. Using the linked conics method, interplanetary missions have initial and final locations at planets and a TOF that creates a conic heliocentric trajectory between these planets. The Lambert targeter can generate a two-body trajectory solution for nearly all relative position configurations and flight times. Designing an interplanetary trajectory with a Lambert targeting tool facilitates the use of gravity assists, midcourse maneuvers, and accurate planetary positions. These capabilities are necessary while seeking an optimal trajectory solution.

Lambert's problem has been solved a variety of different ways and has been coded into countless programming languages. The following discussion will focus on the use of a Lambert targeting program, while the theory and mathematical techniques involved in solving Lambert's problem can be found in other texts. Possibly

the most rigorous and descriptive solutions are seen in *An Introduction to Mathematics and Methods of Astrodynamics, Revised Edition* (Battin 1999) [10] and *Fundamentals of Astrodynamics and Applications* (Vallado 2001) [1].

A typical multiple revolution Lambert targeting problem requires seven inputs that describe the conditions of the orbital problem so that the desired outputs can be found.

Inputs:

1.  $\mu$ : gravitational parameter of the central body
2.  $\vec{r}_1$ : radius vector of initial location with respect to gravitational body
3.  $\vec{r}_2$ : radius vector of final location with respect to gravitational body
4. TOF: time of flight between initial and final locations
5.  $d$ : direction of travel about the central body
6.  $m$ : number of orbital revolutions around the central body
7.  $p$ : short or long period of multiple revolution orbit

Outputs:

1.  $\vec{V}_1$ : velocity vector at initial location
2.  $\vec{V}_2$ : velocity vector at final location

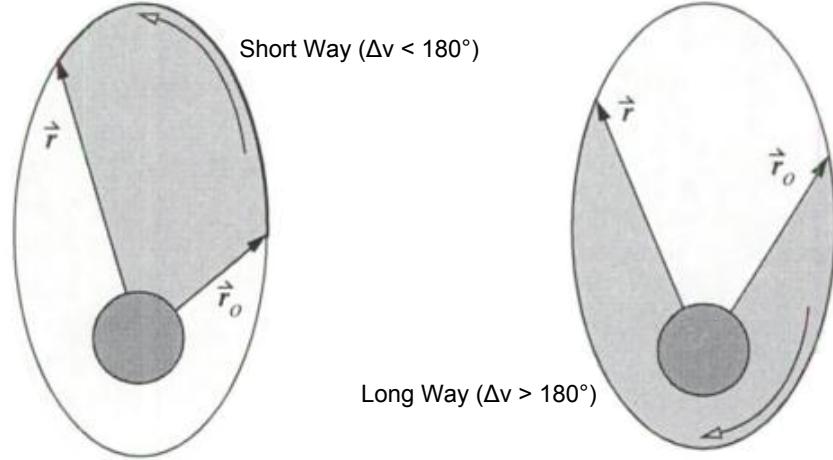


Figure 2.4: Transfer types for Lambert’s problem [Source: Vallado [1]]

Assuming an interplanetary trajectory, the gravitational parameter is of the Sun. The initial location  $\vec{r}_1$  is the position vector of the spacecraft about the Sun at the beginning of the orbital transfer, while  $\vec{r}_2$  is the spacecraft’s position vector at the end of the orbital transfer. The plane defined by the three points,  $\vec{r}_1$ ,  $\vec{r}_2$ , and the origin (Sun) is the orbital transfer plane. With a given set of  $\vec{r}_1$ ,  $\vec{r}_2$ , and TOF, multiple transfer solutions exist. If the transfer is not a multiple revolution trajectory, then only two solutions exist, identified by the transfer direction [16]. The transfer direction is defined in most scripts as either short-way or long-way, which can be seen in Figure 2.4 to describe the relative angle that the transfer arc sweeps out. A short-way path refers to the direction of flight for the shorter transfer arc that is less than  $180^\circ$  (also known as a type 1 transfer), while the long-way path refers to the longer transfer arc direction that is greater than  $180^\circ$  (also known as a type 2 transfer). Though this method is sufficient for a fixed set of orbital parameters, the spacecraft locations and flight times in an iterative environment

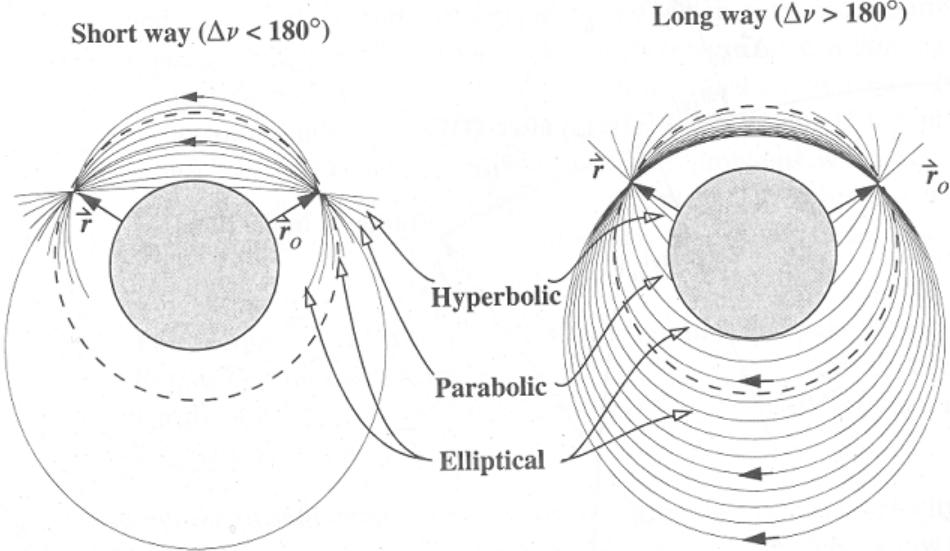


Figure 2.5: Varying time of flight for Lambert trajectory [Source: Vallado [1]]

can change enough that maintaining a short-way or long-way flight direction is not desirable for all iterations [1]. Figure 2.5 shows an example of the potential orbital arcs (elliptical, parabolic, and hyperbolic) created between two fixed locations with varying TOF.

In most interplanetary missions, the heliocentric angular momentum of each trajectory leg is preserved in the same general direction, prograde or retrograde. Since all planets (and most minor bodies) orbit the Sun in a prograde direction, with the angular momentum vector pointing in the northern ecliptic hemisphere, most spacecraft trajectories are also in the prograde direction throughout their entire mission. A change in heliocentric angular momentum direction would have the spacecraft reverse its velocity direction, requiring an unrealistic amount of propellant to accomplish such a large  $\Delta\vec{V}$ . A Lambert targeting tool with a direction specified as short-way or long-way may initially produce prograde trajectory seg-

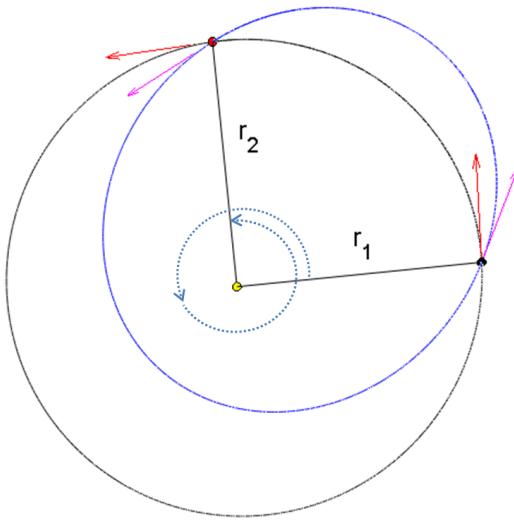


Figure 2.6: Multiple revolution Lambert solution pair for prograde direction with long and short period trajectories identified by the larger and smaller orbit respectively

ments. Yet, as the iterative design loop continues, the orbital parameters may vary enough to yield a retrograde trajectory segment even though the same short-way or long-way direction has been maintained. Therefore, it is preferred that a Lambert targeter script be modified to use prograde and retrograde as the choice for direction of travel.

Multiple revolution transfer solutions are identified by the number  $m$  of revolutions the trajectory completes before arriving at the final location and which of the two possible transfer types is desired, long period or short period (also known as left branch or right branch). The  $m$  parameter is a positive integer value, where zero identifies a trajectory segment that does not travel a full orbital revolution about the central body between initial and final locations. A nonzero value of  $m$  refers to the number of circular or elliptical orbits that are completed [16]. An example

multiple revolution solution pair for a prograde direction is shown in Figure 2.6, where the larger orbit (black) is the long period solution and the smaller orbit (blue) is the short period. Both orbits transfer approximately  $450^\circ$  between  $r_1$  and  $r_2$  and both are prograde. There exist two additional retrograde solutions. These solutions from  $r_1$  to  $r_2$  require the spacecraft to transit approximately  $630^\circ$  in the retrograde direction but are not shown.

The outputs of a Lambert targeting routine are the initial and final velocity vectors,  $\vec{V}_1$  and  $\vec{V}_2$ , that yield the desired transfer. With the initial and final states (position and velocity vectors) of the spacecraft, the orbital transfer segment is fully defined. The segment information can be used to determine the  $\Delta\vec{V}$  required to enter into the transfer from a previous orbit and the  $\Delta\vec{V}$  necessary to exit the transfer into the next orbit segment.

As with most tools, there are limitations to the use of some Lambert targeting algorithms. Most formulations of the Lambert targeting algorithm have an undefined solution when the  $\vec{r}_1$  and  $\vec{r}_2$  vectors are collinear (angles of  $0^\circ$  or  $180^\circ$  apart). This singularity occurs because the three points usually used to define the transfer plane are all along the same line, and infinitely many planes can be chosen to complete the transfer. A modified Lambert targeting algorithm would require a specified plane for the transfer or a metric to determine a unique transfer plane solution, such as minimizing the associated  $\Delta V$  maneuvers.

### 2.3.2 Hyperbolic terminal velocity problem

A very useful but seemingly under-utilized BVP algorithm in trajectory design is the hyperbolic terminal velocity problem. The BVP is particularly useful for

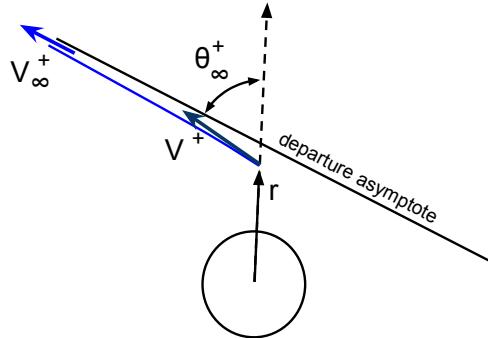


Figure 2.7: Hyperbolic terminal velocity problem diagram for departing trajectory

linked conics for handling the planetocentric hyperbolic trajectory legs, such as a flyby, a hyperbolic arrival trajectory to insert into a parking orbit, or a hyperbolic departure trajectory from a parking orbit. The hyperbolic terminal velocity problem seeks to determine the hyperbolic conic trajectory between a given departure or arrival (depending on desired scenario)  $\vec{V}_\infty$  asymptote vector and a given  $\vec{r}$  position vector about the encountered body. The  $\vec{r}$  location would be where the hyperbolic conic trajectory intersects the parking orbit for arrival or departure. The  $\vec{r}$  location would also be where the incoming hyperbolic leg of a flyby connects with the outgoing hyperbolic leg, thereby completing the flyby transfer trajectory. The resulting solution of the algorithm is the required  $\vec{V}$  velocity direction and magnitude associated with the  $\vec{r}$  location to complete the hyperbolic trajectory, as shown in Figure 2.7. The BVP assumes that the  $\vec{V}_\infty$  asymptote is not at a predefined position or offset distance with respect to the central body, allowing the algorithm to determine the trajectory required to complete the transfer boundary conditions. The BVP is solved in the following chapters to determine the incoming and outgoing hyperbolic legs of a flyby, connecting between the given incoming  $\vec{V}_\infty^-$  and outgoing  $\vec{V}_\infty^+$  vectors.

It can also be used to determine the hyperbolic departure leg from a parking orbit to the outgoing  $\vec{V}_\infty^+$  vector (or the hyperbolic arrival leg to a parking orbit from the incoming  $\vec{V}_\infty^-$  vector).

The solution presented below was initially developed by Battin [10] and has been reformulated for two cases (arrival/incoming or departure/outgoing) in the current work for clarity and ease of computation. The hyperbolic boundary value problem was solved using hodograph analysis, where velocity diagrams are constructed to determine the relative magnitude and direction of the problem's associated velocities. The details of this process can be found in Battin's work [10]. The relationships result in an analytic solution for the velocity  $\vec{V}^+$  at location  $\vec{r}$  of an outgoing hyperbolic trajectory that reaches the required departing  $\vec{V}_\infty^+$  asymptote conditions, Equation 2.13. The  $\vec{V}^+$  vector is built up by a component along the  $\hat{V}_\infty^+$  asymptote unit vector direction and a component along the  $\hat{r}$  radial position unit vector direction. The transfer angle  $\theta_\infty^+$  between the  $\vec{r}$  position vector direction and the  $\vec{V}_\infty^+$  asymptote direction is computed as  $\theta_\infty^+ = \arccos(\hat{r} \cdot \hat{V}_\infty^+)$ .

$$\begin{aligned}\vec{V}^+ &= \left(D + \frac{1}{2}V_\infty^+\right)\hat{V}_\infty^+ + \left(D - \frac{1}{2}V_\infty^+\right)\hat{r} \\ D &= \sqrt{\frac{\mu}{r(1+\cos\theta_\infty^+)} + \frac{V_\infty^{+2}}{4}}\end{aligned}\tag{2.13}$$

The solution formulation for the arrival case is an incoming hyperbolic trajectory that determines the  $\vec{V}^-$  velocity vector at  $\vec{r}$  after transferring from the arriving  $\vec{V}_\infty^-$  asymptote. The parameter  $D$  is the similar for this case, but with incoming variables. The  $\vec{V}^-$  velocity vector is constructed with a negative radial component, Equation 2.14. This is to correct for the radial direction on incoming trajectories

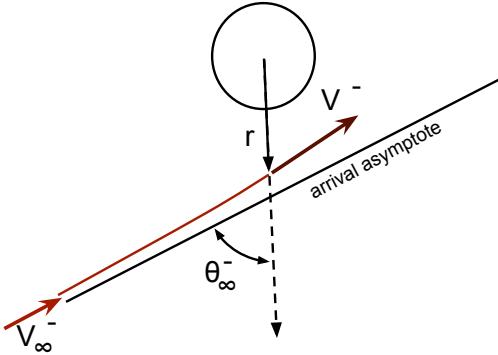


Figure 2.8: Hyperbolic terminal velocity problem diagram for arriving trajectory

inherently opposing the velocity direction, which can be seen in Figure 2.8.

$$\begin{aligned}\vec{V}^- &= \left(D + \frac{1}{2}V_\infty^-\right)\hat{\mathbf{V}}_\infty^- - \left(D - \frac{1}{2}V_\infty^-\right)\hat{\mathbf{r}} \\ D &= \sqrt{\frac{\mu}{r(1+\cos\theta_\infty^-)}} + \frac{V_\infty^{-2}}{4}\end{aligned}\tag{2.14}$$

Upon further study the hyperbolic terminal velocity problem algorithm exhibits qualities similar to that of Lambert targeting. For a given set of boundary conditions  $\vec{V}_\infty^-$  and  $\vec{r}$ , there exist two trajectory solutions that complete such a hyperbolic transfer. The two solutions to the problem are similar to the short and long way solutions to Lambert's problem, shown in Figures 2.9 and 2.10. By definition the magnitude of the  $\vec{V}^\pm$  solution ( $V_\infty^-$  and  $V_\infty^+$ ) for both short and long way is the same. The algorithms in Equations 2.13 and 2.14 provide the short way solution, while the long way solution is found by reversing the sign of the  $D$  parameter. Using a negative  $D$  in the incoming and outgoing algorithms reverses the direction of motion for the long way solution. The short way solution may or may not transit through periapse depending on the BVP geometry, but the long way solution always passes through periapse. Note that the “short way” and the “long way” solutions are

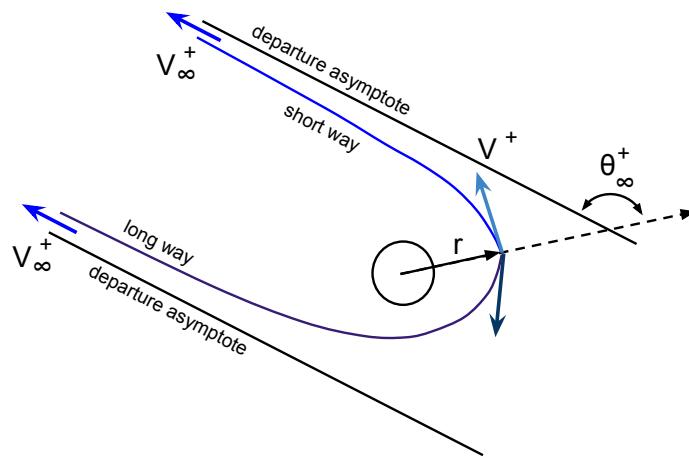


Figure 2.9: Short and long way hyperbolic terminal velocity solutions for outgoing transfer

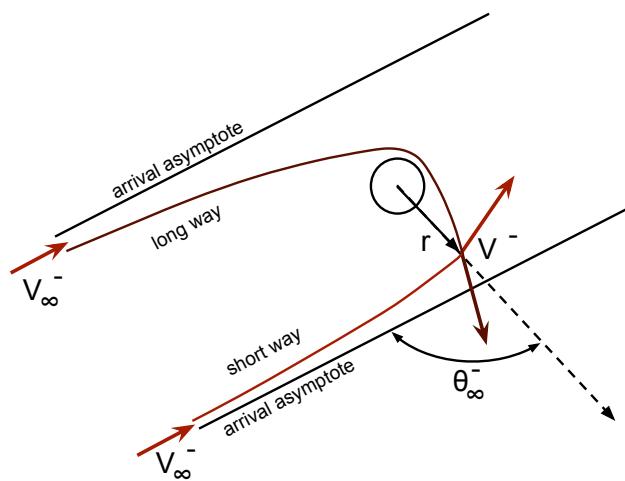


Figure 2.10: Short and long way hyperbolic terminal velocity solutions for incoming transfer

not segments of the same hyperbolic flyby. Instead, they are two separate trajectory solutions that connect between a specific location  $\vec{r}$  and a hyperbolic asymptote excess velocity vector,  $\vec{V}_\infty$ . Considering the difference in the short way and long way solutions, a planetocentric hyperbolic transfer trajectory can be found with a specified angular momentum direction by changing the sign on  $D$  when needed. The periapse distance is unconstrained for these solutions, therefore care should be taken when implementing the long way solution to avoid an unacceptably low periapse distance.

During an iterative trajectory design process, the general regions of concern for maintaining a useful and feasible hyperbolic transfer is when the specified  $\vec{r}$  location transitions from one side of the periapse to the other, requiring  $D$  to switch signs in order to maintain the same transfer direction. Also, when  $\theta_\infty^\pm$  ( $\theta_\infty^-$  or  $\theta_\infty^+$ ) passes through  $\pi$  or 0, the  $D$  parameter sign should be switched. Although extra consideration should be taken when passing through  $\theta_\infty^\pm = 0$ , because in order to maintain the hyperbolic orbit's angular momentum direction through the transfer, the solution switches from a very short path trajectory and a very long path. Depending on the specific problem scenario at hand this drastic change in trajectory just to maintain a specific angular momentum direction may or may not be desirable. It would be up to the user implementing this BVP algorithm to take care in understanding the potential trajectory solutions of the hyperbolic BVP and tailoring it to meet their needs.

Just as Lambert's problem is undefined at  $n\pi$  transfer angles, the hyperbolic terminal velocity problem is undefined when  $\theta_\infty^\pm$  is exactly  $\pi$ . This is because there are an infinite number of planes to transfer through to satisfy the trajectory boundary

conditions; a preference or metric is required to select a unique transfer solution. When  $\theta_{\infty}^{\pm}$  is exactly 0, then by the problem's definition  $\hat{r} \Rightarrow \hat{r}_{\infty}$  and  $\vec{V}^{\pm} \Rightarrow \vec{V}_{\infty}^{\pm}$ , despite the magnitude of  $r$ . Essentially, the hyperbolic terminal velocity problem is a special extension of Lambert's problem, where  $\vec{r}_2 \Rightarrow \vec{r}_{\infty}$  and TOF  $\Rightarrow \infty$ .

## 2.4 Planetary encounters

In an linked conic interplanetary trajectory, most  $\Delta V$  maneuvers occur within the planetary encounters. The typical types of planetary encounters are departing from or arriving to parking orbits about planets, as well as, flybys of planets. All of these types include hyperbolic transfers and therefore employ the hyperbolic terminal velocity problem solution to determine the required  $\vec{\Delta V}$ . Most encounter orbits are defined in a planetocentric reference frame. The planet's pole and equator orientation define the planetocentric reference frame, thereby providing a sense of latitude. For preliminary trajectory design phases, the rotation of the planet need not be considered nor accounting for any particular longitude with respect to a planet-specific reference. The subsections below provide the details and examples for each type of encounter as an introduction to the topics in later chapters.

### 2.4.1 Parking Orbits

Determining the departure from a parking orbit about a planet requires information about the parking orbit and the departing  $\vec{V}_{\infty}$  asymptote. The mission designer must supply the parking orbit data, while the  $\vec{V}_{\infty}$  is already computed from the heliocentric trajectories. Only a subset of the parking orbit information is required, while the remaining parameters can be optimized to determine the minimum

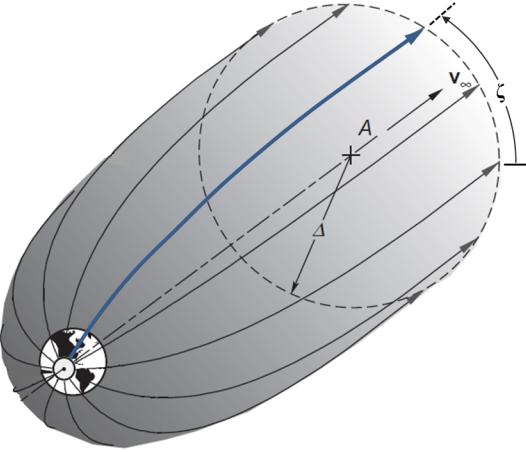


Figure 2.11: Hyperbolic departure trajectories from peripase to  $\vec{V}_\infty$  asymptote defined by asymptote offset  $\Delta$  and departure plane angle  $\zeta$  [Source: Curtis [5]]

$\Delta V$  maneuver required to achieve the  $\vec{V}_\infty$  asymptote direction and magnitude.

Typical parking orbit data provided is semimajor axis, eccentricity, and inclination. Of the remaining orbit orientation parameters, right ascension of ascending node  $\Omega$  can be determined analytically, while the argument of peripase  $\omega$  must be optimized, assuming the maneuver occurs at peripase  $v = 0$  [10]. Most parking orbit transfer algorithms require that the inclination of the parking orbit be greater than or equal to the declination of the hyperbolic  $\vec{V}_\infty$  asymptote to ensure a coplanar transfer and a velocity-aligned collinear  $\Delta V$  maneuver. For each different parking orbit orientation and maneuver location, the hyperbolic terminal velocity problem is solved again in order to compute the required  $\vec{\Delta V}$ . Each different hyperbolic transfer trajectory achieves these asymptote requirements with the available degrees of freedom, the asymptote aiming radius  $\Delta$  and the departure plane angle  $\zeta$  about the  $\vec{V}_\infty$  asymptote direction. The surface of potential hyperbolic departure trajectories is provided in Figure 2.11, where the transfer plane is defined from some reference

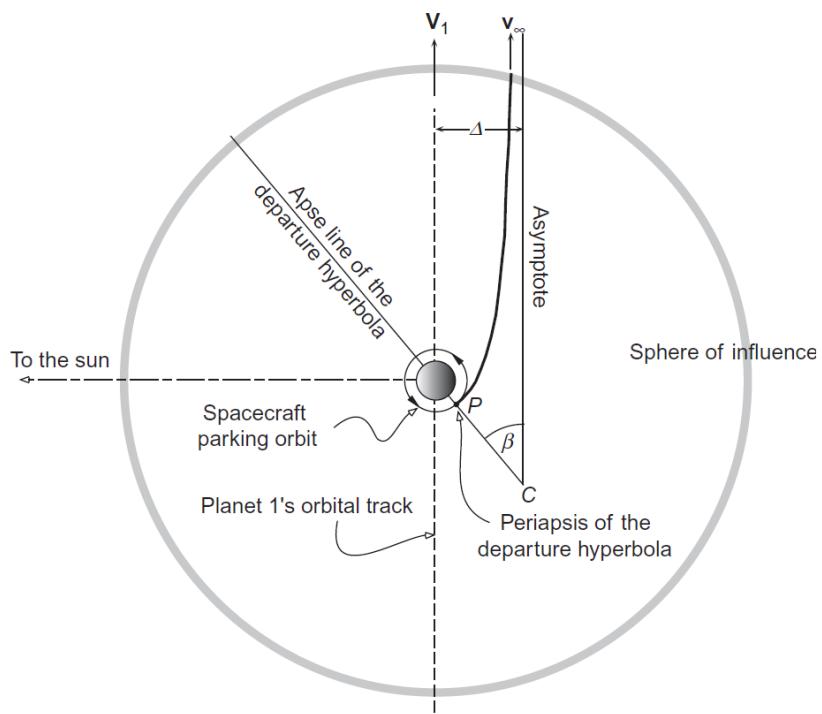


Figure 2.12: Example departure trajectory from a parking orbit about a planet  
[Source: Curtis [5]]

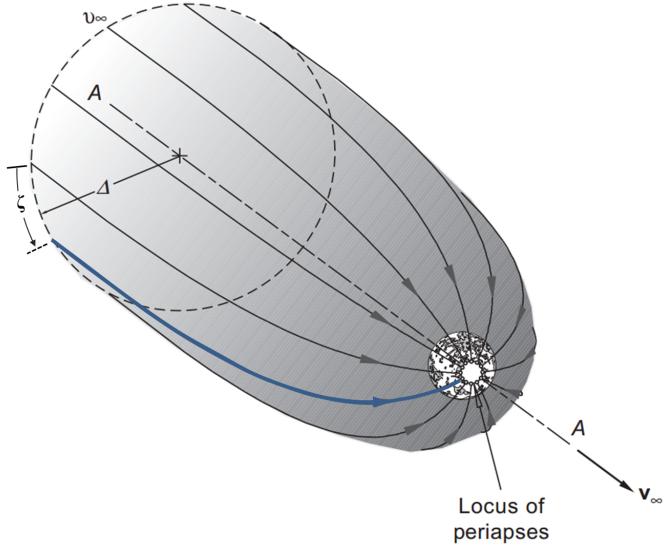


Figure 2.13: Hyperbolic arrival trajectories from  $\vec{V}_\infty$  asymptote to periapse, as defined by the asymptote offset  $\Delta$  and arrival plane angle  $\zeta$  [Source: Curtis [5]]

plane by angle  $\zeta$ . Linked conics provides this freedom by assuming the desired offset distance  $\Delta$  and transfer plane can be achieved by a infinitesimal maneuver performed at an infinite distance from the planet, or at least by a negligible maneuver at the sphere of influence. The analytic solution of the hyperbolic terminal velocity problem inherently determines the appropriate offset and planet angle to complete the trajectory. An example of a departure trajectory from a parking orbit about a planet is shown in Figure 2.12.

Most planetary arrivals are in the form of a closed parking orbit about the planet, the main exceptions being a hyperbolic entry or a landing trajectory. The arrival conditions are similar to the departure, where orbit data is required and the  $\vec{V}_\infty$  asymptote is already determined. The same optimization process is performed to determine the remaining orbit information and the minimum required  $\Delta V$ . The

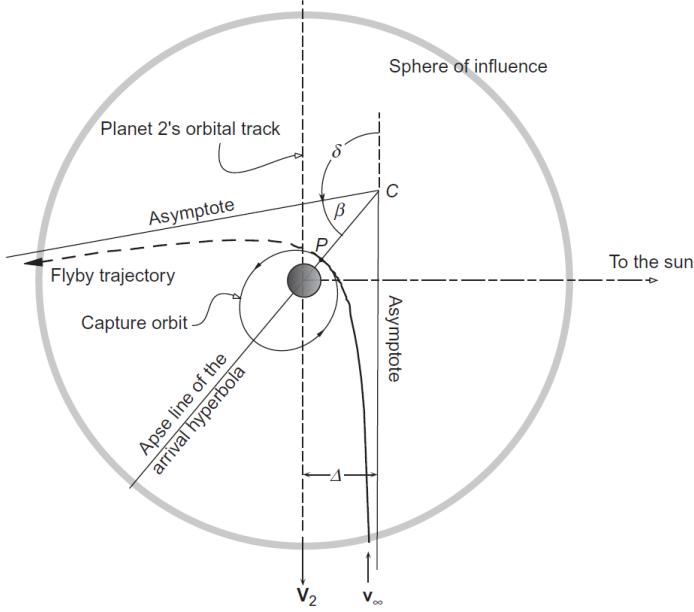


Figure 2.14: Example arrival trajectory into a parking orbit about a planet [Source: Curtis [5]]

potential hyperbolic arrival transfers are similar to the departure trajectory surface of revolution about the incoming  $\vec{V}_\infty$  asymptote, displayed in Figure 2.13. The arrival trajectory diagram in Figure 2.14 depicts the hyperbolic transfer into a parking orbit, as well as the ballistic flyby trajectory if the maneuver was not performed [5].

#### 2.4.2 Ballistic Flybys

A few assumptions are commonly adopted when approaching the gravity assist problem. The flyby is assumed to be a coplanar transfer, where the plane is defined by the incoming and exiting  $\vec{V}_\infty$  asymptotes about the planetary center. It is assumed that the flyby maintains a hyperbolic energy level throughout the entire encounter. The short trajectory time associated with hyperbolic flyby maneuvers

supports the linked conic assumptions that each planetary encounter is effectively instantaneous compared to the heliocentric trajectory flight times [17].

It is also assumed that the flyby trajectory periapse never falls below a minimum radius. Though this assumption is commonly stated in gravity assist algorithms, it is often not strictly enforced on both legs of the hyperbolic transfer. The gravity assist algorithms developed in the following chapters rigorously maintain this requirement, yielding continuous and feasible encounter trajectories. A less common assumption but just as important requirement is that the periapse of the flyby trajectory never exceeds the planet's sphere of influence. This enforces the linked conic assumptions that the encounter with the planet takes place within the planet's sphere of influence and that the planet's gravity alters the trajectory.

A gravity assist maneuver is applied in an interplanetary trajectory to use a planet's gravitational field and momentum to change the spacecraft's heliocentric orbital velocity. In the planet centered reference frame, the ballistic hyperbolic trajectory (unpowered gravity assist) does not change the orbital energy in planet centered coordinates, but is simply redirected from the entering  $\vec{V}_\infty^-$  to the exiting  $\vec{V}_\infty^+$ . When converting the planetocentric entering and exiting  $V_\infty^\pm$  to a heliocentric spacecraft velocity, the change in heliocentric orbital velocity is apparent, particularly when visualized in Figures 2.15 and 2.16. From this point in the dissertation onward, the planetocentric turning angle of a gravity assist (powered or unpowered) will be represented by  $\Phi$  and a specified subscript. This will help to accommodate the turning angles of powered gravity assists and delineate it from the ballistic hyperbolic turning angle  $\delta$  defined previously [10, 5].

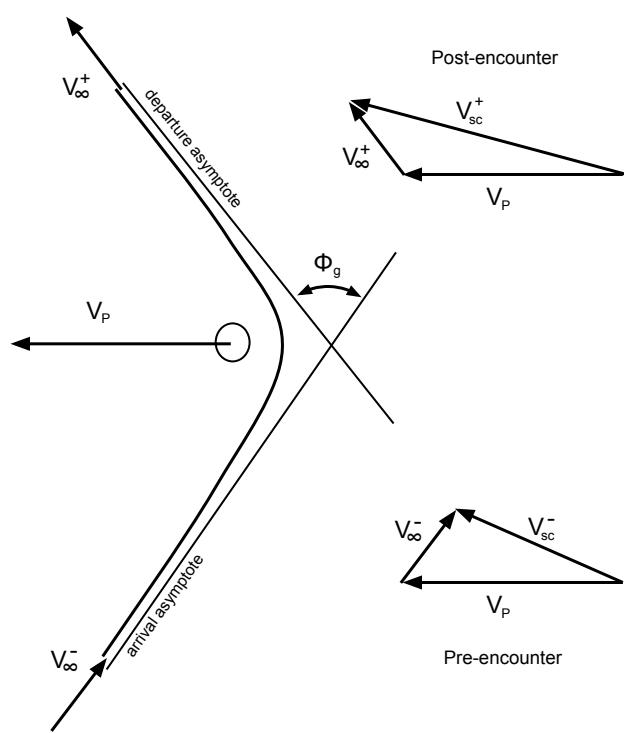


Figure 2.15: Increasing velocity gravity assist on trailing side of planet

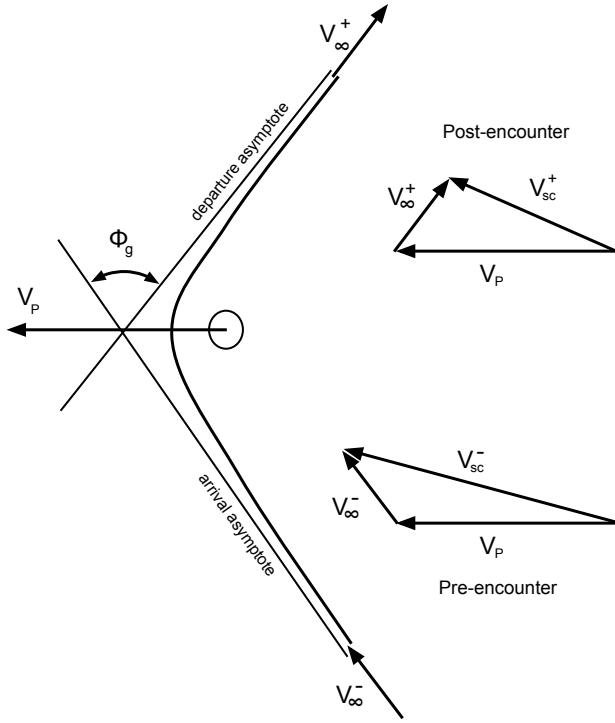


Figure 2.16: Decreasing velocity gravity assist on leading side of planet

$\vec{V}_{sc}^-$  : Spacecraft heliocentric velocity pre-encounter

$\vec{V}_{sc}^+$  : Spacecraft heliocentric velocity post-encounter

$\vec{V}_P$  : Planet heliocentric velocity

$\Phi_g$  : Gravitational turning angle

The heliocentric velocity of the spacecraft ( $\vec{V}_{sc}$ ) resulting from the gravity assist is increased or decreased depending on how the maneuver is performed. If the hyperbolic periapsis occurs on the trailing side of the planet with respect to the planet's heliocentric velocity, then the spacecraft heliocentric velocity will be in-

creased by the gravity assist. This is due to the flyby's resulting  $\vec{V}_\infty^+$  being turned more inline with the planet's velocity vector, therefore adding into a greater space-craft velocity  $\vec{V}_{sc}^+$ , and displayed in Figure 2.15. If the hyperbolic periapsis occurs on the leading side of the planet with respect to the planet's heliocentric velocity, then the spacecraft heliocentric velocity will be decreased by the gravity assist, Figure 2.16. Similarly, the flyby's resulting  $\vec{V}_\infty^+$  is being turned against the planet's velocity vector, therefore adding into a smaller spacecraft velocity  $\vec{V}_{sc}^+$ .

These gravity assist configurations apply to most trajectory circumstances, but if encountering a planet with a heliocentric spacecraft velocity in exact alignment with the planet's velocity (in the same direction or opposite direction), an interesting result occurs. In the case of encountering a planet with an heliocentric spacecraft velocity less than the planet's velocity (such as a gravity assist of an outer planet) and in the same direction (a Hohmann transfer), the planet catches up to the spacecraft from behind. After this type of flyby configuration, the exiting heliocentric velocity is increased no matter the location of the periapse, Figure 2.17. Comparing the same flyby configuration with a leading or trailing location of the periapse, the magnitude of the increase in heliocentric velocity is the same, but the resulting heliocentric velocity direction is different. This type of gravity assist is characteristic of a Hohmann transfer to an outer planet, where the intention of the maneuver is typically to gain heliocentric orbital energy. If encountering a planet with a heliocentric spacecraft velocity greater than the planet's (such as a gravity assist of an inner planet) and in the same direction, the spacecraft catches up to the planet from behind. After this type of flyby, the exiting heliocentric velocity is decreased whether the periapsis is on the trailing or leading side, Figure 2.18.

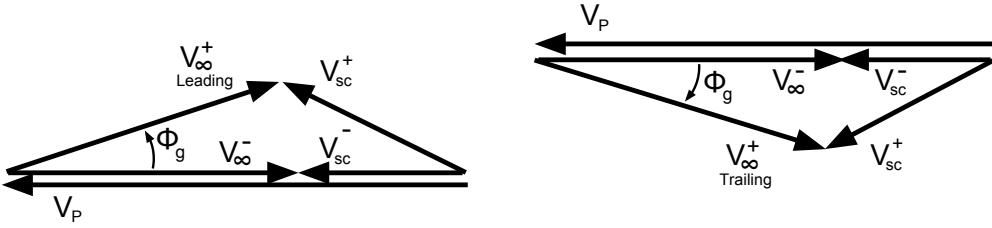


Figure 2.17: Special case gravity assist of  $V_{sc} < V_p$  with leading and trailing configuration

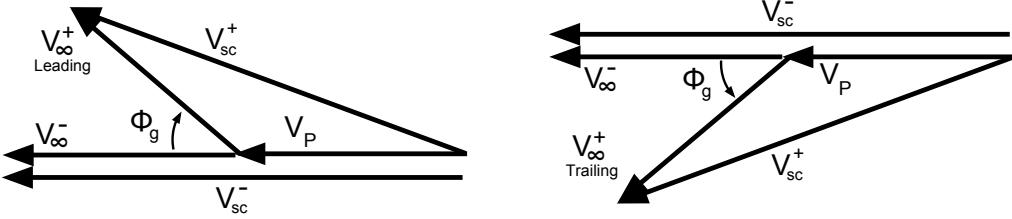


Figure 2.18: Special case gravity assist of  $V_{sc} > V_p$  with leading and trailing configuration

Similarly, the magnitude of the decrease in heliocentric velocity is the same, but the resulting direction is different. Again, this type of gravity assist is characteristic of a Hohmann transfer but to an inner planet, where the intention of the maneuver is typically to reduce heliocentric orbital energy [10, 5, 17].

The gravitational turning angle ( $\Phi_g$ ) that occurs during a gravity assist affects how much the resulting heliocentric spacecraft velocity is changed. Important factors in determining how large or small the resulting turning angle are the incoming  $V_\infty^-$  and the periapsis radius ( $r_p$ ) of the gravity assist. With a given fixed  $r_p$ , the magnitude of  $V_\infty^-$  directly affects  $\Phi_g$ . As  $V_\infty^-$  increases the flyby trajectory flattens out with decreasing  $\Phi_g$ , and as  $V_\infty^-$  decreases the flyby trajectory bends further with increasing  $\Phi_g$ . Alternatively assume a fixed  $V_\infty^-$ , as  $r_p$  increases the flyby

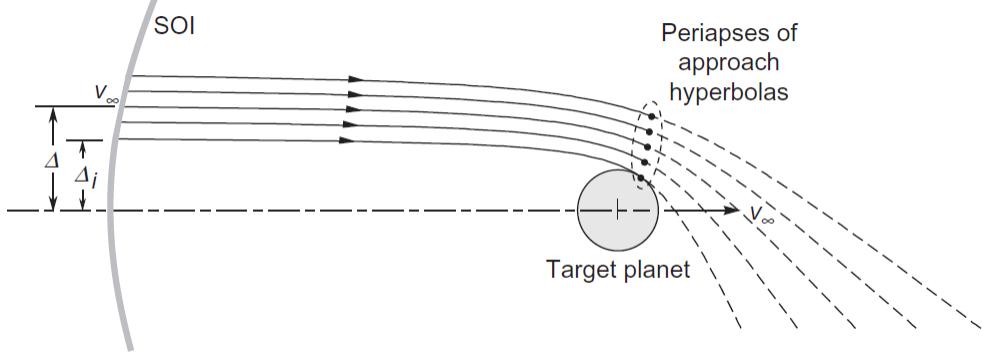


Figure 2.19: Varying periapsis radius changes the gravitational turning angle of the gravity assist [Source: Curtis [5]]

trajectory flattens out with decreasing  $\Phi_g$ , and as  $r_p$  decreases the flyby trajectory bends further with increasing  $\Phi_g$ . These results are displayed in Figure 2.19 [5]. The diagram shows the effect of increasing the hyperbolic offset  $\Delta$  on the raising the resulting periapse distance. The  $\Delta_i$  shows the minimum offset distance of the hyperbolic trajectory just before impacting the surface of the planet [5, 17].

Though a ballistic flyby remains in-plane throughout a two body problem hyperbolic orbit, the plane of that flyby has the same plane angle  $\zeta$  degree of freedom described in previous sections. The flyby encounter can be rotated about the incoming  $\vec{V}_\infty^-$  asymptote by an infinitesimal impulse at  $r_\infty^-$ , providing the desired plane angle  $\zeta$  and the offset distance  $\Delta$ . An example of potential trajectory planes for a fixed  $\Delta$  offset distance is shown in Figure 2.20.

A few constraints are assumed for a flyby in order to ensure that the plane-to-centric trajectory is mission feasible and abides by the linked conics assumptions. One of the most active constraints in flyby trajectories is to require that the trajectory never passes below a minimum radius threshold  $r_{min}$ . This constraint is applied

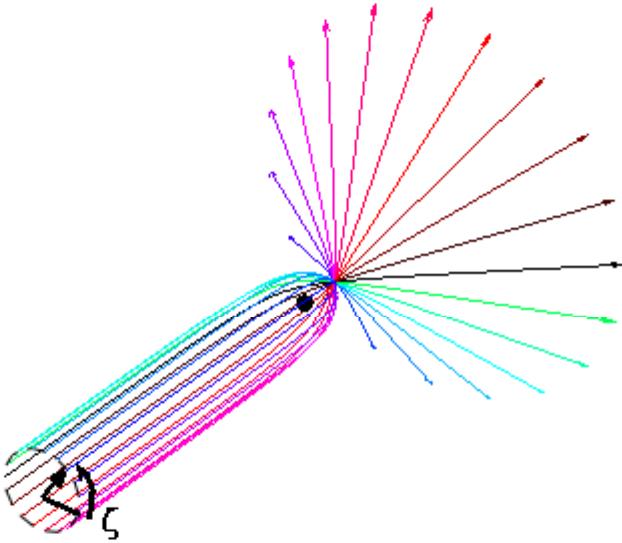


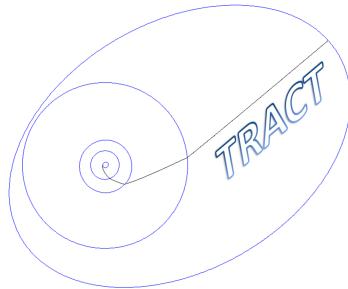
Figure 2.20: Flyby potential trajectories for a fixed  $\Delta$  offset distance and a revolution of hyperbolic transfer planes by plane angle  $\zeta$

by requiring the flyby periapse  $r_p$  be at or above the minimum radius:  $r_p \geq r_{min}$ . Another constraint is that the flyby periapse must not exceed the planet's sphere of influence radius  $r_p \leq r_{SOI}$ . Linked conics assumes that the flyby encounter must enter into the sphere of influence of the planet and have some influence on the trajectory [17, 10]. More discussion on these flyby constraints is found in Chapters 4 and 5.

During a flyby trajectory, an impulsive  $\Delta V$  maneuver can be performed to change the outgoing hyperbolic leg trajectory characteristics (direction and energy) from that of the incoming hyperbolic leg. The utility of incorporating a  $\Delta V$  maneuver during a flyby is thoroughly described in Chapter 4.

## Chapter 3

### TRACT: TRAjectory Configuration Tool



TRACT is a trajectory design tool developed for preliminary interplanetary mission trajectory design and optimization and is one of the main contributions of this dissertation. This chapter discusses TRACT's formulation and its distinguishing features. TRACT has proven to be a helpful tool for any experience level, including graduate academic research and professional industry mission planning. The unique bi-level architecture of TRACT sets it apart from other trajectory design tools. TRACT features separate boundary value problem (BVP) algorithms for every planetocentric and heliocentric trajectory segment, allowing each to be solved efficiently and reliably. This is achieved by using a standardized trajectory parameterization that enables a wide variety of optimization schemes, while still being adaptable enough to incorporate new trajectory solution algorithms (for heliocentric or planetocentric BVPs). This BVP methodology also reduces the amount of

information required from the user, while still maintaining an efficient and robust trajectory computation and optimization.

A predecessor to TRACT, known simply as the Trajectory Design Tool (TDT) was developed to aid undergraduate student senior design projects for space-craft missions [18]. TDT had limited trajectory design capabilities, providing only a few trajectory maneuver options. Each trajectory maneuver computation was inefficient and slow. The gravity assists were modeled by using an inefficient powered periapse-only gravity assist algorithm. Also, TDT was only configured for unconstrained trajectory optimization schemes, severely restricting the tool's trajectory optimization performance. Despite all these limitations, TDT was the inspiration for the powerful and capable mission design tool presented here, TRACT.

Compared to its predecessor TDT, TRACT has been completely rewritten, modularized, and many additional features have been added, resulting in a highly customizable trajectory design tool with significant computational performance increases. TRACT's modular structure allows easy implementation of new maneuver types and sequences. Due to its adaptability, TRACT can be used as a testbed for new trajectory algorithms, such as the optimal powered gravity assist algorithm and asymptote corrected gravity assist algorithms developed in this dissertation. The other primary contributions of this dissertation are characterizing the powered gravity assist boundary value problem, developing a new optimal powered gravity assist algorithm, and extending this work to an asymptote corrected ballistic gravity assist algorithm. Incorporating these capabilities into TRACT has allowed for thorough testing, analysis, and validation of the optimal single-impulse powered gravity assist algorithm and the asymptote maneuver algorithm for producing ballistic gravity

assist trajectories. Uniting these contributions together support each other to create more of an impact than any single contribution alone.

### 3.1 TRACT summary

TRACT determines and optimizes interplanetary trajectories using precise planetary ephemerides and accurate linked conic methods. Written in the MATLAB computer programming language and application, the design tool computes the user defined interplanetary trajectory with inputs including various options for departure and arrival from planets (and user defined bodies), gravity assists, and deep space maneuvers (DSM). One of the features of TRACT is its architecture of linked boundary value problems to determine all heliocentric and planetocentric trajectories in a Bi-level optimization scheme. The details of the Bi-level optimization scheme are discussed later in the chapter. After evaluating the trajectory, the user may choose a variety of organized Excel data outputs and preconfigured MATLAB visual outputs, trajectory plots, and trajectory videos with three dimensional vantage point controls. MATLAB provides an adaptable structure for the design tool so that it is easily used by novice trajectory designers while also providing enough options to model complex trajectories for experienced designers.

As a useful mission example, the Juno mission to Jupiter will be referenced when necessary throughout this chapter to demonstrate TRACT's usage and capabilities. The Juno spacecraft launched from Earth August 5, 2011 and over a year later performed two DSMs to redirect its trajectory before returning to Earth for a gravity assist. The Earth gravity assist boosts the spacecraft velocity in order to

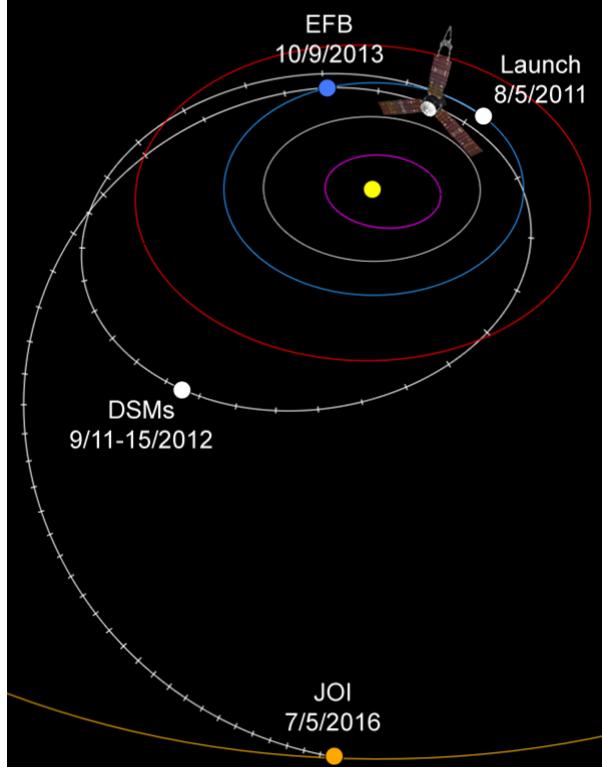


Figure 3.1: Juno mission trajectory to Jupiter using DSMs and an Earth flyby [6]

reach Jupiter as shown in Figure 3.1 excerpt from a Juno mission article [6]. In this example mission the two DSMs will be combined into just a single DSM for simplicity. In the Figure, EFB indicates the Earth flyby and JOI indicates Jupiter orbit insertion.

### 3.1.1 Assumptions

Each segment of the trajectory is modeled using two body motion, where heliocentric legs are determined by a Lambert targeting routine and the planetocentric legs are determined by efficient and robust BVP algorithms. The trajectory determination process is formulated so that at each iteration of optimization procedure

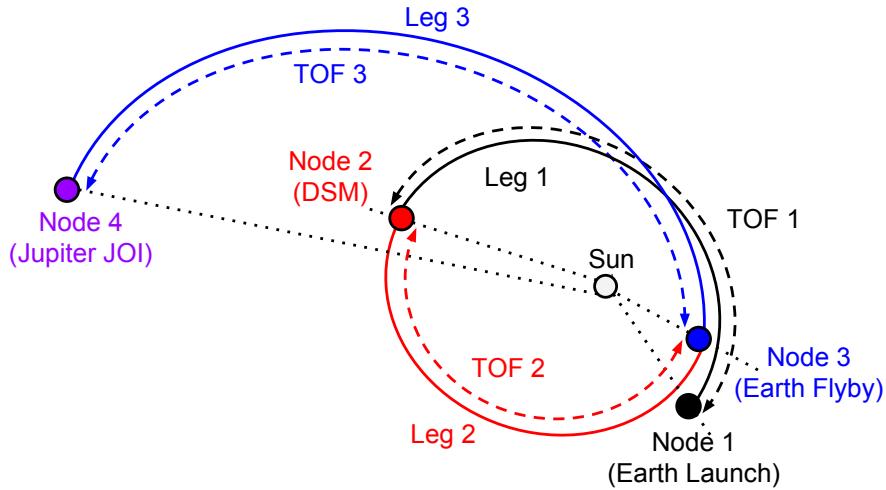


Figure 3.2: Juno trajectory configuration sequence example and nomenclature (relative scale slightly adjusted for clarity)

the entire trajectory is complete, continuous (with the exceptions inherent to linked conics), and a feasible candidate for final design selection. This is ensured in part by satisfying typical trajectory constraints of the planetocentric encounter within the BVP. Only velocity state discontinuities  $\vec{\Delta V}$  are allowed as deterministic maneuvers. The trajectory formulations assume impulsive maneuvers to accomplish the  $\vec{\Delta V}$  state discontinuities. Maintaining the continuity of the linked trajectory positions is accomplished by adopting BVP solutions for each segment of the trajectory, where the initial and final locations of each trajectory segment are explicitly connected in the linked conics methodology.

To facilitate the design process, a heliocentric trajectory is broken into segments called legs, as depicted in the Juno example Figure 3.2. The initial and ending points of each trajectory leg are locations for a potential maneuver where a mission

event would occur. For Juno these mission events are an Earth launch, a DSM, an Earth flyby, and a Jupiter orbit insertion. The terminal points of the legs are called nodes and are identified by coinciding with a planet, celestial object, or heliocentric location. The date when the trajectory begins defines the time leaving the first node, while the TOF for each leg defines all other nodes. The Juno example trajectory diagram is color coded by each leg, TOF, and preceding node to clearly identify their associated trajectory segments.

The planetary state position and velocity vectors are determined from JPL ephemerides functions. The heliocentric trajectory legs are determined by a multiple revolution Lambert Targeting routine. The heliocentric trajectory vectors are defined in rectangular Cartesian coordinates relative to the mean ecliptic and equinox of J2000 as defined in the *Explanatory Supplement to the Astronomical Almanac* [15]. The planetocentric trajectory vectors are defined in rectangular Cartesian coordinates relative to the equator and right ascension of the ascending node of the specific planet at the encounter's current Julian date (JD). The rotation of planets' surfaces is not considered. The TOFs only apply to heliocentric trajectory segments, while planetocentric segments are assumed to occur instantaneously.

### 3.1.2 TRACT architecture

The TRACT structure is made up of the mission driver interface, the central trajectory configuration function, and various mission event modules containing the boundary value problem algorithms. The mission driver interface is a MATLAB script that provides inputs to the trajectory configuration function, while all other parts of the design tool are written as MATLAB functions which use inputs to

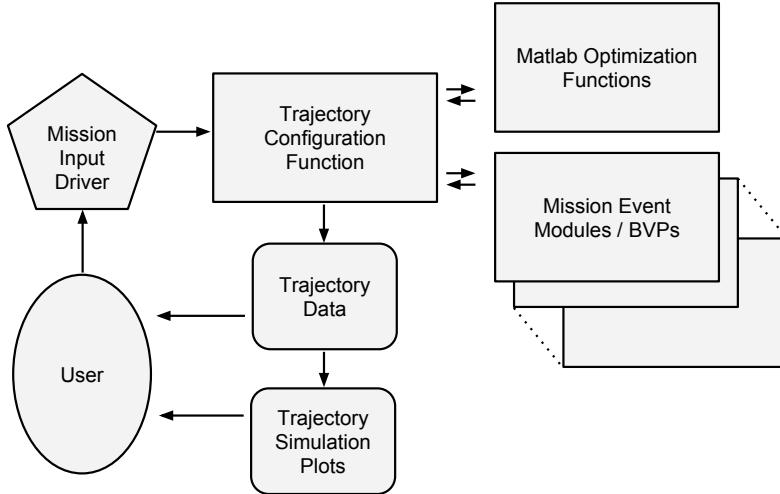


Figure 3.3: Trajectory design tool flowchart

perform calculations and deliver outputs as necessary. The trajectory configuration function uses the mission driver information to direct which modules are called, what optimization scheme is executed, what constraints are applied, and which outputs options are generated.

The general order of processes performed by TRACT can be seen in Figure 3.3. After the user has entered all the appropriate information to describe the trajectory, optimization process, constraints, and outputs, the mission driver script is run in MATLAB to begin the computation process. The input information is given to the trajectory configuration function, which starts the optimization process and builds the trajectory at each iteration by calling the mission event modules. Once the optimization process is complete, the resulting trajectory information is provided to the user, as well as the selected additional outputs: a neatly formatted Excel mission data spreadsheet and/or MATLAB trajectory simulation plots/movies.

### 3.1.2.1 Mission driver interface

The mission driver interface is a formatted MATLAB script that allows the user to input the desired trajectory configuration and optimization instructions. The user can also identify all pertinent output options desired from TRACT, such as trajectory plots or mission data spreadsheets. Each input parameter entered by the user into the mission driver script is a numerical value, a decimal number or integer. The numerical identification is essential in computational efficiency and organization. The numerical identification also allows TRACT to be operated by another program, or easily adjusted inside a loop (for-loop, while-loop, etc.). As an example the Juno mission launch Julian date ( $JD0$ ) would be input as  $JD0 = 2455778.5$ , the TOFs input as  $tof = [ \ 393.5 \ 402.5 \ 1000 \ ]$  in days, and the initial DSM location would be input as  $DSM = [ \ -1.8 \ 1.4 \ 0 \ ]$  assuming the heliocentric [X Y Z] coordinates are in AU. These trajectory values are relatively close to the actual Juno trajectory values.

Some of the trajectory maneuver options available to the user are:

- Launch vehicle departure conditions
- Depart/arrive with parking orbit about planet
- Depart/arrive with parking orbit about Sun
- Hyperbolic entry arrival conditions
- Deep Space Maneuvers
- Powered gravity assists

- Ballistic gravity assists

The full list of the mission driver interface options are provided in Appendix G.

### 3.1.2.2 Trajectory configuration function

The trajectory configuration function is the central programming script that takes the users inputs and manages the trajectory computation and optimization process. The primary optimization parameters are standardized to accommodate nearly all interplanetary mission trajectories in a uniform way by only using the heliocentric trajectory parameters. This is done using linked conics as one level of the computation and optimization process. The planetocentric trajectories are configured to be solved by independent BVPs without initial guesses from the user. The heliocentric optimization parameters are the trajectory times of flight (TOFs) and the Julian date at the first node (JD0). The initial values of these parameters are provided by the user via the mission driver script. The heliocentric location of the DSMs are also optimization parameters if the user includes DSMs in the trajectory. The first module called in the trajectory configuration function is midcourse maneuver guess function, which checks whether any DSMs are included in the trajectory and computes location estimates for the DSMs if no location is provided by the user. The midcourse maneuver location guess generator uses the locations of the nodes before and after the maneuver to find an average radius and angle between the nodes as a rough initial guess for the DSM. With all optimization parameters defined, the optimization process is initialized by formatting the parameters for input into the selected optimization function. The MATLAB optimization schemes available to use in TRACT are fmincon, fminunc, fminsearch, fzero, lsqnonlin, as

well as other optimization options from MATLAB’s Global Optimization Toolbox (if available). More details on these optimization options and schemes are discussed in the following TRACT Capabilities section.

The optimization function uses the total scalar  $\Delta V$  of the trajectory as its performance index, which is minimized throughout the process. The trajectory information must be computed for both the heliocentric legs and the nodes (mission events) in order to evaluate the  $\Delta V$  required for each mission event. The Julian dates for the nodes are found from the initial node Julian date and the TOFs. These Julian dates are used to determine the location of the planets from a JPL ephemerides function (DE405 [19, 20] or 2<sup>nd</sup> order [21]). The Lambert targeter function calculates the heliocentric velocity vector of the spacecraft’s trajectory at each node [22, 23, 24]. For the Juno mission the planetary states  $\vec{r}_P$  and  $\vec{V}_P$  are found for Earth at launch JD0, Earth at the flyby Julian date, and Jupiter at arrival Julian date. Then the Lambert targeter computes the spacecraft heliocentric velocity  $\vec{V}_1^-$  at Earth departure, the velocity before and after the DSM  $\vec{V}_2^-$  and  $\vec{V}_2^+$ , before and after the Earth flyby  $\vec{V}_3^-$  and  $\vec{V}_3^+$ , and the spacecraft heliocentric velocity at Jupiter arrival  $\vec{V}_4^-$ .

Next, the specific mission event modules must be called to determine the  $\Delta V$  maneuver at each node. A loop iterates through the potential mission event function calls for each node, satisfying a conditional statement to select the desired mission event for that particular node. When the mission event is selected, the module is run and the  $\Delta V$  for that node is computed. The loop continues for the total number of nodes and sums all the  $\Delta V$ s and applying any specified weights (default is 1) as directed by the user input. For the Juno mission the mission events

used would be the Depart Launch, DSM, Optimal Powered Gravity Assist, and Arrive Parking Orbit. At each of these mission event modules the BVP is solved to determine the  $\Delta V$  maneuver. The Juno DSM  $\Delta V$  maneuver computation is simply  $\Delta V = |\vec{V}_2^+ - \vec{V}_2^-|$ .

The performance index with the totaled  $\Delta V$  for the iteration is analyzed by the selected optimization function alongside any linear constraints (equality or inequality) applied to the optimization parameters. The optimization function may also call a nonlinear constraint function that recomputes any necessary trajectory information (planet/body states, heliocentric and planetocentric BVP, etc) in order to determine the specified nonlinear inequality and/or equality constraints,  $H(\vec{X})$  and  $G(\vec{X})$ . The optimization function uses all of the computed data to determine how the optimization parameters will be adjusted in order to satisfy the constraints and reduce the total  $\Delta V$  performance index. The optimization parameters JD0, all the TOFs, and any DSM locations are updated, starting the trajectory building process again for the next iteration. The optimization process continues until the function's analysis of the performance index and constraints determines that it has converged upon a feasible minimum  $\Delta V$  solution that satisfies all the constraints. The a detailed flowchart diagram of the optimization process is seen in Figure 3.4.

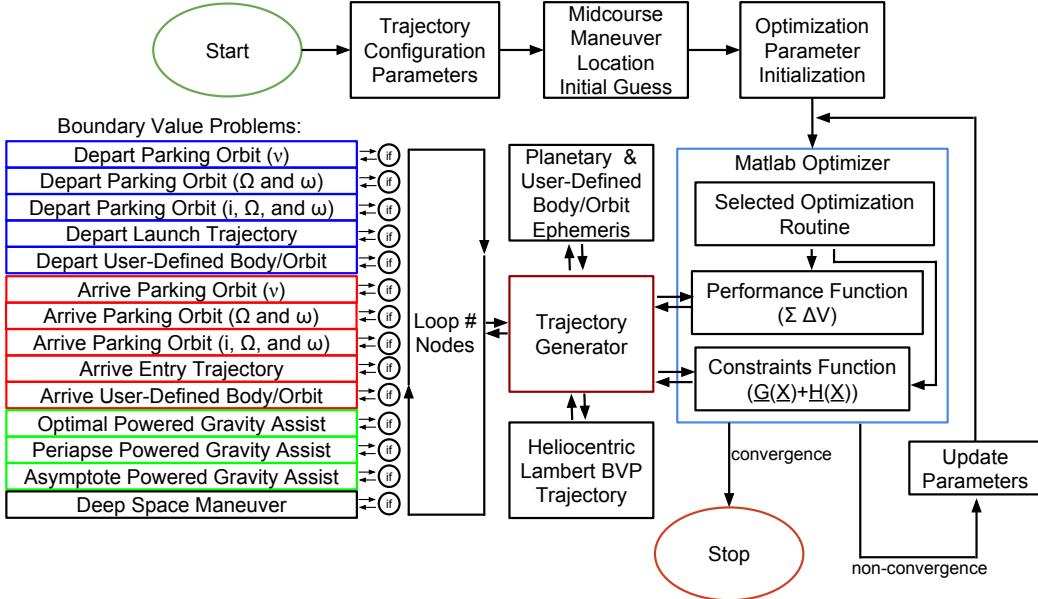


Figure 3.4: Trajectory optimization flowchart depicting the performance and constraint functions interacting with the various heliocentric and planetocentric boundary value problems through the trajectory generator loop

### 3.1.3 TRACT capabilities

#### 3.1.3.1 Planetary state ephemerides

The location and velocity vectors of the planets and the dwarf planet Pluto are retrieved from one of the two JPL ephemerides function options, the DE405 high fidelity ephemeris [19, 20] or a faster 2<sup>nd</sup> order ephemeris of lower fidelity[21]. The DE405 data file provides accurate ephemerides for the time range of 1600 AD to 2200 AD and accepted to have inner planet errors of about 0.001 arcseconds (about 1km at Mars distance) and about 0.1 arcseconds error for outer planets (about 360 km at Jupiter distance and 1450 km at Neptune distance) [19]. The 2<sup>nd</sup> order ephemeris have two different date ranges of applicability, 1850-2050 AD and 3000

BC - 3000AD. The 1850-2050BC range was found to produce roughly accurate positions compared to the DE405 ephemeris, where the inner planets had mean relative errors of about 5000 km or better, while the outer planets had mean relative errors of about 1 million km or better. When outside this typical mission trajectory date range, the 3000 BC - 3000AD ephemeris uses averaged planetary data and will have an even more degraded accuracy. These errors can be considered acceptable for rough trajectory planning considering the large planetary distances on the order of 100's of millions to billions of km. The 2<sup>nd</sup> order ephemeris provides about 10 times faster computation speed compared to the DE405 ephemeris.

### **3.1.3.2 Optimization schemes**

An effective optimization routine can manage the applied trajectory constraints, while minimizing the trajectory's total  $\Delta V$ , producing an acceptable and feasible trajectory. Some difficult trajectory designs for complex missions may prevent certain optimization schemes from reaching the desired trajectory solutions. Despite whether the difficulty is from the trajectory complexity or from a poor initial guess of the trajectory parameters (JD0, TOFs, and DSM [X Y Z]) by the user, a different optimization routine may be able to find the optimal minimum  $\Delta V$  trajectory solution. One of the major benefits of TRACT's uniform and modular architecture is that various different optimization routines can easily be incorporated. Fortunately, MATLAB provides many various optimization functions with vastly different capabilities and all with a standardized input/output structure. This makes adding them into TRACT effortless. An additional benefit of utilizing MATLAB's optimization functions, is that they offer easy access to many customizable

optimization options, such as adjusting the underlying search algorithm or even performing parallel computing. The list of available MATLAB optimization schemes configured in TRACT is found in Appendix E.

### 3.1.3.3 Trajectory constraints

Many different types of trajectory constraints can be applied to the appropriate optimization scheme, but they must be formatted to work correctly with the MATLAB constraint input structure. Fortunately, the constraint input structure is the same for every constrained optimization function implementing the specific constraint. The most efficient and effective constraints handled by the optimization functions are those applied directly to the optimization variables (JD0, TOFs, and DSM locations). Commonly used heliocentric trajectory constraints that apply to the optimization variables are provided, such as an upper bound on the total flight time or fixing the launch date JD0 to a specific value, etc. The list of linear constraints available in TRACT can be found in Appendix F. Any additional specialized linear constraints can be applied to the optimization parameter without difficulty by following the MATLAB guidelines.

Other nonlinear trajectory constraints can be applied through a separate nonlinear constraint function called by the optimization routine. Commonly used nonlinear trajectory constraints are provided, such as restricting a specific node's  $V_\infty$ , or applying traditional ballistic gravity assist constraints. The details on all of the nonlinear constraints available in TRACT can be found in Appendix F. Any additional specialized nonlinear constraints can be applied to the trajectory optimization process with minimal effort by adding the trajectory relationship in the standard form

outlined in MATLAB documentation [25].

Each planetocentric trajectory also incurs constraints in order to ensure the trajectory is mission feasible and abides by the linked conics assumptions. One of the most active constraints in planetocentric trajectories is to require that the trajectory never passes below a minimum radius threshold  $r_{min}$ . This constraint is applied by requiring the trajectory periapse be at or above the minimum radius:  $r_p \geq r_{min}$ . Another planetocentric constraint is that the periapse not exceed the planet's sphere of influence radius  $r_p \leq r_{SOI}$ . Similarly, the  $\Delta V$  maneuver locations  $r_{\Delta V}$  are not allowed to exceed the sphere of influence:  $r_p \leq r_{\Delta V} \leq r_{SOI}$ . These sphere of influence constraint enforces the linked conics assumptions that all planetary encounters occur within the sphere of influence.

### 3.1.3.4 Departure/arrival maneuvers

TRACT can model departures and arrivals as rendezvous with a body or planet, transferring with parking orbits about a body or planet, and as launch or entry conditions at a planet. The rendezvous assume the body or planet has no gravitational influence and the spacecraft is essentially performing a rendezvous with a heliocentric orbit at the same location. The parking orbits, launch, and entry conditions are modeled within the planet's frame (mean equator and node of date), using basic planetary north pole relations [15]. The details of these transformation equations can be found in the Masters thesis [18]. The available departure and arrival options were developed assuming common preliminary mission requirements, as outlined below. The transfer maneuver is assumed at the parking orbit periapse unless otherwise noted. The rendezvous maneuver algorithm simply computes the

difference in the heliocentric velocity vectors at the known location.

- Parking orbit with user defined semimajor axis  $a$  and eccentricity  $e$  are fixed. The inclination  $i$  is assumed  $\geq \delta_{\infty}^{\pm}$ , the  $\vec{V}_{\infty}^{\pm}$  asymptote declination. The right ascension of ascending node  $\Omega$  is determined analytically, while the argument of periapse  $\omega$  is optimized (if necessary) in order to find the orbit orientation that minimizes the transfer  $\Delta V$  maneuver.
- Parking orbit with user defined semimajor axis  $a$ , eccentricity  $e$ , and inclination  $i$  are fixed. The right ascension of ascending node  $\Omega$  is determined analytically, while the argument of periapse  $\omega$  is optimized (if necessary) in order to find the orbit orientation that minimizes the transfer  $\Delta V$  maneuver.
- Parking orbit with all fixed orbital elements defined by user except the transfer true anomaly  $v$ . The true anomaly is optimized to find the parking orbit location that minimizes the transfer  $\Delta V$  maneuver.
- Launch/entry condition with user defined available/acceptable  $V_{\infty}$  magnitude, asymptote declination (hyperbolic orbit inclination), and assumed transfer altitude are fixed. The right ascension of ascending node  $\Omega$  for the hyperbolic orbit is determined analytically, while the argument of periapse  $\omega$  is optimized (if necessary) in order to find the orbit orientation that minimizes the transfer  $\Delta V$  maneuver.

Each departure and arrival trajectory algorithm is formulated as a BVP, solving for the optimal transfer maneuver location and/or orbit orientation that yields the minimum  $\Delta V$  maneuver. Research in optimal transfers with parking orbits has led

to efficient BVP algorithms that focus only on providing the optimal  $\Delta V$  magnitude required for the maneuver. The details of the departure and arrival BVP algorithms are provided in the next section.

### 3.1.3.5 Gravity assist maneuvers

TRACT has extensive gravity assist maneuver options that model various powered and unpowered ballistic gravity assist trajectories. The user defines the minimum allowable radius  $r_{min}$  for the flyby periapse of each planetary encounter. The user can also restrict the gravity assist to a fixed periapse radius if desired. The available types of powered and unpowered ballistic gravity assist maneuvers are outlined below.

- Optimal powered gravity assist: optimizes the transfer maneuver location for a global minimum  $\Delta V$  maneuver during the flyby encounter. The algorithm ensures a strict adherence to the  $r_{min}$  constraint throughout the entire flyby transfer trajectory.
- Periapse powered gravity assist: assumes a periapse  $\Delta V$  maneuver during the flyby encounter. The algorithm determines the required periapse radius  $r_p$  and the associated collinear  $\Delta V$  maneuver. If  $r_p < r_{min}$  then one leg of the flyby is restricted as  $r_p = r_{min}$ , and the non-tangential  $\Delta V$  maneuver is determined to complete the transfer.
- Asymptote corrected ballistic gravity assist: assumes a ballistic flyby trajectory between the  $\vec{V}_\infty^-$  and  $\vec{V}_\infty^+$  asymptotes with  $\Delta V$  correction maneuver applied at one asymptote. The algorithm determines the  $\Delta V$  correction maneu-

ver required to complete a flyby and acts as a penalty to handle the ballistic gravity assist constraints. The penalty  $\Delta V$  maneuver will driven to a negligible near zero value, if possible, through the primary trajectory optimization.

- Traditional ballistic gravity assist: requires the traditional gravity assist constraints (Match- $\vec{V}_\infty^\pm$ ) to be applied to align the  $\vec{V}_\infty^-$  and  $\vec{V}_\infty^+$  asymptotes for a ballistic gravity assist. There is no associated  $\Delta V$  maneuver for this gravity assist option during the primary optimization process.

Each powered and unpowered gravity assist algorithm are formulated as a BVP. The optimal powered gravity assist BVP utilizes an optimization scheme tailored for efficiency and accuracy, determining the optimal transfer maneuver location true anomaly  $v_{\Delta V}$  and the minimum  $\Delta V$ . The periapse powered gravity assist BVP determines the optimal periapse radius  $r_p$  and required  $\Delta V$  maneuver. The asymptote corrected ballistic gravity assist BVP is a fast and simple algorithm that determines the required asymptote  $\Delta V$  penalty maneuver. The traditional ballistic gravity assist option does not require a  $\Delta V$  algorithm, but mandates the Match- $\vec{V}_\infty^\pm$  constraints be applied. The algorithms of the optimal powered gravity assist, the periapse powered gravity assist, and the asymptote corrected ballistic gravity assist are developed and presented in Chapter 4.

### 3.1.3.6 Deep space maneuvers

The objective of a DSM is to apply a  $\vec{\Delta V}$  in heliocentric space that redirects the heliocentric trajectory to achieve specific mission goals. The DSM could help synchronize or align the spacecraft's trajectory to properly meet with the next ce-

lestial body. The DSM could also be used to make the entire heliocentric trajectory more  $\Delta V$  efficient. This can be seen as a heliocentric orbital plane change between planetary encounters or when coupled with a gravity assist, the DSM acts as a  $V_\infty$  leveraging maneuver [26]. DSMs can also be used to split up and desensitize a sensitive trajectory segment that is becoming difficult to converge to the desired trajectory configuration. By splitting up a ballistic trajectory segment with DSMs, initially The DSMs will likely have nonzero  $\Delta V$ . But with an effective optimization strategy, the DSMs  $\Delta V$  magnitudes can often be reduced to zero while achieving the desired ballistic trajectory configuration [27, 28].

The DSM options are either the user explicitly defines the initial guess location for the DSM, or the midcourse maneuver function estimates initial guess locations for the user. The DSM maneuver is evaluated simply as the difference of the heliocentric incoming and outgoing velocity vectors.

$$\vec{\Delta V} = \vec{V}_{sc}^+ - \vec{V}_{sc}^- \quad (3.1)$$

### 3.1.3.7 MATLAB trajectory visualization output

One of the easiest ways for a mission designer to interpret and present a trajectory solution is by visualizing the physical trajectory in plots, which makes TRACT's trajectory visualization performance essential to its utility and success. Another important advantage of a MATLAB based trajectory design tool is the myriad plotting functions and graphics capabilities. TRACT has adaptable preconfigured heliocentric and planetocentric trajectory visualizations. Each trajectory simulation plot is represented in an interactive 3D graphical window, where the

user can zoom in/out and rotate perspectives in all directions. All trajectory plots can provide data sampling with “point-and-click” access on the orbital paths. Also any trajectory segment can be highlighted, labeled, or even removed for better interpretation, as well as countless other customization options.

The preconfigured trajectory visualization capability of TRACT provides a wealth of mission information to the user in an easy to digest format. This allows the user to quickly judge the performance of a trajectory design and adjust any necessary trajectory configuration options or optimization options. With TRACT’s fast computation time and instant trajectory visualization results, the user can become a seamless part of the optimization process. This powerful combination allows even novice mission planners or trajectory analysts to use trial and error methods to hone their trajectory design techniques.

To demonstrate the visualization capabilities of TRACT, the Juno mission to Jupiter is modeled. The Juno mission launched from Earth August 2011, performed a DSM as a  $V_{\infty}$ -leveraging maneuver before an Earth gravity assist. This maneuver combination enabled the Juno spacecraft to efficiently reach Jupiter, where an orbit insertion maneuver setup its highly eccentric polar orbit about the gas giant.

The preconfigured heliocentric visualizations provide the physical spacecraft trajectory path plotted among the orbits of encountered planets and any user defined orbits. The mission events and planetary encounters are depicted along the trajectory path, as well as the initial planetary locations for alignment perspective. An example with Juno mission’s heliocentric trajectory is shown in Figure 3.5. The spacecraft trajectory after launch continues outward to aphelion, where the DSM

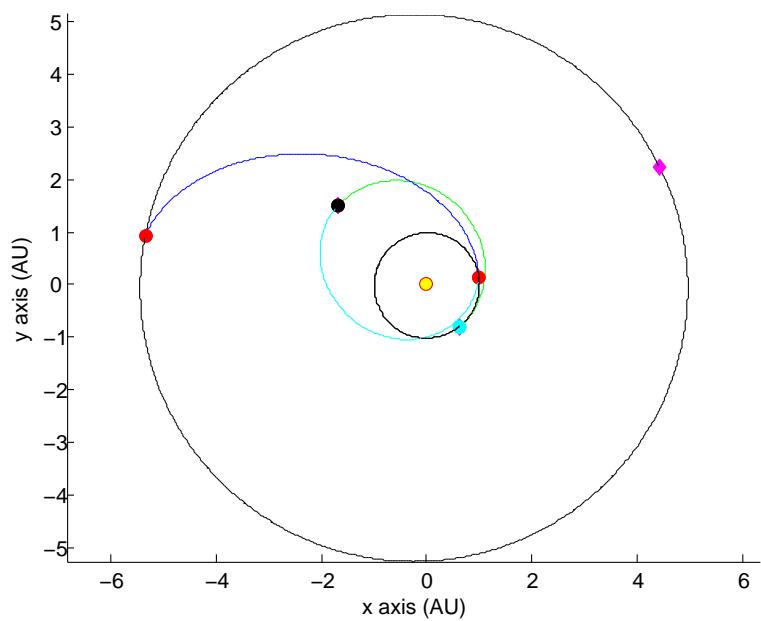


Figure 3.5: Juno mission heliocentric trajectory with Earth and Jupiter orbits depicting the launch (cyan), DSM (black), Earth gravity assist and Jupiter orbit insertion (red), and Jupiter's alignment at launch (magenta)

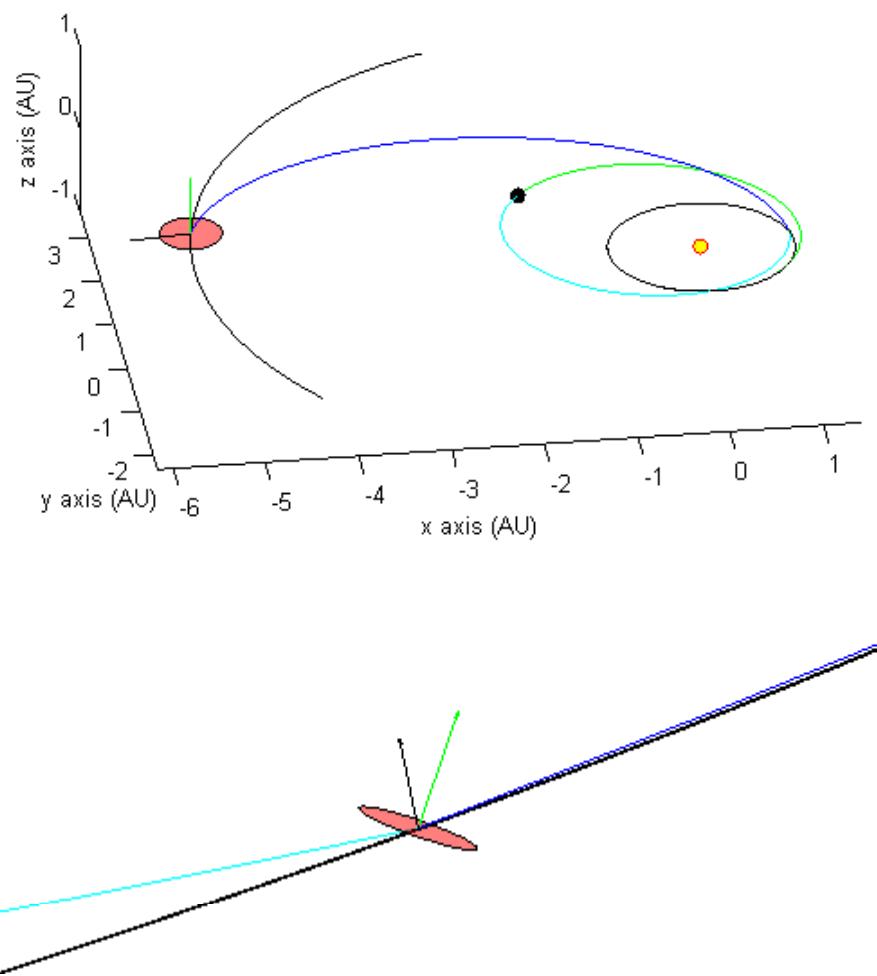


Figure 3.6: Juno mission heliocentric trajectory (top) with Earth and Jupiter orbits, depicting each planetary encounter by its SOI disk oriented normal to the planet's pole vector and showing the encounter angular momentum vector. Close up of Earth gravity assist displayed below

reduces the perihelion to below Earth’s orbit. The DSM  $V_\infty$ -leveraging maneuver adjusts the Earth flyby incoming  $\vec{V}_\infty^-$  orientation and magnitude to increase the gravity assist boost from Earth, enough to reach Jupiter.

Each planet’s SOI can be depicted at the encounter location along with the planet’s north pole axis vector and the spacecraft’s encounter angular momentum vector, Figure 3.6. These important markers aid the mission designer in clearly visualizing the entire trajectory’s heliocentric and planetocentric interactions on a single heliocentric trajectory simulation. Another useful feature is the trajectory motion simulation, which provides an interactive 3D graphical simulation environment showing accurate relative motion of the trajectory system. The user still maintains the ability to zoom and rotate the simulation perspective throughout the trajectory motion.

The preconfigured planetocentric plots depict the hyperbolic transfers and any parking orbits about a 3D planetary sphere of accurate scale. The  $\vec{V}_\infty^\pm$  asymptote direction vectors are presented as radial arrows in line with the planet center. The  $\vec{\Delta V}$  direction vector is shown at the transfer maneuver location. Also, the spacecraft and planet’s heliocentric velocity vectors are represented (transformed into the planetocentric frame) for orientation and clarity. An example of a departure is displayed by the Juno mission’s launch from Earth in Figure 3.7. The Juno gravity assist about Earth is shown in Figure 3.8. Also, the arrival into the highly eccentric Jupiter polar parking orbit is provided in Figure 3.9.

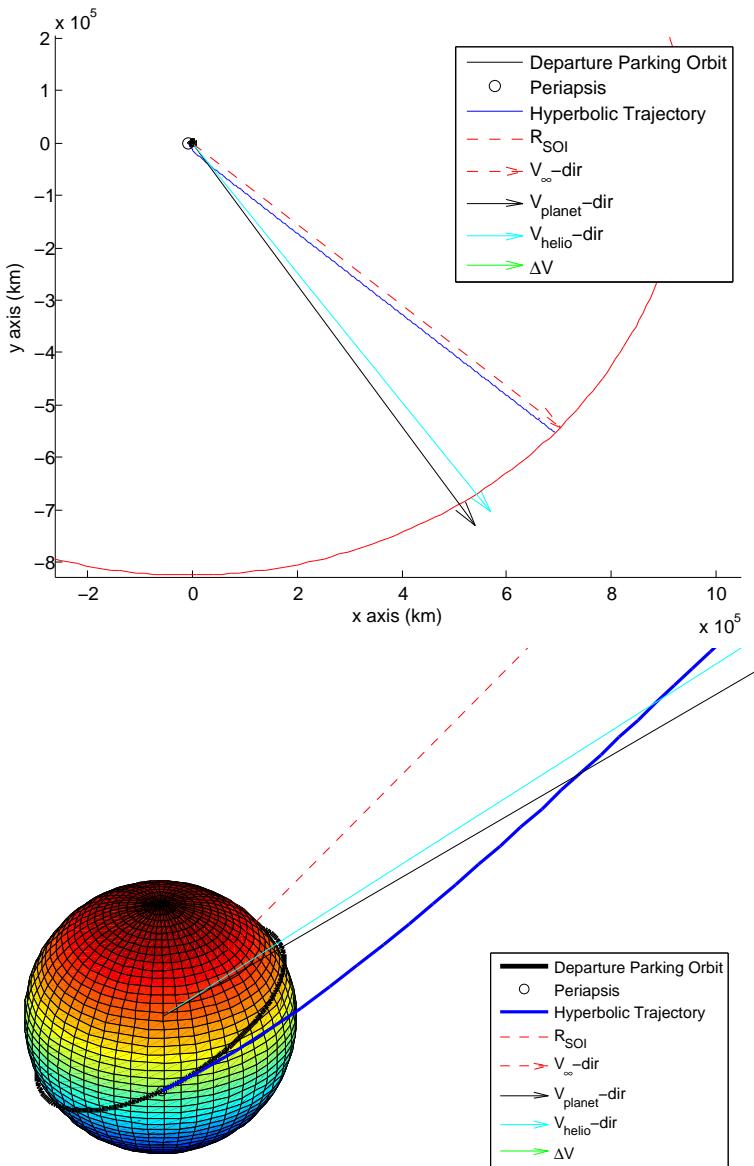


Figure 3.7: Juno mission launch trajectory from Earth, depicting the SOI and the various orientation vectors (top). A close up view of the hyperbolic launch trajectory is shown along with the launch “parking orbit” to reference the desired orientation of the launch plane and periapse altitude.

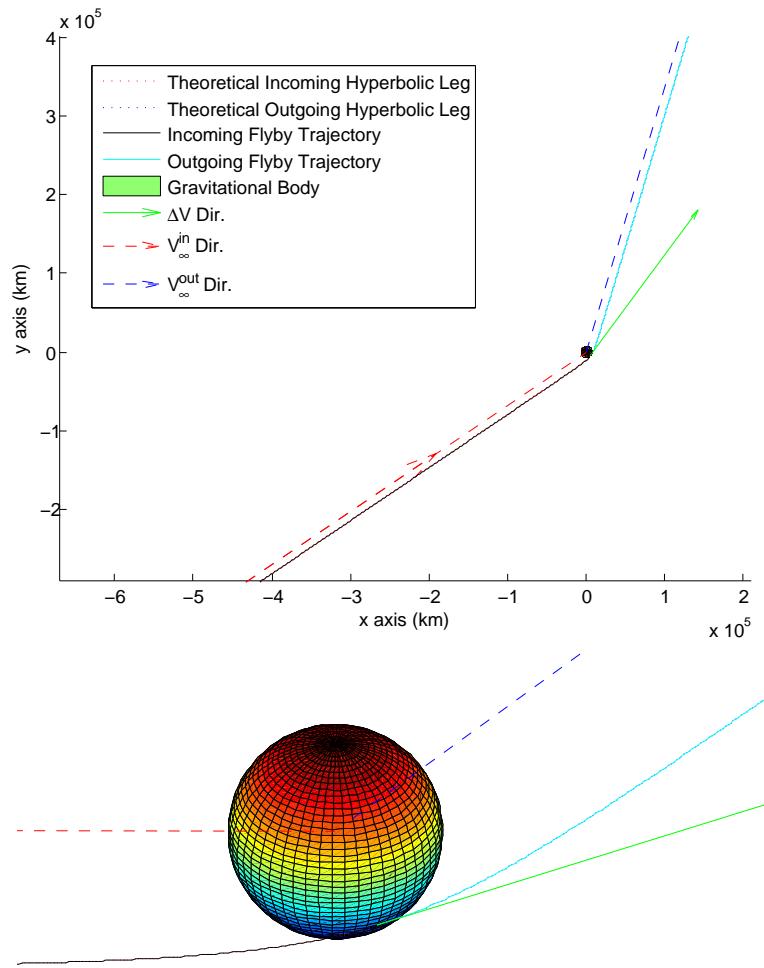


Figure 3.8: Juno mission Earth gravity assist, showing the incoming and exiting hyperbolic asymptotes as radial vectors from the planet (top). The negligible periapse  $\Delta V$  maneuver direction vector (green arrow) is also depicted. A close up view of the 800 km close encounter above the 3D spherical Earth model is shown below.

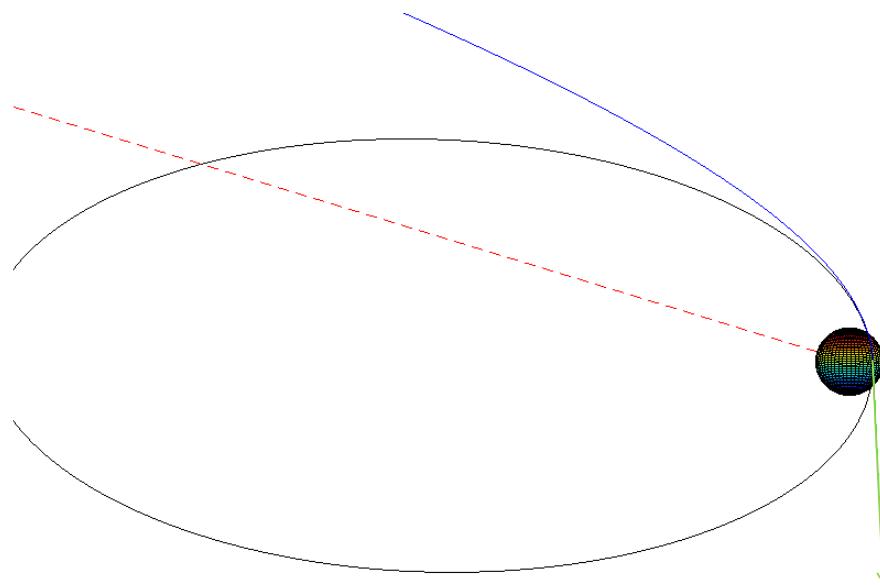
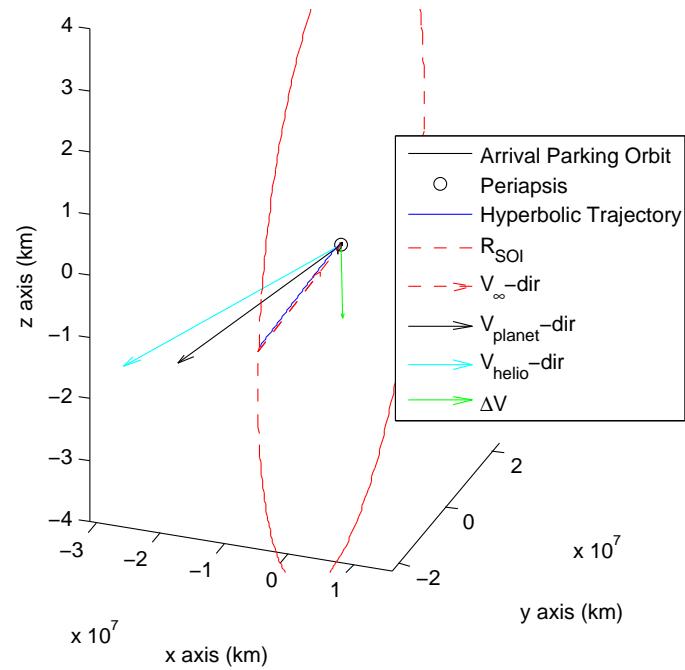


Figure 3.9: Juno mission arrival into the highly eccentric polar parking orbit about Jupiter depicting the SOI and the various orientation vectors (top). A close up view of the orbit insertion at periapse is shown about the 3D spherical Jupiter model.

### 3.1.3.8 Excel data output

|                         |            |                      |          |                      |              |                       |          |
|-------------------------|------------|----------------------|----------|----------------------|--------------|-----------------------|----------|
| <b>Trajectory Case:</b> | Juno.xlsm  | <b>Date Written:</b> | 06/20/15 | <b>Execute Time:</b> | 1.714 s      | <b>Exit Flag:</b>     | 2        |
| <b>Iterations:</b>      | 27         | <b>Func. Count:</b>  | 332      | <b>Step Size:</b>    | 0.000000     | <b>1st Order Opt:</b> | 0.0000   |
| <b>Ephemeris:</b>       | DE405      |                      |          |                      |              |                       |          |
| <b>Total ΔV</b>         | 1.707 km/s |                      |          | <b>Total TOF</b>     | 1719.96 days |                       | 4.71 yrs |

General Trajectory Performance

|               | <b>Planet</b> | <b>Date</b> | <b>Julian Date</b> | <b>TOF</b>  | <b>ΔV (km/s)</b> | <b><math>V_{\infty}^-</math> (km/s)</b> | <b><math>V_{\infty}^+</math> (km/s)</b> |
|---------------|---------------|-------------|--------------------|-------------|------------------|---|---|
| <b>Node 1</b> | 3             | 08/03/11    | 2455777.25         |             | 0.00000          |   | 5.550E+0                                |
| <b>Leg 1</b>  |               |             |                    | 392.56 days |                  |   |   |
| <b>Node 2</b> | 0             | 08/30/12    | 2456169.81         |             | 0.66883          | 15.531E+0                               | 14.863E+0                               |
| <b>Leg 2</b>  |               |             |                    | 400.16 days |                  |   |   |
| <b>Node 3</b> | 3             | 10/04/13    | 2456569.97         |             | 0.00000          | 10.072E+0                               | 10.072E+0                               |
| <b>Leg 3</b>  |               |             |                    | 927.24 days |                  |   |   |
| <b>Node 4</b> | 5             | 04/18/16    | 2457497.21         |             | 1.03863          | 5.578E+0                                |   |
|               |               |             |                    |             |                  |   |   |

Figure 3.10: Juno mission general trajectory data Excel spreadsheet example

Another useful tool for mission design is an organized spreadsheet of all the trajectory data frequently inspected and managed. TRACT takes advantage of MATLAB’s data export to Excel feature, providing all necessary trajectory and optimization data to a new Excel workbook. The Excel workbook includes a macro function that formats the data in clearly labeled and organized spreadsheets. The general trajectory data is categorized into the “General” tab spreadsheet, including the optimization information, trajectory calendar dates, Julian dates, TOF,  $V_{\infty}^{\pm}$  magnitudes, and associated  $\Delta V$  maneuvers. An example general trajectory data tab is shown for the Juno mission in Figure 3.10.

The departure and arrival trajectory information is organized into the “Dep. & Arr.” tab spreadsheet, including any parking orbit elements, required  $\Delta V$  maneuver, as well as any launch or entry data. All of the values presented in the spread-

sheets pertain to the final converged trajectory values, unless otherwise noted. The Juno mission example in Figure 3.11 shows the launch conditions, the Jupiter arrival parking orbital elements, orbit insertion  $\Delta V$  maneuver, etc. There are no departure parking orbit parameter values entered, because the mission launched directly into the interplanetary trajectory. Similarly, there are no arrival entry parameters, because the Juno mission did not perform an entry maneuver. The gravity assist trajectory information is found in the “Flybys” tab spreadsheet, including the flyby optimization type,  $V_{\infty}^{\pm}$  magnitudes, periapse radius  $r_p$ , hyperbolic turning angles  $\delta^{\pm}$ , required  $\Delta V$  maneuver, maneuver location radius  $r_{\Delta V}$ , etc. The Juno mission flyby information spreadsheet is displayed in Figure 3.12. It can be noted that the optimal periapse value converged upon is the minimum periapse. Any trajectory DSMs data is exported into the “DSM” tab spreadsheet, including the DSM location initial guess  $[X, Y, Z]$  coordinates, the optimized DSM location  $[X, Y, Z]$  coordinates, the incoming and exiting heliocentric velocities  $V_{\Delta V}^{\pm}$ , and the optimized DSM  $\Delta V$  maneuver, etc. The Juno mission DSM spreadsheet is shown in Figure 3.13.

| Departure Parameters   |  |                   |        |   |   |          |          |
|--|--|-------------------|--------|---|---|----------|----------|
| Planetary/Solar Parking Orbits (Mean Equator and Node of Date) |  |                   |        |   |   |          |          |
|  | $\mu$ (km <sup>3</sup> /s <sup>2</sup> ) | $\Delta V$ (km/s) | a (km) | e | i | $\Omega$ | $\omega$ |
|  |  |                   |        |   |   |          |          |
|  |  |                   |        |   |   |          |          |

| Planetary Launch |  |                   |                       |                       |          |                       |
|------------------|--|-------------------|-----------------------|-----------------------|----------|-----------------------|
|                  | $\mu$ (km <sup>3</sup> /s <sup>2</sup> ) | $\Delta V$ (km/s) | $V_{inf}$ Avl. (km/s) | $V_{inf}$ Req. (km/s) | Launch i | Alt <sub>p</sub> (km) |
| <b>Node 1</b>    | 3.986E+5                                 | 0.000E+0          | 5.550E+0              | 5.550E+0              | 28.500 ° | 2.00E+2               |
|                  |  |                   |                       |                       |          |                       |
|                  |  |                   |                       |                       |          |                       |

| Arrival Parameters   |  |                   |          |          |          |           |           |
|--|--|-------------------|----------|----------|----------|-----------|-----------|
| Planetary/Solar Parking Orbits (Mean Equator and Node of Date) |  |                   |          |          |          |           |           |
|  | $\mu$ (km <sup>3</sup> /s <sup>2</sup> ) | $\Delta V$ (km/s) | a (km)   | e        | i        | $\Omega$  | $\omega$  |
| <b>Node 4</b>  | 1.267E+8                                 | 1.039E+0          | 1.432E+6 | 9.471E-1 | 90.000 ° | 276.088 ° | 176.007 ° |
|  |  |                   |          |          |          |           |           |
|  |  |                   |          |          |          |           |           |

| Planetary Entry |  |                   |                       |                       |          |                       |
|-----------------|--|-------------------|-----------------------|-----------------------|----------|-----------------------|
|                 | $\mu$ (km <sup>3</sup> /s <sup>2</sup> ) | $\Delta V$ (km/s) | $V_{inf}$ Avl. (km/s) | $V_{inf}$ Req. (km/s) | Launch i | Alt <sub>p</sub> (km) |
|                 |  |                   |                       |                       |          |                       |
|                 |  |                   |                       |                       |          |                       |

Figure 3.11: Juno mission departure and arrival trajectory data Excel spreadsheet example

| Powered / Unpowered Flyby Summary    |              |  |                   |                    |                    |                        |                         |
|--------------------------------------|--------------|--|-------------------|--------------------|--------------------|------------------------|-------------------------|
| Orbital State at $\Delta V$ maneuver |              |  |                   |                    |                    |                        |                         |
|                                      | Event        | $\mu$ (km <sup>3</sup> /s <sup>2</sup> ) | $\Delta V$ (km/s) | $V_{inf}^-$ (km/s) | $V_{inf}^+$ (km/s) | $R_{\Delta V}$ (radii) | $V_{\Delta V}^-$ (km/s) |
| <b>Node 3</b>                        | Powered Peri | 3.986E+5                                 | 4.974E-8          | 10.072E+0          | 10.072E+0          | 1.125E+0               | 14.577E+0               |
|                                      |              |  |                   |                    |                    |                        |                         |
|                                      |              |  |                   |                    |                    |                        |                         |

| Bend Angle ( $\phi = 2\beta$ ) |           |           |            |            |                    |                    |
|--------------------------------|-----------|-----------|------------|------------|--------------------|--------------------|
|                                | $\beta^-$ | $\beta^+$ | $\delta^-$ | $\delta^+$ | $t_{soi}^-$ (days) | $t_{soi}^+$ (days) |
| <b>Node 3</b>                  | 69.283 °  | 69.283 °  | 41.435 °   | 41.435 °   | 1.044E+0           | 1.044E+0           |
|                                |           |           |            |            |                    |                    |
|                                |           |           |            |            |                    |                    |

|               | Minimum             | Optimized           | Periapsis Velocities |                | Aiming Distance    |                    |
|---------------|---------------------|---------------------|----------------------|----------------|--------------------|--------------------|
|               | $R_{p-min}$ (radii) | $R_{p-opt}$ (radii) | $V_p^-$ (km/s)       | $V_p^+$ (km/s) | $\Delta^-$ (radii) | $\Delta^+$ (radii) |
| <b>Node 3</b> | 1.125E+0            | 1.125E+0            | 14.577E+0            | 14.577E+0      | 1.629E+0           | 1.629E+0           |
|               |                     |                     |                      |                |                    |                    |
|               |                     |                     |                      |                |                    |                    |

Figure 3.12: Juno mission flyby trajectory data Excel spreadsheet example

| Space Burn / Midcourse Maneuvers |                                    |           |                        |                              |                                     |                                     |
|----------------------------------|------------------------------------|-----------|------------------------|------------------------------|-------------------------------------|-------------------------------------|
|                                  | ΔV RA & DEC in Mean Ecliptic J2000 |           |                        | Orbital State at ΔV maneuver |                                     |                                     |
|                                  | Event                              | ΔV (km/s) | RA <sub>ΔV</sub> (deg) | DEC <sub>ΔV</sub> (deg)      | V <sub>ΔV</sub> <sup>-</sup> (km/s) | V <sub>ΔV</sub> <sup>+</sup> (km/s) |
| <b>Node 2</b>                    | SB                                 | 6.688E-1  | 48.893E+0              | 26.175E-2                    | 15.531E+0                           | 14.863E+0                           |

|               | Maneuver Location Guess |          |           | Maneuver Location Final |          |           |
|---------------|-------------------------|----------|-----------|-------------------------|----------|-----------|
|               | X (AU)                  | Y (AU)   | Z (AU)    | X (AU)                  | Y (AU)   | Z (AU)    |
| <b>Node 2</b> | -1.779E+0               | 1.407E+0 | -1.136E-4 | -1.771E+0               | 1.416E+0 | -1.135E-4 |

Figure 3.13: Juno mission DSM trajectory data Excel spreadsheet example

## 3.2 Bi-level optimization

### 3.2.1 Bi-level problem definition

Bi-level optimization consists of an outer loop primary optimization scheme and nested inner loop secondary optimization sub-problems. For each iteration of the outer loop primary optimization, each inner loop secondary optimization routine is run until converged upon the sub-problem solution, where the results aid the primary optimization search. The primary level (outer loop) optimization variables are high-level parameters that define and affect many aspects of the entire problem set. Each secondary level (inner loop) optimization routine utilizes the high-level parameters as constants to define the sub-problem and/or initialize the optimization. The secondary level optimization routines are typically used to solve complex dynamical problems separately to provide only optimal and/or feasible solutions for the primary level (outer loop) optimization to judge the set of high-level parameters that best solves all the individual sub-problems. The bi-level optimization process continues to iterate the outer loop primary level optimization variables and rerun the secondary level inner loop routines until convergence on the optimal set of primary

optimization parameters [29, 30].

Recent trajectory research has utilized bi-level optimization for global trajectory optimization, where the outer loop optimizes the discrete trajectory sequence and the inner loop is a typical single-level trajectory optimization scheme [31, 32, 33]. Research by Lantukh and Russell involves developing pathfinding methods and trajectory BVPs to systematically search the various trajectory sequences for those that meet the optimization criteria (minimum  $\Delta V$ , minimum TOF, etc.) [34, 35, 36]. Other research employs multi-level optimization techniques to solve complex multiple gravity assist continuous trajectory systems and incorporating low thrust trajectories into these difficult problems [37, 38]. The current study extends the scope of a bi-level optimization scheme for a fixed trajectory sequence, where heliocentric trajectory parameters are optimized by the outer loop primary optimization routine and each planetocentric trajectory is optimized independently in the inner loop secondary optimization routines. This is compared to the typical interplanetary trajectory optimization schemes having a single optimization routine that handles all of the heliocentric and planetocentric variables at the same time.

The trajectory optimization problem as formulated with linked conics is a prime candidate for bi-level optimization, where the high-level trajectory variables (JD0, TOFs, and DSM locations) define each heliocentric and planetocentric trajectory segment as separate BVPs. Most linked conics trajectory optimization tools utilize some form of a Lambert targeting BVP routine to determine the heliocentric trajectory segments, which could be considered a secondary level (inner loop) optimization routine. But many of these tools parameterize the planetocentric trajectories with variables to be included in a single-level monolithic optimization routine

[39, 40, 41, 42, 43], where all of the heliocentric and planetocentric parameters are included in the single-level optimization routine. There are some advantages to having all trajectory segment parameters (heliocentric and planetocentric) being optimized by a single optimization scheme, such as fast convergence when all of the optimization parameters are already close to the solution (usually minimizing total  $\Delta V$  of the trajectory). Although, the bi-level optimization method presented in this research offers an alternative set of advantages that include easily handling rough or poor initial guess trajectory values and allowing the optimization process to take large steps towards reaching the optimal trajectory configuration solution. The details of the different advantages and disadvantages of these optimization strategies are discussed in the Optimization Performance section below.

A powerful alternative method implemented in TRACT is to solve all trajectory segments, including planetocentric trajectories, as separate BVP algorithms. Using separate and independent BVPs to determine trajectory segments enables adaptive solution strategies within each BVP algorithm to determine the optimal minimum  $\Delta V$  trajectory segment solution. An important impact of using BVPs for all trajectory segments is substantially reducing the number of primary level optimization parameters. Additionally, the planetocentric trajectory BVPs do not require any initial guess parameters from the user. This alternative method is only possible through the use of linked conics and other assumptions previously defined. The key feature that justifies this strategy is that the trajectory solution of each planetocentric trajectory has no affect on the any other trajectory segment solutions, heliocentric or planetocentric, for a given set of heliocentric conditions. Considering this property, there are many advantages to only providing the primary

optimization scheme with optimal minimum  $\Delta V$  planetocentric trajectory solutions and satisfying all planetocentric constraints, some of which are mentioned above.

### 3.2.2 Linked BVPs architecture

A major advantage of a bi-level optimization scheme is an adaptable modular architecture. TRACT's linked BVPs structure provides versatility to add new trajectory BVP algorithm modules (DSMs, flybys, arrivals, etc.) effortlessly. With modular and independent BVPs to be computed, MATLAB's easy to implement parallel computing functions can be used effectively. Using modular BVPs also facilitates a standardized set of heliocentric optimization variables for the primary optimization scheme, where the optimization algorithm can be completely switched without any change to the underlying BVPs.

Each planetocentric BVP algorithm has been developed to determine the optimal minimum  $\Delta V$  planetocentric trajectory solution without any initial guess provided by the user. With the primary optimization routine only handling the heliocentric trajectory variables (JD0, TOFs, and DSM locations) the number of optimization variables is substantially reduced. Therefore, using a bi-level optimization scheme with linked BVPs in TRACT reduces the amount of trajectory information and initial guess values to be provided by the user. A single-level optimization routine accommodating all trajectory variables and constraints typically will require the user to provide initial guess values for all of these parameters, putting a heavier burden on the mission planner or trajectory designer.

A form of the linked BVP methodology is also seen in trajectory design tools developed by Lantukh and Russell, where the focus is on determining the he-

heliocentric trajectories with BVPs to automatically include  $V_\infty$ -leveraging maneuvers and ballistic gravity assists [34, 35, 36]. Similar pathfinding strategies using linked heliocentric BVPs is frequently utilized in global optimization problems, such as those posed by the Global Trajectory Optimization Competition (GTOC) [44].

### 3.2.3 Optimization performance

Most single-level optimization schemes can perform better if the problem is well formulated to reduce the number of optimization variables [45, 46]. With a bi-level optimization architecture, the number of primary level (outer loop) optimization variables is further reduced by using secondary level optimization algorithms. Using a bi-level optimization scheme, each primary level heliocentric optimization iteration is provided the optimal (minimum)  $\Delta V$  for all associated planetocentric segments. Therefore the primary level optimization is given a more direct and accurate measure of the  $\Delta V$  performance of each set of heliocentric trajectory parameter values and can better guide the heliocentric optimization search.

Even with a reduced set of primary level optimization and always using optimal planetocentric trajectories it is not certain whether a bi-level optimization scheme using gradient-based Newton-like optimization algorithms would achieve better convergence performance for a specific problem compared to a single-level optimization scheme [30, 47]. This specific topic is discussed in more detail below. But there can be more obvious benefits when using gradient-free optimization algorithms in a bi-level scheme. Gradient-free optimization algorithms such as directed methods (simplex, pattern search, etc.) or even stochastic methods (genetic algorithms, particle swarm, etc.) explore the solution space of the problem and judge

the performance for each newly generated set of parameter values [48]. These directed and/or stochastic methods are an effective strategy for difficult optimization problems and frequently implemented in trajectory design [49, 50, 33].

In TRACT, gradient-free optimization algorithms have better performance with a reduced set of optimization parameters. By requiring stochastic algorithms to include planetocentric trajectory variables such as in a single-level optimization scheme, each candidate heliocentric trajectory configuration may be misrepresented by an inflated sub-optimal  $\Delta V$  performance, due to the sensitive planetocentric trajectory parameters not being exactly at their optimal condition. In single-level gradient-based optimization schemes, the planetocentric trajectory parameters can be better guided towards their optimal states as the heliocentric trajectory changes. Yet, many stochastic algorithms cannot reliably provide this characteristic for each iteration and suffer excessive optimization iterations and degraded performance in order to converge upon optimal heliocentric and planetocentric trajectory solutions together [48].

It is well documented that when trajectory initial guess values are close to the optimal conditions, the solution space can be approximated as a quadratic, and therefore Newton-like optimization algorithms quickly converge to the optimal solution [45, 46]. Single-level optimization trajectory segment functions are only computed once per optimization iteration, and with these near optimal initialization conditions, even their planetocentric trajectories are likely corrected well enough per iteration for each to be driven to its optimal minimum  $\Delta V$  trajectory solution quickly. Although the bi-level technique potentially has multiple function calls for the planetocentric trajectory segments, each heliocentric iteration is guar-

anteed to be provided the minimum total  $\Delta V$  performance for that heliocentric configuration. This can lead to more accurate primary level heliocentric optimization search directions and may reach the optimal solution in fewer iterations compared to the single-level optimization scheme. Considering these characteristics it cannot be certain which trajectory optimization strategy provides the fastest convergence generally, and therefore must be investigated on a specific case-by-case basis. This investigation is beyond the scope of this dissertation, but is intended for future research. Although, other bi-level optimization convergence studies not specific to spacecraft trajectory problems have presented similar optimization comparisons as stated above and have proven superlinear convergence for the bi-level algorithms analyzed [47, 48, 30].

Another benefit of TRACT's bi-level optimization scheme is its ability to efficiently take large steps in the primary level heliocentric optimization procedure. When provided rough (inaccurate) initial guess estimates to the heliocentric variables, the primary optimization procedure will likely begin to take large steps navigating the search space of these heliocentric variables. These large changes in the heliocentric trajectory lead to very large changes in the boundary conditions for the planetocentric trajectories. The solution space dynamics of the planetocentric trajectories can be very sensitive to the changes in the boundary conditions. Since TRACT's planetocentric BVP algorithms are tailored to efficiently and robustly solve for the optimal trajectory solution for any boundary conditions, these large changes are inconsequential. Yet it can be very difficult for a single-level optimization scheme to accurately estimate such sensitive planetocentric dynamics through large heliocentric steps. This can be especially true in the initial iterations

of Newton-like algorithms, which commonly rely on building up second derivative information about each variable (the system's Hessian) in order to make more accurate estimates of variables for the next iterations [46].

An option for improving the convergence performance of the overall bi-level optimization scheme is to allow the secondary level optimization procedures to be initialized using the previous iteration's solution. If used for conditions close to the optimal solutions, this modification would reduce the number of iterations required for the secondary optimization algorithms and effectively mimic the Newton-like convergence of the overall problem. Such an option could be used from the onset of the bi-level optimization process or turned on when certain conditions are detected. The uninitialized BVPs would likely be more effective at the onset of the bi-level optimization and then switching to the initialized BVPs when in the neighborhood of an optimal solution. Though this option is not yet implemented in TRACT, it is intended for future work.

### 3.3 Primary level optimization procedure

The primary level optimization procedure for TRACT is standardized to accommodate a variety of different optimization algorithms. The user provided initial guess values for heliocentric trajectory (JD0, TOFs, and DSM locations) are input as the primary optimization variables for the commonly used deterministic optimization algorithms, such as fminunc, fminsearch, and fmincon, etc. There are many other unconstrained and constrained MATLAB optimization routines configured for use in TRACT, the full list of these options can be found in Appendix E.

The most popular unconstrained optimization routines are fminunc using a quasi-Newton gradient based optimization algorithm and fminsearch using the Nelder-Mead simplex method[51]. The most popular optimization algorithm that applies constraints is fmincon, which uses a variety of Newton-like gradient based optimization algorithms (interior-point, SQP, etc.) and allows for linear and nonlinear constraints to be applied to the problem.

The selected primary optimization algorithm iterates on these heliocentric parameter values and immediately applies any linear constraints on the optimization variables, then providing them to the secondary level BVPs. For each set of heliocentric parameters, the heliocentric and planetocentric trajectory BVPs are executed to provide the optimal minimum  $\Delta V$  performance of the interplanetary trajectory configuration. For gradient-based optimization routines, the gradient of heliocentric trajectory parameters is also computed using numerical-differences. The primary level optimization procedure then computes the nonlinear constraints function (if applicable) which recomputes any necessary heliocentric and planetocentric trajectory data in order to determine the constraint performance equations. Depending on the optimization routine selected, the primary optimization algorithm uses the resulting  $\Delta V$  performance value, its gradient information, and the nonlinear constraints results to determine the adjustments to be made to the heliocentric trajectory parameters. This process continues until the primary optimization scheme determines that convergence conditions have been met while satisfying all constraints or other stopping conditions . Other stopping conditions can include exceeding the maximum number of iterations, or function evaluations, etc.

When using gradient-free optimization methods, the required computation

per iteration is substantially reduced by avoiding the intensive gradient computations. Although, these gradient-free methods typically complete many iterations and therefore still end up requiring a large number of trajectory function evaluations. Some of the stochastic optimization routines from MATLAB’s global optimization toolbox do not require initial guess values, such as the genetic algorithm or particle swarm optimization routines. These stochastic global optimization algorithms only require the number of parameters along with as upper and lower bound on each parameter value.

In order for the user to quickly determine if the trajectory parameter initial guesses are close to producing the desired trajectory, TRACT offers the option to bypass the optimization process and determine the trajectory with the fixed input values. The trajectory produced is guaranteed to be continuous and feasible in location throughout the entire trajectory due to utilizing BVPs for every heliocentric and planetocentric leg. Many other trajectory design tools allow for location discontinuities in their trajectory and/or allow infeasible planetocentric locations, such as a gravity assist periapse falling below the planet’s  $r_{min}$  threshold. These trajectory design tools require their optimization routines to manage these discontinuities and constraints for a feasible trajectory. TRACT’s fixed trajectory option allows the user to quickly generate trajectories, judge what the effects are of changing specific inputs, and find a better initial trajectory configuration for the optimization process. This option can also be used as a sensitivity tool to evaluate the characteristics of the optimization space by trying various boundary conditions of the trajectory configuration. This gives the user a better idea of the entire optimization space.

## 3.4 Secondary level BVPs

The secondary level inner loop trajectory optimization problems are formulated as separate BVP algorithms. The heliocentric trajectory segments are determined by a Lambert targeting BVP algorithm. When given the  $\vec{V}_\infty^\pm$  boundary conditions, each planetocentric trajectory segment is determined by a specialized algorithm developed for efficiently and reliably computing the optimal minimum  $\Delta V$  encounter trajectory. Each planetocentric BVP algorithm is formulated to inherently satisfy the  $r_{min}$  and  $r_{SOI}$  constraints required for a feasible trajectory and supporting the linked conics methodology. Each planetocentric BVP algorithm is described below.

### 3.4.1 Departure/arrival transfer with parking orbit

The parking orbit departure/arrival options are designed to address the common interplanetary trajectory design plans while restricting the BVP algorithm to a single optimization variable. In most parking orbit options the single optimization variable is the orbit orientation parameter, argument of periapse  $\omega$ . In the parking orbit option with all fixed orbit parameters the only optimization variable available is the true anomaly of the departure/arrival  $\Delta V$  maneuver  $v_{\Delta V}$ . Well known orbital relationships of the parking orbit transfer to a hyperbolic escape trajectory problem are employed to analytically compute necessary orbit elements and initialize the remaining orbital element with a good estimate for optimization. When the conditions are available, an analytic algorithm is used to determine the optimal transfer  $\Delta V$  maneuver. Each algorithm is briefly described below, but further details can be found in the Masters thesis [18] and other resources on this common orbital

problem [42, 10].

The BVP algorithm is simple for departure/arrival transfer with a parking orbit assuming inclination  $i \geq \delta_\infty^\pm$ , while  $\Omega$  and  $\omega$  are free to be determined for minimizing the transfer  $\Delta V$  maneuver. The algorithm assumes the transfer maneuver occurs at the parking orbit periapse. These assumptions allow the parking orbit orientation free to be coplanar with the  $\vec{V}_\infty^\pm$  asymptote, which is the optimal orientation for a minimum  $\Delta V$  collinear maneuver [42]. Since the primary goal of the BVP is to deliver the minimum required  $\Delta V$  magnitude, then the algorithm is simply the difference in periapse velocity magnitudes:

$$\Delta V = \sqrt{\frac{2\mu}{r_p} + V_\infty^2} - \sqrt{\mu \left( \frac{2}{r_p} - \frac{1}{a} \right)} \quad (3.2)$$

The BVP algorithm for departure/arrival transfer with a parking orbit of fixed inclination  $i$  and free  $\Omega$  and  $\omega$  is more complicated. First the  $\vec{V}_\infty^\pm$  asymptote declination  $\delta_\infty^\pm$  is checked for the coplanar condition  $i \geq \delta_\infty^\pm$ ; if so then the simple algorithm above is used. But if  $i < \delta_\infty^\pm$ , then the transfer is non-coplanar and orbit's orientation must be optimized by varying  $\omega$  to find the minimum  $\Delta V$  transfer configuration. The orbit  $\Omega$  can be determined analytically from the orbit node vector  $\hat{n}$  defined by  $\hat{V}_\infty^\pm$  direction vector in Equation 3.3, where  $\beta = \pi/2 - \delta_\infty$ . The optimal orientation of the orbit's angular momentum vector projection on the planet frame's x-y plane is exactly opposite of the  $\hat{V}_\infty^\pm$  asymptote direction vector projection [10].

$$\begin{aligned} \hat{n} &= \text{sign}(\delta_\infty) \left[ \frac{\hat{V}_{\infty Y}}{\sin \beta} \hat{X} - \frac{\hat{V}_{\infty X}}{\sin \beta} \hat{Y} \right] \\ \Omega &= \text{ATAN2}(\hat{n}_Y, \hat{n}_X) \end{aligned} \quad (3.3)$$

A Newton-like optimization scheme is used to optimize  $\omega$ . A good initial guess estimate is found to be  $\omega = -\text{sign}(\delta_\infty) \frac{\pi}{2} + \frac{\pi}{4}$ , since it is known that the optimal orientation of the parking orbit periapse should lie between right at the ascending node  $\omega = 0$  and about  $\frac{\pi}{2}$  away from the ascending node.

Though it is always more desirable for a parking orbit inclination to be  $i \geq \delta_\infty^\pm$  for an efficient maneuver, this condition cannot be guaranteed for a fixed inclination when the heliocentric trajectory is potentially undergoing large changes in the optimization process. Therefore, the non-coplanar maneuver must be implemented for these sub-optimal conditions. The larger  $\Delta V$  maneuver magnitude required for a non-coplanar maneuver is found to be extremely effective at influencing the primary level optimization of the heliocentric trajectory to avoid these conditions. An example of a difficult fixed inclination parking orbit to design a mission for is an equatorial orbit about the planet Uranus, which has its equator oriented nearly perpendicular to the ecliptic plane. This means that if the heliocentric trajectory is not precisely aligned the planetocentric orbit insertion is always non-coplanar. Therefore a parking orbit BVP such as that described above is required to provide strong incentive for the optimization procedure to align the heliocentric trajectory for a coplanar orbit insertion, without having to resort to applying complex nonlinear constraints.

The BVP algorithm for the departure/arrival transfer with a fixed parking orbit except for the  $\Delta V$  maneuver location true anomaly  $v_{\Delta V}$ . This restricted orbit orientation always requires optimizing  $v_{\Delta V}$  for minimum transfer  $\Delta V$  maneuver. A Newton-like optimization scheme is used. A good initial guess estimate is found to be similar to the above methodology, where the  $\hat{V}_\infty^\pm$  projection can be used as a

reference. The transfer maneuver location estimate on the parking orbit projection is at an angle  $\frac{\pi}{2}$  from the  $\hat{V}_\infty^\pm$  projection. The periapse vector location is found using the fixed orbital elements and the  $v_{\Delta V}$  is initialized accordingly.

### 3.4.2 Departure/arrival transfer with launch/entry conditions

The BVP algorithm for launch/entry conditions is similar to the parking orbit BVPs. With the given  $V_\infty^\pm$  magnitude available to the launch/entry, the required inclination  $i$ , and the assumed maneuver periapse altitude, first the declination of the actual  $\vec{V}_\infty^\pm$  is compared with the required inclination. If the coplanar condition exists  $i \geq \delta_\infty^\pm$  then the maneuver is collinear and the simple  $\Delta V$  algorithm is the difference in hyperbolic periapse velocities.

$$\Delta V = \sqrt{\frac{2\mu}{r_p} + V_\infty^2|_{av}} - \sqrt{\frac{2\mu}{r_p} + V_\infty^2} \quad (3.4)$$

For the non-coplanar condition  $i < \delta_\infty^\pm$ , the hyperbolic transfer orbit orientation is optimized similar to the parking orbits, where the orbit elements  $a$ ,  $e$ , and  $i$  are known and  $\Omega$  is determined analytically, while  $\omega$  must be optimized to find the minimum  $\Delta V$  configuration, assuming a periapse transfer maneuver location. The initial guess estimate for  $\omega$  is the same as above. Also the user can specify if  $V_\infty|_{av}$  (the available  $V_\infty$ ) condition should be met exactly as  $V_\infty^\pm = V_\infty|_{av}$  and the  $\Delta V$  computed as stated above. The other option is if lower  $V_\infty^\pm$  magnitudes are acceptable where  $V_\infty^\pm \leq V_\infty|_{av}$  and the maneuver is  $\Delta V = 0$  for lesser  $V_\infty^\pm$  magnitudes.

### 3.4.3 Gravity assists

TRACT provides four options for modeling gravity assists in an interplanetary trajectory: optimal powered gravity assist, periapse powered gravity assist, asymptote corrected ballistic gravity assist, and ballistic gravity assist constraints. Each of these algorithms are briefly described below, but further details on the algorithms, their development, and associated analysis can be found in Chapter 4. The definition, characterization, and implementation of these four gravity assist options constitute major contributions of this research.

The optimal powered gravity assist algorithm uses an efficient and robust cubic interpolation optimization method to determine the optimal hyperbolic flyby maneuver location and the associated minimum  $\Delta V$  maneuver. The algorithm is designed to minimize optimization iterations and determine the optimal solution analytically when possible. It features strict enforcement of the  $r_{min}$  and  $r_{SOI}$  constraints. The resulting optimal powered gravity assist trajectories are determined either as unconstrained solutions, where the flyby periapse is between  $r_{min}$  and  $r_{SOI}$ , or as constrained solutions where the periapse lies on one of the constraint boundaries.

The periapse powered gravity assist algorithm uses Newton's root-finding algorithm to efficiently determine the required periapse radius  $r_p$  for a hyperbolic powered flyby with the incoming and outgoing legs meeting at their common periapse, where a collinear  $\Delta V$  maneuver occurs. While the periapse is above the minimum level  $r_p > r_{min}$  and below the maximum  $r_p < r_{SOI}$ , the transfer uses a collinear  $\Delta V$  maneuver. Otherwise a non-tangential maneuver is computed at the

constrained periapse  $r_p$  of one hyperbolic leg.

The asymptote corrected ballistic gravity assist BVP is a fast analytic algorithm intended to account for the ballistic gravity assist conditions by providing a correction  $\Delta V$  maneuver at one of the hyperbolic asymptotes,  $\vec{V}_\infty^\pm$ . The maneuver acts as a penalty for gravity assist BVP conditions that don't support a ballistic flyby. This algorithm also accounts for the  $r_{min}$  and  $r_{SOI}$  constraints on the flyby periapse. The purpose of this algorithm is to effectively drive the heliocentric trajectory optimization towards ballistic gravity assists without needing to incorporate nonlinear ballistic gravity assist constraints. The following flyby option describes the standard nonlinear ballistic gravity assist constraints.

The last gravity assist option is to not have a BVP computed and only use the standard gravity assist constraints incorporated into the primary level optimization nonlinear constraints function. These consist of an equality constraint on the incoming and exiting excess velocities  $V_\infty^-$  and  $V_\infty^+$  magnitudes and an inequality constraint on the available turning angle of the hyperbolic legs,  $\Phi_{av}$  to be greater than the required flyby turning angle  $\Phi_r$ , shown below.

$$\begin{aligned} |V_\infty^-| &= |V_\infty^+| \\ \Phi_r &\leq \Phi_{av} \end{aligned} \tag{3.5}$$

### 3.5 Chapter conclusions

The form of TRACT's architecture and capabilities follows its function as a preliminary interplanetary trajectory design tool. The primary motivation for

TRACT is to make the difficult task of designing interplanetary trajectories more accessible, especially to novice users like undergraduate students. By using a bi-level optimization architecture, TRACT does not require as much trajectory information and initial guess values from the user. The bi-level scheme also allows TRACT's optimization performance to be more stable for rough initial guess values, as is common for novice and expert trajectory designers alike. TRACT also enhances the interactivity between the user and the trajectory optimization process, by providing fast and easy to digest trajectory results in the form of MATLAB's driver script, organized Excel spreadsheets, and MATLAB's adaptable trajectory plotting visualizations/simulations.

TRACT provides many common interplanetary trajectory options in order to accurately model the desired mission trajectory, such as departing/arriving with parking orbits, using launch vehicle trajectories, applying DSMs, and various gravity assist types. New efficient trajectory designs could potentially enable previously unreachable spacecraft mission objectives. In order for users to have all the possible tools available to achieve such mission trajectory goals, the author developed ballistic and powered gravity assist options. These efficient and robust gravity assists models allow users to effectively build the next generation of interplanetary trajectories without the need for advanced trajectory design experience. The next chapter lays out the research and development of these powerful and enabling gravity assist models incorporated in TRACT.

## Chapter 4

# Powered Gravity Assist Boundary Value Problem

### 4.1 Motivation for powered gravity assist

The ultimate goal of a gravity assist is to adjust the exiting  $\vec{V}_\infty$  to achieve a specific goal for the encounter. The encounter may be designed to achieve any of the following: increasing the heliocentric velocity by rotating  $\vec{V}_\infty^+$  in the direction of the planet's velocity, decreasing the heliocentric velocity by rotating  $\vec{V}_\infty^+$  opposite the direction of the planet's velocity, and/or changing the heliocentric orbit inclination by rotating  $\vec{V}_\infty^+$  out of plane with respect to the planet's orbit, Figure 4.1. In all cases, the required outcome for a gravity assist is that the outgoing velocity vector at the planetary encounter location take on the value specified for the next heliocentric trajectory segment. For almost all initial mission scenarios involving gravity assists, the incoming  $V_\infty^-$  and outgoing  $V_\infty^+$  values will not match. Thus, some form of  $\Delta V$  maneuver needs to be incorporated into the flyby.

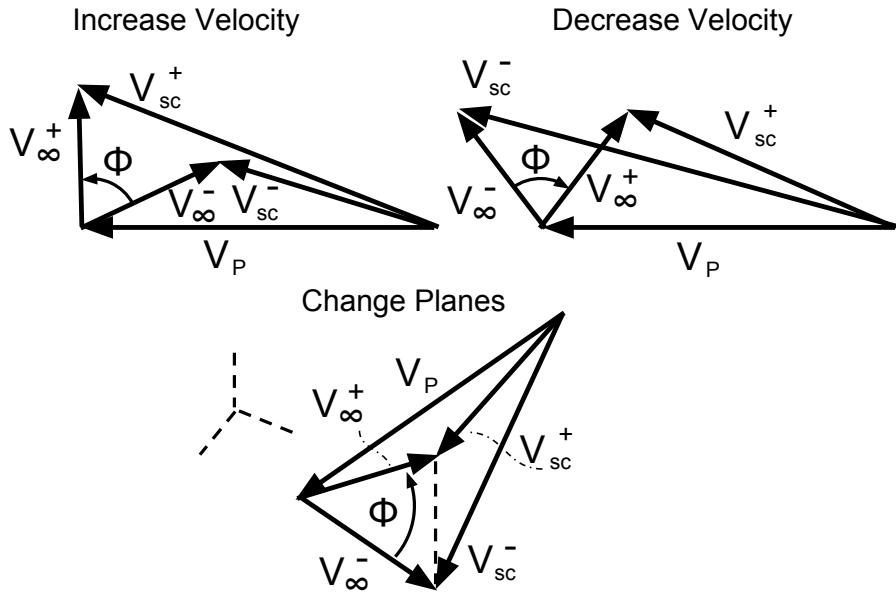


Figure 4.1: Flyby velocity vector diagrams representing the different types of gravity assists

A gravity assist incorporating a  $\Delta V$  maneuver during the flyby encounter is called a powered gravity assist or powered flyby. When on a the hyperbolic flyby trajectory, the  $\Delta V$  can be used to torque the  $\vec{V}_\infty^+$  further towards the intended direction and to increase/decrease the exiting  $V_\infty^+$  magnitude, shown in Figure 4.2. The additional change in exiting excess velocity  $\vec{V}_\infty^+$  due to the  $\Delta V$  maneuver is shown in the diagram as  $\Delta V_\infty$ . The  $\Delta V$  maneuver is not the same as  $\Delta V_\infty$ . A desirable powered flyby configuration would allow a small  $\Delta V$  maneuver produce a larger  $\Delta V_\infty$ . In other words, “buying” more gravity assist performance with a smaller  $\Delta V$  investment. The powered gravity assist dynamics can have a multiplying effect on the overall outcome of the gravity assist. The location of the  $\Delta V$  maneuver can be anywhere along the trajectory between the incoming and exiting asymptotes, at the

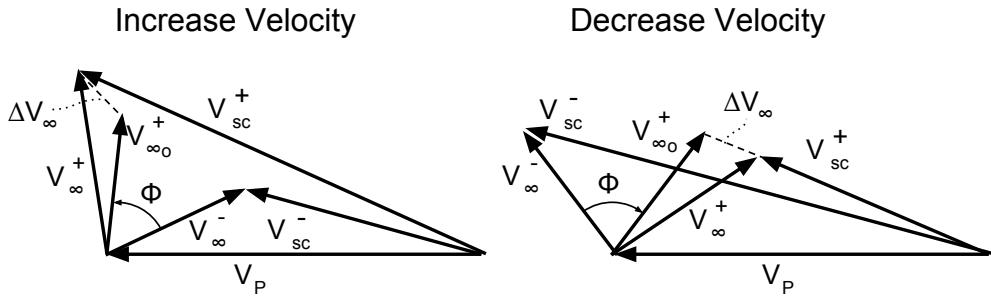


Figure 4.2: Powered flyby velocity vector diagrams showing the improved performance and change in original  $V_{\infty}^+$  to the final  $V_{\infty}^+$  due to the interior  $\Delta V$  maneuver

intersection of the two hyperbolic trajectories. A major component of the powered gravity assist's effectiveness is derived from the well known Oberth effect, where a  $\Delta V$  maneuver changing orbital energy is more efficient at higher velocities. For this reason, many designed powered gravity assist have an optimum maneuver location near perihelion, where the orbital velocity is greatest.

If given a set of flyby conditions about a planet, a hyperbolic BVP can be formulated. It is assumed that the flyby is coplanar, where the plane is defined by the incoming and exiting asymptote vectors  $\vec{V}_{\infty}^-$  and  $\vec{V}_{\infty}^+$  and the center of the encounter planet. The fixed boundary conditions defining the BVP are incoming  $V_{\infty}^-$ , exiting  $V_{\infty}^+$  magnitudes, and the angle  $\Phi_r$  between the asymptotes (as defined from  $\vec{V}_{\infty}^-$  and  $\vec{V}_{\infty}^+$ ). The desired solution of the powered flyby BVP is the minimum  $\Delta V$  and its maneuver location,  $v_{\Delta V}$  or  $r_{\Delta V}$ . In general there are two independent (free) variables for the BVP optimization that define the incoming and exiting trajectories. The independent variables could be any pair of hyperbolic orbit parameters ( $\Delta$ ,  $e$ ,  $r_p$ ,  $\delta$ ,  $v_{\Delta V}$ ,  $r_{\Delta V}$ , etc.) for each side of the flyby transfer trajectory shown in Figure 4.3.

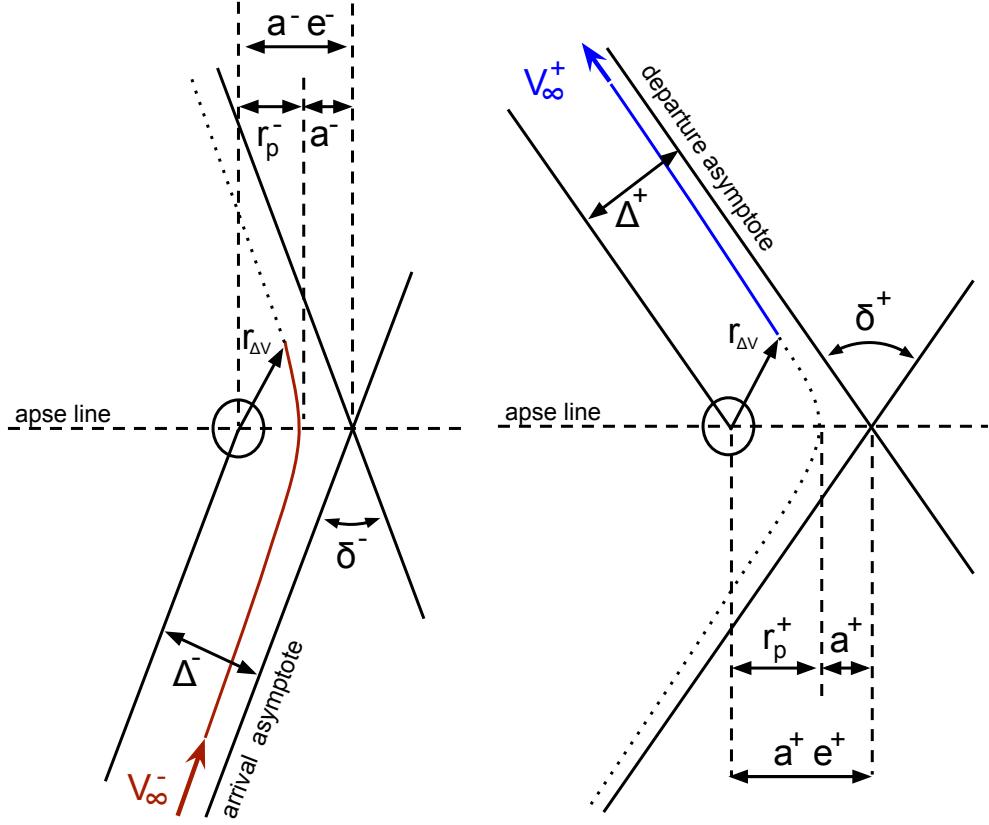


Figure 4.3: Hyperbolic boundary value problem setup for powered flyby with incoming hyperbola on left and outgoing hyperbola on right

The hyperbolic asymptote offset distance  $\Delta$  or turning angle  $\delta$  can be used to define the trajectory leg, directly relating to the hyperbolic legs eccentricity  $e$  and periapse radius  $r_p$  with basic two body relationships provided in Chapter 2. As an example of the complete powered flyby transfer, the diagram in Figure 4.4 shows the asymptote offset distances  $\Delta_-$  and  $\Delta_+$  used to fully define the transfer maneuver location. Later in the chapter, a useful set of hyperbolic variables are the periapse  $r_p$  of one leg and the  $\Delta V$  maneuver location  $r_{\Delta V}$  of the other leg to define the optimal powered gravity assist. The powered flyby diagrams show that a powered gravity

assist is simply two different hyperbolic orbits spliced together at an intersection point [52].

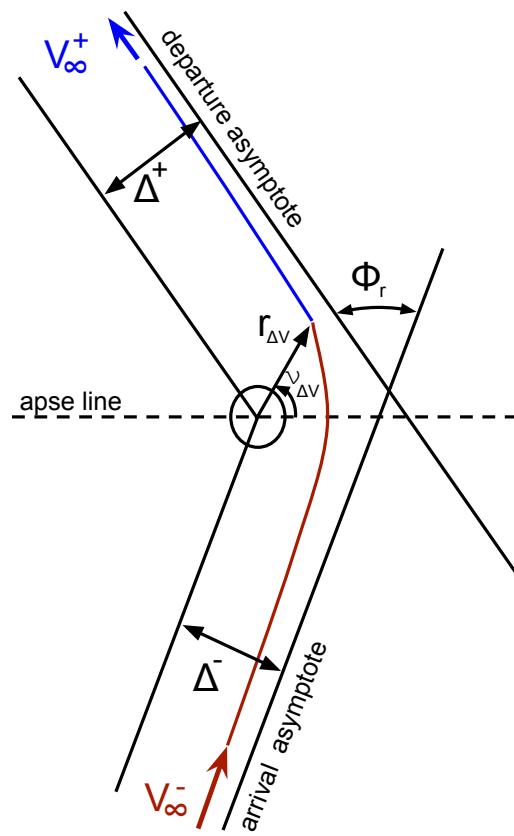


Figure 4.4: General hyperbolic boundary value problem diagram for powered flyby

In a linked conic interplanetary trajectory design scheme, the gravity as-

sist connects the two heliocentric legs at the encounter planet. The flyby BVP is the bridge that completes this connection and provides a quantitative measure of its efficiency,  $\Delta V$ . When optimizing the entire interplanetary mission trajectory, a powered flyby for every planetary encounter should be assumed. However, it is sometimes the case that the  $\Delta V$  maneuver for a powered flyby at a given planet may be reduced to zero in the optimization, yielding an unpowered gravity assist.

## 4.2 Background on types of flyby BVPs

### 4.2.1 Typical powered gravity assist strategy (impulse at periapse)

Most preliminary gravity assist trajectory schemes account for a gravity assist either by applying constraints on the  $\vec{V}_\infty^-$  and  $\vec{V}_\infty^+$  conditions about each encounter planet or by determining the required  $\Delta V$  to complete the planetocentric hyperbolic transfer. In practice, the former method is more useful for designing preliminary mission trajectories because of the reduction in number of constraints and the direct measurement of the gravity assist's efficiency, the  $\Delta V$  maneuver bridging the connection. This is especially true when considering the trial and error approach often used for honing mission/trajecotry requirements.

In order to simplify the computational process, a gravity assist trajectory is often determined by restricting the connection point of the incoming and outgoing hyperbolic legs to be at their common periapse, producing a tangential  $\Delta V$  maneuver along the direction of motion. An iterative procedure is performed to find the required radius for the common periapse for the incoming and outgoing trajectory legs. The discontinuity in the velocity at the converged coincident point is the required  $\Delta \vec{V}$  impulse to complete the prescribed flyby. The left hand plot in

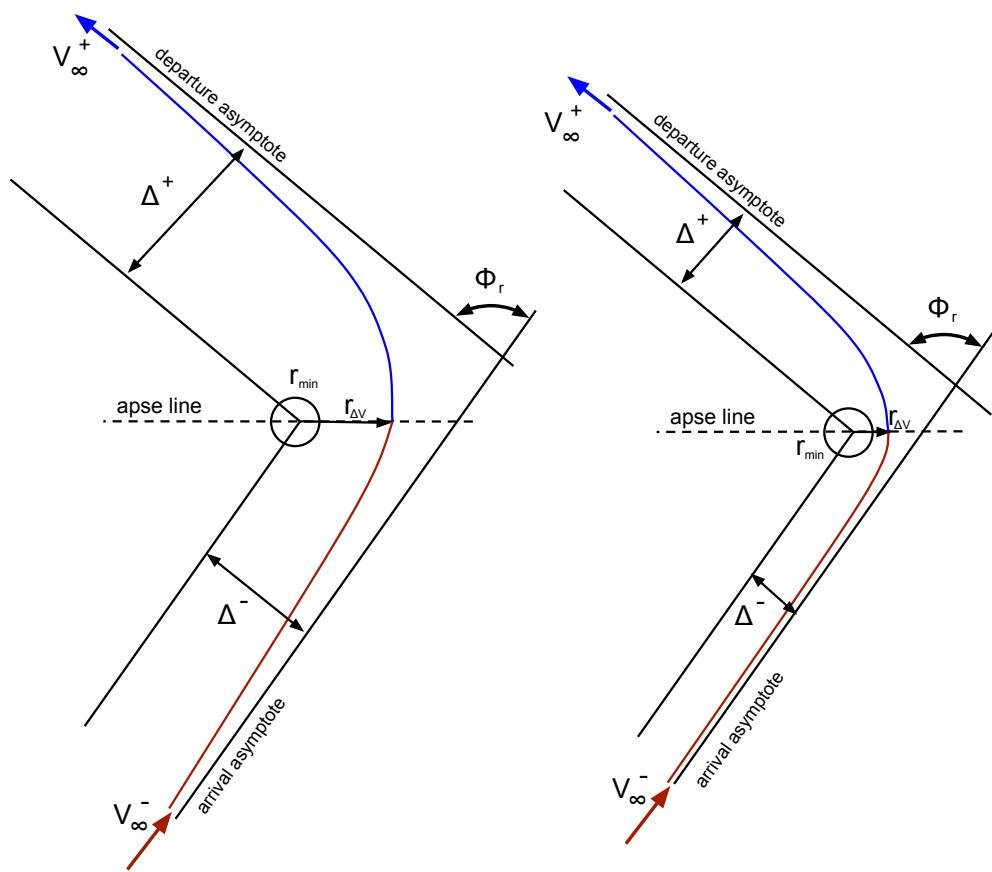


Figure 4.5: Periapsis restricted powered flyby before (left) and after (right) determining common periapsis

Figure 4.5, shows two hyperbolic trajectory legs of the flyby intersecting at a non-periapse point. The right hand plot shows the set of hyperbolic trajectory legs with a converged common periapse. Here, a specialized algorithm has been tailored for reliable optimization performance and fast convergence.

A popular numerical strategy in the literature is to determine the common periapse radius by iterative substitution [42, 31, 33]. The given flyby configuration parameters  $\mu$ ,  $V_\infty^-$ ,  $V_\infty^+$ , and  $\Phi_r$  are used to determine the half turning angle  $\delta/2$  for each leg of the powered gravity assist, summing to the overall effective turning angle in Equation 4.1 as a function of  $r_p$ . The equation can be reformulated into a single expression presented by Horsewood [42], Equation 4.2. Each leg's eccentricity is defined by  $r_p$ , seen from Equation 2.8. Given the required turning angle  $\Phi_r$ , an estimate for  $r_p$  can be iteratively substituted into these equations until converging upon the appropriate  $r_p$ . Horsewood also provides a clever initial estimate of  $r_p$  shown in Equation 4.3, where the  $V_\infty^2$  term of a ballistic hyperbolic orbit is approximated by the product of the given excess velocities  $V_\infty^- V_\infty^+$ . The approximate  $r_p$  can be used to initialize the iterative procedure.

$$\begin{aligned}\Phi_g &= \frac{\delta^-}{2} + \frac{\delta^+}{2} = \arcsin\left(\frac{1}{e_-}\right) + \arcsin\left(\frac{1}{e_+}\right) \\ &= \arcsin\left(\frac{\mu}{\mu+r_p V_\infty^{-2}}\right) + \arcsin\left(\frac{\mu}{\mu+r_p V_\infty^{+2}}\right)\end{aligned}\tag{4.1}$$

$$\cos \Phi_g = \frac{\sqrt{(e_-^2 - 1)(e_+^2 - 1)} - 1}{e_- e_+}\tag{4.2}$$

$$r_p = \frac{\mu}{V_\infty^2} \left( \sqrt{\frac{2}{1 - \cos \Phi_r}} - 1 \right) \approx \frac{\mu}{V_\infty^- V_\infty^+} \left( \sqrt{\frac{2}{1 - \cos \Phi_r}} - 1 \right)\tag{4.3}$$

A more efficient and reliable root-finding algorithm is Newton's method, which uses the derivative information to improve the iterations, as presented by Ceriotti [53]. In order to find the required  $r_p$  for both hyperbolic legs to share a common periapse, the function to be searched is formulated as the cosine difference in required turning angle  $\Phi_r$  and the available turning angle  $\Phi_g$  of the current  $r_p$  estimate, Equation 4.4. The function derivative with respect to  $r_p$  is provided in Equation 4.5, followed by the defined constant  $C_p$ .

$$\begin{aligned} f(r_p) &= \cos \Phi_g - \cos \Phi_r \\ &= \frac{\sqrt{(e_-^2 - 1)(e_+^2 - 1)} - 1}{e_- - e_+} - \cos \Phi_r \end{aligned} \quad (4.4)$$

$$\begin{aligned} \frac{df}{dr_p} &= [r_p^3 (V_\infty^{-6} V_\infty^{+2} + V_\infty^{-2} V_\infty^{+6}) + 3\mu r_p^2 V_\infty^{-2} V_\infty^{+2} (V_\infty^{-2} + V_\infty^{+2}) + \\ &\quad 2\mu^2 r_p V_\infty^{-4} V_\infty^{+2} (C_p + 2) + \mu^3 C_p (V_\infty^{-2} + V_\infty^{+2})] / [\mu^4 C_p e_-^2 e_+^2] \end{aligned} \quad (4.5)$$

$$C_p = \sqrt{r_p^2 V_\infty^{-2} V_\infty^{+2} (r_p V_\infty^{-2} + 2\mu) (r_p V_\infty^{+2} + 2\mu) / \mu^4}$$

The root-finding algorithm updates an  $r_p$  estimate using the function and derivative information as seen below. The iterative method performs well when initialized by the  $r_p$  estimate above, Equation 4.3. The algorithm can easily be modified to check and maintain the  $r_{min}$  and  $r_{SOI}$  constraints. Although, it will become obvious that when the  $r_{min}$  constraint is active only one of the hyperbolic leg's periapse can be held at  $r_{min}$ , while the other leg's periapse can potentially fall below  $r_{min}$ . A similar scenario can occur when the  $r_{SOI}$  constraint is active.

$$r_p^{(i+1)} = r_p^{(i)} - \frac{f(r_p^{(i)})}{\left.\frac{df}{dr_p}\right|_{r_p^{(i)}}} \quad (4.6)$$

Though the periapse impulse method for incorporating gravity assists is a relatively easy algorithm to set up, the resulting  $\Delta V$  performance of the flyby is quite sensitive to the heliocentric configuration changes when optimizing the entire interplanetary trajectory. Consequently, the periapse impulse method requires a good initial guess of the overall mission configuration for the trajectory to converge on a desirable solution – one with minimal total  $\Delta V$  and minimal (if not zero)  $\Delta V$  for each gravity assist. A better representation of the  $\Delta V$  performance for a gravity assist can be found in the following sections, that does not restrict the impulse to be only at periapse. The literature on powered gravity assists recognizes that the best  $\Delta V$  performance may occur at a maneuver location other than periapse. The current methods in literature for determining the optimal non-periapse powered flyby  $\Delta V$  maneuvers are found to be computationally inefficient and cumbersome. Also, some of the current methods found in literature do not span the entire flyby solution space, resulting in sub-optimal local minimum  $\Delta V$  maneuvers, while a better global minimum  $\Delta V$  maneuver exists elsewhere in the solution space. Yet, some of the globally optimal powered gravity assist solutions in literature result in non-realizable trajectories for spacecraft missions. The goal of the new powered gravity assist algorithms developed in Section 4.3 is to provide an efficient and robust method for determining the global optimal powered flyby  $\Delta V$  maneuver point, while maintaining a mission feasible flyby trajectory.

#### 4.2.2 Literature review: optimal multiple impulse flyby problem

The powered flyby problem has been well studied in classic literature as transfers between two hyperbolas or hyperbolic asymptotes. The prior research in this area has focused on determining the globally optimum powered gravity assist as an academic exercise, in which some optimal solutions are impossible to fly on a realistic mission. In this section, it will be shown that many studies focus on producing globally optimal (minimum total  $\Delta V$ ) gravity assist trajectories with characteristics that make them infeasible for a realistic spacecraft mission. These mission infeasible characteristics include allowing the flyby periapse to fall below the minimum allowable radius  $r_{min}$  (such as the radius of the planet), having multiple  $\Delta V$  maneuvers during the relatively short hyperbolic encounter, and requiring an excessive amount of time to complete the flyby. Also, allowing the flyby periapse to be excessively far from the planet, beyond  $r_{SOI}$ , is infeasible with respect to the linked conics assumption, which will be discussed in detail later. It will then be shown that formulating a consistent powered gravity assist method using only a single  $\Delta V$  maneuver provides an effective mission feasible strategy. This strategy also allows for the development of a comprehensive powered gravity assist solution algorithm.

In one study by Edelbaum, the overall optimum transfer between two hyperbolas has been solved generally for the case where the radius of the gravitational body is assumed to be zero. Thus the impulses are performed at the center of attraction [54], and are of negligible, infinitesimal  $\Delta V$  magnitude. The resulting “optimal” powered flyby requires four infinitesimal impulses for a bi-parabolic transfer solution. The flyby’s incoming leg performs the first impulse at the center

of attraction and reduces the trajectory to parabolic energy. Then as the resulting parabolic leg exits at an infinite distance, a second impulse closes the orbit, allowing the trajectory to align for another encounter. Once the trajectory is aligned, an impulse inserts into an incoming parabolic trajectory. The final impulse at the center of attraction creates the required exiting hyperbolic trajectory [55]. Though this bi-parabolic transfer is globally optimal, it requires infinite time and an unrealistic point-mass gravitational body [54]. The next studies provide a finite minimum radius  $r_{min}$  constraint to improve the realistic nature of the transfers.

Gobetz presents an initial study for powered gravity assists with a constraint on the closest approach distance and numerical results of applying various methods for the transfer [7]. Again, the flyby problem is defined as a hyperbolic BVP, where the transfer trajectory is between two hyperbolic asymptotes, and the asymptote offset distances  $\Delta^\pm$  are free. The paper considers three different types of transfer solutions: an optimum (unconstrained periapse) one-impulse flyby, a  $r_{min}$  constrained optimum one-impulse flyby, and a four-impulse transfer. An iterative numerical procedure was used to solve each type of transfer and compare the solutions for the minimum  $\Delta V$ . For the one-impulse solutions, the two independent optimization variables used were the half turning-angles  $(\psi^\pm = \frac{\delta^\pm}{2})$  for each leg of the flyby. The four-impulse solution is similar to Edelbaum's bi-parabolic transfer previously described, although Gobetz assumes the entire transfer remains within the SOI (sphere of influence) of the planet encountered, shown in Figure 4.6. The  $\Delta V$  required for the four-impulse bi-parabolic transfer was fully determined by the minimum periapse radius and is solved analytically [7].

It is this paper that first suggests a common characteristic in the optimal

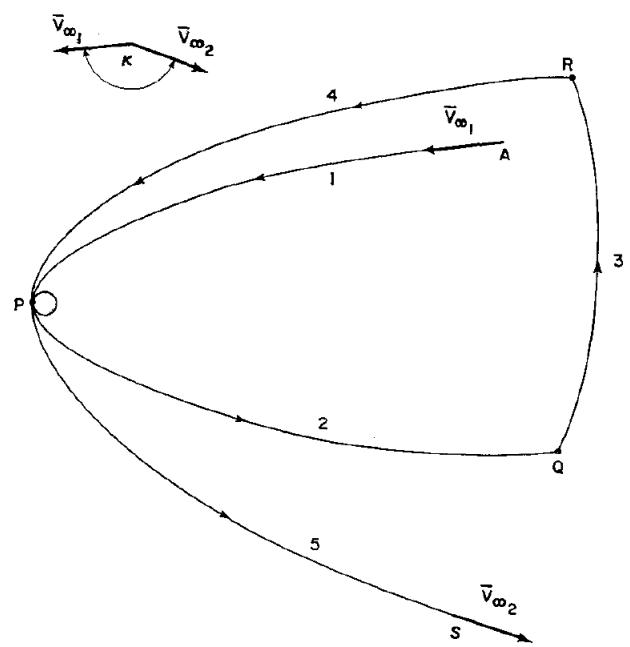


Figure 4.6: Four-impulse bi-parabolic transfer between hyperbolic asymptotes  
[Source: Gobetz [7]]

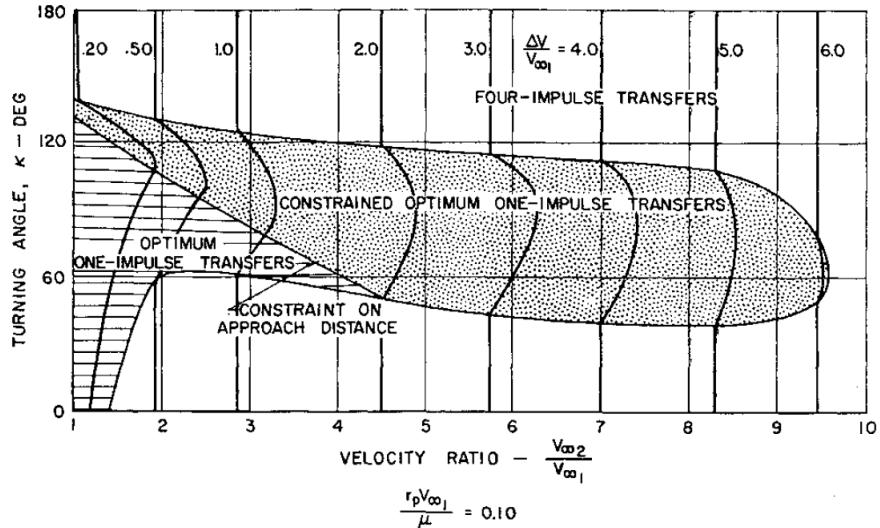


Figure 4.7: Diagram of resulting optimal solution cases under varying flyby conditions with a minimum radius constraint [Source: Gobetz [7]]

single impulse solutions; if the optimal transfer  $\Delta V$  maneuver occurs before the incoming leg reaches periapse, then the exiting leg will always begin before its own periapse and traverse through it. The opposite is true; if the optimal transfer  $\Delta V$  maneuver occurs after the incoming leg has traversed its periapse, then the outgoing leg will begin after its periapse. Additionally, the paper recognized that the flyby BVP solutions are reversible. This insight allows the flybys to be categorized and the solution process improved, as seen in the next papers and later in this dissertation (Section 4.3). The paper assumes that the one-impulse iterative procedure initializes with the  $\Delta V$  maneuver at periapse (as an initial guess to the solution location). Gobetz claims that the optimal single-impulse powered gravity assist maneuver location is always close to periapse, but research presented later in this dissertation proves otherwise for the general powered flyby BVP.

Gobetz provides a helpful diagram of the optimal solution cases in Figure 4.7, showing that under certain conditions the four-impulse bi-parabolic transfer gives the minimum required  $\Delta V$ . The four-impulse transfer is seen to be optimal when turning angles are relatively large or relatively small for very different incoming and exiting  $V_\infty^\pm$  magnitudes  $\left(\frac{V_{\infty 2}}{V_{\infty 1}} \gg 1\right)$ , as seen in the diagram. By definition, the periapse of the minimum  $\Delta V$  one-impulse solutions are found at some optimal distance above the  $r_{min}$  constraint. When the flyby BVP conditions become more extreme, the optimal one-impulse periapse drops below the  $r_{min}$  constraint, therefore requiring the constrained optimum one-impulse solution or the bi-parabolic solution [7].

The optimal solution methods were later expanded by the works of Walton, Marchal, and Culp to include a more definitive list of the global optimum impulsive powered gravity assist solution types for the general BVP minimizing total  $\Delta V$ , [2, 3, 4]. The first stage of categorizing the solution types requires a reference flyby trajectory related to the minimum radius constraint. The minimum radius  $r_{min}$  constraint directly limits the maximum turning angle able to be achieved by a ballistic hyperbolic trajectory. This angle is defined as  $\Phi_p$ . The maximum turning angle  $\Phi_p$  powered gravity assist is a periapse maneuver flyby with a minimum periapse radius  $r_{min}$ , as depicted in Figure 4.8. The incoming and outgoing hyperbolic legs are met at their common  $r_{min}$  periapse, where each leg contributes half of its maximum hyperbolic turning angle  $\delta^\pm$ , Equation 4.7. Since each leg meets at the common periapse, the connecting  $\Delta V$  maneuver is collinear along the velocity direction at the common periapse.

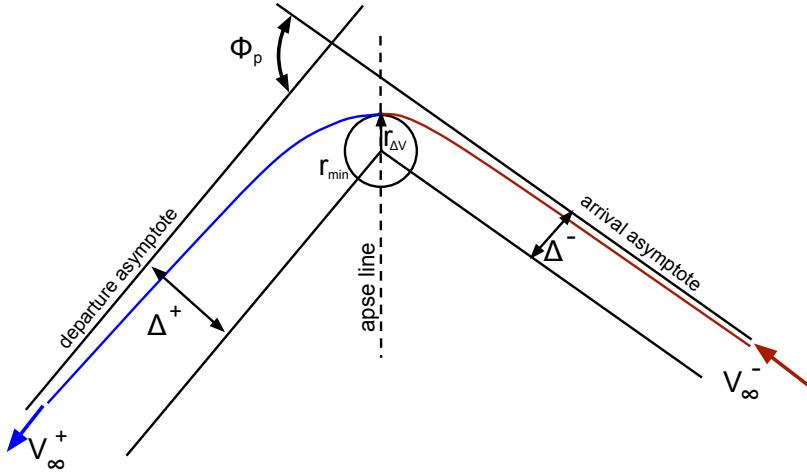


Figure 4.8: The maximum natural turning angle  $\Phi_p$  powered flyby reference trajectory used to categorize all flybys

$$\begin{aligned}
 \Phi_p &= \frac{\delta_{max}^-}{2} + \frac{\delta_{max}^+}{2} \\
 &= \arcsin\left(\frac{\mu}{\mu+r_{min}V_\infty^{-2}}\right) + \arcsin\left(\frac{\mu}{\mu+r_{min}V_\infty^{+2}}\right)
 \end{aligned} \tag{4.7}$$

This study by Walton, Marchal, and Culp found that comparing possible flyby BVP configurations to this reference trajectory, the problem is divided neatly into two classes . The first class is defined by the required turning angle being less than the natural maximum,  $\Phi_r < \Phi_p$ , wherein the flyby periapse is likely to be unconstrained by the minimum radius and the unpowered ballistic flyby may exist. The second class is then defined by  $\Phi_r > \Phi_p$ , where it is guaranteed that the flyby is constrained by the minimum radius, therefore the periapse is fixed as  $r_{min}$  [2, 3, 4]. Example powered gravity assist trajectories of these two classes is shown in Figure 4.9.

The studies by Walton, Marchal, and Culp first focused on developing the

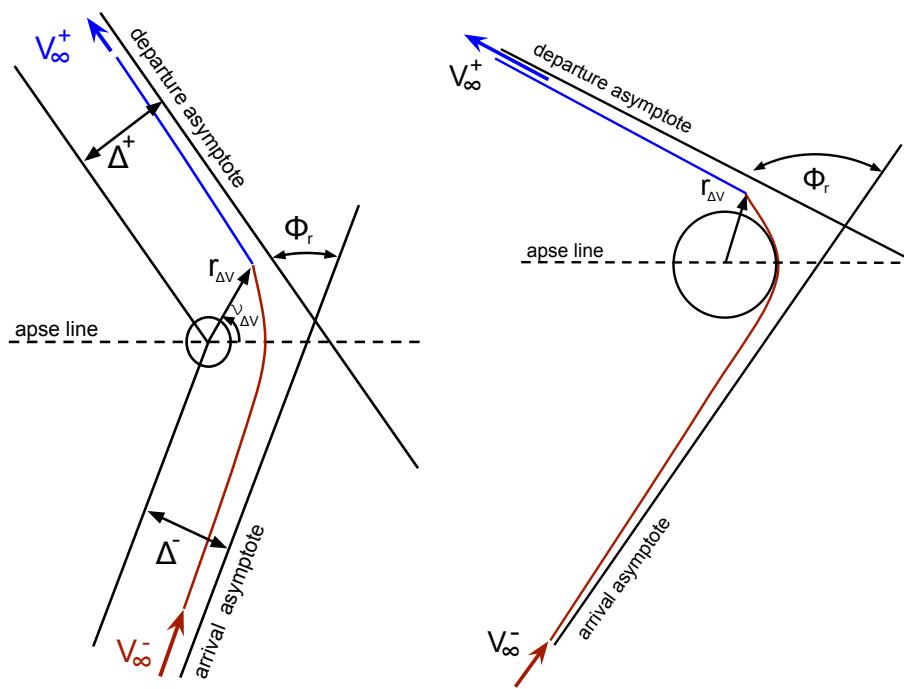


Figure 4.9: The two classes of powered gravity assist trajectories are split into the unconstrained peripase flyby (left) and constrained peripase flyby (right)

| Type             | Powered flyby transfer description   |
|------------------|--|
| Type F           | Grazing $\Delta V$ at a finite distance above $r_{min}$                                |
| Type $F_\infty$  | Type F flyby along with a $\Delta V$ at outgoing $r_\infty$ asymptote                  |
| Type RF          | Graze $r_{min}$ at periapse, then $\Delta V$ at finite distance above $r_{min}$        |
| Type $RF_\infty$ | Type RF flyby along with a $\Delta V$ at outgoing $r_\infty$ asymptote                 |
| Type PNP         | Bi-parabolic transfer with 2 $\Delta V$ at periapse and 2 $\Delta V$ at $r_\infty^\pm$ |

Table 4.1: Optimal powered flyby BVP solutions for the case of  $\Phi_r < \Phi_p$  and  $V_\infty^- > V_\infty^+$  [2, 3, 4]

set of optimal solution types for the case where the required turning angle is less than the maximum available ballistic turning angle  $\Phi_r < \Phi_p$ . These sets of gravity assists were thought to be more likely implemented in a real spacecraft mission as opposed to the set of flybys with the periapse constrained to be at the minimum radius [2, 3, 4]. Though this suggestion might be true for more mature mission trajectory designs, flyby BVP configurations producing both classes are frequently encountered in the trajectory planning and optimization process before reaching the final trajectory design.

The set of five solution types developed by Walton, Marchal, and Culp for the  $\Phi_r < \Phi_p$  flyby configurations are shown in the following Table 4.1 for the case of  $V_\infty^- > V_\infty^+$ . The alternate case of  $V_\infty^- < V_\infty^+$  have similar solution types simply in reversed order, as identified earlier. Each of these flyby types, besides the bi-parabolic (Type PNP), are adjusting the hyperbolic flyby trajectory with a finite  $\Delta V$  maneuver at a location within the vicinity (the sphere of influence) of the planet and a possible  $\Delta V$  maneuver at the exiting asymptote to adjust the  $\vec{V}_\infty^+$  directly. The Gobetz study just considered the Type F case when just a single maneuver is used, the Type RF case where the flyby periapse is constrained to be  $r_{min}$  and the single

maneuver is executed after periapse somewhere along the exiting leg, and the Type PNP bi-parabolic transfer. The additional maneuver cases determined by Walton, Marchal, and Culp include impulses at  $r_\infty$ . The  $\Delta V$  impulse applied directly to  $V_\infty^+$  or  $V_\infty^-$  allows for a better placement of the periapse and is more effective at adjusting the asymptote direction due to the lower velocity magnitude at  $r_\infty$ . The identifier “F” in the flyby types indicates a maneuver at a “Finite” distance above  $r_{min}$ . The identifier “R” in the flyby types indicates that the periapse is restricted to graze the  $r_{min}$  sphere, therefore  $r_p = r_{min}$ . Finally, the identifier “ $\infty$ ” indicates that a second maneuver is located at  $r_\infty$ , applying the  $\Delta V$  directly to the asymptote excess velocity  $V_\infty$ .

Of all the potential solution types, the single impulse flybys accounted for the majority of the solution space, followed by the bi-parabolic (type PNP) transfer [2], which was identified previously by Gobetz [7]. The small transition areas between solution types contain the two-impulse cases. The solution space diagrams can be found in the related papers [2, 3, 4]. The additional  $\Delta V$  impulse at the  $r_\infty$  asymptote typically provides a reduction in turning angle in order to match the asymptote conditions, while a majority of the required velocity magnitude adjustments were completed by the more energy efficient impulse closer to the body.

The remaining set of periapse constrained powered flyby classifications for those cases where the required turning angle is greater than the maximum available ballistic turning angle  $\Phi_r > \Phi_p$  were later developed in Walton’s dissertation [3] and proceeding papers [4]. The periapse constrained powered flybys were categorized into six different types, similar to the previous set, as seen in Table 4.2. The most common type of solution in this set is the four-impulse bi-parabolic transfer (type

| Type                     | Powered flyby transfer description   |
|--------------------------|--|
| Type R $\infty$          | Graze $r_{min}$ at periapse, then $\Delta V$ at outgoing $r_\infty$ asymptote          |
| Type $\infty$ R $\infty$ | Type R $\infty$ flyby with additional $\Delta V$ at incoming $r_\infty$ asymptote      |
| Type RF                  | Graze $r_{min}$ at periapse, then $\Delta V$ at finite distance above $r_{min}$        |
| Type RF $\infty$         | Type RF flyby along with a $\Delta V$ at outgoing $r_\infty$ asymptote                 |
| Type $\infty$ RF         | Type RF flyby along with a $\Delta V$ at incoming $r_\infty$ asymptote                 |
| Type PNP                 | Bi-parabolic transfer with 2 $\Delta V$ at periapse and 2 $\Delta V$ at $r_\infty^\pm$ |

Table 4.2: Optimal powered flyby BVP solutions for the case of  $\Phi_r > \Phi_p$  and  $V_\infty^- < V_\infty^+$  [3, 4]

PNP), which accommodates the most versatile turning angles between incoming and outgoing asymptotes. The second most common transfer type is the single-impulse flyby. There is an increased role of the two-impulse flybys. Since this set of flyby transfers is defined by requiring excessive turning angles, there is a greater need for additional trajectory torquing, which can frequently be more effective with a  $\Delta V$  impulse at the asymptote.

All of the preceding solutions were developed by Walton, Marchal, and Culp through hodograph analysis and applying Lawden's primer vector theory[56], verifying the solutions types as primer optimal. Each flyby transfer type was determined analytically when possible or using a simple iterative procedure otherwise. Though diagrams of the solution space were given in the literature depicting the flyby configurations for each optimal flyby transfer type, there wasn't a simple and reliable method for selecting the appropriate solution at the onset of the problem. Therefore, to ensure the overall optimum flyby transfer type is selected, all of the potential types of the flyby set must be computed and compared for minimum  $\Delta V$ . This inefficiency is only one of many issues in implementing optimal powered gravity assists in preliminary mission trajectory design. Though the bi-parabolic transfer

was found to be one of the more common solutions, most missions implementing gravity assist trajectories are interested in reducing the overall time of flight as well as the total  $\Delta V$ . The excessive transfer time of the bi-parabolic gravity assist not only pushes against the mission goals, but also violates the foundational assumption of linked-conics, which assumes a relatively short planetocentric transfer time.

The multiple impulse flybys are also of concern, because in a real mission each engine burn would require post maneuver analysis to reevaluate the next engine burn and potentially add clean-up maneuvers in between. The time required to accomplish such tasks while within a planet's sphere of influence is prohibitive. More to the point, the additional maneuvers at the asymptotes are acting as DSMs to correct an undesirable configuration of  $\vec{V}_\infty^-$  and/or  $\vec{V}_\infty^+$ , which would likely be more effectively altered by a heliocentric DSMs well before the planetary encounter, such as a  $V_\infty$  leveraging maneuver [26]. Therefore these second maneuvers at  $r_\infty$  should be omitted from the flyby and a DSM included in the heliocentric trajectory. Many times these DSMs will be driven to a negligible  $\Delta V$  magnitude anyway [28, 27].

Considering these limitations an efficient and robust algorithm is needed to fill the void in developing optimal powered gravity assist trajectories for preliminary mission trajectory design. The next section details the development of such an algorithm for producing single impulse powered gravity assists that extends and clarifies the foundational research above.

## 4.3 Optimal single impulse flyby BVP

The single impulse gravity assist configuration was selected as the basis for creating a mission feasible optimal powered gravity assist solution method. By limiting the number of  $\Delta V$  maneuvers to just a single impulse during a flyby, the resulting gravity assist trajectory will remain hyperbolic throughout the encounter. This ensures that the planetocentric flyby encounter occurs in a relatively short timeframe as required by the linked conics assumptions. Optimal multiple-impulse flyby trajectories, as described in the prior section, can potentially fall below a hyperbolic energy during the flyby encounter and require an excessive planetocentric flight time, breaking the linked conics assumptions. Even the case of using multiple maneuvers during an entirely hyperbolic flyby encounter, such as the Type  $F_\infty$ , Type  $RF_\infty$ , etc., are infeasible. This is because a realistic mission would not space multiple maneuvers in such a short timeframe inherent of a flyby. Therefore only a single impulse is allowed during each planetocentric encounter to maintain mission feasibility.

### 4.3.1 Assumptions

The single-impulse powered flyby BVP is rigorously investigated in the following development, where an emphasis is placed on mission feasibility and computational efficiency. The minimum radius  $r_{min}$  constraint is applied to both incoming and outgoing legs of the hyperbolic transfer. An explicit maximum radius constraint is also placed on the flyby, where the impulsive maneuver is only considered within the sphere of influence of the planet,  $r_{SOI}$ . Each leg of the flyby is assumed to be in the hyperbolic energy regime,  $e > 1$ . This section combines ex-

isting methods with new algorithms to develop a robust and efficient technique for solving the optimal single impulse flyby BVP.

#### 4.3.2 Optimal powered gravity assist $\Delta V$ maneuver solution neighborhoods

The first step in efficiently solving the optimal gravity assist problem is to sort the potential solutions into specialized cases. This is accomplished using one of the most important and useful conclusions found from the literature: “A general rule holds for the less than optimal deviation turn angle transfers. An impulse before periapse invariably increases velocity, while an impulse after periapse invariably decreases velocity. The contrary is true for the greater than optimal deviation turn angle transfers” [4]. This result means that the solution space can be split into two categories based on whether the required turning angle  $\Phi_r$  is greater or less than the maximum available ballistic turning angle  $\Phi_p$  (with a periapse at  $r_{min}$ ). Then, each category can be further split into sub-categories depending on whether the  $\Delta V$  occurs before or after passage of periapse (this sub-categorization could also be made alternatively on the basis of comparison of the incoming  $V_\infty^-$  and outgoing  $V_\infty^+$ ), Equation 4.8. The flyby configurations with  $\Phi_r < \Phi_p$  pertain to the Type F flyby, whose periapse is not necessarily constrained by  $r_{min}$  (the majority of cases having periapse above  $r_{min}$ ). The flyby configurations with  $\Phi_r > \Phi_p$  pertain to the Type RF or Type FR flyby, whose periapse guaranteed to be constrained by the minimum radius, therefore  $r_p = r_{min}$ .

$$\begin{aligned} \Phi_r < \Phi_p & \quad \left\{ \begin{array}{l} \text{Type F: } V_{\infty}^- < V_{\infty}^+ \Rightarrow \Delta V \text{ before } r_p \\ \text{Type F: } V_{\infty}^- > V_{\infty}^+ \Rightarrow \Delta V \text{ after } r_p \end{array} \right. \\ \Phi_r > \Phi_p & \quad \left\{ \begin{array}{l} \text{Type RF: } V_{\infty}^- < V_{\infty}^+ \Rightarrow \Delta V \text{ after } r_p \\ \text{Type FR: } V_{\infty}^- > V_{\infty}^+ \Rightarrow \Delta V \text{ before } r_p \end{array} \right. \end{aligned} \quad (4.8)$$

The turning angle comparison  $\Phi_r \leq \Phi_p$  identifies if the flyby BVP is a periapse constrained case (Type RF or FR) or a potentially unconstrained periapse case (Type F). The  $\Phi_r < \Phi_p$  condition is considered to be a “potentially” unconstrained periapse case, because not all of the  $\Phi_r < \Phi_p$  optimal powered gravity assist solutions result in an unconstrained flyby. Although much of the  $\Phi_r < \Phi_p$  solution space results in Type F unconstrained periapse flybys, there exist special cases where the periapse must be restricted by the  $r_{min}$  constraint or the  $r_{SOI}$  constraint. These special cases are discussed in detail in the proceeding sections.

The excess velocity comparison  $V_{\infty}^- \leq V_{\infty}^+$  identifies which leg will contain the periapse and for constrained cases it distinguishes it as either Type RF or Type FR. Examples of these potential trajectory solution configurations are seen in Figures 4.10 & 4.11. These results are hinted at in other literature as well [57][58] [59], but were never well characterize nor fully expanded into the rigorous analysis of powered gravity assist.

Subtypes of these groupings have been developed by the author and will be found useful in later analysis. The subtypes are identified by the individual incoming and outgoing leg’s maximum hyperbolic turning angle ( $\delta_{max}^-$  and  $\delta_{max}^+$ ) as compared with the required powered flyby turning angle  $\Phi_r$ . The first sorting level uses  $\Phi_p$ , having contributions of  $\delta_{max}^-/2$  and  $\delta_{max}^+/2$ , but the portion of  $\Phi_p$  from each leg depends on the relative magnitudes of  $V_{\infty}^{\pm}$ . Both unconstrained periapse

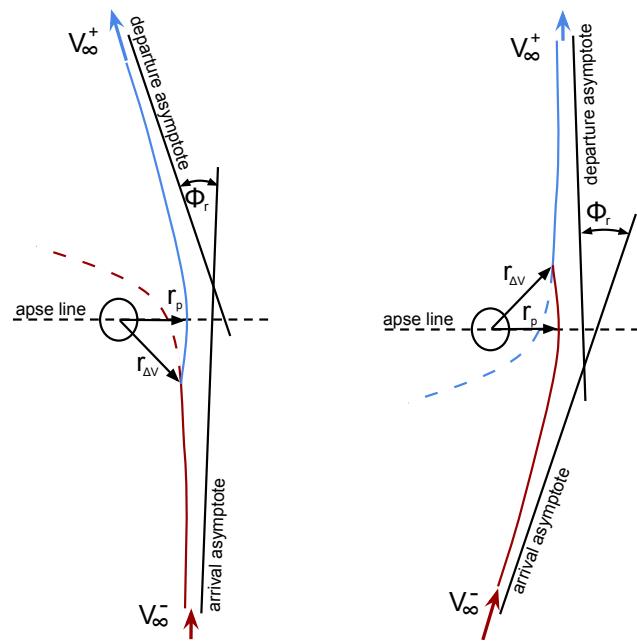


Figure 4.10: Single impulse powered flyby example solutions for unconstrained periapse (type F) cases of  $\Phi_r < \Phi_p$  with  $V_\infty^- < V_\infty^+$  (left) and  $V_\infty^- > V_\infty^+$  (right)

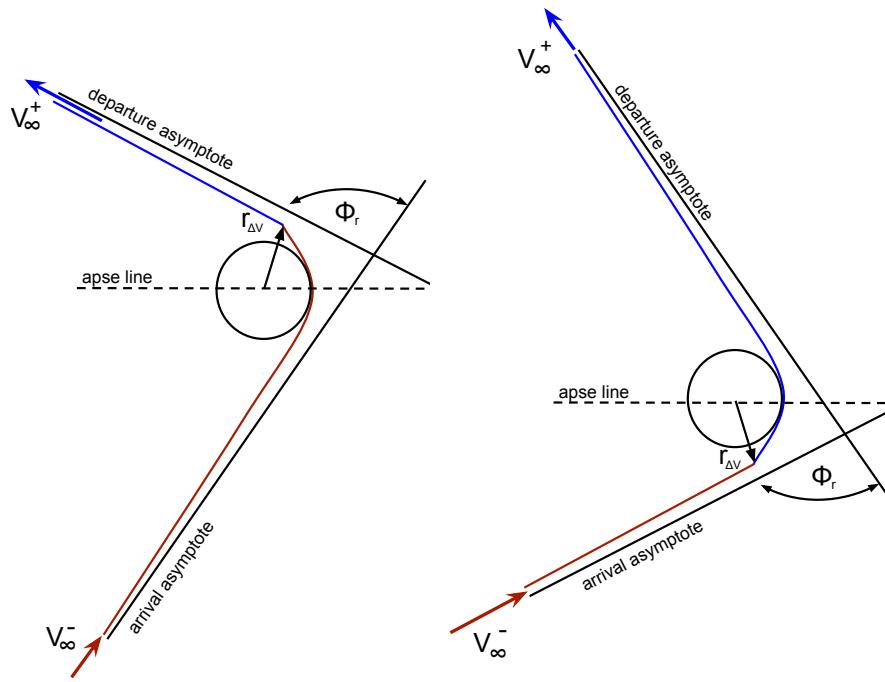


Figure 4.11: Single impulse powered flyby example solutions for  $r_{min}$  constrained periape cases of  $\Phi_r > \Phi_p$  with  $V_\infty^- < V_\infty^+$  (type RF: left) and  $V_\infty^- > V_\infty^+$  (type FR: right)

and constrained periapse groups have 3 subtypes hereafter known as Type 1, Type 2, and Type T (indicating the “Transition” between Type 1 and Type 2). The unconstrained periapse set is shown by Equation 4.9, while the constrained set is shown by Equation 4.10.

$$\Phi_r < \Phi_p : \begin{cases} V_{\infty}^- < V_{\infty}^+ & \begin{cases} \text{Type 1 : } (\Phi_r < \delta_{max}^-) & \text{and } (\Phi_r < \delta_{max}^+) \\ \text{Type T : } (\Phi_r < \delta_{max}^-) & \text{and } (\Phi_r = \delta_{max}^+) \\ \text{Type 2 : } (\Phi_r < \delta_{max}^-) & \text{and } (\Phi_r > \delta_{max}^+) \end{cases} \\ V_{\infty}^- > V_{\infty}^+ & \begin{cases} \text{Type 1 : } (\Phi_r < \delta_{max}^-) & \text{and } (\Phi_r < \delta_{max}^+) \\ \text{Type T : } (\Phi_r = \delta_{max}^-) & \text{and } (\Phi_r < \delta_{max}^+) \\ \text{Type 2 : } (\Phi_r > \delta_{max}^-) & \text{and } (\Phi_r < \delta_{max}^+) \end{cases} \end{cases} \quad (4.9)$$

$$\Phi_r > \Phi_p : \begin{cases} V_{\infty}^- < V_{\infty}^+ & \begin{cases} \text{Type 1 : } (\Phi_r > \delta_{max}^-) & \text{and } (\Phi_r > \delta_{max}^+) \\ \text{Type T : } (\Phi_r = \delta_{max}^-) & \text{and } (\Phi_r > \delta_{max}^+) \\ \text{Type 2 : } (\Phi_r < \delta_{max}^-) & \text{and } (\Phi_r > \delta_{max}^+) \end{cases} \\ V_{\infty}^- > V_{\infty}^+ & \begin{cases} \text{Type 1 : } (\Phi_r > \delta_{max}^-) & \text{and } (\Phi_r > \delta_{max}^+) \\ \text{Type T : } (\Phi_r > \delta_{max}^-) & \text{and } (\Phi_r = \delta_{max}^+) \\ \text{Type 2 : } (\Phi_r > \delta_{max}^-) & \text{and } (\Phi_r < \delta_{max}^+) \end{cases} \end{cases} \quad (4.10)$$

The Type 1 flybys have both hyperbolic leg’s  $\delta_{max}^{\pm}$  comparing similarly with  $\Phi_r$ , both  $> \Phi_r$  or both  $< \Phi_r$ . In contrast, Type 2 flybys have each  $\delta_{max}^{\pm}$  comparing differently with  $\Phi_r$ , where one  $\delta_{max}^{\pm} > \Phi_r$  and the other  $\delta_{max}^{\mp} < \Phi_r$ . The transition between these cases is defined as Type T, where the limiting hyperbolic leg’s turning angle is equal to the required turning angle,  $\delta_{max}^{\pm} = \Phi_r$ . The details and impact of these flyby subtypes are inspected in later sections.

### 4.3.3 Analytic solution for unconstrained periapse powered gravity assist

With the powered gravity assist BVP configuration types sorted, specialized solution methods are developed to solve the single-impulse flyby for the optimal minimum  $\Delta V$  maneuver location. The location of the  $\Delta V$  maneuver will be defined either by its radius or true anomaly interchangeably. These variables are monotonic under the reduced solution space (divided at periapse) provided by the flyby BVP configuration sorting characteristics described above. The optimal powered gravity assist cases unconstrained by  $r_{min}$  with  $r_p > r_{min}$  is identified in the previous section as Type F. The Type F flybys have an exact analytical solution developed by Walton, Marchal, and Culp [2, 3, 4]. This analytic solution is adopted into the optimal single-impulse powered gravity assist algorithm and reformulated, where the primary focus is determining the  $\Delta V$  magnitude of the maneuver and its location as  $r_{\Delta V}$  or  $v_{\Delta V}$ . The analytic Type F flyby BVP solution algorithm is a fast and accurate method to determine the optimal powered gravity assist maneuver location without any consideration for constraints on the flyby trajectory, including the flyby periapse. This analytic algorithm is used as a tool to quickly compute the Type F unconstrained periapse solutions and then check that solution to determine if its periapse breaches any constraints such as the  $r_{min}$  or  $r_{SOI}$  constraints. Handling these constraint conditions is discussed in the next sections.

The type F flyby algorithm was defined by the angles F and G as depicted in the powered gravity assist diagram, Figure 4.12. The angles F and G measure the angular position of the optimal  $\Delta V$  maneuver from each respective asymptote, as depicted in the diagram. The optimal maneuver location  $r_{\Delta V}$  for the minimum  $\Delta V$  was found by variational analysis and constructing an optimal powered flyby hod-

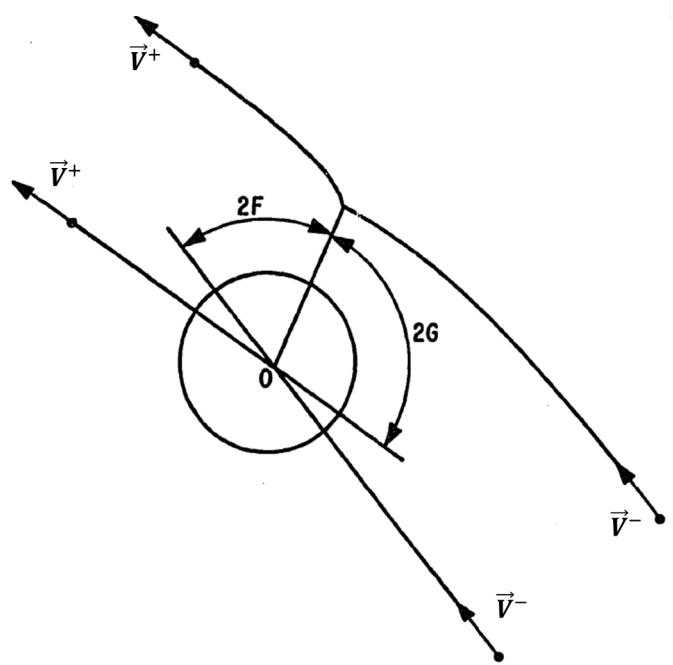


Figure 4.12: Type F powered gravity assist formulation of transfer maneuver location for analytic BVP solution [Source: Walton [2, 3, 4]]

graph diagram. Details on Walton's hodograph diagram and solution development can be found in his dissertation [3]. To aid the analytic solution form, angles  $\Delta$  and  $\Sigma$  are defined from the F and G angles in Equation 4.11 below.

$$F = \Sigma - \Delta, \quad G = \Sigma + \Delta, \quad (4.11)$$

where,  $\Sigma = (\pi - \Phi_r)/4$ ,  $\tan \Delta = \frac{V_\infty^- - V_\infty^+}{V_\infty^- + V_\infty^+} \tan \Sigma$

The  $\Delta V$  maneuver magnitude is computed by Equation 4.12, as well as the location  $r_{\Delta V}$  of the minimum impulse. Additional relationships in the hodograph and details of reaching this solution can be found in Walton's dissertation and other literature [2, 3, 4].

$$\begin{aligned} \Delta V &= (V_\infty^- + V_\infty^+) \sin \Delta \\ r_{\Delta V} &= \frac{2\mu \sin^2 F}{V_\infty^{+2} \cos^2 \Delta (2 \cos^2 \Sigma - \cos^2 \Delta)} \end{aligned} \quad (4.12)$$

#### 4.3.4 Unconstrained periapse powered gravity assist solution analysis

The analysis of the unconstrained periapse case (Type F) for the single-impulse powered gravity assist is presented in this section. The goal of this analysis is to verify the solutions of the analytic algorithm for Type F powered gravity assists, developed by Walton et al. [2, 3, 4]. Since this analytic solution to the Type F powered gravity assist is to be adopted into the optimal single-impulse powered gravity assist algorithm, it is necessary to determine the range of flyby configurations for which the analytic algorithm computes the correct optimal powered gravity

assist solution, while avoiding the  $r_{min}$  and  $r_{SOI}$  restrictions. In addition, it is necessary to find the conditions where the analytic unconstrained algorithm computes infeasible powered gravity assist solutions with a flyby periapse below  $r_{min}$  or above  $r_{SOI}$ . It will be shown that the analytic solution determines the exact optimal unconstrained powered gravity assist solution, but in some cases this optimal trajectory breaches one of the radius constraints. The cases where the analytic solution is infeasible are easily found by checking the periapse of the unconstrained solution to see if it falls below  $r_{min}$  or if it exceeds  $r_{SOI}$ . These infeasible cases of the analytic Type F solutions are presented as special cases, where the feasible solutions are determined iteratively with a constrained periapse single-impulse powered gravity assist algorithm presented in the next major section.

In this and proceeding sections, the location of the powered gravity assist  $\Delta V$  maneuver, connecting the two legs of the flyby, is defined as the transfer location, transfer maneuver location, or simply the maneuver location. Similarly a  $\Delta V$  maneuver performed at periapse is referenced as a periapse maneuver or a periapse transfer maneuver, while the periapse  $\Delta V$  maneuver location is referenced as the periapse transfer location or the periapse maneuver location. This nomenclature is intended to clarify the different powered gravity assist maneuvers and their locations.

#### 4.3.4.1 Introductory solution space dynamics

The analytic solution to the unconstrained periapse case can be analyzed in a variety of ways. The simplest method is to show an example solution space for a single impulse powered flyby BVP configuration and demonstrate the optimality of

the solution found from the algorithm. The  $\Delta V$  solution space for an unconstrained periapse powered flyby about Earth is displayed in Figure 4.13 and 4.14 by varying the assumed independent variables: the flyby periapse distance  $r_p$  (in units of Earth radii) and the location of the  $\Delta V$  impulse maneuver along the trajectory as true anomaly  $v_{\Delta V}$ . The solution space shows the  $\Delta V$  maneuvers required for various possible transfer maneuver locations  $v_{\Delta V}$  and associated values of the periapse radius  $r_p$ . The  $\Delta V$  solution space was computed in two halves with respect to the periapse maneuver case, where the transfer maneuver locations on the incoming leg have negative  $v_{\Delta V}$  and the maneuver locations on the outgoing leg have positive  $v_{\Delta V}$ . The overhead view of the solution space clearly shows the global minimum  $\Delta V$  maneuver (yellow dot) located slightly away from the periapse. The minimum  $\Delta V$  periapse maneuver is shown in the diagrams (cyan dot) to require slightly higher  $\Delta V$ . The globally optimal powered gravity assist solution exactly matches that of the analytic Type F algorithm.

These 3D surface plots use the different color gradients to also show the relative  $\Delta V$  magnitudes, where the solution space locations with colors in the red spectrum are the higher  $\Delta V$  magnitudes and the transfer maneuver locations with the lowest  $\Delta V$  magnitudes are shown with colors in the blue spectrum. These 3D solution space plots also show that the Type F optimal powered gravity assist maneuver location is usually not far from the minimum periapse  $\Delta V$  maneuver location. If only considering Type F powered gravity assists, these findings support the results from previous studies which state that the general optimal powered gravity assist maneuver location always lies near periapse. However, when considering any arbitrary flyby configuration, the optimal single-impulse powered gravity assist may

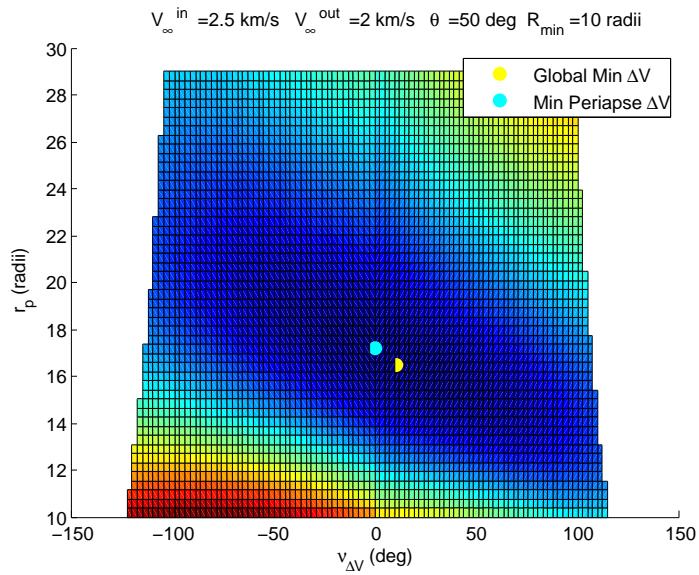


Figure 4.13: Solution space of unconstrained peripuse case with minimum peripuse  $\Delta V$  and overall minimum  $\Delta V$  (at positive  $v_{\Delta V}$ ) solutions shown (overhead view)

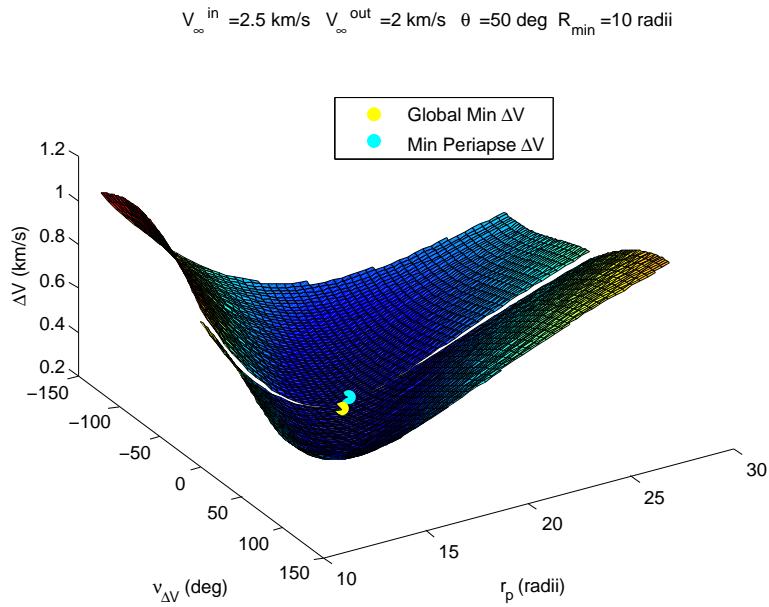


Figure 4.14: Solution space of unconstrained peripuse case with minimum peripuse  $\Delta V$  and overall minimum  $\Delta V$  solutions shown (3D view)

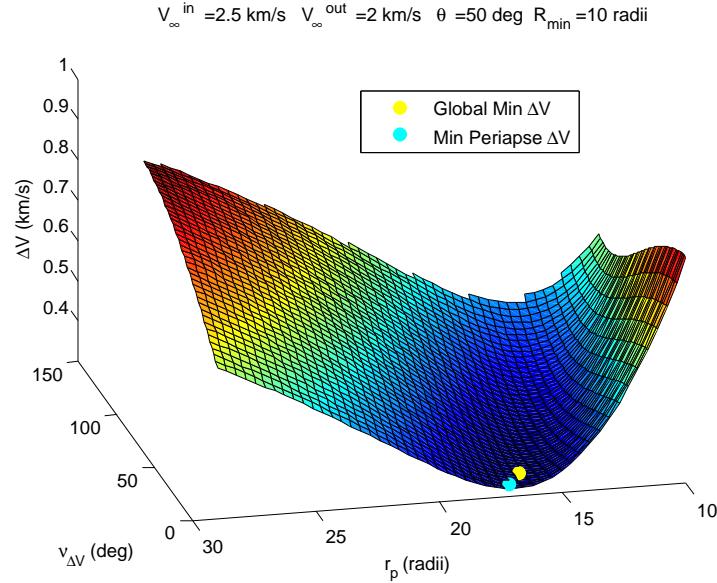


Figure 4.15: Solution neighborhood of unconstrained periapse case containing minimum periapse  $\Delta V$  and overall minimum  $\Delta V$  solutions. Only half of the potential solution space is displayed here, focusing on the positive true anomalies where the optimal maneuver location occurs

be periapse constrained but have global minimum  $\Delta V$  maneuver locations very far from periapse.

In some cases, the three dimensional view shows a slight discontinuity of the solution space along the periapse maneuver line ( $v_{\Delta V} = 0^\circ$ ) for varying periapse distances. The discontinuity can be seen as a white empty space between the solution spaces along the periapse line. The discontinuity occurs because the periapse location of the incoming hyperbolic leg with positive  $v_{\Delta V}$  maneuvers is different from the periapse location of the exiting hyperbolic leg with negative  $v_{\Delta V}$  maneuvers. The discontinuity between the halves of the solution space increases for peri-

apse transfer maneuvers farther away from the minimum. An alternate view of the exact same solution space as described above is shown with only half of the solution space to clearly see the neighborhood around the minimum  $\Delta V$  in Figure 4.15. This different perspective gives a clear view of the periapse maneuver locations and their corresponding  $\Delta V$  magnitudes. The transition between solution space halves is smooth and continuous at the minimum  $\Delta V$  periapse solution location, where the incoming and outgoing legs intersect at their common periapse. This smooth and continuous solution space point is unique for flybys with  $V_{\infty}^- \neq V_{\infty}^+$ . Flybys with  $V_{\infty}^- = V_{\infty}^+$  have continuous solution spaces at all periapse maneuver  $r_p$  magnitudes (having the same periapse location), but is only smooth (having a continuous gradient  $\vec{\nabla}(\Delta V)$ ) at the minimum  $\Delta V$  periapse location. This smooth and continuous periapse maneuver point on the  $V_{\infty}^- = V_{\infty}^+$  solution space is also the ballistic unpowered flyby conditions, which is discussed further below.

Further investigation can be done by varying the boundary parameters such as  $V_{\infty}^-$ ,  $V_{\infty}^+$ , or  $\Phi_r$  and analyzing the change in the optimal maneuver properties. It is sufficient that following presented analysis only demonstrates flybys with  $V_{\infty}^- < V_{\infty}^+$ , where the relationships and results found can also applied to the  $V_{\infty}^- > V_{\infty}^+$  cases by using the reversibility of the flyby problem. The incoming asymptote was fixed for an Earth powered flyby at an arbitrary  $\Phi_r$ , while the outgoing  $V_{\infty}^+$  magnitude was varied. The optimal solution and minimum periapse impulse solution were recorded for each configuration in Figure 4.16. All cases investigated support what is presented in these diagrams, where the globally optimal minimum  $\Delta V$  delivers greater performance than the minimum  $\Delta V$  periapse maneuver solution as  $V_{\infty}^+$  magnitude diverges from  $V_{\infty}^-$ .

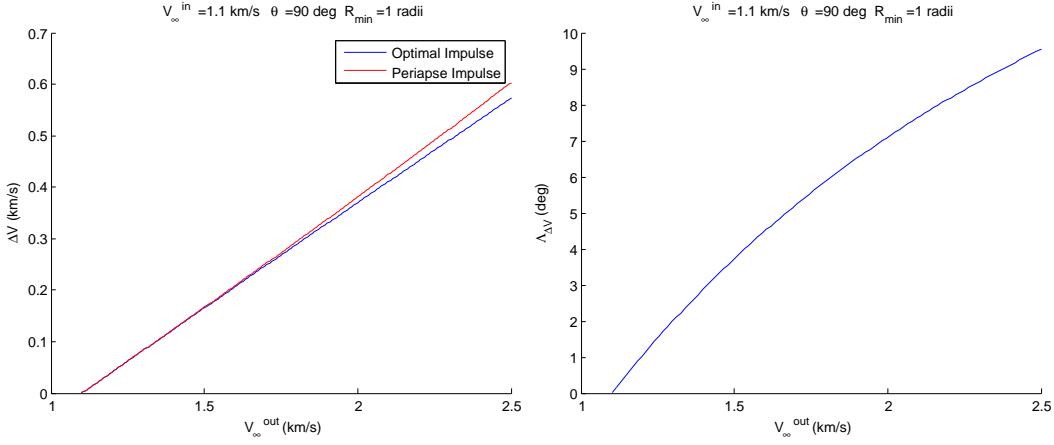


Figure 4.16: Optimal minimum impulse  $\Delta V$  solution compared with collinear peripuse cases (left) and optimal case impulse angle  $\Lambda_{\Delta V}$  (right) for varying outgoing  $V_\infty^+$  magnitudes, showing the convergence of both solution types to coincide when  $V_\infty^- = V_\infty^+$  for Type F flybys

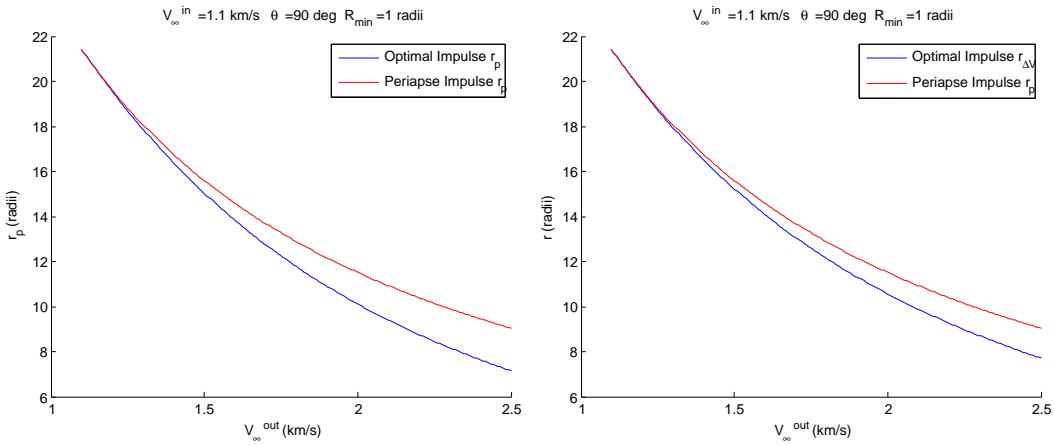


Figure 4.17: Optimal solution peripuse radius  $r_p$  (left) and  $\Delta V$  maneuver radius  $r_{\Delta V}$  (right) for varying outgoing  $V_\infty^+$  magnitudes, showing the convergence of both solution types to coincide when  $V_\infty^- = V_\infty^+$  for Type F flybys

A limiting case is when the periapse impulse maneuver is also the optimal maneuver location and therefore the global minimum  $\Delta V$  solution, as shown in Figures 4.16 and 4.17. Note that the optimal  $\Delta V$  impulse maneuver and the periapse  $\Delta V$  impulse maneuver coincide when  $V_\infty^- = V_\infty^+$ . Additional details about this limiting case is covered later in the discussion. These optimal unconstrained periapse maneuvers are driven to zero  $\Delta V$  at periapse when  $V_\infty^- = V_\infty^+$ , which is the ballistic unpowered flyby condition. The periapse radius  $r_p$  and the radius of the maneuver location  $r_{\Delta V}$  converge together between the two solution types, Figure 4.17. The direction of the  $\Delta V$  impulse with respect to the velocity vector before the maneuver, is defined as the impulse angle  $\Lambda_{\Delta V}$ . The  $\Lambda_{\Delta V}$  for the optimal  $\Delta V$  maneuver is also shown to be driven to zero, while the common periapse solution inherently has a collinear  $\Delta V$  impulse of  $\Lambda_{\Delta V} = 0$  for all Type F flybys. The  $\Delta V$  direction for a single impulse maneuver is only tangent to the trajectory velocity direction ( $\Lambda_{\Delta V} = 0$ ) at periapse, but this is also where the  $\Delta V$  magnitude goes to zero. This identifies the same conclusion Gobetz found that finite  $\Delta V$  powered flyby maneuvers are generally non-tangential [7].

To better understand the solution space, we will now compare the optimal unconstrained periapse (Type F) powered gravity assist with flybys using collinear (i.e. tangential)  $\Delta V$  maneuvers to accomplish the powered gravity assist. As an example, the solution space for an Earth gravity assist is presented with a  $V_\infty^- = 10$  km/s, a  $V_\infty^+ = 11$  km/s, and a turning angle  $\Phi_r = 35^\circ$ . The solution space is highlighted with the locus of minimum  $\Delta V$  maneuver locations (cyan) for each value of  $r_p$  alongside the locus of collinear  $\Delta V$  maneuver locations (red), Figure 4.18. Essentially, if the flyby periapse were fixed at each specific  $r_p$  magnitude,

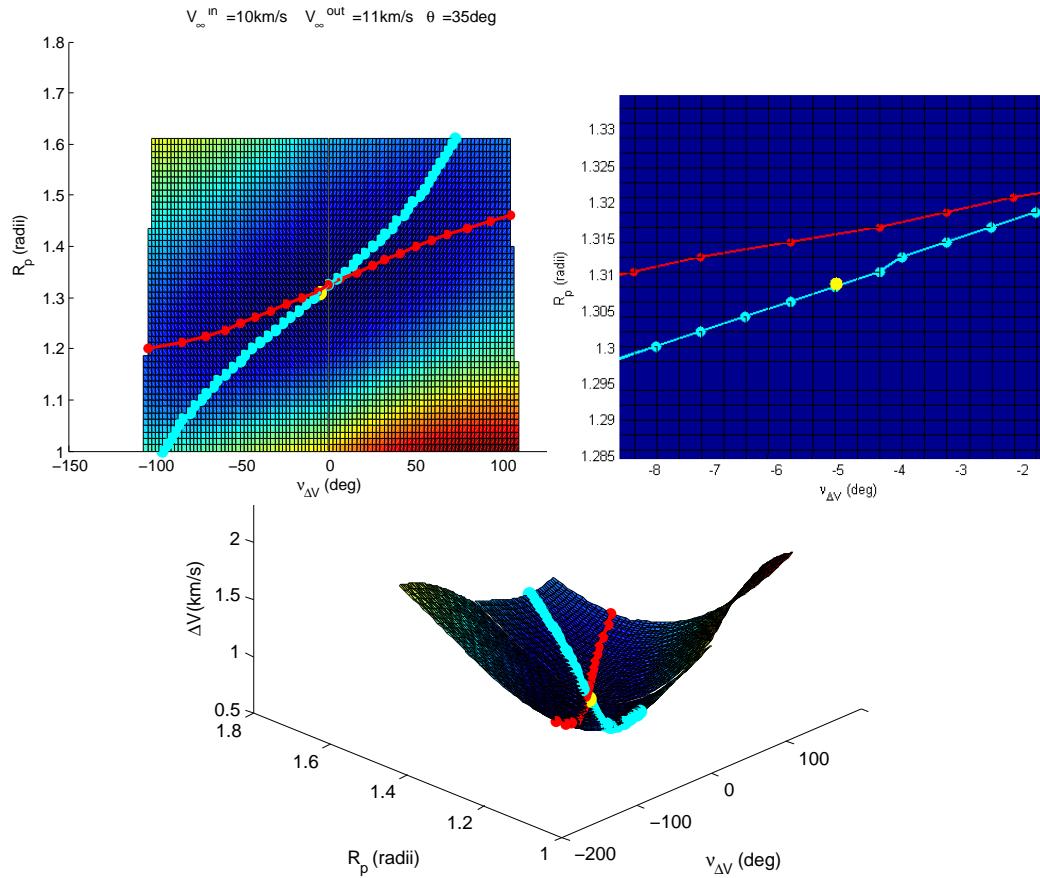


Figure 4.18: Unconstrained periapse powered gravity assist solution space depicting locus of minimum  $\Delta V$  maneuvers (cyan) and locus of collinear/tangential  $\Delta V$  maneuvers (red), as well as the global minimum  $\Delta V$  location (yellow) about Earth: above view (top left), zoomed at optimal  $\Delta V$  maneuver (top right), and 3D view (bottom)

the most efficient minimum  $\Delta V$  maneuver location  $v_{\Delta V}$  is shown by the cyan dot. Similarly, the locus of collinear maneuver locations for each  $r_p$  is represented by the red line. The right hand figure shows a zoomed in look at the area around the optimum maneuver point. The global minimum  $\Delta V$  impulse and maneuver location is shown by the yellow dot in the negative  $v_{\Delta V}$  region near periapse.

It is clear from the left hand graph that collinear maneuvers only exist for a small range of  $r_p$  values, while optimal maneuvers exist for a much wider range of values of  $r_p$ . The optimal maneuver location for the global minimum  $\Delta V$  is only collinear for finite  $\Delta V$  impulse locations when at the maximum natural turning angle condition previously identified (Figure 4.21). The collinear maneuver is always sub-optimal compared to the actual minimum  $\Delta V$  maneuver for the same  $r_p$ . These 3D solution space plots show how the cyan locus of optimal  $\Delta V$  maneuver locations for each  $r_p$  magnitude can diverge from the collinear  $\Delta V$  maneuver locations. This demonstrates the potential for such stark differences in powered gravity assist maneuver locations and  $\Delta V$  magnitudes between the collinear and optimal  $\Delta V$  maneuvers.

#### 4.3.4.2 Strict $r_{min}$ constraint handling for transfer maneuvers near periapse

This section discusses the special occasions where the analytic Type F flyby trajectory solution breaches the  $r_{min}$  constraint by producing a powered gravity assist trajectory with a periapse below  $r_{min}$ . Though it is guaranteed that flyby configurations with excessive turning angles  $\Phi_r > \Phi_p$  always have a periapse constrained at  $r_{min}$ , it cannot be guaranteed that flybys with  $\Phi_r < \Phi_p$  are always unconstrained. A majority of the  $\Phi_r < \Phi_p$  flyby configurations have optimal single-impulse flyby

solutions with a periapse above the  $r_{min}$  constraint. But there exist flyby configurations close that of the maximum turning angle  $\Phi_p$  condition where the flyby periapse falls below the  $r_{min}$  constraint.

The Type F flyby algorithm was designed by Walton et al. to determine the unconstrained periapse optimal powered gravity assist maneuver location without any consideration for the  $r_{min}$  constraint [3, 4]. The sorting procedure identifies the obvious periapse constrained  $r_p = r_{min}$  cases by checking if  $\Phi_r > \Phi_p$ . Yet, the  $\Phi_r < \Phi_p$  cases are more difficult to know if they are constrained or unconstrained without further computation.

Therefore, the procedure developed to handle the seemingly periapse unconstrained  $\Phi_r < \Phi_p$  cases is to first compute the analytic Type F solution and then check if the periapse is below  $r_{min}$ . For the unconstrained periapse sorted cases of  $\Phi_r < \Phi_p$  as the required turning angle  $\Phi_r$  increases towards  $\Phi_p$ , the optimal powered gravity assist has its periapse constrained by  $r_{min}$  just before reaching the transition point of  $\Phi_r = \Phi_p$ , where any greater  $\Phi_r$  and the flyby is guaranteed to be periapse constrained as  $\Phi_r > \Phi_p$ . More details and analysis on the  $r_{min}$  constraint handling can be found in Appendix A.

The  $r_{min}$  constrained case solutions utilize the methodology and algorithms developed in the next major section to efficiently and reliably solve the optimal maneuver location for a minimum  $\Delta V$ , while maintaining the  $r_{min}$  restriction. The  $r_{min}$  constrained optimal gravity algorithm is initialized using the infeasible  $v_{\Delta V}$  result from the unconstrained periapse analytic algorithm. The analysis of the solution space near the optimal periapse constrained flyby confirms that this initialization

point is always close to the desired feasible solution, providing fast convergence.

#### 4.3.4.3 Linked conics $r_{SOI}$ constraint handling

There exists a second set of special case solutions to the single-impulse gravity assist problem where the required turning angle is less than the maximum available turning angle ( $\Phi_r < \Phi_p$ ). One of the major foundational assumptions of linked conics is that the encountered gravitational body must have an effect the trajectory. This is achieved by limiting the flyby periapse distance to a maximum at the sphere of influence  $r_{SOI}$ . Without such a restriction, extreme flyby configurations might have a periapse that diverges to an unrealistically large distance. In order to handle this situation, when the unconstrained periapse Type F flyby periapse exceeds the maximum radius  $r_{SOI}$ , it is reevaluated as a constrained flyby with  $r_p = r_{SOI}$  and the optimal  $\Delta V$  maneuver location is determined.

The maneuver location for this  $r_{SOI}$  constrained periapse flyby can potentially extend far out beyond the vicinity of the planet's sphere of influence, so an upper bound must be set on the maneuver location. The maximum transfer maneuver radius is set to  $r_{max} = 2r_{SOI}$  in this algorithm. These maximum flyby radius constraint methods have been proven to work well in practice to drive the trajectory optimization away from such extreme flyby conditions and toward more effective gravity assist configurations. The group of  $r_{SOI}$  constrained solutions occur at very low flyby turning angles with respect to the asymptote excess velocities. More details and analysis on the  $r_{SOI}$  and  $r_{max}$  constraint handling can be found in Appendix B.

These are extreme cases where the flyby is to be minimally affected by the

presence of the encounter planet. These flyby configuration conditions could be judged by a mission planner to remove the gravity assist encounter from the trajectory, although the intent of the mission planner could be to use this flyby configuration as a guide to finding a more useful flyby configuration. The later intent is assumed in the optimal single-impulse powered gravity assist algorithm developed here, so that the overall spacecraft mission trajectory optimization routine can find the best possible trajectory with the selected gravity assist encounters. It is left to the mission planner to manually remove the gravity assist encounter from the spacecraft mission trajectory if judged necessary.

The set of candidate single-impulse gravity assist solutions for the  $\Phi_r < \Phi_p$  flyby types are completely determined by the unconstrained periapse Type F powered flyby and the two special cases using constrained periapse powered flyby solution methods. It can easily be seen that the neighborhood around the constrained cases would benefit by the multiple impulse transfer solutions discussed in prior sections: Type  $F_\infty$ , Type  $RF_\infty$ , and Type PNP. For the two-impulse transfers, the total  $\Delta V$  is reduced by using the impulse at infinity (or at least  $r_{SOI}$ ) to adjust the periapse location for a more efficient impulse in the vicinity of the planet. The same concept is true for the Type PNP transfers, but usually for more extreme circumstances. Because these multiple-impulse solution case have lower total  $\Delta V$  magnitudes compared to the constrained single-impulse powered flyby cases, the single-impulse powered flyby problem was never characterized for these particular conditions in the literature reviewed by the author. Therefore, the characterization of these constrained periapse powered gravity assists of the type  $\Phi_r < \Phi_p$  and the developed solution methods are considered new and part of the major contributions

of the dissertation.

#### 4.3.5 Constrained periapse powered gravity assist $\Delta V$ function and derivative development

Many past methods require multiple optimization variables to find the optimal  $\Delta V$  maneuver location and must constantly check to maintain the  $r_{min}$  constraint. These often rely on an iterative substitution approach or numerical derivatives in the optimization process, which are less accurate and usually require more computation than analytic derivatives. These iterative methods often use poor initial guess points to begin the iterative optimization process, leading to sub-optimal local minima or boundary cases. Other simpler methods simply assume a periapse maneuver for a preselected hyperbolic leg, where  $r_p$  is fixed at  $r_{min}$  or  $r_p$  is optimized in order to maintain the  $r_{min}$  constraint for the entire flyby. Another common issue with past algorithms is that though one leg's periapse is restricted at the  $r_{min}$  constraint, the other leg of the flyby may be allowed to fall below  $r_{min}$ .

In this major section, the optimal single impulse constrained periapse powered gravity assist BVP is presented and discussed in detail. Previous studies and literature are reviewed to establish the state-of-the-art strategies for solving the constrained periapse powered gravity assist BVP prior to the current research. The primary shortcomings of these prior strategies are their lack of considering mission feasible constraints for the powered gravity assist (such as the number of  $\Delta V$  maneuvers, constraints on transfer time, loose  $r_{min}$  constraint handling, etc.). Their single-impulse constrained gravity assist solution methods are only applicable to a small sub-set of the potential flyby configurations encountered in practice. This

section develops new equations for the constrained periapse single-impulse gravity assist, determining the  $\Delta V$  maneuver as a function of the maneuver location ( $r_{\Delta V}$  or  $v_{\Delta V}$ ) and its derivative. The next major section analyzes the solution space of the constrained periapse single-impulse gravity assist problem using the new equations, resulting in an efficient and robust BVP solution algorithm for optimal single-impulse constrained periapse gravity assists.

#### 4.3.5.1 Development of the $\Delta V$ function and derivative

The development of the  $\Delta V$  function and its derivatives is presented below in the following order: the reduction of optimization variables, the derivation of an effective  $\Delta V$  magnitude equation and its derivative, an analysis of the problem's solution space, and finally, its implementation into a BVP solution algorithm.

Approaching the boundary value problem with a generalized algorithm for all flyby configurations  $\Phi_r \gtrless \Phi_p$  and  $V_\infty^- \gtrless V_\infty^+$  requires at least two independent optimization variables as described in Section 4.1, analogous to Gobetz's approach with  $\psi_1$  and  $\psi_2$ . The solution neighborhood sorting procedure derived in Section 4.3.2 enables the constrained periapse powered gravity assist algorithm to have only one independent optimization variable to define the maneuver location, either the radius,  $r_{\Delta V}$ , or the true anomaly,  $v_{\Delta V}$ , of the maneuver. The flyby configuration  $\Phi_r > \Phi_p$  implies that the flyby trajectory must be turned more than is available ballistically, and that a  $\Delta V$  maneuver is needed. In this situation, the optimal gravity assist must graze the  $r_{min}$  sphere, meaning that one leg of the hyperbolic transfer is fully determined by the constrained periapse,  $r_p = r_{min}$ . The remaining leg of the flyby can have a transfer maneuver at any point along the determined leg trajectory

from periapse out to the sphere of influence ( $r_{SOI}$  or  $v_{SOI}$ ) and be able to achieve the appropriate  $\vec{V}_\infty$  conditions, Figure 4.19. The goal is to find the unique hyperbolic transfer trajectory that minimizes the required  $\Delta V$  maneuver. This periapse constrained powered flyby diagram shows examples of candidate hyperbolic transfer maneuver locations and the associated  $\Delta V$  relative magnitudes, where the  $\Delta V$  magnitudes near periapse are clearly larger than those located farther out. The sorting process removes any ambiguity associated with the use of radius or true anomaly as the optimization search variable, since the solution is predetermined to graze  $r_{min}$  on one leg and to have a  $\Delta V$  maneuver on the other side of the conic, where  $r_{\Delta V} \geq r_p$  and  $v_{\Delta V} \geq 0$  (or  $v_{\Delta V} \leq 0$ ).

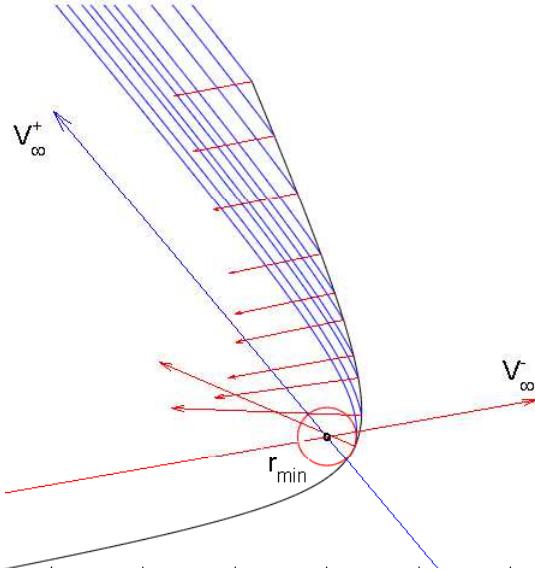


Figure 4.19: Example potential outgoing transfer maneuver locations and trajectories (blue) for  $V_\infty^- < V_\infty^+$  and  $\Phi_r > \Phi_p$  of a constrained periapse flyby configuration, showing the  $\Delta V$  maneuver direction and relative scale (red)

An efficient and robust solution algorithm for the constrained periapse pow-

ered gravity assist BVP requires a thoughtful choice of defining the  $\Delta V$  equation to be used in the iterative optimization process. Many deterministic optimization routines benefit from additional problem information in the form of the performance metric derivative with respect to the optimization variables, in this case  $\frac{d\Delta V}{dr}$  or  $\frac{d\Delta V}{dv}$ . By defining the  $\Delta V$  equation appropriately, its computational efficiency and that of its analytic derivative  $\frac{d\Delta V}{dr}$  or  $\frac{d\Delta V}{dv}$  can be improved. An efficient scalar formulation was selected as a function of flight path angles  $\gamma^\pm$  at the transfer maneuver location, Equation 4.13. The velocity magnitudes at the maneuver location are known, as well as the constrained leg's flight path angle, from basic orbital relationships.

$$\Delta V = \sqrt{V^{+2} + V^{-2} - 2V^+V^- \cos(\gamma^+ - \gamma^-)}$$

(4.13)

$$\text{where } V^\pm = \sqrt{\frac{2\mu}{r} + V_\infty^{\pm 2}}$$

Next, an efficient scalar derivation of the unconstrained leg's  $\gamma$  as a function of maneuver location  $r$  or  $v$  is required, while satisfying the flyby terminal velocity conditions. A thorough search for alternative formulations of the hyperbolic transfer BVPs in terms of  $\gamma$  failed to yield a viable method. All standard formulae explored for  $\gamma$  includes some form of eccentricity  $e$ . For the present problem, eccentricity can only be computed inefficiently using the  $\vec{e}$  vector Equation 4.14, requiring the terminal velocity vector Equation 2.14 or 2.13. Such a formulation leads to exceedingly computationally inefficient derivatives  $\frac{d\Delta V}{dr}$  and  $\frac{d\Delta V}{dv}$ . Finally, the hodograph approach of Battin was investigated to derive an alternative scalar formulation of  $\gamma$  [10].

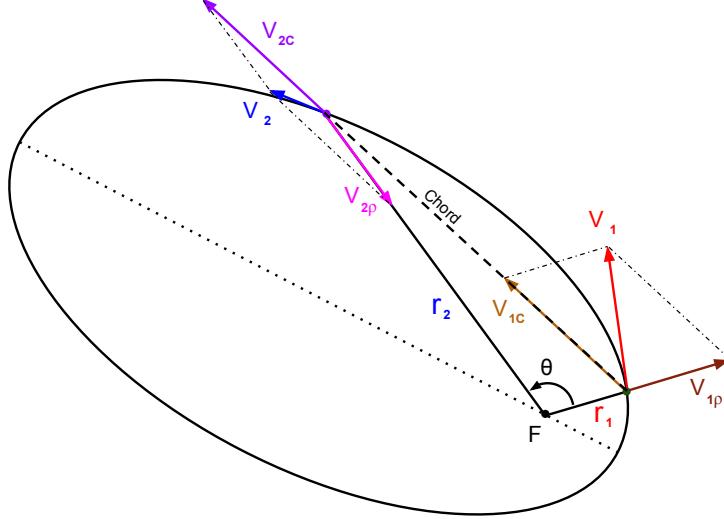


Figure 4.20: Lambert problem with defined radial and chordal skewed velocity components

$$\vec{e} = \frac{V^2 \vec{r}}{\mu} - \frac{(\vec{r} \cdot \vec{V}) \vec{V}}{\mu} - \frac{\vec{r}}{r} \quad (4.14)$$

The foundation of Battin's terminal velocity hodograph analysis is Lambert's problem introduced in Section 2.3.1. A common approach to Lambert's orbital transfer problem is to define a chord between the initial and final positions ( $C = \|\vec{r}_2 - \vec{r}_1\|$ ) in order to derive the associated velocity vectors  $\vec{V}_1$  and  $\vec{V}_2$  at these end points of the conic trajectory. Rather than employing a standard coordinate system for  $\vec{V}_1$  and  $\vec{V}_2$ , an interesting and pivotal relation results from using a skewed local basis of radial and chordal components, as shown in Figure 4.20. The surprising relationship is that chordal components  $\vec{V}_{1C}$  and  $\vec{V}_{2C}$  of the end points are not only parallel but are also the same magnitude [10].

$$\begin{aligned}
V_r &= \frac{\sqrt{\mu p}}{r} \cot \gamma = \sqrt{\frac{\mu}{p}} e \sin v \\
V_\theta &= \frac{\sqrt{\mu p}}{r} \\
\vec{V} &= V_r \hat{r} + V_\theta \hat{\theta}
\end{aligned} \tag{4.15}$$

$$\begin{aligned}
\cos \gamma &= \frac{V_\theta}{V} \\
\sin \gamma &= \frac{V_r}{V}
\end{aligned} \tag{4.16}$$

For comparison, the common form of the spherical components  $V_r$  and  $V_\theta$  are given in Equation 4.15, being the typical parameters used to find flight path angle, Equation 4.16. The radial and chordal components  $V_\rho$  and  $V_C$  are found in Equation 4.17. Note the radial component of the skewed basis  $V_\rho$  is not the same as the radial component of the spherical system  $V_r$ . An additional noteworthy relationship can be derived from the skewed components in Equation 4.18, where the product  $V_C V_\rho$  does not involve any orbital parameters, just the configuration of the boundary value problem [10].

$$\begin{aligned}
V_\rho &= \sqrt{\frac{\mu}{p}} \frac{1 - \cos \theta}{\sin \theta} \\
V_C &= \frac{C \sqrt{\mu p}}{r_1 r_2 \sin \theta} \\
\vec{V}_{1,2} &= V_\theta \hat{C} \pm V_\rho \hat{r}
\end{aligned} \tag{4.17}$$

$$V_C V_\rho = \frac{\mu C}{2 r_1 r_2} \sec^2(\theta/2) \tag{4.18}$$

Extending the skewed coordinates of Lambert's problem to the hyperbolic terminal velocity problem is key in developing an effective equation for flight path

angle  $\gamma$ . The powered gravity assist BVP is defined with the terminal velocity transfer angle  $\theta_\infty$  for the incoming or outgoing transfer maneuver in Figure 4.21. These powered flyby diagrams depict the hyperbolic transfer parameters ( $\theta_\infty$ ,  $r_{\Delta V}$ ,  $v_{\Delta V}$ , etc.) as related to a powered gravity assist problem. In these diagrams the parameter  $v_\infty$  is the true anomaly angle at  $r_\infty$ , used to measure the asymptote direction angle with respect to the  $r_{min}$  constrained periapse. The powered flyby transfer maneuver was parameterized with a focus on the skewed radial and chordal velocity components to derive a new form of  $\gamma$ . A diagram of the velocity and components at the transfer maneuver location is provided in Figure 4.22, where the horizontal velocity component  $V_\theta$  and  $\gamma$  are easily related to the skewed velocity components.

Using the terminal velocity transfer angle  $\theta_\infty$  and the circumferential velocity  $V_\theta$  in the form of chordal velocity  $V_C$ , the flight path angle  $\gamma$  can be determined, Equation 4.19. Despite the coupled nature of components in a skewed basis, the chordal velocity component retains direction of motion information with respect to flight path angle.

$$\begin{aligned} V_\theta &= V_C \sin \theta_\infty \\ \cos \gamma &= \frac{V_C \sin \theta_\infty}{V} \end{aligned} \tag{4.19}$$

Next, an efficient equation for the chordal velocity component  $V_C$  is required to determine the flight path angle  $\gamma$ . Understanding the hyperbolic terminal velocity BVP as an extension of the Lambert problem, many of the features of the Lambert problem correlate to the hyperbolic transfer as well. Specifically, the chordal velocity component at the powered flyby maneuver location (as outlined in Figure 4.22 for the hyperbolic terminal velocity transfer BVP) is in the same direction as

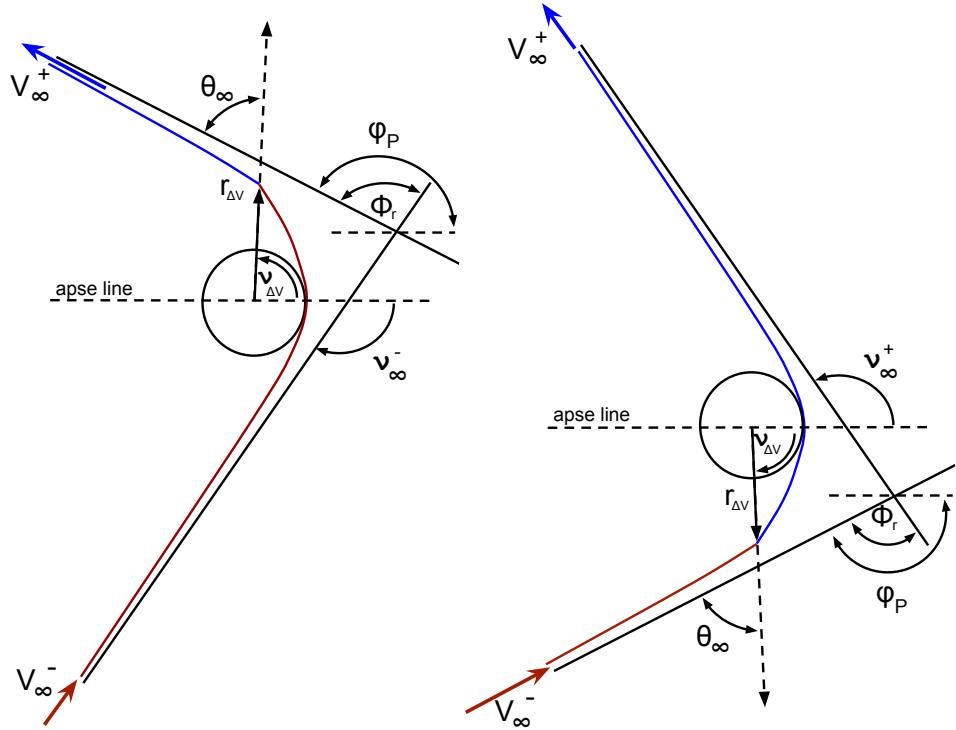


Figure 4.21: Constrained powered gravity assist of  $\Phi_r > \Phi_p$  with  $V_{\infty}^- < V_{\infty}^+$  (type RF: left) and  $V_{\infty}^- > V_{\infty}^+$  (type FR: right) incorporating outgoing and incoming hyperbolic terminal velocity transfers respectively.

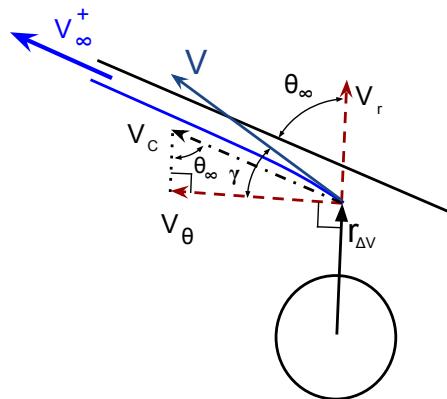


Figure 4.22: Outgoing hyperbolic terminal velocity problem diagram of velocity components as related to flight path angle  $\gamma$  at the transfer maneuver location, where the flyby trajectory view is isolated just on the outgoing hyperbolic transfer leg

the leg's  $\vec{V}_\infty^+$  asymptote. Recognizing this relationship and using the hyperbolic terminal velocity algorithm derived by Battin [10] in Equation 4.20 below, the  $\hat{V}_\infty^+$  component term is also the chordal velocity component  $V_C$  of the flyby hyperbolic velocity  $\vec{V}^+$  at the transfer maneuver location  $r$ . Therefore the  $V_C$  equation is determined and shown in Equation 4.21. In order to verify this relationship, the hyperbolic terminal velocity problem is analyzed in detail, including constructing the velocity vector hodograph diagram for the hyperbolic transfer and deriving the chordal velocity component equation. The analysis and derivation confirms the  $V_C$  equation presented below. The details of this analysis can be found in Appendix C.

$$\begin{aligned}\vec{V}^+ &= (D + \frac{1}{2}V_\infty^+) \hat{V}_\infty^+ + (D - \frac{1}{2}V_\infty^+) \hat{r} \\ D &= \sqrt{\frac{\mu}{r(1+\cos\theta_\infty^+)} + \frac{V_\infty^{+2}}{4}}\end{aligned}\tag{4.20}$$

$$V_C = \sqrt{\frac{\mu}{r(1+\cos\theta_\infty)}} + \frac{V_\infty^2}{4} + \frac{V_\infty}{2}\tag{4.21}$$

Finally, the flight path angle equation can be found as a function of the hyperbolic transfer maneuver parameters, Equation 4.22 below. When implementing the equation, there is potential for cases to occur where  $\gamma^-$  and  $\gamma^+$  should be of different signs, which significantly impacts the  $\Delta V$  equation. The constrained leg is guaranteed to pass through periapse and have a known flight path angle (magnitude and sign), although the unconstrained leg's flight path angle is less easily determined. If the constrained leg's periapse is the only apsidal transit during a flyby, then both  $\gamma^-$  and  $\gamma^+$  will be of the same sign. In extreme flyby configuration cases, the unconstrained leg may also pass through its own periapse, resulting in  $\gamma^-$

and  $\gamma^+$  of different signs. Most forms of flight path angle only provide a positive magnitude value and don't accommodate the sign change. Additional analysis produced a robust formulation using already computed terms seen in Equation 4.23. The term  $S_\gamma$  determines the sign of  $\gamma$  by using the radial velocity component information found from the hyperbolic terminal velocity equation. The preceding sign  $\pm$  of  $S_\gamma$  is chosen by flyby type: if Type RF then + and if Type FR then -.

$$\cos \gamma = \left( \sqrt{\frac{\mu}{r(1+\cos \theta_\infty)} + \frac{V_\infty^2}{4}} + \frac{V_\infty}{2} \right) \frac{\sin \theta_\infty}{V} \quad (4.22)$$

$$\begin{aligned} \gamma &= S_\gamma \arccos \left( \left( S_\theta \sqrt{\frac{\mu}{r(1+\cos \theta_\infty)} + \frac{V_\infty^2}{4}} + \frac{V_\infty}{2} \right) \frac{\sin \theta_\infty}{V} \right) \\ S_\gamma &= \pm \text{sign} \left[ S_\theta (1 + \cos \theta_\infty) \sqrt{\frac{\mu}{r(1+\cos \theta_\infty)} + \frac{V_\infty^2}{4}} + (\cos \theta_\infty - 1) \frac{V_\infty}{2} \right] \end{aligned} \quad (4.23)$$

$$S_\theta = \text{sign}(\pi - \theta_\infty)$$

The last variable to be determined as a function of  $r$  or  $v$  is the hyperbolic transfer angle  $\theta_\infty$ . The hyperbolic transfer angle  $\theta_\infty$  is defined between the maneuver location radial  $\hat{r}$  and the hyperbolic  $\hat{V}_\infty$  asymptote, as displayed in Figure 4.21. Using the common vector definition  $\theta_\infty^+ = \arccos(\hat{r} \cdot \hat{V}_\infty^+)$  would require determining the  $\vec{r}$  vector, which dramatically increases the equation's complexity and computation requirements. A more efficient scalar form of  $\theta_\infty$  was derived using known hyperbolic orbit data of both legs. The periapse of the constrained leg was used as a reference direction for the following angles. The transfer angle  $\varphi_p$  is the angle between the constrained leg's periapse and the other leg's  $\hat{V}_\infty$  asymptote

direction, also shown in Figure 4.21. The transfer angle  $\varphi_p$  is found by the geometry of the flyby configuration, using the turning angle  $\Phi_r$  and the constrained leg's asymptotic true anomaly  $v_\infty$ , Equation 4.24. An alternative geometric relationship for  $\varphi_p$  is derived from the flyby configuration by the constrained leg's  $\Delta V$  maneuver location true anomaly  $v_{\Delta V}$  and the transfer angle  $\theta_\infty$ , Equation 4.25. The  $\theta_\infty$  equation can be in terms of  $r_{\Delta V}$  by substituting for  $v_{\Delta V}$ .

$$\begin{aligned}\varphi_p &= \Phi_r + \pi - v_\infty \\ &= v_{\Delta V} + \theta_\infty\end{aligned}\tag{4.24}$$

$$\begin{aligned}\theta_\infty &= \Phi_r - v_\infty - v_{\Delta V} + \pi \\ &= \Phi_r - v_\infty - \arccos\left(\frac{\sqrt{\mu p}}{er_{\Delta V}} - \frac{1}{e}\right) + \pi\end{aligned}\tag{4.25}$$

Collecting all the derived parameters and terms, the  $\Delta V$  equation is fully determined by the maneuver location, the flyby configuration, and the constrained leg's orbital parameters. The flyby type (either Type RF or Type FR) identifies whether the corresponding + or - trajectory leg parameters are used. As a reminder, a Type RF flyby has its incoming leg constrained at  $r_{min}$  and a maneuver after periapse passage, while Type FR has the outgoing leg constrained by  $r_{min}$  and the maneuver located before periapse passage.

$$\begin{aligned}
\Delta V &= \sqrt{V^{+2} + V^{-2} - 2V^+V^- \cos(\gamma^+ - \gamma^-)} \\
V^\pm &= \sqrt{\frac{2\mu}{r} + V_\infty^{\pm 2}} \\
\gamma^\pm &= \pm \arccos \frac{\sqrt{\mu p^\pm}}{r V^\pm} \\
\gamma &= S_\gamma \arccos \left( \left( S_\theta \sqrt{\frac{\mu}{r(1+\cos\theta_\infty)} + \frac{V_\infty^2}{4}} + \frac{V_\infty}{2} \right) \frac{\sin\theta_\infty}{V} \right) \\
\theta_\infty &= \Phi_r - v_\infty^\pm - \arccos \left( \frac{\sqrt{\mu p^\pm}}{e^\pm r_{\Delta V}} - \frac{1}{e^\pm} \right) + \pi
\end{aligned} \tag{4.26}$$

Next, the analytic derivatives of  $\Delta V$  with respect to the maneuver radius and true anomaly were derived for  $\frac{d\Delta V}{dr}$  and  $\frac{d\Delta V}{dv}$  in order to compare complexity and computational advantage. It was found that  $\frac{d\Delta V}{dr}$  resulted in a more efficient derivative form, seen in Equation 4.27. The alternate derivative  $\frac{d\Delta V}{dv}$  could be found if needed by chain rule, Equation 4.28. The signs  $\mp$  in  $\frac{d\Delta V}{dr}$  again are chosen by flyby type,  $-$  if Type RF and  $+$  if Type FR.

$$\begin{aligned}
\frac{d\Delta V}{dr} &= \frac{1}{2\Delta V} \left[ -\frac{4\mu}{r^2} + \left( \frac{8\mu^2}{r^3} + \frac{2\mu(V_\infty^{-2} + V_\infty^{+2})}{r^2} \right) \frac{\cos(\gamma^+ - \gamma^-)}{V^- V^+} \right. \\
&\quad \mp 2V^- V^+ \sin(\gamma^+ - \gamma^-) \left[ \left\{ -S_\theta \left( \frac{1}{r} + \frac{p^\pm c_1 \sin\theta_\infty}{\mu r e^\pm \sin v} \right) \frac{c_1 \sin\theta_\infty}{V^+ \sqrt{4c_1 + V_\infty^{\pm 2}}} \right. \right. \\
&\quad \left. \left. \mp \frac{p^\pm \cos\gamma^\pm \cot\theta_\infty}{e^\pm r^2 \sin v} + \frac{\mu \cos\gamma^\pm}{r^2 v^{\pm 2}} \right\} \frac{1}{\sin\gamma^\pm} + \frac{(rV_\infty^{\pm 2} + \mu) \cot\gamma^\pm}{r^2 V^{\pm 2}} \right] \right]
\end{aligned} \tag{4.27}$$

where,

$$c_1 = \frac{\mu}{r(1+\cos\theta_\infty)}$$

$$\frac{d\Delta V}{dv} = \frac{d\Delta V}{dr} \frac{dr}{dv} = \frac{d\Delta V}{dr} \frac{r e^\pm \sin v}{1 + e^\pm \cos v} \tag{4.28}$$

The derived  $\Delta V$  function and its derivative are non-linear, and the behavior

of these functions must be analyzed over the spectrum of potential boundary value problems encountered. The next section investigates potential constrained periapse powered flyby boundary value problem configurations and provides insight about the solution space characteristics. The analysis concludes by presenting an efficient and robust minimum  $\Delta V$  optimization algorithm for the constrained periapse powered gravity assist BVP.

#### **4.3.6 Constrained periapse powered gravity assist solution analysis**

Using the constrained periapse powered gravity assist  $\Delta V$  maneuver equation and its derivative, the  $\Delta V$  maneuver solution space of the powered gravity assist problem can be investigated. It will be shown in this section that the constrained periapse powered gravity assist problem can potentially be multimodal with respect to the  $\Delta V$  magnitude as a function of transfer maneuver location ( $v_{\Delta V}$ ), where there can exist a maximum and a minimum stability point along the solution space. It will also be shown that the  $\Delta V$  solution space can be better modeled as a cubic function, therefore quadratic optimization techniques can be replaced by cubic methods. A key feature of the solution space dynamics is that the  $\frac{d\Delta V}{dv}$  derivative at  $r_{SOI}$  can indicate if the optimal  $\Delta V$  maneuver location occurs at the  $r_{SOI}$  boundary or if the optimal maneuver location exists at an interior point between  $r_{SOI}$  and periapse. Finally, the  $r_{min}$  infeasible maneuver locations are shown to occur near periapse and an efficient algorithm is developed to determine the infeasible region.

The main variable to for these studies is the maneuver location, which is best represented graphically by true anomaly  $v_{\Delta V}$ . The initial investigations identify local maneuver location dynamics for various fixed flyby configuration cases, where

$r_p$ ,  $V_\infty^\pm$ , and  $\Phi_r$  are constant for a given planet encounter. The next level of analysis utilizes three dimensional (3D) surface plots to portray the powered gravity assist solution space for various flyby configurations. The 3D plots enable investigating the effects of flyby configuration parameters on  $\Delta V$  performance and maneuver location  $v_{\Delta V}$ , where two parameters are varied and displayed with  $\Delta V$ .

Similar to the unconstrained periapse analysis, it is sufficient that following investigations were done for flybys with  $V_\infty^- < V_\infty^+$ , where the relationships and results found can also be applied to the  $V_\infty^- > V_\infty^+$  cases by using the reversibility of the flyby problem. The solution space dynamics are relative to the flyby configuration parameters, so similar trends and relationships can be found at low energy flyby configurations (low  $V_\infty^\pm$  relative to  $\Phi_r$  and  $r_{min}$ ) as at high energy flyby configurations (high  $V_\infty^\pm$  relative to  $\Phi_r$  and  $r_{min}$ ). Both energy regimes will be represented in the analyses, initially showing lower energy solution spaces followed by the higher energy solution spaces.

#### 4.3.6.1 Introductory solution space dynamics

The following constrained periapse flyby analysis depicts all potential maneuver locations occurring between the incoming asymptote at the  $v_{SOI}^-$  and the outgoing asymptote at  $v_{SOI}^+$ . Each of the solution space plots are set at specific flyby configurations in order to demonstrate specific solution space characteristics. The flyby  $\Delta V$  maneuver locations are split into two sets, negative and positive true anomaly maneuver locations  $v_{\Delta V}$ . The positive  $v_{\Delta V}$  maneuver locations are for Type RF flybys where the incoming leg is constrained at  $r_{min}$ , and the  $\Delta V$  maneuver occurs between periapse and  $v_{SOI}^+$ . The negative  $v_{\Delta V}$  maneuver locations are for

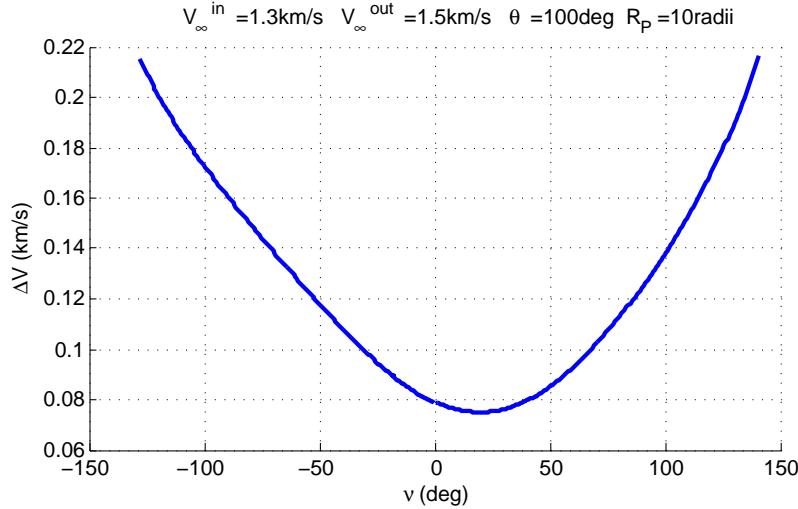


Figure 4.23: Example constrained periapse powered gravity assist with minimum  $\Delta V$  maneuver near periapse

Type FR flybys where the outgoing leg is constrained at  $r_{min}$ , and the  $\Delta V$  maneuver occurs between periapse and  $V_{SOI}^-$ .

An example case representative of what past literature considers a typical constrained periapse powered flyby is shown in Figure 4.23, where the minimum  $\Delta V$  maneuver occurs near the periapse. The solution space appears to be somewhat quadratic and unimodal, also commonly expected from literature. The optimal maneuver location indeed lies on the positive true anomaly side of the incoming leg, which confirms the expected Type RF configuration for a periapse constrained  $V_\infty^- < V_\infty^+$  flyby. If we increase the exiting  $V_\infty^+$  magnitude while holding the  $V_\infty^-$  level fixed, the solution space shifts the minimum  $\Delta V$  location away from periapse, as seen in Figure 4.24. This plot also reveals a discontinuity at periapse between the two sets of maneuver locations on either side of the periapse. This discontinuity is always present in a periapse constrained Type RF or Type FR flyby, except for the

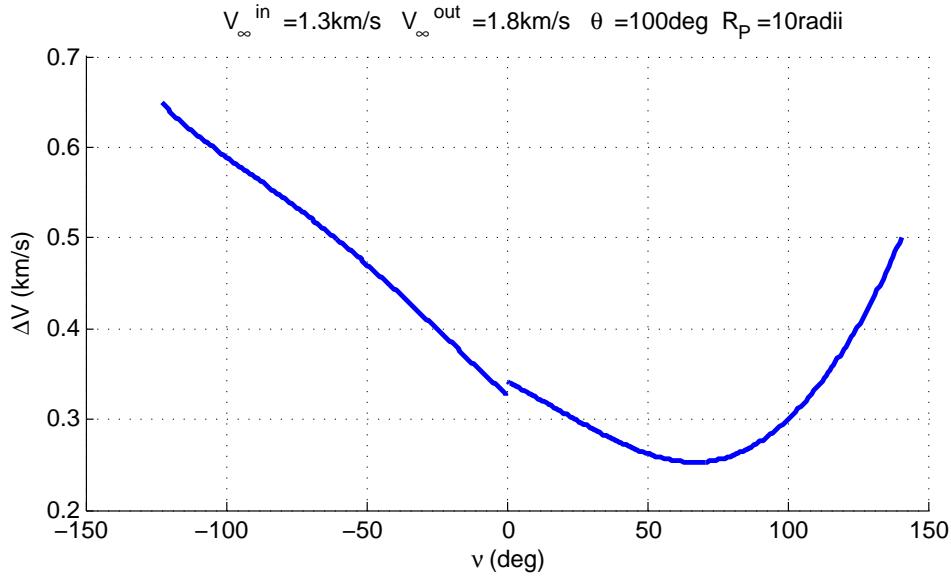


Figure 4.24: Similar example constrained periapse powered gravity assist with minimum  $\Delta V$  maneuver location receding from periapse

limiting case of a maximum natural turning angle  $\Phi_p$  flyby with the maneuver at the  $r_{min}$  periapse. The discontinuity exists because the constrained periapse location is different between the Type RF and Type FR flybys, Figure 4.25. In the constrained flyby diagram, the periapse vector  $r_p$  along the apse line is clearly shown in separate locations for the cases of the outgoing leg being constrained by  $r_{min}$  or the incoming leg being constrained by  $r_{min}$ . The only flyby configuration where either constrained legs share a common periapse vector is at the  $\Phi_p$  flyby configuration case, which is the transition point between the  $r_{min}$  constrained solutions (Type RF and FR) and the potentially unconstrained periapse solutions (Type F).

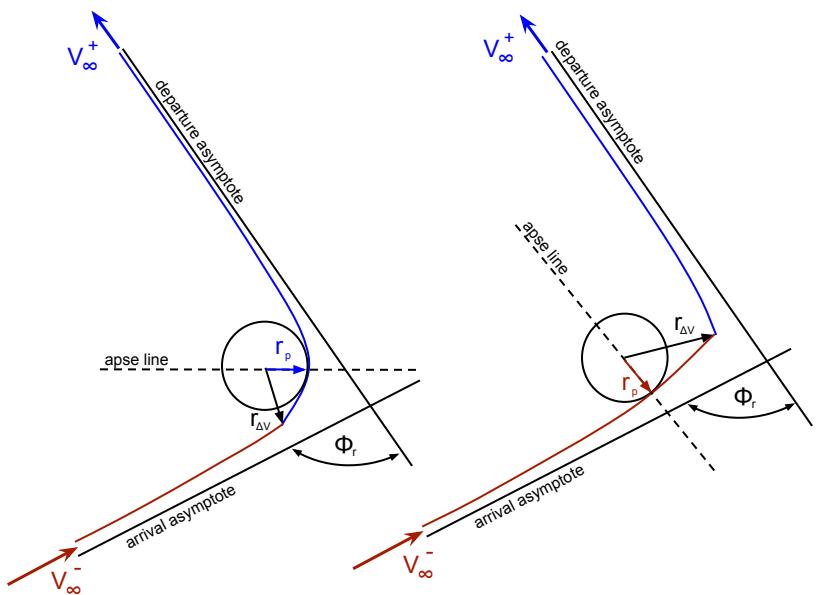


Figure 4.25: Candidate flyby types for constrained powered gravity assist analysis with periapse restriction either applied to the outgoing leg (left) or applied to the incoming leg (right)

#### 4.3.6.2 Strict $r_{min}$ constraint handling for periapse constrained powered flybys

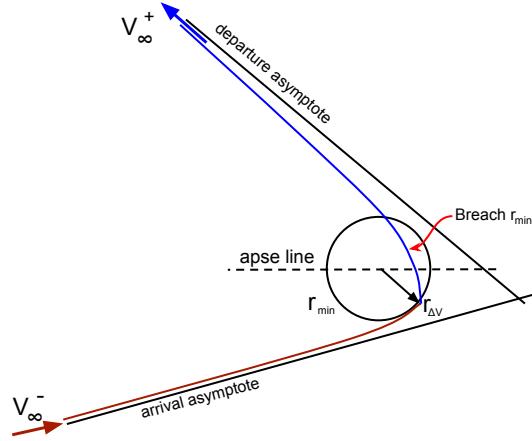


Figure 4.26: Constrained gravity assist infeasible maneuver location of maneuver that breaches  $r_{min}$  constraint

Focusing on flyby configurations with maneuver locations near periapse, it was found that all constrained periapse powered flyby maneuvers at and near periapse will breach the  $r_{min}$  constraint, Figure 4.26. The only exception is again the limiting case of the maximum natural turning angle  $\Phi_p$  flyby configuration. Each of these infeasible unconstrained periapse legs encounter their own periapse after the maneuver, thus allowing the breach to occur. The range of the infeasible maneuver locations about periapse varies by the flyby boundary value problem conditions. Generally, the more extreme the constrained periapse powered flyby configuration, the larger the infeasible maneuver neighborhood near periapse. The diagram in Figure 4.27 demonstrates the infeasible and feasible maneuvers for each set of the analysis.

There exists a minimum feasible maneuver location  $r_{Fmin}$  (or  $v_{Fmin}$ ), where

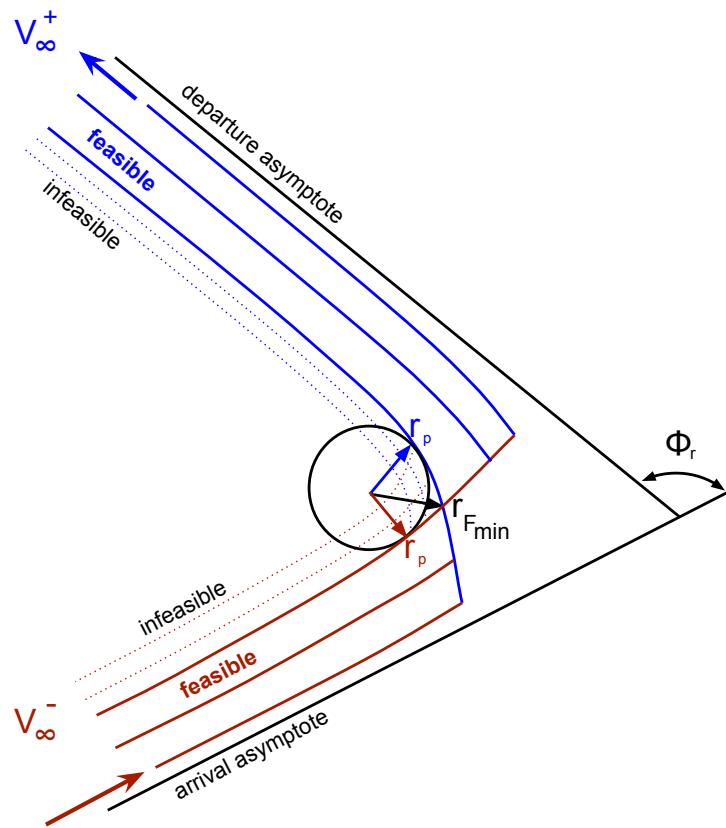


Figure 4.27: Constrained gravity assist depicting the minimum feasible maneuver location near periapse of each flyby solution set

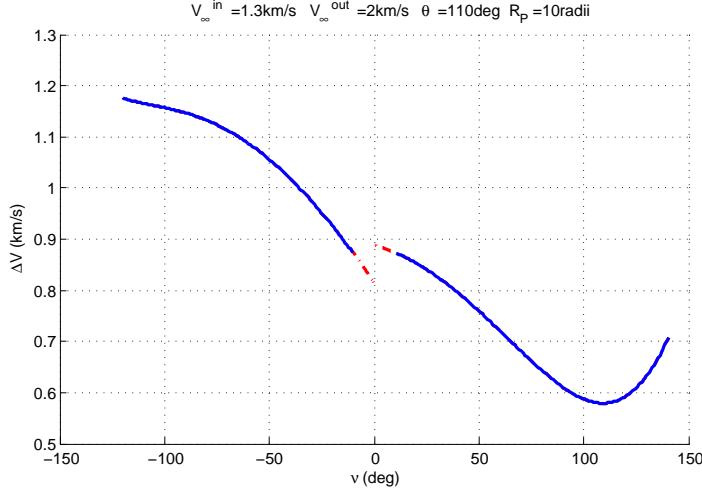


Figure 4.28: Example constrained periapse powered gravity assist reduced solution set with feasible maneuver locations (blue) and infeasible maneuver locations (red) shown

the periapse of both legs is restricted to  $r_{min}$ . This is used as a lower bound for the feasible candidate maneuver locations to accomplish the constrained periapse powered flyby. It can be seen in Figure 4.27 that the minimum feasible maneuver location defined by  $r_{Fmin}$  is a unique trajectory within both sets of analyzed powered flyby solutions, where any lower maneuver location of either set is infeasible. Therefore if exploring only feasible maneuver locations sequentially departing along one constrained leg from the sphere of influence down towards periapse, the minimum feasible maneuver location  $r_{Fmin}$  is the transition condition from one set to the other.

The periapse constrained powered flyby solution spaces are modified to clearly show the infeasible maneuver region near periapse. The infeasible maneuver region is shown in 4.28 as a dotted red line, but in most future plots will simply

be omitted, leaving just empty plot space. Since the minimum feasible maneuver location  $r_{Fmin}$  is the same trajectory in both sets, the associated  $\Delta V$  is the same at this minimum feasible transition point of both sets. Even though the  $\Delta V$  maneuver solution plots show the infeasible gap along true anomaly near periapse, the  $\Delta V$  level will continue through the set transition without discontinuity. This can clearly be seen in the current plot as the end point of the blue feasible line near periapse on the negative  $v_{\Delta V}$  side is the same  $\Delta V$  magnitude as the end point of the blue feasible line on the positive side, picking up where the other left off. The diagram solution space in Figure 4.28 also shows that by increasing the  $V_{\infty}^+$  farther away from  $V_{\infty}^-$  the  $\Delta V$  trough containing the minimum  $\Delta V$  stability point is pushed out towards SOI, far from periapse.

Predetermining the minimum feasible maneuver location  $r_{Fmin}$  for the particular flyby geometry required by the linked conic reduces the solution space range and ensures that any candidate powered flyby investigated always maintains a feasible trajectory above  $r_{min}$ . Both legs of the minimum feasible maneuver flyby trajectory are restricted by  $r_p = r_{min}$ , (i.e., the trajectory must stay at or above  $r_{min}$  on both legs) and the only unknowns are the true anomaly angles of the maneuver location with respect to each periapse,  $v_{Fmin}^-$  and  $v_{Fmin}^+$ . One of the true anomaly angles was solved as a function of the other, reducing the problem to just a single unknown, Equation 4.29.

$$\begin{aligned} r_{min} &= \frac{a_-(1-e_-^2)}{1+e_- \cos v_{Fmin}^-} = \frac{a_+(1-e_+^2)}{1+e_+ \cos v_{Fmin}^+} \\ v_{Fmin}^+ &= \arccos \left( \frac{a_+(1-e_+^2)}{a_-(1-e_-^2)} \frac{(1+e_- \cos v_{Fmin}^-)}{e_+} - \frac{1}{e_+} \right) \end{aligned} \quad (4.29)$$

$$\Phi_r + \pi - v_{\infty}^- = v_{Fmin}^- + v_{Fmin}^+ + v_{\infty}^+$$

$$0 = v_{\infty}^- + v_{\infty}^+ - \Phi_r - \pi + v_{Fmin}^- + \arccos \left( \frac{a_+(1-e_+^2)}{a_-(1-e_-^2)} \frac{(1+e_- \cos v_{Fmin}^-)}{e_+} - \frac{1}{e_+} \right) \quad (4.30)$$

The same periapse transfer angle relation (Equation 4.24) can be used to account for the required flyby turning angle  $\Phi_r$  along side  $v_{Fmin}^-$  and  $v_{Fmin}^+$ , Equation 4.30. Unfortunately when attempting to solve for the root, it was found to be transcendental in  $v_{Fmin}^-$ . An efficient and effective numerical root-finding procedure was implemented. The second order Halley's method [60] was chosen due to its rapid cubic convergence rate and the relatively simple function and derivatives, Equation 4.31. A good initial guess for  $v_{Fmin}^-$  was found to be half of the angle between the periapsides:  $v_{Fmin_0}^- = (v_{Fmin}^- + v_{Fmin}^+)/2 = (\Phi_r + \pi - v_{\infty}^- - v_{\infty}^+)/2$ .

$$f(v_{Fmin}^-) = v_{\infty}^- + v_{\infty}^+ - \Phi_r - \pi + v_{Fmin}^- + \arccos K$$

$$\frac{df}{dv_{Fmin}^-} = 1 + H \frac{e_- \sin v_{Fmin}^-}{\sqrt{1-K^2}}$$

$$\frac{d^2 f}{dv_{Fmin}^{-2}} = \left[ -K \left( \frac{df}{dv_{Fmin}^-} - 1 \right)^2 + H e_- \cos v_{Fmin}^- \right] / \sqrt{1-K^2} \quad (4.31)$$

where,

$$H = \frac{a_+(1-e_+^2)}{a_- e_+(1-e_-^2)}$$

$$K = H (1 + e_- \cos v_{Fmin}^-) - \frac{1}{e_+}$$

#### 4.3.6.3 Optimal transfer maneuvers near SOI boundary

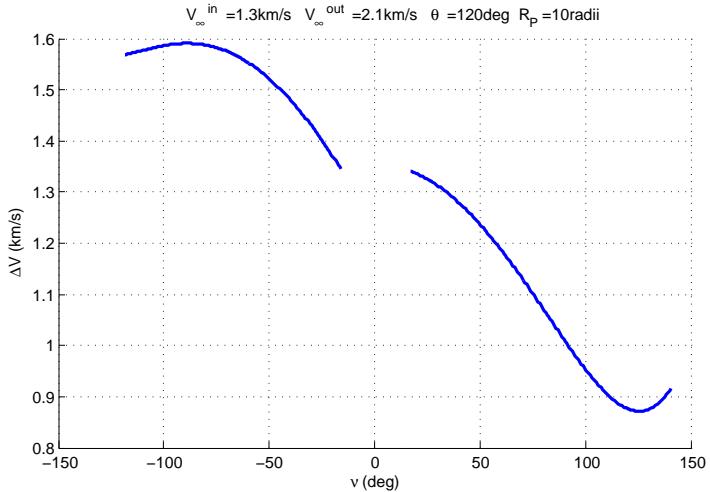


Figure 4.29: Example constrained periapse powered gravity assist with minimum  $\Delta V$  maneuver near sphere of influence  $v_{SOI}$

The analysis is continued by focusing on powered gravity assist  $\Delta V$  maneuver solutions near the SOI boundary. The trough containing the minimum  $\Delta V$  can occur near the maximum allowable maneuver distance, the sphere of influence  $v_{SOI}$ . This occurs when the required turning angle  $\Phi_r$  becomes larger. The trajectory takes advantage of the fact that the velocity decreases as the radius increases and turning a smaller velocity through a given angle is more efficient. An example of this configuration is shown in Figure 4.29, where the minimum  $\Delta V$  is reached just before  $v_{SOI}$ . This set of powered flybys for positive  $v_{\Delta V}$  transfer maneuvers are also shown in Figure 4.30, where the scaled red  $\Delta V$  maneuver vectors become smaller out toward  $r_{SOI}$ . This trend continues until the trough is beyond SOI, yielding a local minimum  $\Delta V$  at the  $v_{SOI}$  boundary, seen in Figure 4.31 and 4.32.

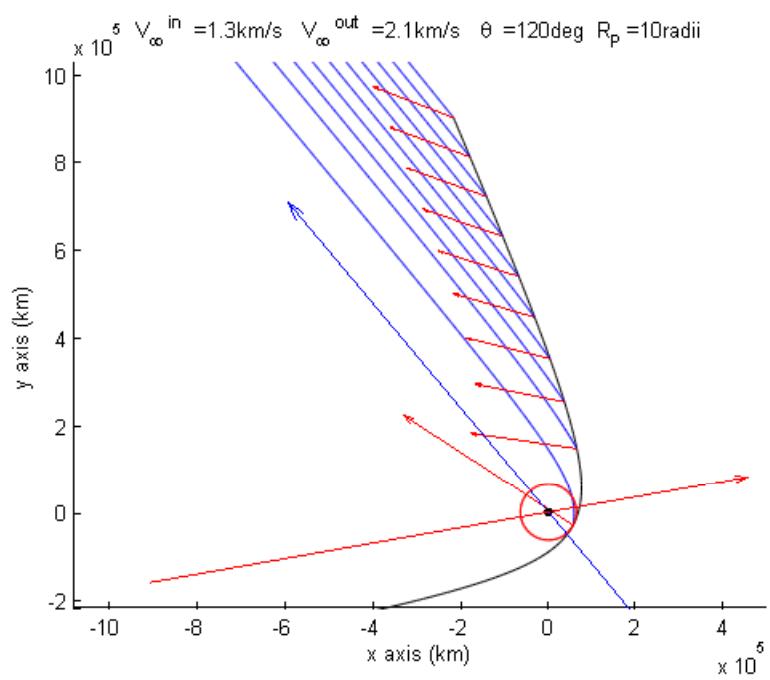


Figure 4.30: Example of candidate constrained peripuse powered gravity assist transfer solutions with minimum  $\Delta V$  maneuver transfer location near sphere of influence  $v_{SOI}$

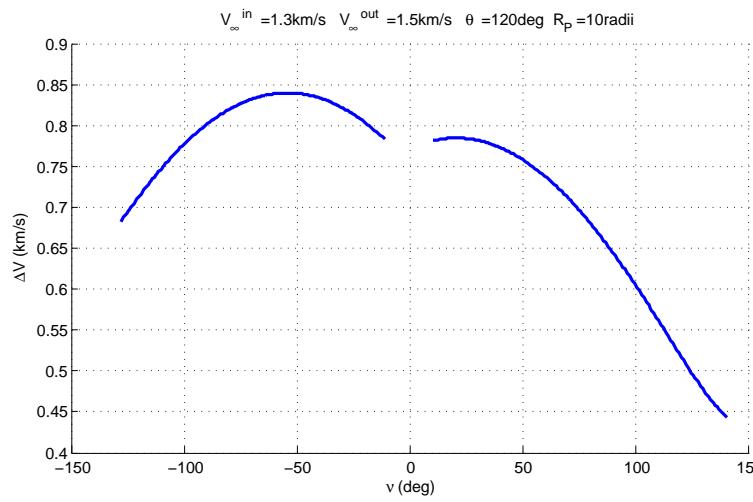


Figure 4.31: Example constrained periapse powered gravity assist with minimum  $\Delta V$  maneuver on the boundary at sphere of influence  $v_{SOI}$

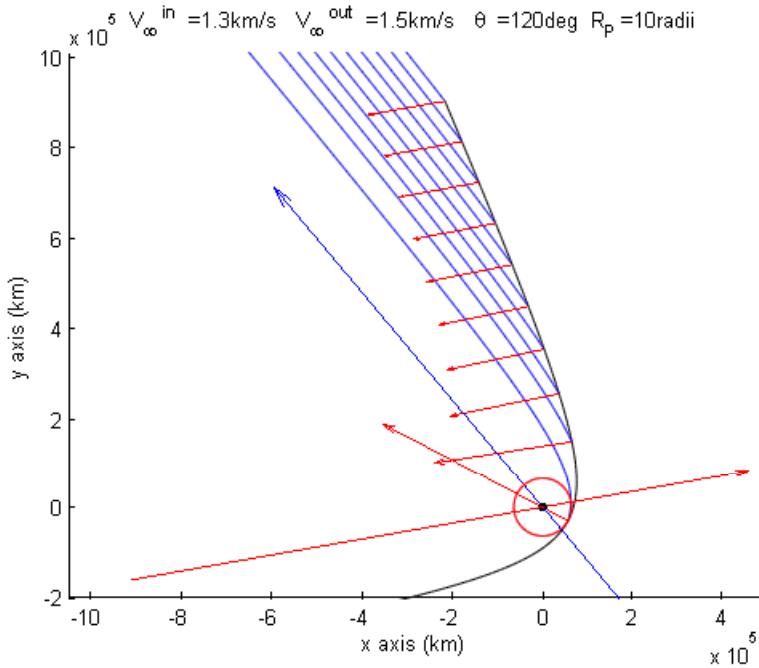


Figure 4.32: Example of candidate constrained periapse powered gravity assist transfer solutions with minimum  $\Delta V$  maneuver transfer location on the boundary at sphere of influence  $v_{SOI}$

The analysis above leads to several observations about the constrained periapse powered gravity assist problems. First, the optimal  $\Delta V$  maneuver for a constrained periapse powered gravity assist may potentially be located very far from periapse, even beyond the planet's SOI. This finding counters statements found in much of literature on this subject, which state that optimal powered gravity assist maneuvers are always located near periapse [7], [55], [61], etc. The incomplete modeling of powered gravity assists has led to many powered gravity assist algorithms to assume the impulsive maneuver location strictly at periapse [31], [53], [62], etc. In some cases the periapse  $\Delta V$  can be drastically larger than the optimal

located farther away. The examples in Figures 4.29 and 4.31 show feasible maneuvers near periapse with  $\Delta V$  values around 70% and 80% larger than the optimal maneuvers near SOI.

A second observation is that many iterative algorithms attempting to search for the optimal powered gravity assist maneuver location use the periapse as the initial guess for the maneuver location[7],[55], [42], etc. A few authors have hinted at the potential benefit of a gravity assist with a maneuver at SOI or beyond [63, 58, 57], but none of these studies have reported a characterization of the powered gravity assist cases that result in a  $\Delta V$  maneuvers at or near SOI.

#### 4.3.6.4 Multimodal solution space dynamics

In this section, the multimodal nature of the solution space is discussed further and analyzed for many more cases by extruding the 2D plot types shown in the prior section into 3D surface plots. The 3D surface plots show multiple different  $\Delta V$  maneuver solution lines for varying outgoing  $V_\infty^+$  magnitudes as the extra dimension. The goal of this section is to characterize what types of flyby configurations do the constrained periapse powered gravity assist solution space exhibit multimodal traits. The analysis also identifies the general location of the optimal transfer maneuver for each flyby configuration types. The different flyby configuration types defined at the beginning of the chapter will be used heavily in this section in order to characterize and clearly identify the solution types. As a reminder, the Type RF flyby indicates a flyby with the incoming leg's periapse constrained at  $r_{min}$  and a  $\Delta V$  transfer maneuver occurring after the periapse. In a reverse manner, a Type FR flyby indicates a flyby with the outgoing leg's periapse constrained at  $r_{min}$  and a  $\Delta V$

transfer maneuver occurring before reaching periapse. Also, the constrained flyby  $\Phi_r > \Phi_p$  types are displayed again now for reference below. Each of these types can also be referred to with the Type RF or FR nomenclature, such as Type RF-1, Type FR-2, Type RF-T, etc. The characteristics for each type will be described in detail along with the corresponding 3D solution space plot depicting the locus of optimal  $\Delta V$  maneuver locations for each of the varying  $V_\infty^+$  flyby configurations.

$$\Phi_r > \Phi_p : \begin{cases} V_\infty^- < V_\infty^+ & \begin{cases} \text{Type 1 : } (\Phi_r > \delta_{max}^-) \text{ and } (\Phi_r > \delta_{max}^+) \\ \text{Type T : } (\Phi_r = \delta_{max}^-) \text{ and } (\Phi_r > \delta_{max}^+) \\ \text{Type 2 : } (\Phi_r < \delta_{max}^-) \text{ and } (\Phi_r > \delta_{max}^+) \end{cases} \\ V_\infty^- > V_\infty^+ & \begin{cases} \text{Type 1 : } (\Phi_r > \delta_{max}^-) \text{ and } (\Phi_r > \delta_{max}^+) \\ \text{Type T : } (\Phi_r > \delta_{max}^-) \text{ and } (\Phi_r = \delta_{max}^+) \\ \text{Type 2 : } (\Phi_r > \delta_{max}^-) \text{ and } (\Phi_r < \delta_{max}^+) \end{cases} \end{cases} \quad (4.32)$$

A second important characteristic found from the analysis of optimal flyby maneuvers near the SOI boundary is that the solution space near periapse can incur a change in gradient direction due to the emergence of a local maximum. In other words, conditions exist that produce a multimodal constrained periapse powered gravity assist solution space, as shown in Figure 4.33. A multimodal solution space can create ambiguity and uncertainty in the resulting flyby solution, depending on the optimization procedure used. In particular, a gradient based algorithm such as Newton's method may be driven toward a local minimum/boundary solution, while the global minimum solution occurs elsewhere. The following analysis shows that the multimodal features maintain a consistent polynomial "direction" or "shape", where a local maximum stability point ( $\frac{d\Delta V}{dv} = 0$ ) only arises near periapse whenever the global minimum exists further out toward SOI or beyond.

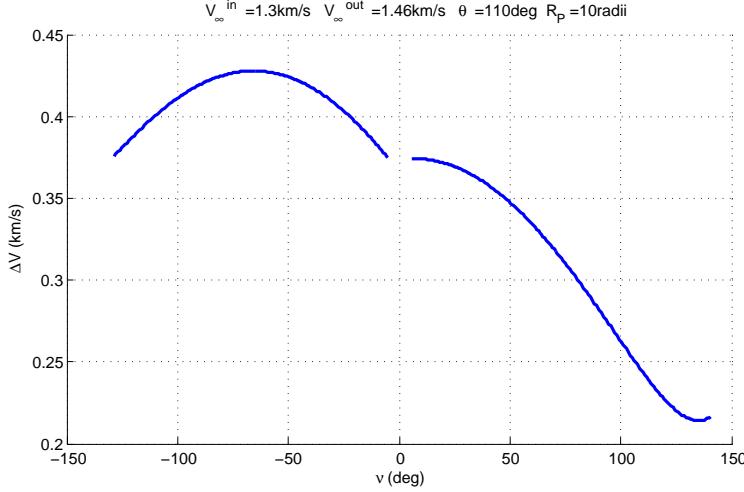


Figure 4.33: Example constrained periapse powered gravity assist multimodal solution space depicting both a maximum and minimum  $\Delta V$  maneuver locations near periapse and sphere of influence respectively

The breadth of the analysis is improved by inspecting an additional variable with 3D surface plots. While holding a fixed flyby  $V_{\infty}^-$  and  $\Phi_r$ , the maneuver location  $v_{\Delta V}$  and  $V_{\infty}^+$  are varied then visualized by the associated  $\Delta V$  maneuver. A family of constrained periapse powered flyby Type RF solutions with  $r_{min} = 1.5$  radii about Earth is analyzed. The range of exiting  $V_{\infty}^+$  is limited to only those associated with a constrained periapse gravity assist ( $\Phi_r > \Phi_p$ ) while still greater than  $V_{\infty}^-$ , isolating only the optimal maneuver location of positive  $v_{\Delta V}$  to be investigated. The example solution family is shown first with the lower flyby configuration energy (lower  $V_{\infty}^-$  and  $V_{\infty}^+$ ) maneuver location near periapse seen in the top two plots of in Figure 4.34. The left side plots are overhead views and the right side plots are 3D perspective views of the surface plot. For the fixed  $V_{\infty}^-$  and  $\Phi_r$ , the  $\Delta V$  magnitude and maneuver location increases with  $V_{\infty}^+$ . Again, the color gradient of the

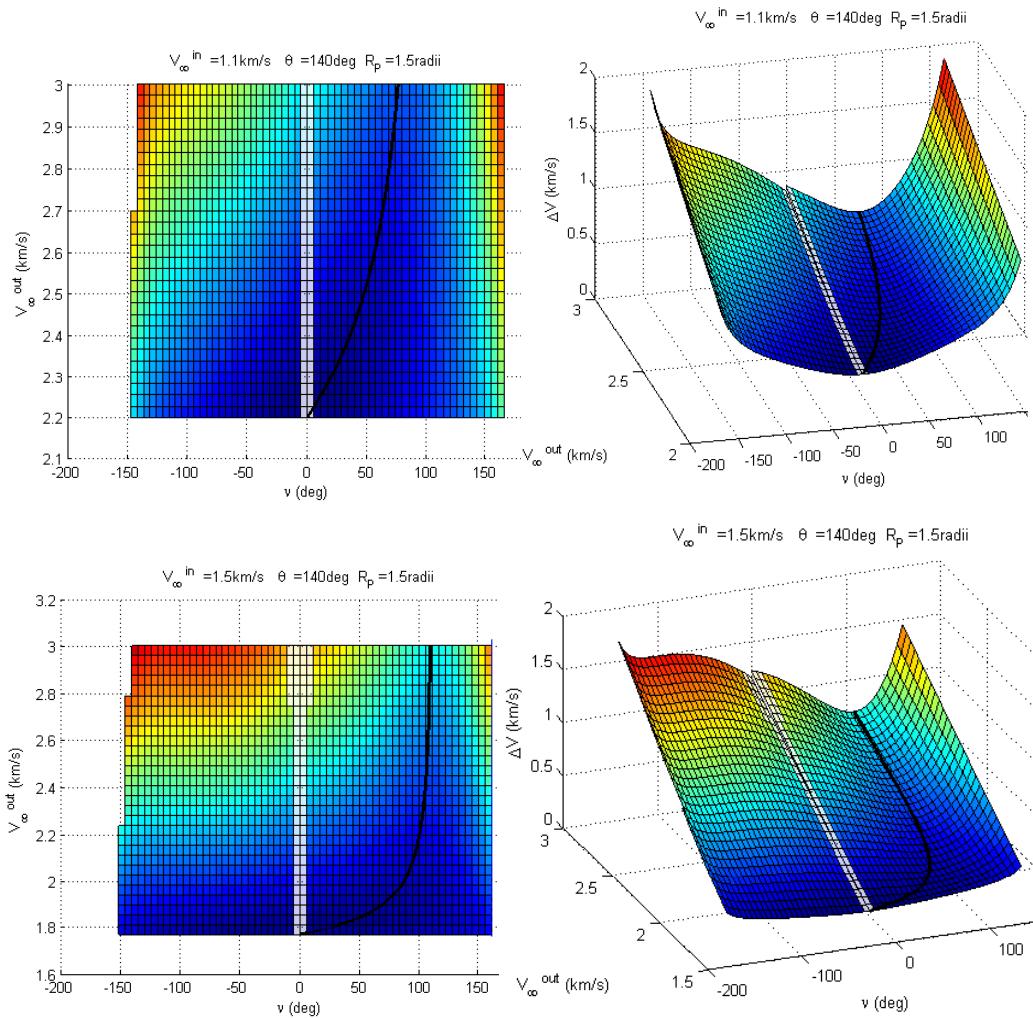


Figure 4.34: Family of Type RF-2 constrained periape powered gravity assists of  $V_{\infty}^- = 1.1 \text{ km/s}$  case (top)  $1.5 \text{ km/s}$  case (bottom) with optimal maneuver location locus for each exiting  $V_{\infty}^+$  shown along black line near periape and outward (overhead views on left and 3D view on right)

surface plots displays higher  $\Delta V$  magnitudes in the red spectrum regions and lower relative  $\Delta V$  magnitudes in the blue spectrum regions.

The diagram also verifies the predetermined optimal  $\Delta V$  maneuver neighborhood and constrained leg, which is the  $V_\infty^-$  hyperbolic leg (because  $V_\infty^- < V_\infty^+$  for the example cases investigated). The diagram displays the feasible solution space with the colored surface plot, while the infeasible portion of the solution space surrounding periapse is made transparent. The solution space and locus of optimal maneuver locations bends farther from periapse as the  $V_\infty^-$  is increased. The set of solutions that include maneuvers near periapse are Type 2 (specifically Type RF-2): the constrained leg's turning angle  $\delta > \Phi_r$ , while the unconstrained leg's  $\delta < \Phi_r$ . The constrained Type RF-2 maneuvers (Figure 4.34) for lower flyby configuration energies (lower  $V_\infty^-$  and  $V_\infty^+$ ) occur near periapse in order to reduce the over-turning effect of the constrained leg's larger  $\delta^-$ , while taking advantage of the more efficient maneuvers at high velocities. In this and all other Type RF-2 flybys exhaustively explored, only a single stability point exists (the global minimum) among the candidate maneuver locations along the predetermined constrained incoming leg.

This trend of optimal maneuver locations occurring near periapse for lower  $V_\infty^+$  flybys and moving away from periapse for higher  $V_\infty^+$  flyby configurations continues with increasing  $V_\infty^-$  until the associated  $\delta^-$  matches the required  $\Phi_r$ , where the solution space transitions from Type 2 periapse maneuvers to Type 1 (specifically Type RF-1) SOI maneuvers for the lower flyby configuration energies (lower  $V_\infty^-$  and  $V_\infty^+$ ). At this Type T transition condition, the optimal  $\Delta V$  maneuver is performed at a point between periapse and SOI, Figure 4.35, balancing benefits of increased turning angle contribution at  $v_{\Delta V}$  farther out from periapse and maneuver

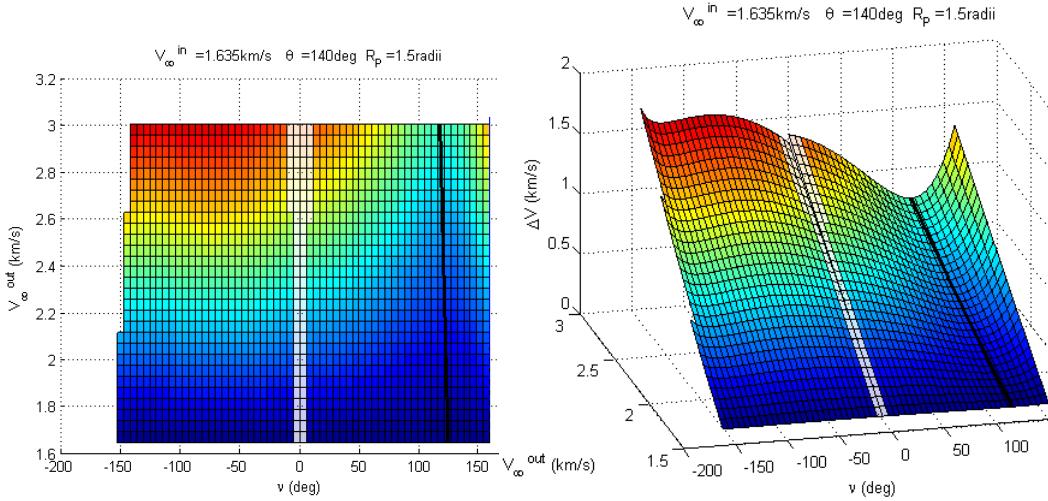


Figure 4.35: Family of Type RF-T constrained periapse powered gravity assists with optimal maneuver location locus shown at transition condition at  $V_{\infty}^- = 1.635$  km/s, where flyby turning angle is nearly that of the constrained leg (overhead views on left and 3D view on right)

efficiency closer to periapse. Past this transition condition are Type RF-1 flybys, where the  $\Delta V$  maneuver location for lower outgoing  $V_{\infty}^+$  values becomes more efficient when performed at SOI or beyond, transiting most of the incoming hyperbola, Figure 4.36. In constrained Type 1 flybys, both hyperbolic legs  $\delta^{\pm}$  are insufficient to complete the required  $\Phi_r$ . Therefore the optimal powered gravity assists at lower energy flyby configurations fully utilizes the maximum available hyperbola turning angle and performs the  $\Delta V$  maneuver at SOI or beyond. In this configuration, the benefits of larger turning angle contribution for SOI maneuvers coupled with the efficiency of changing direction at slower velocities outweighs the energy change efficiency at higher velocities closer to periapse. It can be seen in the present example (4.36) that only Type 1 flybys exhibit multiple stability points and in the same configuration: a local maximum near periapse and the global minimum near SOI or

beyond. Through exhaustive analysis it is found that not all Type 1 flybys have the multimodal features, but all by definition have the global minimum  $\Delta V$  maneuver located farther from periapse, potentially near SOI or beyond.

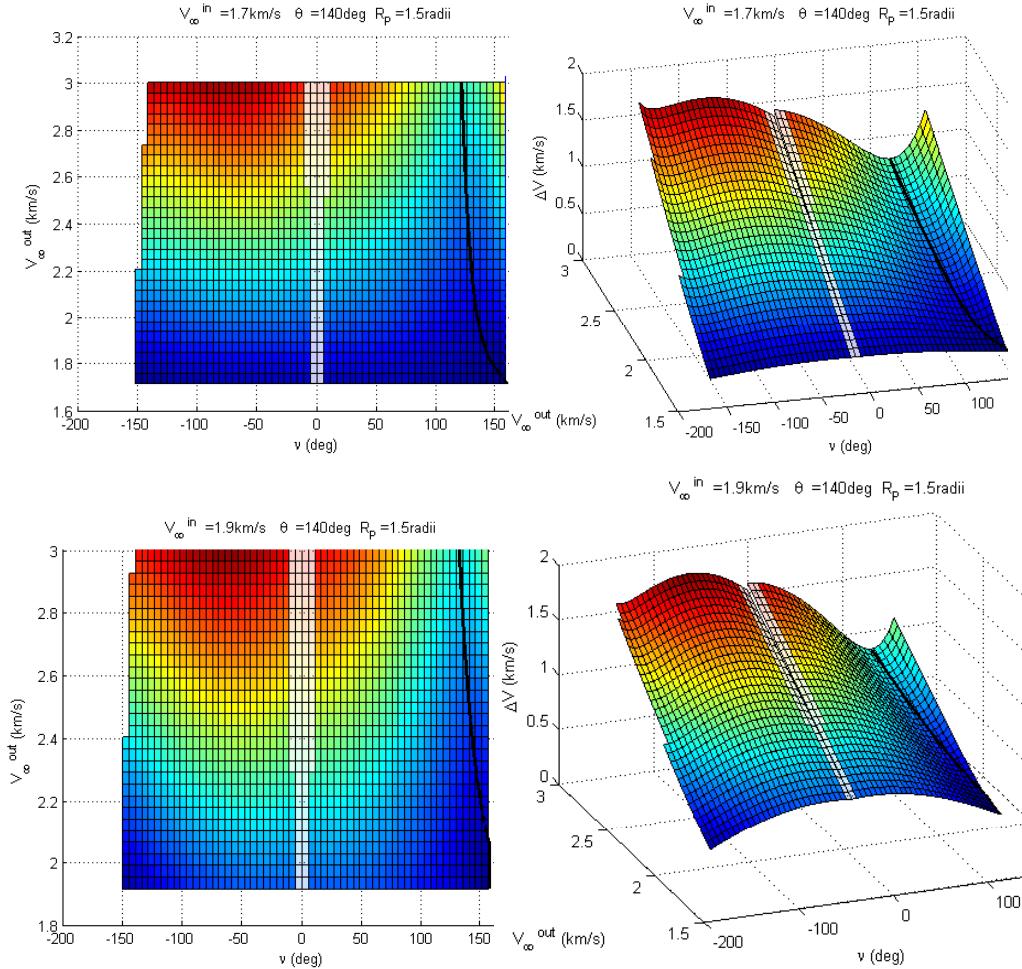


Figure 4.36: Family of Type RF-1 constrained periapse powered gravity assists of  $V_{\infty}^- = 1.7$  km/s cases (top) and 1.9 km/s cases (bottom) with optimal maneuver location locus shown from beyond SOI and inward as  $V_{\infty}^+$  increases (overhead views on left and 3D view on right)

The same span of constrained periapse powered gravity assist families is

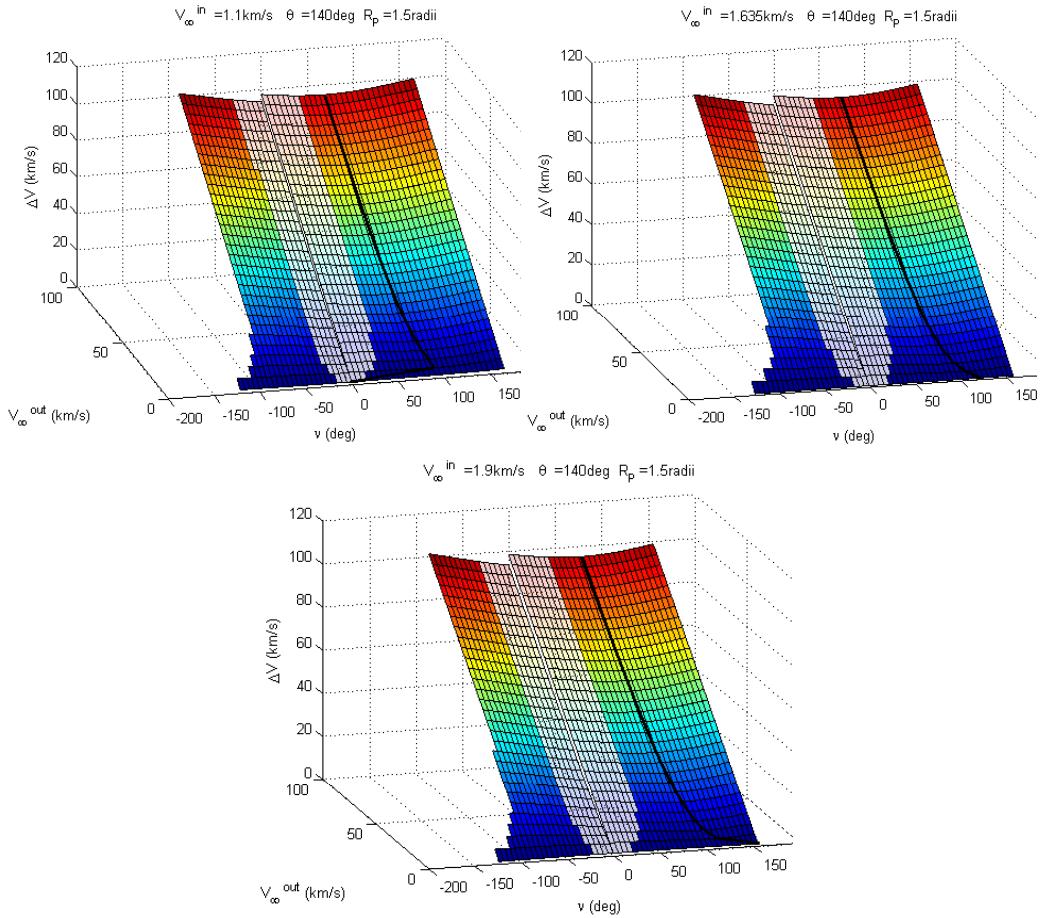


Figure 4.37: Each family of constrained periapse powered gravity assist's locus of optimal solutions asymptotically stagnates at  $\Delta V$  maneuver location for increasingly extreme flyby configurations, clockwise from bottom: Type RF-1, Type RF-2, and Type RF-T

extended to much larger  $V_{\infty}^+$  values in Figure 4.37. The previous range of results can still be identified at the lower end of the plots. The more extreme flyby configurations reveals a less dynamic region of the solution space. The locus of optimal maneuvers is driven toward a maneuver location asymptote seen in Figure 4.37, where exceedingly high energy flyby configurations would have nearly the same maneuver location. As the unconstrained leg's  $V_{\infty}^+$  increases for a given  $V_{\infty}^-$  and  $\Phi_r$ , the higher energy exiting hyperbolas flatten out in shape, becoming more and more like a straight line flyby. These high energy legs don't contribute very much turning angle to the overall flyby. The optimal maneuver location is a balance between the velocities at the maneuver location being more aligned at higher maneuver distances and the efficiency of velocity change at lower distances. With a fixed available turning angle  $\delta^-$  and limited  $\delta^+$  to accomplish  $\Phi_r$ , all the high energy flyby configurations for each family converge upon the similarly balanced maneuver locations. Despite the widely varying trends at lower flyby configuration energies, each type of flyby has very similar high energy locus, where the maneuver location asymptote is simply shifted to slightly higher  $v_{\Delta V}$  (toward  $v_{SOI}$ ) as  $V_{\infty}^-$  increases. This trend continues due to the decreasing available  $\delta^-$  until the locus of all powered flyby maneuver locations are bounded by the SOI. The asymptotic and consistent behavior of these maneuvers at higher flyby configuration energies ensures that no additional analysis is necessary for higher and higher flyby configuration energies ad infinitum.

#### 4.3.6.5 General solution space dynamics

Collecting the results of prior sections, the solution space behavior can be mapped to key characteristics and leveraged for an efficient and robust constrained periapse powered gravity assist solution algorithm. The main results learned from the previous sections are outlined in the following list for clarity before moving forward in the analysis.

1. The powered gravity assist optimal transfer maneuver location may occur far from periapse for the minimum  $\Delta V$  solution and may be a bounded minimum solution at the  $r_{SOI}$  boundary constraint
2. The infeasible transfer maneuver locations always exist for constrained powered flybys and occur in the region about periapse, where the resulting flyby trajectories breach the  $r_{min}$  constraint.
3. The  $\Delta V$  solution space of the constrained powered gravity assist can be multimodal, where a maximum  $\Delta V$  stability point can occur near periapse while a minimum  $\Delta V$  stability point is location farther out towards  $r_{SOI}$  or beyond (if it exists at all)

Applying the linked conic methodology, the acceptable range of  $\Delta V$  maneuver locations for a constrained periapse powered gravity assist is between periapse and SOI. It has been shown that the minimum  $\Delta V$  maneuver rarely occurs exactly at periapse. This behavior allows the dynamics of the solution space to naturally help enforce the periapse boundary constraint. In contrast, the SOI boundary is a much more active constraint. The previous investigation identified the consistency of the

solution space mode configuration; if a maximum exists it would always lie near periapse, while the minimum exists as a stability point toward SOI or as a bounded minimum at the SOI boundary. Furthermore, when an interior minimum solution exists, the derivative  $\frac{d\Delta V}{dv}$  or  $\frac{d\Delta V}{dr}$  at the SOI boundary has a predictable sign (+ for Type RF and – for Type FR), while the periapse location is less reliable. In this manner the derivative at SOI is a key indicator if an interior minimum solution exists.

In preliminary trajectory design and general numerical trajectory optimization , it is likely that many iterates of the process may generate flyby configurations for the Type 1 SOI boundary solutions. As depicted earlier, these Type RF-1 or FR-1 SOI boundary solutions occur for more energetic flyby configurations, such as large  $\Phi_r$  and relatively high  $V_\infty^\pm$ , where  $\delta^\pm < \Phi_r$ . An example of the derivative  $\frac{d\Delta V}{dv}$  at SOI is presented in Figure 4.38, where only the derivative’s  $\Delta V$  component is shown vertically to emphasize the scale and direction. A Type RF-2 trajectory is displayed (top left) with all positive SOI derivatives and interior locus of optimal solution locations as expected, while a higher energy Type RF-1 flyby with higher  $V_\infty^\pm$  and  $\Phi_r$  (top right) has all negative derivatives with SOI boundary solutions also shown. An intermediate case with the same high  $V_\infty^\pm$  but lower  $\Phi_r$  is also provided (bottom left and right) where the lower energy flyby configurations have SOI boundary solutions and the derivative smoothly transitions to positive as the interior solutions appear. An alternate zoomed perspective view of the intermediate case is displayed near the SOI boundary where the  $\frac{d\Delta V}{dv}$  derivative direction switches from negative (indicating an SOI boundary solution) to positive (indicating that the optimal maneuver location for a minimum  $\Delta V$  transfer occurs interior of the SOI

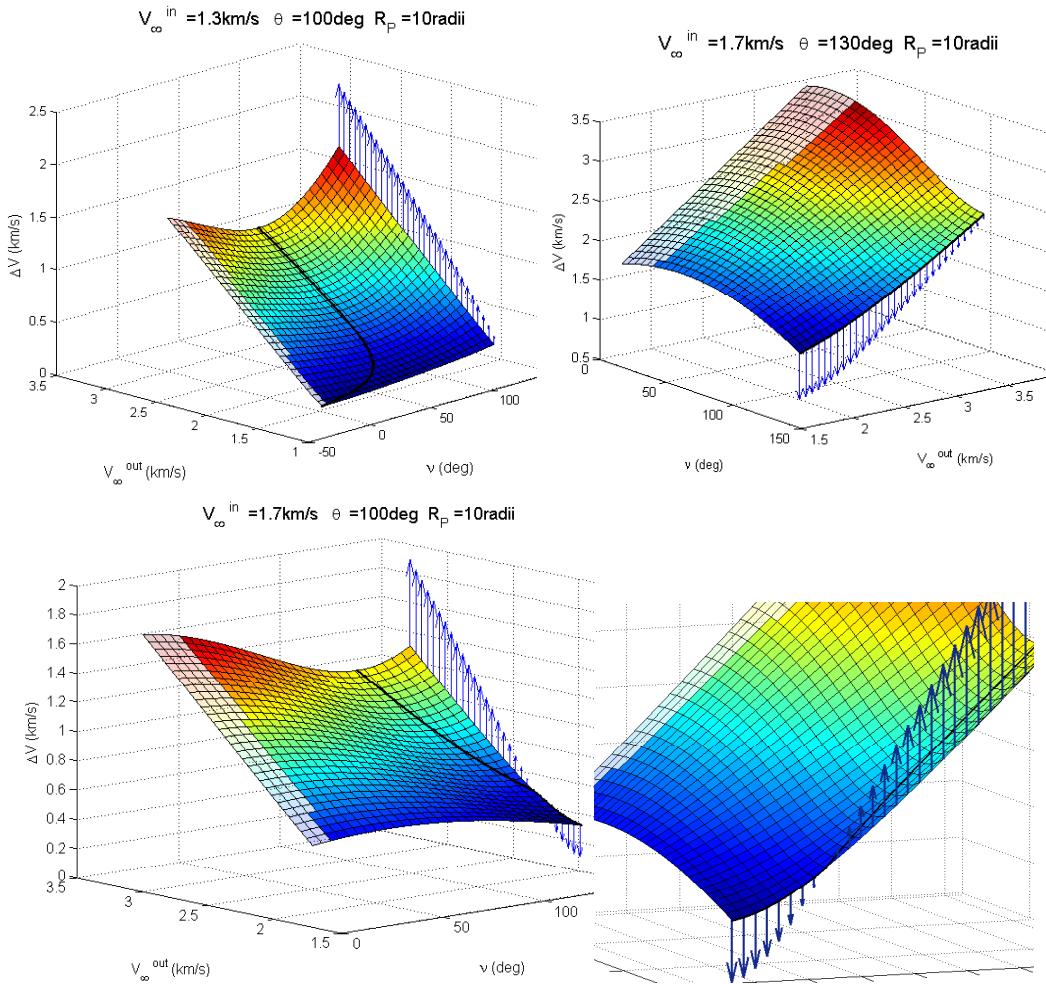


Figure 4.38: Example Type RF reduced solution spaces with SOI derivative  $\frac{d\Delta V}{dv}$  depicted vertically along  $\Delta V$  direction with relative scale, displaying all interior and all boundary solutions (top left and right), as well as a transitioning case from boundary to interior solutions (center) with an additional focused perspective (bottom).

boundary). The term “interior” references a point inside the SOI boundary, as any point between periapse and SOI.

If an interior solution is known to exist for a general constrained periapse powered gravity assist, some factors that can impact a numerical optimization procedure are avoiding the infeasible region near periapse that break the  $r_{min}$  constraint, as well as avoiding the suboptimal gradient-based search directions that can arise near periapse due to the multimodal nature of the solution space. If drawn into this periapse region of a multimodal solution space, the gradient will drive the numerical solution to a local minimum bounded at periapse, while the global minimum lies farther out towards SOI. In all of these solution space examples, the infeasible region near periapse changes with  $V_\infty^\pm$ ,  $\Phi_r$ , and  $r_{min}$  for a selected encounter planet. As the flyby configuration becomes more energetic by increasing its  $V_\infty^\pm$  and/or  $\Phi_r$ , etc., the infeasible region also increases, pushing the minimum feasible maneuver location farther away from periapse. Changes in the infeasible region are shown in Figure 4.39 using higher  $V_\infty^\pm$ , representative of typical Earth flybys performed in past interplanetary missions. The solution spaces in the figure have progressively increased infeasible region (from left to right) when  $V_\infty^\pm$  is increased, because of the reduced  $\delta^\pm$  turning angle capability. The top two solution spaces plots in the figure show the  $V_\infty^-$  increased from 10 km/s to 15 km/s (top left to right), pushing the locus of optimal maneuver locations out to the SOI boundary. Then the infeasible region is also expanded by increasing the required encounter turning angle  $\Phi_r$  from 30° to 42° (top right to bottom left).

Finally, a larger  $r_{min}$  forbidden sphere increases the infeasible region as well, due to its effects on the available turning angle  $\delta^\pm$ , increasing  $r_{min}$  from 1.1 to

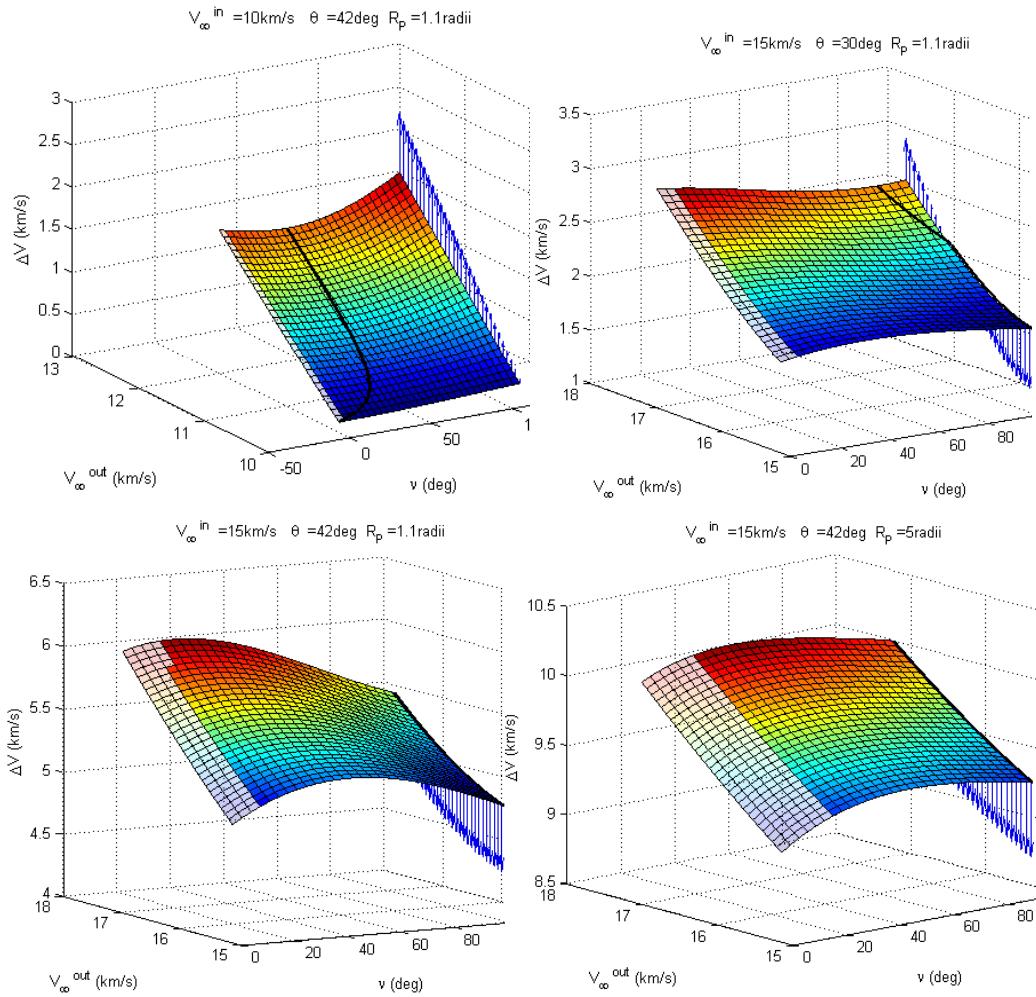


Figure 4.39: Examples of increasing infeasible regions for Earth Type RF flybys with varying flyby configurations (top left to right: increasing  $V_{\infty}^{in}$ , top right to bottom left: increasing  $\Phi_r$ , and bottom left to right: increasing  $r_{min}$ )

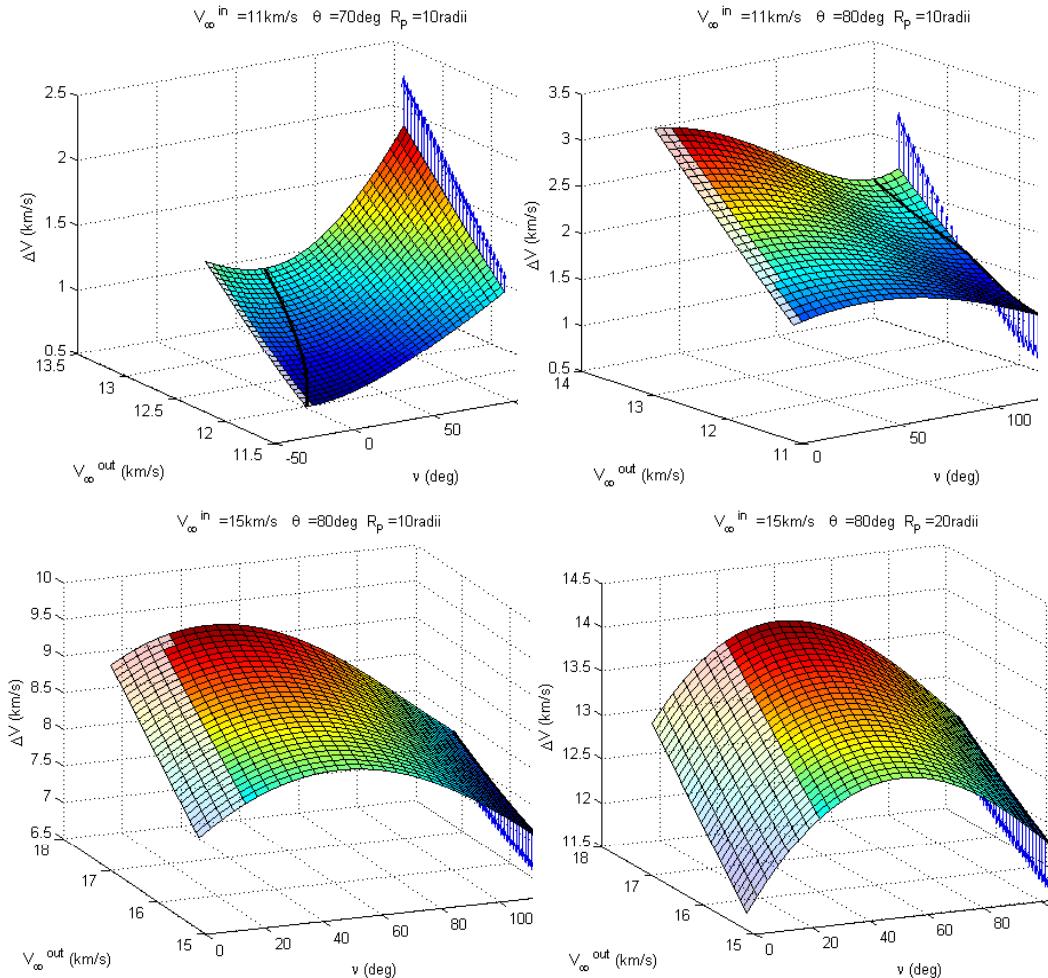


Figure 4.40: Examples of increasing infeasible regions for Jupiter Type RF flybys with varying flyby configurations (top left to right: increasing  $\Phi_r$ , top right to bottom left: increasing  $V_{\infty}^-$ , and bottom left to right: increasing  $r_{min}$ )

5 Earth radii (bottom left to right). Similar relationships are demonstrated for a Jupiter encounter at typical past mission  $V_{\infty}^{\pm}$  levels in Figure 4.40. For the Jupiter encounters figure, the top plots increase the infeasibility region and move the locus of optimal maneuver locations to the SOI boundary as  $\Phi_r$  increases from  $70^\circ$  to  $80^\circ$  (top left to right). The infeasibility region is further increased by increasing  $V_{\infty}^-$  from 11 km/s to 15 km/s (top right to bottom left). The bottom row shows an even larger increase in the infeasibility region by increasing the encounter  $r_{min}$  requirement from 10 to 20 Jupiter radii (bottom left to right). This also demonstrates that these trends and characteristics can be translated to many other planetary flyby scenarios, just with different relative flyby configurations.

The behavior of the infeasible region matches that of the optimal maneuver location, where the minimum  $\Delta V$  trough shifts away from periapse as the flyby configuration becomes more energetic as described above. The same behavior occurs with the maximum crest location, shifting according to the flyby configuration parameters relative values. In effect, all modes transition with the solution space toward SOI as the flyby configuration requirements become more difficult. These relationships are displayed throughout these analyses and can be seen in the provided solution space examples.

A common presumption for impulsive transfers is that the most efficient impulse direction is along the velocity vector, tangential with the direction of motion, or that such a condition/location is at least close to the optimal solution. A few example analyses are provided to compare the collinear maneuver for a constrained periapse powered gravity assist and the global minimum  $\Delta V$  maneuver. The examples presented are typical mission flyby configurations for Earth flybys

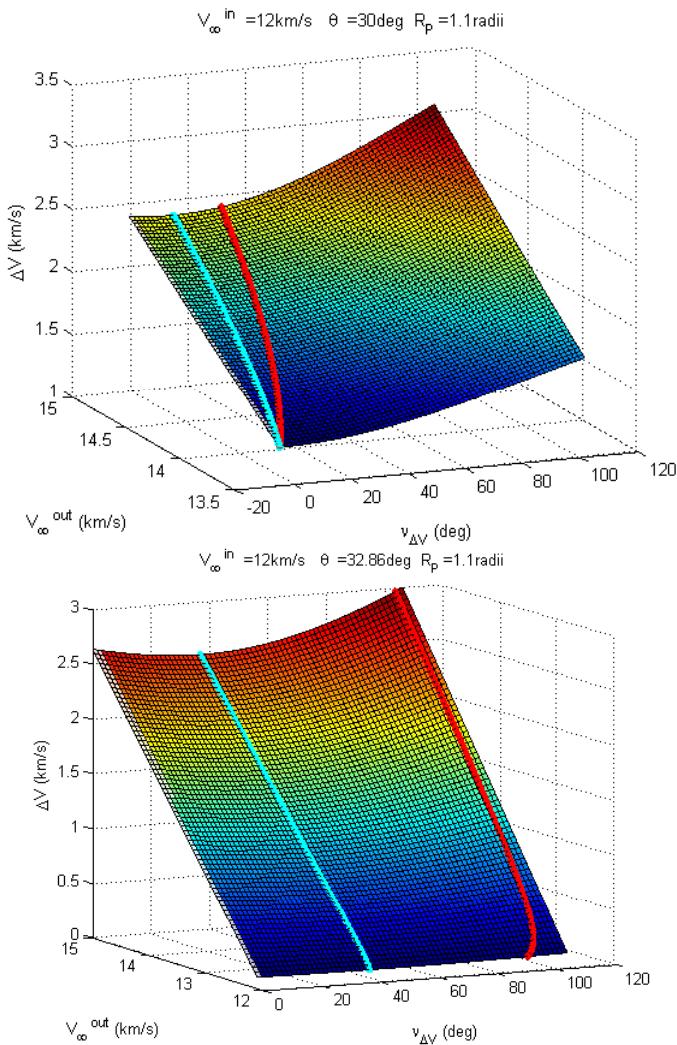


Figure 4.41: Constrained powered gravity assist solution space with minimum  $\Delta V$  maneuver locus (cyan) and collinear  $\Delta V$  locus (red) for Earth encounters

and Jupiter flybys due to their popular use in gravity assist missions. The example flybys in Figure 4.41 show the locus of tangential maneuvers (red) alongside the non-tangential optimal  $\Delta V$  maneuver locus (cyan). Just as with the previous 3D solution space plots, the vertical axis displays the  $\Delta V$  magnitude for each corresponding  $V_\infty^+$  and transfer maneuver location  $v_{\Delta V}$ . Similar to the unconstrained periapse cases, the collinear maneuvers are always sub-optimal compared to the non-tangential minimum  $\Delta V$  solutions. The collinear maneuvers of some flyby configurations do offer a close estimate to the optimal  $\Delta V$  and maneuver location. Contrastingly, the second graph of the Earth encounters in Figure 4.41 show a vast difference between the minimum  $\Delta V$  and the collinear maneuver. A key difference between the unconstrained and constrained tangential flyby maneuver solutions is that the unconstrained periapse case has an extra degree of freedom (periapse radius  $r_p$ ) which potentially allows for multiple solutions for any particular flyby boundary value problem configuration. Yet, the constrained flyby BVP only allows for a single unique collinear  $\Delta V$  maneuver solution, if one exists at all. The periapse constrained collinear maneuver solution locus tends to follow the behavior of the optimal locus, starting near periapse and passing outward toward SOI for Type 2 flybys. As the Type 2 flyby optimal maneuver locus shifts farther toward SOI for varying flyby configurations, so too does collinear maneuver location, until reaching the SOI boundary. It was found that the collinear maneuver location consistently occurs farther from periapse than the optimal maneuver location, and therefore reaches the SOI boundary first.

Once a collinear maneuver reaches the SOI boundary, any higher energy flyby configuration would push the tangential maneuver location past the SOI fea-

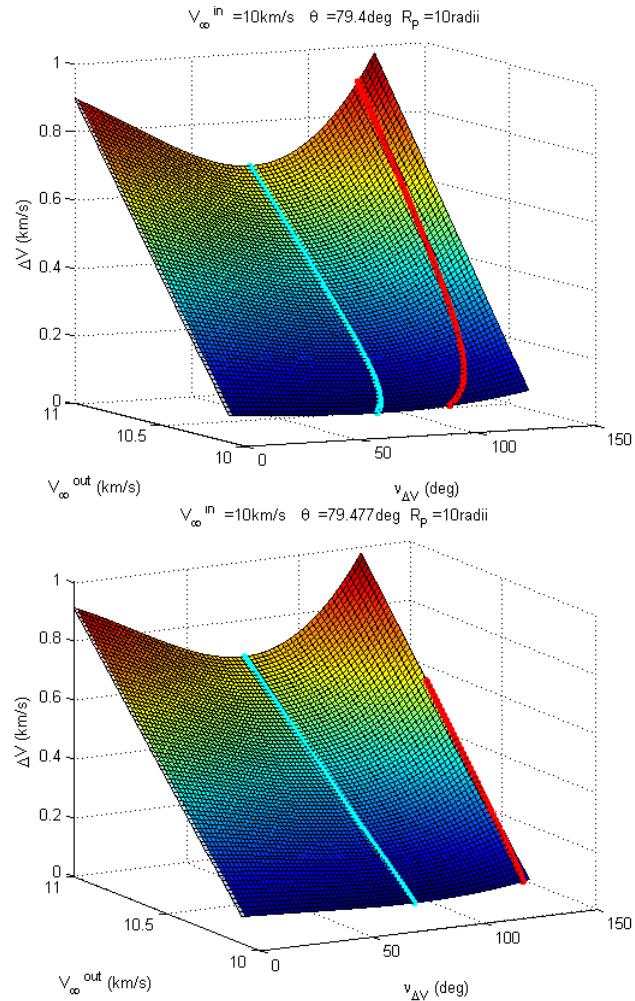


Figure 4.42: Constrained powered gravity assist solution space with minimum  $\Delta V$  maneuver locus (cyan) and collinear  $\Delta V$  locus (red) for Jupiter encounters

sible boundary or a collinear solution may not even exist. This transition can be seen with the Jupiter encounter example in Figure 4.42. The Type RF-2 Jupiter powered gravity assist has the collinear maneuver locus extended near the SOI boundary in the first graph. The collinear solutions are pushed beyond the SOI boundary in the second graph, leaving some flyby configurations without a collinear maneuver option in the feasible region. Considering the potential inability to provide a flyby maneuver solution, coupled with its suboptimal  $\Delta V$  maneuvers, the collinear powered gravity assists were not considered for further investigation. With this and previous section's studies on collinear maneuvers and optimal minimum  $\Delta V$  maneuvers during powered gravity assists, it is the authors intent to clarify their occasional similarities and their stark differences.

#### **4.3.7 Optimal periapse constrained powered gravity assist solution algorithm**

##### **4.3.7.1 Implementation of optimal periapse constrained powered gravity assist method**

The preceding analysis and results were used to develop an efficient and robust constrained periapse powered gravity assist algorithm to determine the optimal maneuver location for the minimum  $\Delta V$  maneuver. Although an analytic solution is the most desirable, the complexity of the boundary value problem yields transcendental  $\Delta V$  and derivative functions with respect to maneuver location  $r_{\Delta V}$  or  $v_{\Delta V}$ , Equations 4.26 and 4.27. The functions are also of higher degree, leading to multiple roots to be investigated and sorted through if attempting to solve in an analytic manner. Therefore an iterative numerical optimization procedure is required to determine the minimum  $\Delta V$  magnitude and maneuver location. Similar to the previous sections, the following analysis assumes a Type RF flyby (incoming flyby

leg constrained by  $r_{min}$  and  $\Delta V$  maneuver after periapse), but the results are easily converted to Type FR (outgoing flyby leg constrained by  $r_{min}$  and  $\Delta V$  maneuver before periapse) by the problem's reversibility.

Some of the most popular and efficient numerical optimization methods incorporate the function as well as its derivative/gradient in the procedure. The addition of slope information for a one dimensional optimization problem enhances the search direction and step size toward the minimum [45]. Though the analytic function derivative ( $\frac{d\Delta V}{dr}$  or  $\frac{d\Delta V}{dv}$ ) is known, it is worth considering if the numerical derivative alternative is beneficial. Numerical derivatives in the form of forward or central step have a degraded accuracy, while a complex step derivative is able to provide similar precision as the analytic derivative. The accuracy of the derivative has a direct impact on some numerical procedures, such as step size estimates, so any benefit of the numerical derivative alternative must rely on function computational efficiency compared to the analytic derivative. Most numerical derivative forms require two or more function evaluations to estimate the derivative, such as forward differences, see Equation 4.33 [45]. Even complex step derivatives require more computational time than a single real-valued function evaluation, because it executes the function call with both real and complex set computations, Equation 4.34 [64].

$$\frac{df}{dx} \approx \frac{f(x+h) - f(x)}{h} \quad (4.33)$$

$$\frac{df}{dx} \approx \frac{\Im[f(x+ih)]}{h} \quad (4.34)$$

The analytic derivative consistently requires less time than an extra function call, where the analytic derivative only requires about 75% to 90% of the computation time for a  $\Delta V$  function calculation. This is primarily due to the efficient formulation of the analytic derivative equation, in which most of the required terms and inputs are already computed for the  $\Delta V$  function. Therefore, the analytic derivative method will remain the method of choice in the upcoming development.

Based on an overall understanding of the powered gravity assist solution space, it is advantageous for the optimization procedure to first check the SOI derivative. The sign of  $\frac{d\Delta V}{dr}$  or  $\frac{d\Delta V}{dv}$  at SOI indicates if the minimum  $\Delta V$  is an interior solution or is bounded there at the SOI boundary. If the derivative is negative for Type RF flybys, then the SOI  $\Delta V$  maneuver is the bounded minimum (which is already known from the derivative computation) and the solution is complete, where the optimal  $\Delta V$  maneuver is located at the SOI boundary. For Type FR flybys, the SOI derivative is positive for an optimal  $\Delta V$  maneuver located at the SOI boundary. Otherwise, an interior minimum  $\Delta V$  solution must be found. Considering the difficulties and complexity of the interior solution space near periapse, a numerical optimization algorithm for the powered gravity assist BVP is more effective and reliable if initialized at the SOI. This is convenient since the  $\Delta V$  function and derivative are already computed for this point. It will be shown in the next section that the SOI boundary is often the global minimum solution, especially for more extreme flyby configurations. Providing the initial SOI boundary minimum check allows for quick solution recognition and exit strategy for many gravity assist situations.

The next step in the numerical procedure depends on the chosen optimiza-

tion method. A quasi-Newton method or similar is a common single variable optimization routine, which assumes the function can be sufficiently approximated as a quadratic or the initialization point is close enough to the minimum solution for such. Unfortunately, in many cases, the solution space is of a higher order and a quadratic model is not as good of a fit as a higher order model. In the multimodal analysis, it was observed that at most only two modes occurred. Considering this and the observed changes in the  $\Delta V$  solution space curvature, the solution space is more appropriately modeled as a cubic function. The large changes in curvature seen throughout the solution space analysis may produce problems for a Newton-like method, where a slope at these locations could lead the algorithm to greatly overshoot the minimum, disrupting the search. A cubic function based optimization is an alternative method for solving the problem and is adopted to robustly handle the higher order solution space dynamics. The cubic interpolation method typically enjoys quadratic rate of convergence, as compared to the superlinear rate of quasi-Newton methods [46].

The selected optimization procedure is the cubic interpolation method, which estimates the minimum of a function between two points. The algorithm assumes a minimum stability point exists between the bracketing points (or at the bracket boundaries). The algorithm inputs requires the function value at two points and their associated derivatives. The chosen maneuver location parameter is true anomaly  $v$ , since this formulation demonstrated faster and reliable convergence due to its naturally contained scale:  $0 < |v| < \pi$ . The cubic interpolation algorithm in Equation 4.35 computes the current estimate of the minimum  $\Delta V$  location  $v_{\Delta V_{min}}$  as  $v_{i+1}$ . The  $f(v)$  function represents the  $\Delta V$  magnitude for the particular maneuver location  $v$

(Equation 4.26). Similarly, the derivative function  $f'(v)$  is the value of  $\frac{d\Delta V}{dv}$  for the respective  $v$  bracket point (Equation 4.28). This particular formulation guarantees the existence and uniqueness of the solution estimate despite all the potential combinations of function points and derivative directions, where less robust algorithms fail [46].

$$\begin{aligned}
 v_{i+1} &= v_i - (v_i - v_{i-1}) \left[ \frac{f'(v_i) + D_2 - D_1}{f'(v_i) - f'(v_{i-1}) + 2D_2} \right] \\
 D_1 &= f'(v_{i-1}) + f'(v_i) - 3 \frac{f(v_{i-1}) - f(v_i)}{v_{i-1} - v_i} \\
 D_2 &= \text{sign}(v_i - v_{i-1}) \sqrt{D_1^2 - f'(v_{i-1}) f'(v_i)}
 \end{aligned} \tag{4.35}$$

To execute the cubic interpolation method, a second bracket point  $v_i$  must be selected in a standardized way to guarantee that this point and  $v_{SOI}$  will bracket the minimum for all  $r_{min}$  constrained flyby configurations. In order to ensure that all interior points are feasible and that the minimum is bracketed, the minimum feasible maneuver location  $v_{Fmin}$  is chosen as the second bracket point. Occasionally  $v_{Fmin}$  occurs on periapse side of the maximum, where slope is directed away from global minimum (toward periapse). Such an occurrence may lead some slope-based search algorithms astray, yet the robust cubic interpolation method is inherently designed to accommodate these scenarios. The pseudo-algorithm in Figure 4.43 displays the general procedure developed to solve the constrained periapse powered flyby boundary value problem. An example solution space plot is also shown depicting the feasible  $\Delta V$  maneuver locations in blue and infeasible solutions in red, separated by the minimum feasible true anomaly maneuver location  $v_{Fmin}$ . After the SOI derivative check, an interior solution is estimated using the  $\Delta V$  and  $\frac{d\Delta V}{dv}$

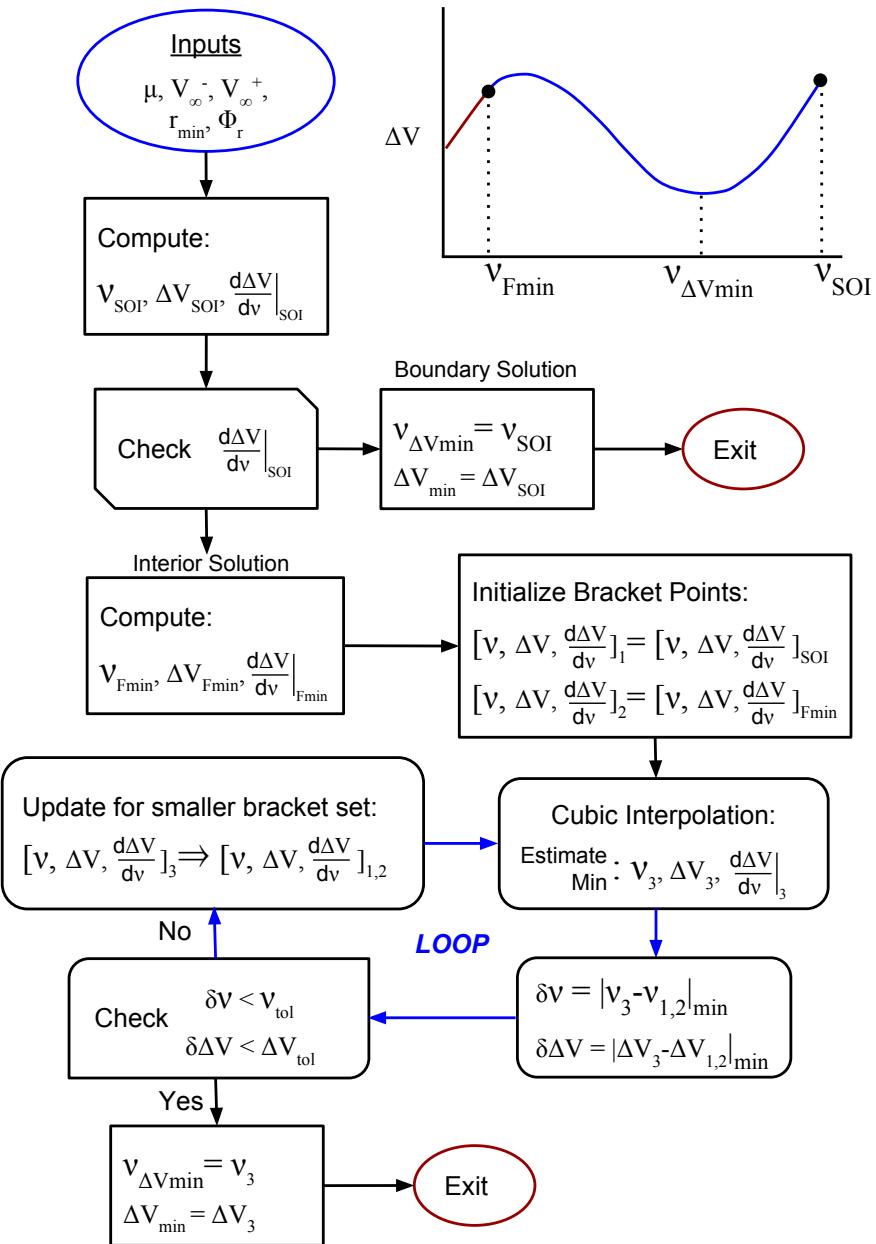


Figure 4.43: Constrained periapse powered gravity assist pseudo-algorithm

information at  $v_{Fmin}$  and  $v_{SOI}$  bracket points. The  $\Delta V$  function and derivative are computed at the new  $v_3$  point and the bracket is updated, replacing the point that shrinks the span of the brackets the most. The iterative method continues where new minimum points are estimated from the updated brackets until converging to the solution point. Convergence is measured by the change in maneuver location  $\delta v$  or the change in maneuver  $\delta\Delta V$  falling below a predefined threshold tolerance. The tolerances adopted are  $v_{tol} = 1E - 6$  radians and  $\Delta V_{tol} = 1E - 6$  km/s. Upon convergence the final estimate  $v_3$  and  $\Delta V_3$  is accepted as the interior minimum solution.

#### **4.3.7.2 Performance analysis of optimal periapse constrained powered gravity assist algorithm**

In this section, the optimization performance of the constrained periapse powered gravity assist algorithm is analyzed and verified for accuracy. The performance investigation is performed at a variety of flyby configurations similar to those of actual flown missions, such as Mariner 10, Voyager, Galileo, Cassini , Juno, New Horizons, etc. The analysis is performed for the most common gravity assist planets: Earth, Venus, and Jupiter. This demonstrates the potential requirements and benefits of a powered gravity assist within range of common flyby scenarios, while also showing associated computational requirements. The analysis of many flyby configurations is executed by varying the incoming and outgoing excess velocities  $V_\infty^\pm$  while holding  $r_{min}$ ,  $\Phi_r$ , and encounter planet fixed. Again, only Type RF flybys are considered, but the results are easily relatable to Type FR flybys. For ease of interpretation, many of the graphical diagrams depict the outgoing  $V_\infty^+$  in the form

$\frac{V_{\infty}^+}{V_{\infty}^-}$ , where Type RF flybys restrict the ratio  $\frac{V_{\infty}^+}{V_{\infty}^-} > 1$ . The selected range of powered gravity assists  $V_{\infty}^{\pm}$  are discretized into 50 equally spaced velocities each, yielding a total of 2500 optimal minimum  $\Delta V$  maneuver solutions.

For the first set of algorithm performance analyses, constrained powered gravity assist solutions are determined for an Earth flyby. The specific range of flyby configuration parameters is selected to represent a scenario similar to past spacecraft mission Earth encounters. The flyby configurations are broadly categorized as lower energy range and higher energy range, primarily references the flyby excess velocity magnitudes  $V_{\infty}^-$  and  $V_{\infty}^+$  (lower  $V_{\infty}^{\pm}$  magnitudes for lower energy and higher  $V_{\infty}^{\pm}$  magnitudes for higher energy).

A common lower energy range of Earth gravity assists have  $V_{\infty} \approx 9 \text{ km/s}$ , similar to those seen in the Galileo and Juno missions [65, 66]. With  $r_{min}$  set at 1.1 Earth radii, a group of constrained periapse powered gravity assists are determined for  $\Phi_r = 49^\circ$ . The optimal  $\Delta V$  magnitude and maneuver location  $v_{\Delta V}$  are shown in Figure 4.44 for exiting  $V_{\infty}^+$  up to 10% larger than  $V_{\infty}^-$ . As expected, the optimal maneuver increases with increasing  $V_{\infty}^-$  and  $V_{\infty}^+$ . The associated maneuver location is generally pushed farther out with increasing  $V_{\infty}^-$  and  $V_{\infty}^+$  until reaching the SOI boundary, seen as the top plateau. The dichotomy between the interior and SOI boundary solutions is seen plainly in Figure 4.45, which displays the span of potential  $\Delta V$  maneuver solutions for each flyby configuration, as well as the number of optimization iterations required to converge upon the minimum interior solution. The bottom plain of zero iterations correlate with the SOI boundary solutions because they are found without iteration.

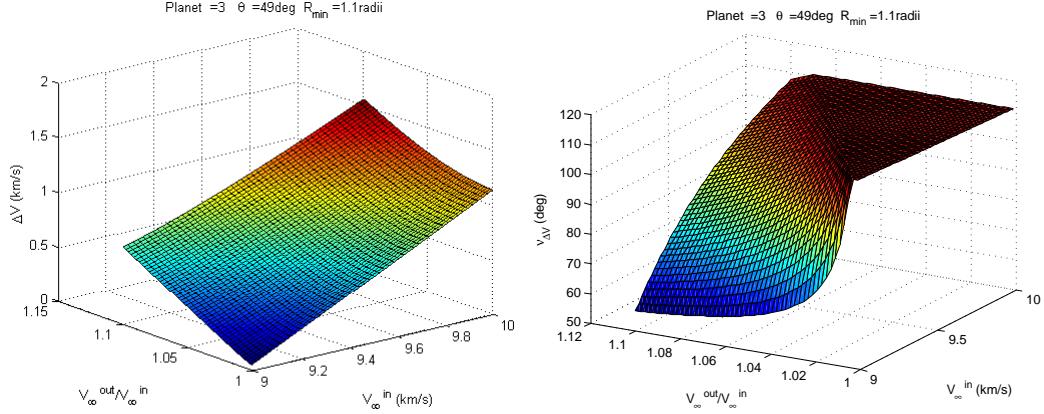


Figure 4.44: Earth Type RF gravity assist optimal  $\Delta V$  magnitude and maneuver location for typical lower  $V_\infty$  range

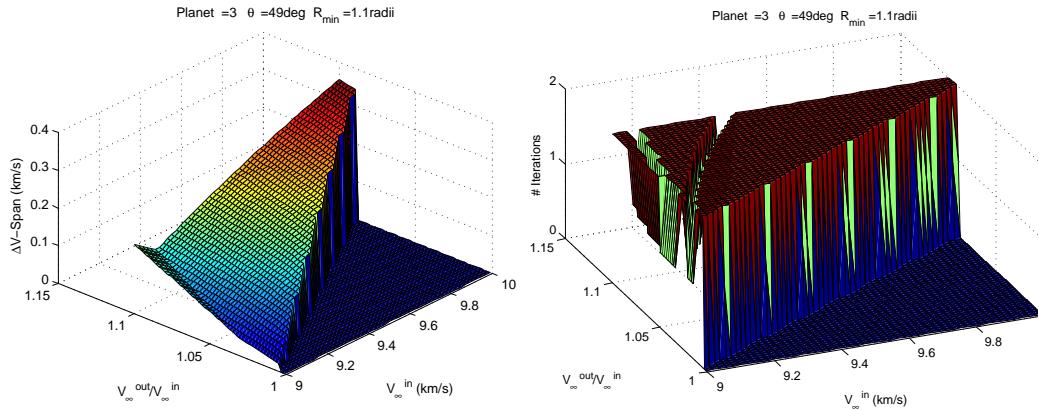


Figure 4.45: Earth Type RF gravity assist  $\Delta V$ -span of the solution space and # of optimization iterations for typical lower  $V_\infty$  range

The  $\Delta V$ -span is defined between the minimum solution and the largest candidate maneuver (usually the  $v_{SOI}$  boundary or near the  $v_{Fmin}$  boundary), therefore the SOI boundary solutions are shown as zero. It was found that in all of the flyby configurations investigated below, the largest candidate maneuver location is usually near the  $v_{Fmin}$  boundary. Although, the largest candidate  $\Delta V$  maneuver location

can switch among the flyby cases plotted, causing a cusp in the  $\Delta V$ -span surface plot. The  $\Delta V$ -span parameter identifies how large the variability is in the potential  $\Delta V$  maneuver solutions. Hence, a gravity assist problem with a large  $\Delta V$ -span, there is great  $\Delta V$  savings to be had by determining the optimal maneuver location. Yet, a gravity assist problem could have a small  $\Delta V$ -span, which indicates that no matter what maneuver location you choose, the resulting  $\Delta V$  magnitude outcome is very similar. In essence, the purpose of the  $\Delta V$ -span parameter is to show the potential impact the optimization procedure can have for a specific gravity assist configuration. The  $\Delta V$ -span graphic shows the range of  $\Delta V$  candidate maneuvers navigated by the iterative optimization procedure to find the minimum solution. For the group presented, 84% of the interior solutions have a  $\Delta V$ -span greater than 100 m/s. The  $\Delta V$ -span cusp seen at the lower  $V_{\infty}^-$  and higher  $\frac{V_{\infty}^+}{V_{\infty}^-}$  corner region indicates the point where the largest candidate  $\Delta V$  maneuver location switches from being near the  $v_{Fmin}$  boundary to being at the SOI boundary. This group of relatively benign interior solutions only require 1-2 iterations to converge at the minimum. The SOI boundary solutions also don't require any optimization iterations, as shown in the iterations plot, Figure 4.45. As the exiting  $V_{\infty}^+$  grows beyond the levels presented, the number of iterations required to converge increases, up to 4 iterations for a majority of extreme cases. By simply increasing the required  $\Phi_r$ , the more extreme flyby configuration results in larger and larger portions of the optimal  $\Delta V$  maneuver solved as SOI boundary minimum solutions.

Though intended as an iterative algorithm, it has been found that the first cubic interpolation estimates the optimal  $\Delta V$  maneuver close to the true minimum  $\Delta V$  for most trials of various flyby configurations. This is seen by the probability

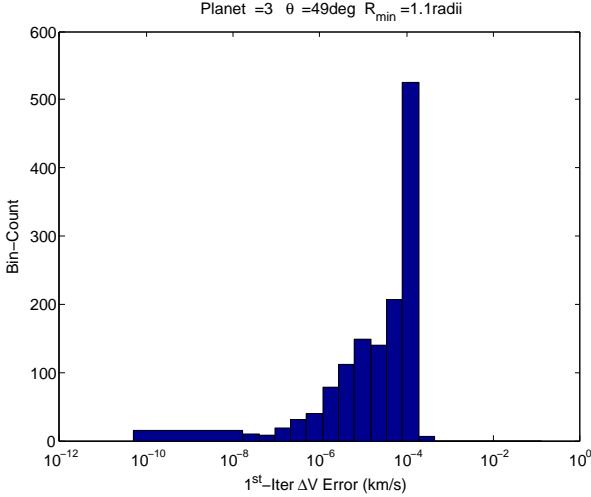


Figure 4.46: Earth Type RF gravity assist 1<sup>st</sup> iteration  $\Delta V$  accuracy for typical lower  $V_\infty$  range

density graph in Figure 4.46. Nearly all (98%) of gravity assist configurations in this group requiring an interior solution estimated the optimal  $\Delta V$  value to an accuracy of 1E-3 km/s or better on the first iteration. This accuracy, 1 m/s or better, is assumed to be good enough to be used for rough trajectory and mission planning. An alternative algorithm would be created to compute without iteration, taking advantage of this acceptable level of accuracy for a rough but fast powered gravity assist boundary value problem solver.

The optimal powered gravity assist solutions were verified by comparing with a reliable but inefficient single variable optimization MATLAB function over a fixed interval, “fminbnd” [25]. The tolerances for these reference solutions were decreased to  $v_{tol} = 1E - 8$  radians and  $\Delta V_{tol} = 1E - 8$  km/s. The optimal  $\Delta V$  error is measured as the difference between the reference “fminbnd” minimum  $\Delta V$  and the cubic algorithm minimum  $\Delta V$ . The largest  $\Delta V$  error for this set was on the order

of 1E-8 km/s, while the average error is of the order 1E-10 km/s.

A common higher energy range of Earth gravity assists have  $V_\infty \approx 16$  km/s, which is similar to that of the Cassini mission [67]. With  $r_{min}$  set at 1.1 Earth radii, a group of constrained periapse powered gravity assists are determined for  $\Phi_r = 21^\circ$ . The results are shown in Figure 4.47. The SOI boundary solutions are a smaller portion of the group shown in the  $\Delta V$  maneuver locations plot (top right) as the plateau region. The iterations plot (bottom right) show interior solutions that require between 1-3 iterations to converge and it depicts the SOI boundary solutions with the flat zero iterations plane. The  $\Delta V$ -span is also reduced to levels ranging from about 5 m/s to 200 m/s (bottom left), where only 22% of interior solutions had a span above 100 m/s. For the smaller  $\Delta V$ -span of candidate solutions, the improvement in minimizing the  $\Delta V$  maneuver has less impact, making the optimization procedure less critical. Note the very large  $\Delta V$ -span cusp indicating where the largest candidate  $\Delta V$  maneuver location switches from being near the  $v_{Fmin}$  boundary (higher  $V_\infty^-$  and lower  $\frac{V_\infty^+}{V_\infty^-}$  region ) to being at the SOI boundary (lower  $V_\infty^-$  and higher  $\frac{V_\infty^+}{V_\infty^-}$  region ). As seen in the diagrams, these solutions are only a small portion of the potential flyby configurations and tend to occur near ballistic flyby solutions. The accuracy of the solutions is confirmed to be maintained at the same levels with most  $\Delta V$  errors between 1E-8 and 1E-10 km/s.

A typical lower energy range of Venus gravity assists have  $V_\infty \approx 6$  km/s and are found in the Cassini and Galileo missions [67, 65]. With  $r_{min}$  set at 1.1 Venus radii, a group of constrained periapse powered gravity assists are determined for  $\Phi_r = 71^\circ$ . The  $\Delta V$ -span is of a similar range, where 92% of interior solution cases had a span above 100 m/s. This indicates that the lower  $V_\infty$  regimes tend to have

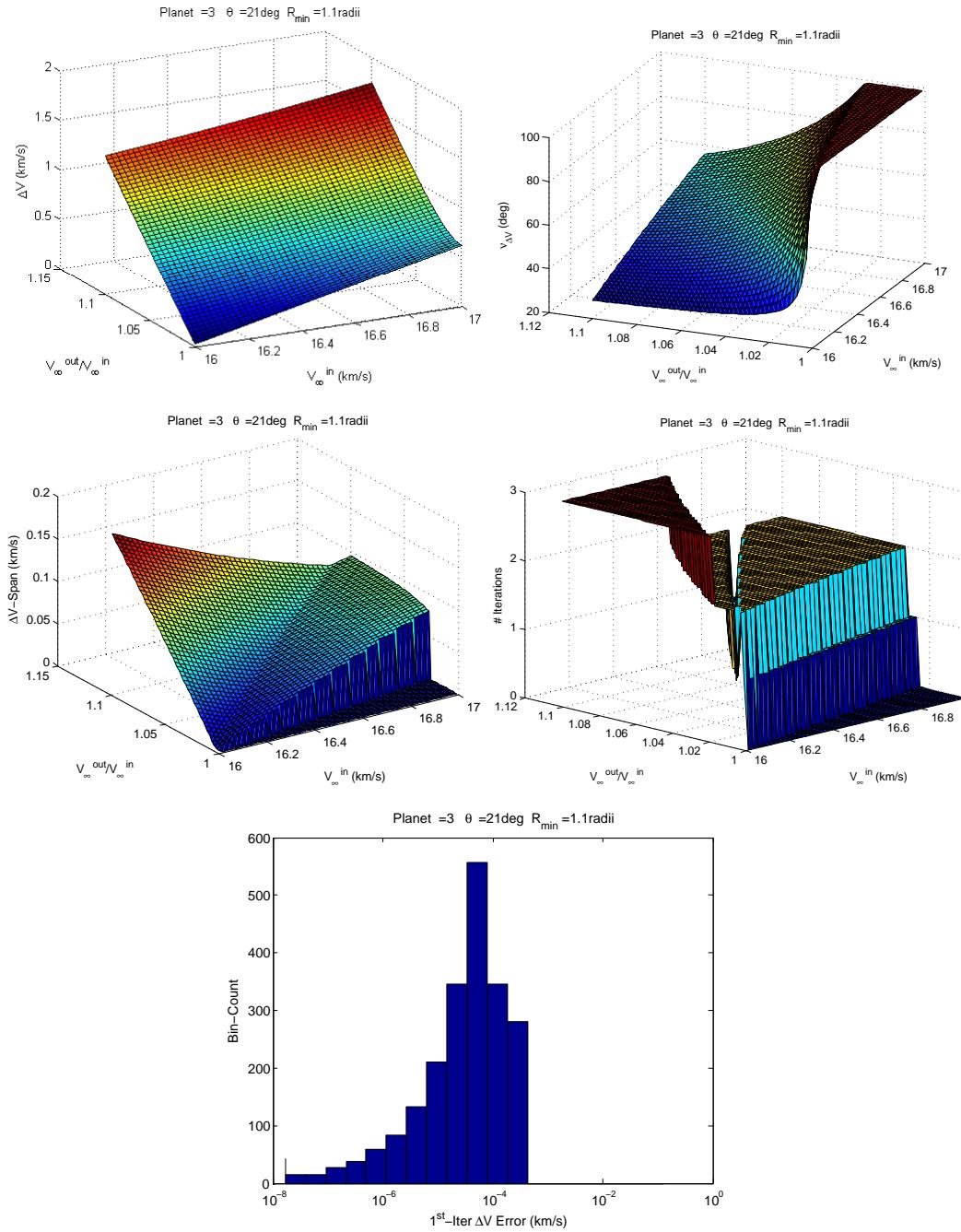


Figure 4.47: Earth Type RF gravity assist optimal maneuver data for typical higher  $V_\infty$  range

greater benefit from the optimization procedure. Also, 79% of the first iteration  $\Delta V$  estimates were within 1 m/s of the true minimum  $\Delta V$  solution. The data is quite similar to the Earth lower energy regime and therefore the plots are not presented.

A typical higher energy range of Venus gravity assists have  $V_\infty \approx 8.5$  km/s, where similar maneuvers can be seen in the Cassini, Mariner 10, and MESSENGER missions [67, 68, 9]. With  $r_{min}$  set at 1.1 Venus radii, a group of constrained periapse powered gravity assists are determined for  $\Phi_r = 48^\circ$ . This set of Venus flybys has conditions commonly seen in missions seeking to reduce their orbital energy for exploring difficult to reach inner planets, such as Mercury [68, 9]. A noteworthy characteristic of this data set is that the first iteration  $\Delta V$  estimates have an error of only 0.09 m/s, an order of magnitude better accuracy than previous cases. This reaffirms the ability of a non-iterative powered gravity assist algorithm alternative to be used for rough estimates of the optimal  $\Delta V$  maneuver. Much of the rest of the data and plots have only minor deviations from those of the lower energy range Earth and Venus cases and therefore the plots are not presented.

Jupiter has long been utilized for gravity assists in spacecraft missions due to its enormous gravitational potential and relative proximity to earth compared to other outer planets. It is worth noting that though Jupiter has the largest gravitational potential and gravity assist potential of all the planets in the solar system, it also has an extreme radiation environment at lower altitudes. This harsh radiation environment prevents many spacecraft missions from utilizing very low flyby altitude encounters with Jupiter.

The following sets of Jupiter powered gravity assist investigations is com-

prised of flyby configurations that are similar to many past missions, spanning a large range of  $V_\infty$  levels. At the lower end of the encounter energy range are gravity assists similar to the Voyager 2 mission with  $V_\infty \approx 8.5$  km/s [69]. With  $r_{min}$  set at 10 Jupiter radii, a group of constrained periapse powered gravity assists are determined for  $\Phi_r = 95^\circ$ . Though the minimum  $\Delta V$  solution graph in Figure 4.48 is similar to the other planetary encounters investigated, the effect of such a large gravitational body is evident in the large  $\Delta V$ -span of the solution space, nearly 1 km/s. This shows the increased sensitivity of a close Jupiter encounter, where a misplaced or delayed hyperbolic transfer maneuver can drastically decrease the performance of a powered gravity assist. The heightened sensitivity is also seen in the slightly decreased accuracy of the first iteration  $\Delta V$  estimate at 6 m/s. Another trend seen here is that the lower  $V_\infty$  powered gravity assists have a larger portion of the optimal transfer maneuver solutions at the SOI boundary, where the impulse is more effective at changing the  $V_\infty$  asymptote direction. The SOI boundary solutions are consistently seen for flyby configurations with very similar asymptote velocities  $V_\infty^\pm$ , but with moderate turning angles (a few degrees larger than the ballistic turning angles  $\delta^\pm$ ), where the  $\Delta V$  impulse is required to contribute more to the overall flyby turning angle with minimal changes in the asymptote velocity. These SOI maneuver solutions always appear as a wedge shape in the solution spaces investigated, where the lower wedge vertex begins along the  $V_\infty^- = V_\infty^+$  line. The SOI maneuver solution wedge expands out with wider  $\frac{V_\infty^+}{V_\infty^-}$  coverage for the fixed  $\Phi_r$  and  $r_{min}$  parameters as  $V_\infty^-$  increases.

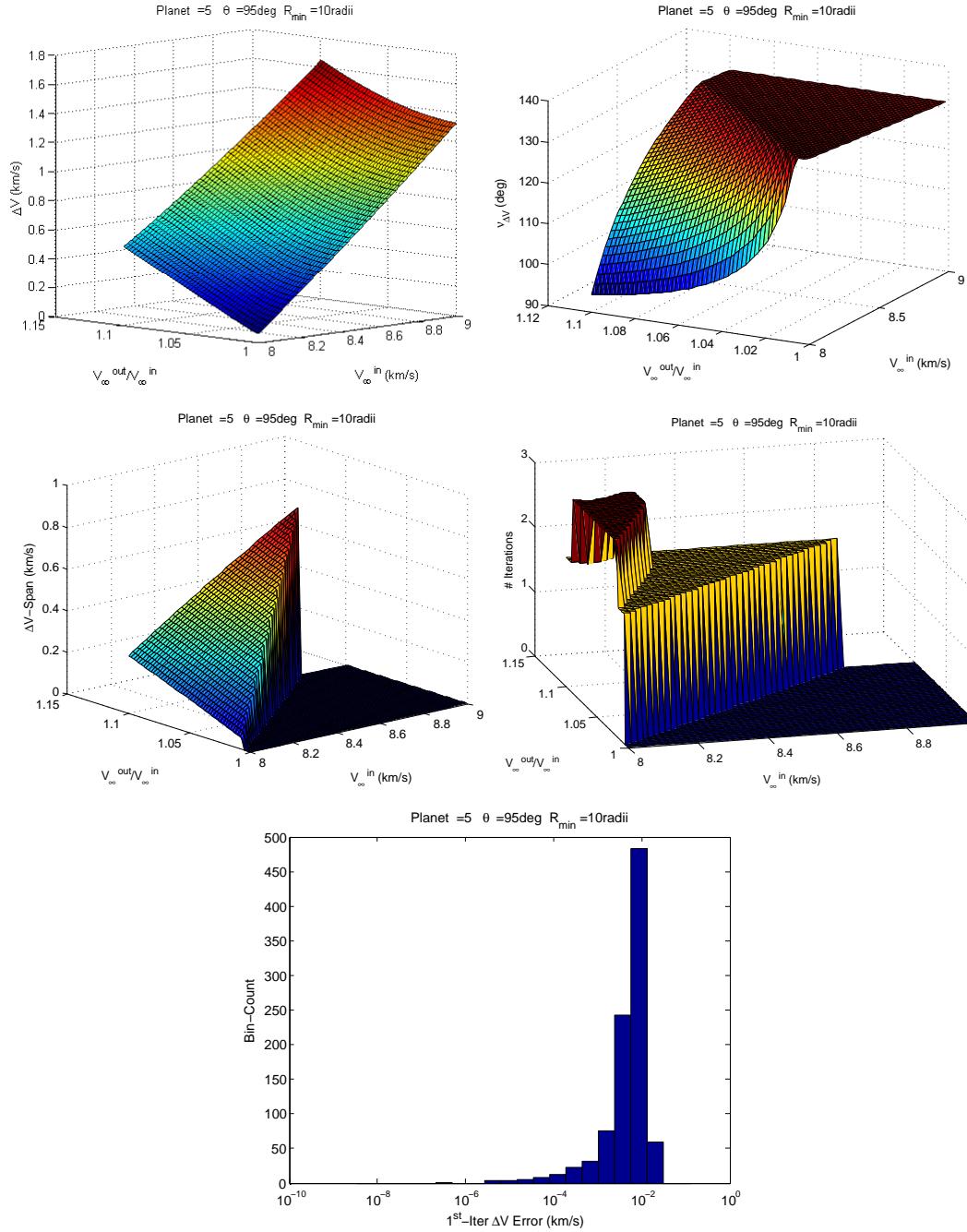


Figure 4.48: Jupiter Type RF gravity assist optimal maneuver data for typical lower  $V_{\infty}$  range similar to Voyager mission

The Ulysses mission provides an interesting example of higher encounter energy range for gravity assists with  $V_\infty \approx 14$  km/s [70]. The overall goal of Ulysses's Jupiter gravity assist was primarily to provide a significant plane change, establishing the spacecraft in a near polar heliocentric orbit. With  $r_{min}$  set at 6.5 Jupiter radii, a group of constrained periapse powered gravity assists are determined for  $\Phi_r = 72^\circ$ . There are many similar features in Figure 4.49 set of gravity assist data plots as compared to the lower Jupiter  $V_\infty$  encounters. There is clear reduction in the required optimization iterations of interior solutions, where only a few require 3 iterations and most have 2 iterations. Also, the first iteration  $\Delta V$  estimate shows slight improvement, having an accuracy of 3 m/s or better.

Finally, an example of a very high energy Jupiter encounter is the New Horizons mission with a Jupiter gravity assist of  $V_\infty \approx 19$  km/s [8]. The New Horizons spacecraft was launched with a very high velocity (the fastest launch velocity to date), producing an approach trajectory to Jupiter that creates such a high relative velocity  $V_\infty$ . This is partially due to the higher heliocentric spacecraft velocity, but mainly due to the large angle between the heliocentric velocities of Jupiter and the spacecraft, at nearly  $70^\circ$ . The Jupiter gravity assist provides a large energy boost for New Horizons spacecraft to reach Pluto on a fast 9.5 year trajectory. The  $r_{min}$  is set at 30 Jupiter radii to simulate New Horizons' minimal radiation dose constraints. The group of constrained periapse powered gravity assists are determined for  $\Phi_r = 17^\circ$ . The larger  $r_{min}$  distance and higher  $V_\infty$  range reduced the span of the solution space as seen by the  $\Delta V$ -span graph in Figure 4.50. These very high energy Jupiter powered gravity assists have an enhanced first iteration  $\Delta V$  accuracy of 0.04 m/s, similar to the higher energy Venus encounters. Note the very large  $\Delta V$ -span

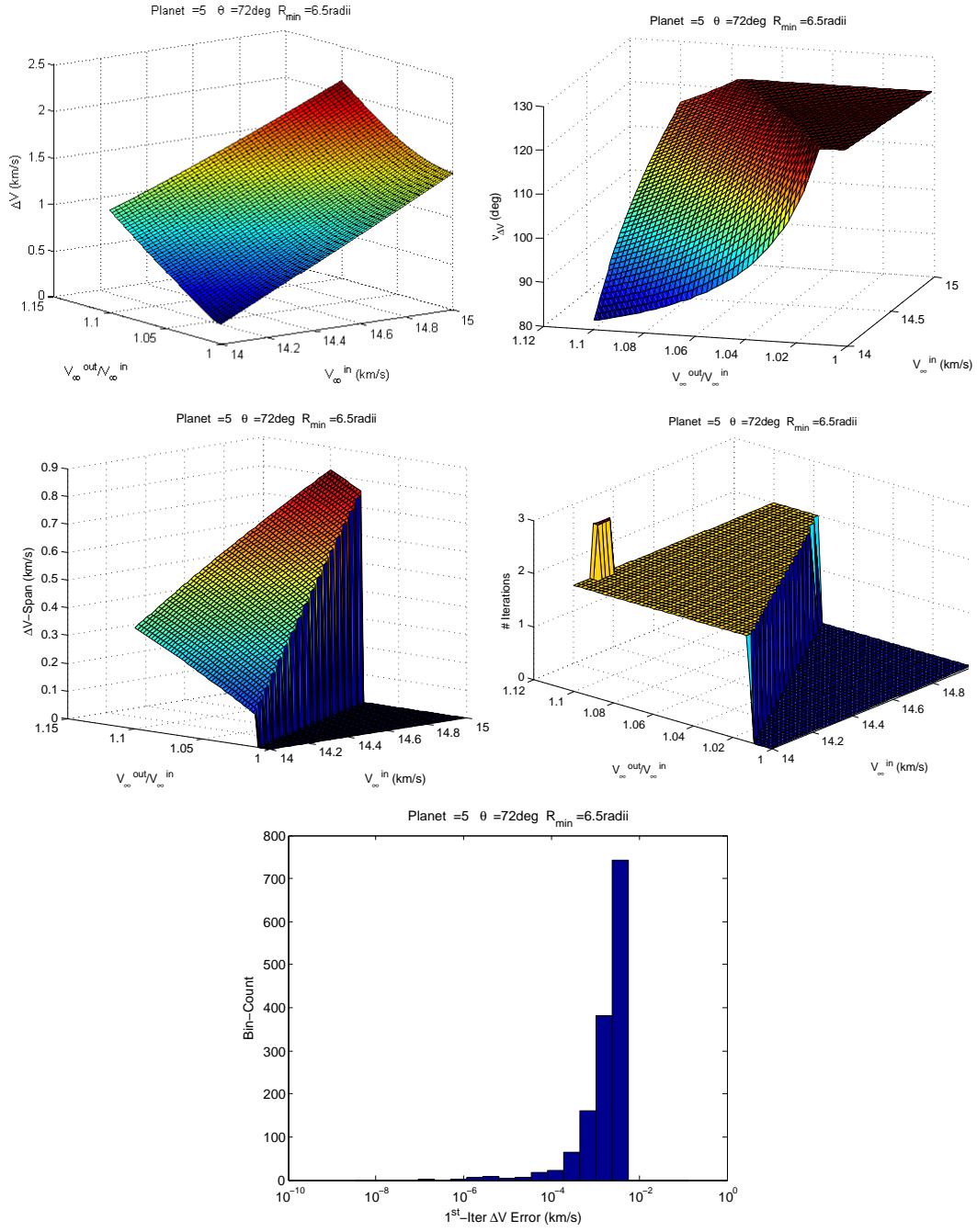


Figure 4.49: Jupiter Type RF gravity assist optimal maneuver data for typical higher  $V_\infty$  range similar to Ulysses mission

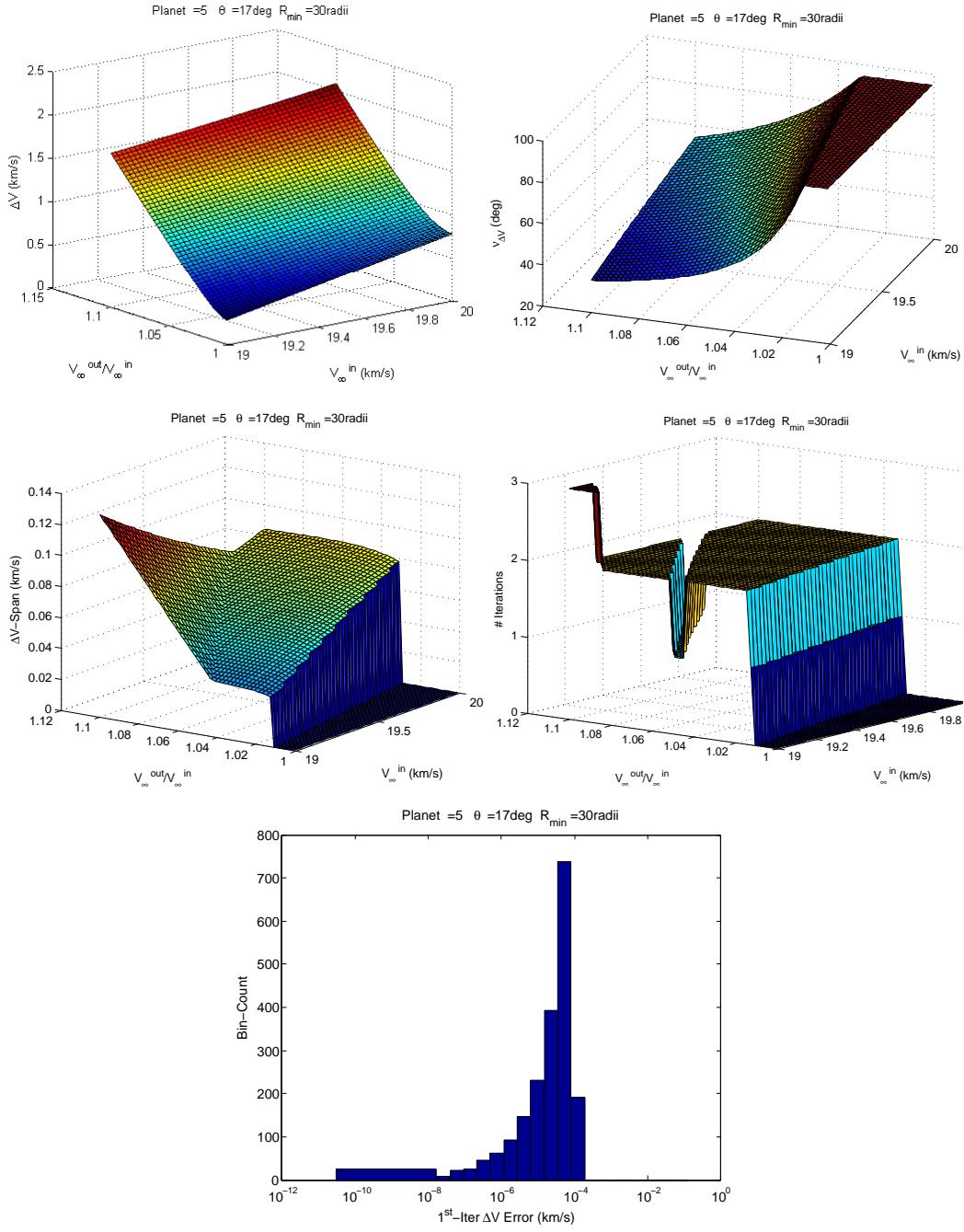


Figure 4.50: Jupiter Type RF gravity assist optimal maneuver data for very high  $V_{\infty}$  range similar to New Horizons mission

cusp indicating where the largest candidate  $\Delta V$  maneuver location switches from being near the  $v_{Fmin}$  boundary (higher  $V_\infty^-$  and lower  $\frac{V_\infty^+}{V_\infty^-}$  region ) to being at the SOI boundary (lower  $V_\infty^-$  and higher  $\frac{V_\infty^+}{V_\infty^-}$  region ).

Throughout the entire optimal powered gravity assist algorithm analysis it was extremely rare to encounter a boundary value problem configuration that required more than 4 iterations of the cubic interpolation procedure. A majority of cases require only 2 iterations or no iterations at all. In mission trajectory scenarios where extreme powered gravity assists occur, the  $\Delta V_{SOI}$  check often avoids requiring any iterations at all. As a comparison of computational execution time, the typical worst case optimal powered gravity assists requiring 3 or 4 iterations is compared to a standard Lambert targeting algorithm. Assumed as an industry standard, the robust Lambert targeting algorithm employed is Gooding's method [24] that provides improvements to the Lancaster and Blanchard algorithm structure [71]. The particular Lambert algorithm is coded by Oldenhuis [22] for speed and reliability in MATLAB. The optimal powered gravity assist convergence tolerance is maintained at  $v_{tol} = 1E - 6$  (radians) and  $\Delta V_{tol} = 1E - 6$  km/s. A variety of cases were investigated for both algorithms, demonstrating that the optimal powered gravity assist trials require at most, 3 or 4 iterations. Both algorithms provide a consistent execution time over randomized 10000 trials per analysis run and for multiple runs, focused at different regions of the boundary value problem. The Lambert targeter algorithm consistently required approximately 1.7E-4 seconds, while the optimal powered gravity assist required approximately 7.8E-5 seconds. This puts the general executing time of the optimal powered gravity assist algorithm at about 45% of the Lambert targeter. The favorable comparison of the worst

case timing reflects the efficiently tailored cubic optimization scheme and robust bracket initialization procedure. The actual execution time of any arbitrary flyby configuration would usually be significantly faster, requiring fewer iterations or if any at all.

## Chapter 5

# Flyby Constraints Utilizing Penalty $\Delta V$ at Asymptote

### 5.1 Typical ballistic flyby constraint handling

The prior chapter began by describing how most gravity assists maneuvers are assumed in trajectory optimization routines as either a set of ballistic flyby constraints or as a  $\Delta V$  maneuver at periapse. The typical constraints used to define a ballistic gravity assist are an equality constraint on the magnitudes of the  $V_{\infty}^{\pm}$  asymptotes and an inequality constraint dictating the required turning angle  $\Phi_r$  be less than some measure of the current available turning angle,  $\Phi_{av}$ , as referenced below in Equation 5.1. The available turning angle is usually measured as the previously defined maximum natural turning angle,  $\Phi_p$  [72] or as the maximum of the hyperbolic leg's turning angles,  $\delta_{max}^{\pm}$  [73]. Depending on the optimization routine being implemented, using constraints to handle gravity assists not only adds more constraint parameters, but increases the difficulty and “struggle” between minimizing the  $\Delta V$  performance index and satisfying the constraints when far from the optimal solution. The best performance of the constraint method is found when in the neighborhood of the optimal solution, where the optimization tends to converge quickly. Also, some constrained optimization routines have a separate nonlinear constraint function, which usually requires much of the trajectory data to be recomputed in order to check and maintain the nonlinear constraints. Considering

these reasons, a preliminary trajectory design investigation may be more efficient and effective by avoiding many complex nonlinear constraints, especially for rough trajectory plans.

$$\begin{aligned} |V_{\infty}^-| &= |V_{\infty}^+| \\ \Phi_r &\leq \Phi_{av} \end{aligned} \tag{5.1}$$

Alternatively, the general optimization performance can be discussed for the commonly used periapse maneuver powered gravity assist method. The periapse maneuver method instead completes the gravity assist by computing the  $\Delta V$  cost of the maneuver without adding any nonlinear constraints to the primary optimization routine. By adding the periapse maneuver powered gravity assist  $\Delta V$  to the optimization performance index, the flyby  $\Delta V$  maneuver acts as a penalty for an inefficient gravity assist. In some cases the optimization scheme is able to reduce the gravity assist  $\Delta V$  to zero and aligns the trajectory for a ballistic flyby. Although, as previously described, there are cases where the periapse maneuver gravity assist enables a reduction in the  $\Delta V$  cost of other maneuvers in the trajectory and improves the overall trajectory performance. If it is desired to have ballistic gravity assists, the energy efficiency of the periapse impulse works against one's interests for those cases.

## 5.2 Asymptote corrected ballistic gravity assist algorithm development

An efficient algorithm for unpowered ballistic gravity assists is developed in this section. The goal of the algorithm is to drive a trajectory optimization rou-

tine to produce ballistic flyby configurations in a similar manner as if applying the ballistic flyby constraints, but without having to add any nonlinear constraints to the primary optimization routine. The strategy of the alternative ballistic gravity assist algorithm is to handle any non-ballistic flyby configurations by using a  $\Delta V$  maneuver located at one of the flyby's asymptote excess velocity vectors ( $\vec{V}_\infty^-$  or  $\vec{V}_\infty^+$ ) to complete the flyby transfer. Essentially, the algorithm chooses one of the flyby leg's trajectories to coast ballistically from one asymptote of the flyby configuration to the other asymptote. The available ballistic trajectory excess velocity vector  $\vec{V}_{\infty-av}$  is the resulting asymptote velocity after completing the hyperbolic orbit. Any misalignment of the resulting  $\vec{V}_{\infty-av}$  vector compared to the required asymptote excess velocity vector  $\vec{V}_{\infty-req}$  is corrected with a  $\Delta V$  maneuver at that particular misaligned asymptote ( $\vec{V}_{\infty-req}^-$  or  $\vec{V}_{\infty-req}^+$ ). As an example, the flyby diagram in Figure 5.1 shows a flyby configuration with  $V_\infty^+ > V_\infty^-$ , where the incoming hyperbolic trajectory is allowed to coast all the way through its encounter to the exiting asymptote. The incoming hyperbolic trajectory periapse is such that the resulting excess velocity available  $\vec{V}_{\infty-av}^+$  at the end of the coasting trajectory is aligned collinear with the required exiting vector  $\vec{V}_{\infty-req}^+$ . Note that the magnitude of the resulting excess velocity available at the exiting asymptote is equal to the incoming magnitude,  $V_{\infty-av}^+ = V_\infty^+$ , as shown in the diagram. Other flyby configurations may result in a non-tangential misalignment of the available excess velocity vectors  $\vec{V}_{\infty-av}$  and required  $\vec{V}_{\infty-req}$ , which will be described later in the discussion below.

Since this correction maneuver is assumed to be restricted to act only on the excess velocity vectors, then we can reference this asymptote maneuver as the change in excess velocity vector  $\Delta V_\infty = \|\vec{V}_{\infty-req} - \vec{V}_{\infty-av}\|$ . Even though this algo-

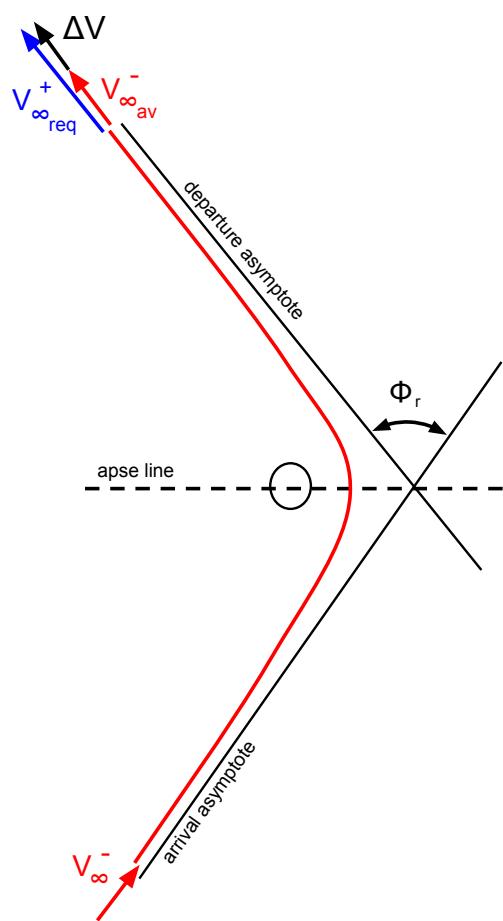


Figure 5.1: Gravity assist example of collinear  $\Delta V$  maneuver at asymptote

rithm performs a  $\Delta V$  maneuver and could be considered a powered gravity assist (with the  $\Delta V$  maneuver restricted to the flyby asymptotes), the intent of the algorithm is to lead an interplanetary trajectory optimization routine toward a ballistic gravity assist flyby configuration. Effectively, the  $\Delta V_\infty$  maneuver acts as a DSM positioned immediately after or before the flyby to adjust the asymptotes for a ballistic flyby. A similar strategy is seen often in multiple gravity assist trajectory optimization tools, where a DSM is automatically placed in heliocentric space at an estimated location after each flyby (assumed ballistic) [74]. Then each DSM [X Y Z] location must be optimized in order to reduce the DSM  $\Delta V$  magnitude to zero. The algorithm presented here models the ballistic flyby accompanied by an asymptote DSM ( $\Delta V_\infty$ ), but the DSM location does not have to be optimized since it is assumed to occur at an asymptote of the flyby. The discussion below details the justification for the maneuver location restricted to either asymptote.

The reasoning for the asymptote corrected ballistic flyby strategy utilizes the understanding gained about powered gravity assist solution space behavior from the previous chapter. The analysis of the powered gravity assist BVP has displayed consistent behavior among varying flyby configurations, from near ballistic trajectories to even extreme flyby configuration cases that require large  $\Delta V$  maneuvers. The optimal unconstrained periapse (Type F) powered gravity assists that are unencumbered by the  $r_{min}$  constraints tend to have the optimal maneuver location relatively near periapse. Difficult flyby configurations for the unconstrained periapse powered gravity assist can occur that require too small of a turning angle  $\Phi_r$  for the  $V_\infty^\pm$  magnitudes, thus determining higher and higher periapse radii, until hitting the  $r_{SOI}$  constraint.

The optimal constrained periapse powered gravity assists also tend to have maneuver locations relatively close to periapse for near ballistic configurations. As the difference in asymptote velocities and/or  $\Phi_r$  increases, the maneuver location is also driven out toward the  $r_{SOI}$  boundary. The dynamics that influence the optimal maneuver location is a trade between greater energy change efficiency closer to periapse and greater impulse directional efficiency (rotating the velocity vector direction with a  $\Delta V$  maneuver) farther from periapse. Also influencing the optimal flyby  $\Delta V$  maneuver location is the natural velocity rotation of the hyperbolic trajectory and the decreasing orbital velocity for locations farther out along the hyperbola. This sensitivity of the  $\Delta V$  performance to maneuver location is leveraged in the asymptote corrected flyby algorithm for handling the ballistic gravity assist constraints.

By restricting the  $\Delta V$  maneuver far from periapse, the energy efficiency aspect of the maneuver is mitigated. Applying the maneuver at the asymptote completely removes the energy efficient feature and greatly reduces the computation to a simple algorithm, presented later in the section. Also, any change in the flyby configuration  $V_\infty^\pm$  asymptotes would then have a direct and proportional affect on the  $\Delta V_\infty$  maneuver. The  $\Delta V_\infty$  maneuver at the appropriate  $V_\infty^\pm$  asymptote provides effective convergence characteristics. If at a region of the solution space with more extreme flyby configurations, the optimal powered gravity assist maneuver location is likely close to the asymptote solution as well. This leads the overall optimization of the asymptote corrected flyby algorithm along a similar solution path as the optimal powered gravity assist algorithm, especially when starting from rough trajectory initialization. When the interplanetary trajectory provides the a flyby

configuration close to that of a ballistic flyby, the asymptote maneuver acts more as a penalty for an unaligned flyby configuration. In a similar near ballistic flyby configuration, the periapse powered flyby and optimal powered flyby algorithms would produce much lower  $\Delta V$  costs for the flyby. This affect gives the optimization routine less incentive to quickly drive the flyby  $\Delta V$  to zero and potentially prevents a zero  $\Delta V$  for the flyby altogether. Therefore, the penalty asymptote  $\Delta V_\infty$  is able to have faster and more consistent convergence near a ballistic flyby solution compared to periapse powered gravity assist or optimal powered gravity assist methods. These general optimization performance characteristics are supported by the interplanetary mission trajectory analysis performed at the end of the chapter.

The required  $\Delta V_\infty$  maneuver is simply the difference in available excess velocity  $\vec{V}_{\infty\_av}^\pm$  of the transited hyperbola and the required  $\vec{V}_{\infty\_req}^\mp$  at the same end point. Conveniently, the magnitude of asymptotic velocities  $V_\infty^-$  and  $V_\infty^+$  are provided as inputs to the boundary value problem, but their angular offset at the  $\Delta V_\infty$  asymptote maneuver end point must be determined. Selecting the transit hyperbola and determining the  $\Delta V$  maneuver is similar to the optimal solution neighborhood sorting technique. The maximum turning angle  $\delta_{max}^\pm$  for each hyperbola is compared and the hyperbola with the largest available turning angle (labeled as  $\Phi_{max}$ ) is selected to be transited. The selection can also be made by choosing the minimum excess velocity ( $V_\infty^-$  or  $V_\infty^+$ ). A collinear asymptote maneuver solution is possible when the transited hyperbola's turning angle  $\delta$  is able to match the required  $\Phi_r$ , taking advantage of the periapse degree of freedom, Figure 5.1. This condition occurs when  $\Phi_{max} > \Phi_r$ . Otherwise, when  $\Phi_{max} < \Phi_r$  the maneuver is not collinear and the offset angle between the endpoint excess velocity vectors is the difference

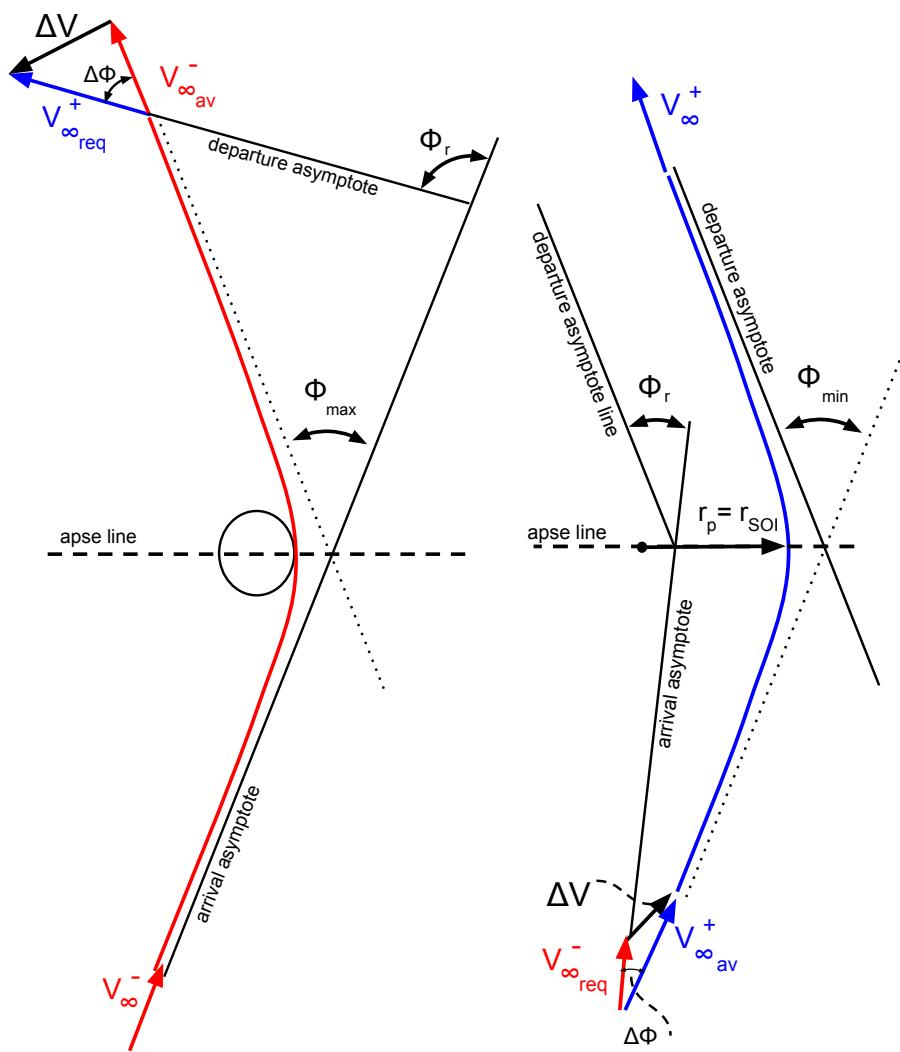


Figure 5.2: Gravity assist example of non-tangential  $\Delta V_\infty$  maneuvers at asymptote:  
 $\Phi_{\max} < \Phi_r$  (left) and  $\Phi_{\min} > \Phi_r$  (right)

in available and required turning angles, as displayed in Figure 5.2. The algorithm is shown below in Equation 5.2.

An exception must be made to account for those collinear cases that require periapse values outside the sphere of influence, eliminating the planet's influence and violating the rules of linked conics. Therefore a minimum turning angle  $\Phi_{min}$  associated with a periapse at  $r_{SOI}$  must be included in the algorithm. The minimum turning angles  $\delta_{min}^\pm$  of each hyperbola for  $r_p = r_{SOI}$  are compared, where the smaller of the turning angles ( $\delta_{min}^-$  or  $\delta_{min}^+$ ) is defined as  $\Phi_{min}$ . More directly, the maximum excess velocity ( $V_\infty^-$  or  $V_\infty^+$ ) can be used to define  $\Phi_{min}$ . By constraining the periapse at  $r_{SOI}$  or below, the flyby for this  $r_p = r_{SOI}$  condition incurs a non-tangential asymptote maneuver  $\Delta V_\infty$  that is larger than the collinear  $\Delta V_\infty$  of the infeasible flyby that has a periapse beyond SOI. Thus, the larger  $\Delta V_\infty$  acts as a penalty for the infeasible flyby configuration, which drives the interplanetary optimization routine away from this condition and toward a ballistic flyby configuration.

$$\begin{aligned} \Phi_{max} > \Phi_r & \quad \left\{ \begin{array}{l} \Phi_{min} < \Phi_r \Rightarrow \Delta V = |V_\infty^- - V_\infty^+| \\ \Phi_{min} > \Phi_r \Rightarrow \Delta V = \sqrt{V_\infty^{-2} + V_\infty^{+2} - 2V_\infty^- V_\infty^+ \cos(\Phi_r - \Phi_{min})} \end{array} \right. \\ \Phi_{max} < \Phi_r & \quad \Rightarrow \quad \Delta V = \sqrt{V_\infty^{-2} + V_\infty^{+2} - 2V_\infty^- V_\infty^+ \cos(\Phi_r - \Phi_{max})} \end{aligned}$$

where  $\Phi_{max} = 2 \arcsin \left( \frac{\mu}{\mu + r_{min} V_\infty^2 |_{min}} \right)$

$$\Phi_{min} = 2 \arcsin \left( \frac{\mu}{\mu + r_{SOI} V_\infty^2 |_{max}} \right) \quad (5.2)$$

### 5.3 Asymptote corrected ballistic gravity assist algorithm analysis

As  $\Phi_r$  varies from above  $\Phi_{max}$  to below  $\Phi_{min}$ , the asymptote  $\Delta V$  maneuver type changes as does the maneuver location. The sequence of diagrams in Figure 5.3 depicts the range of potential  $\Delta V_\infty$  maneuvers assuming  $V_\infty^- < V_\infty^+$ . The maneuver occurs at the black dot in each example case and the  $\Delta V_\infty$  maneuver vector is indicated in red. Starting with the flyby case at right side of the diagram, the asymptote  $\Delta V_\infty$  maneuver is non-tangential for the  $\Phi_{max} < \Phi_r$  case and is located at the end point of the smaller  $V_\infty^-$  hyperbola intersecting the larger  $V_\infty^+$  asymptote. When the required turning angle is just below  $\Phi_{max}$ , the maneuver is collinear and remains at the  $V_\infty^+$  asymptote. As  $\Phi_r$  decreases farther below  $\Phi_{max}$ , it is reduced below the turning angle  $\delta_{max}^+$ , therefore allowing the collinear maneuver to be placed at either asymptote for the same resulting  $\Delta V_\infty$  magnitude. This trend continues until  $\Phi_r$  falls below  $\delta_{min}^-$ , where the maneuver is still collinear but must occur at the opposite  $V_\infty^-$  asymptote. Finally, once  $\Phi_r$  drops below  $\Phi_{min}$ , a collinear maneuver is no longer achievable, requiring an angled  $\Delta V_\infty$  between the excess velocity vectors  $\vec{V}_{\infty-av}^+$  and  $\vec{V}_{\infty-req}^-$ , while remaining at  $V_\infty^-$  asymptote.

Though this complete form of ballistic gravity assist constraint handling for linked conics was developed organically as a result of the optimal powered gravity assist analyses, a similar partially-completed form was later found in one other study; a dissertation by Russell [73]. Russell's form handled the ballistic gravity assist constraints by utilizing the collinear or offset angle  $\Delta V$  maneuver conditions, but did not regard the minimum turning angle condition. This potentially allows for ballistic gravity assist solutions at unrealistic periapse distances in the

Assume  $V_{\infty}^- < V_{\infty}^+ \therefore \Phi_{\max} = \delta_{\max}^- > \delta_{\max}^+ \text{ and } \delta_{\min}^- > \delta_{\min}^+ = \Phi_{\min}$

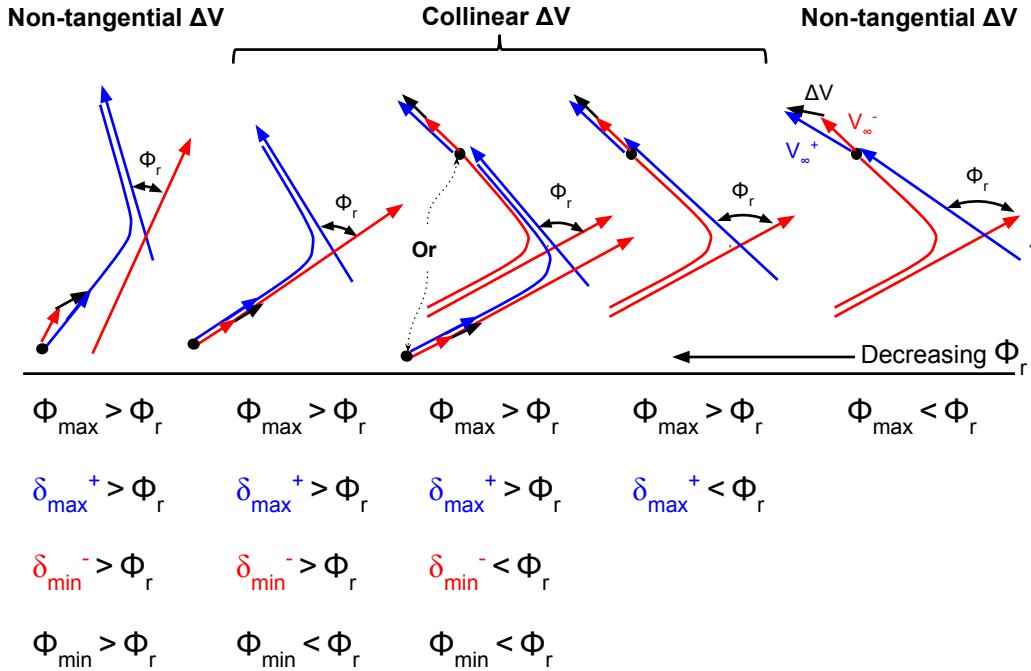


Figure 5.3: Varying asymptote  $\Delta V$  maneuver locations and  $\Delta V$  directions (collinear or non-tangential) for decreasing  $\Phi_r$  assuming  $V_{\infty}^- < V_{\infty}^+$

encounter planet frame. This essentially assumes that the heliocentric trajectory can pass through the vicinity of a planet unperturbed. The result can be useful as an indicator for a mission designer to consider avoiding the particular planetary encounter. However, this yields inconsistencies when allowed within an trajectory optimization procedure, since the large distance would actually require a different heliocentric trajectory and therefore change the encounter  $\vec{V}_\infty^\pm$ . In order to avoid these inconsistencies that violate the linked conic assumptions, the SOI boundary constraint must be implemented on the periapse radius in addition to the  $r_{min}$  constraint,  $r_{min} \leq r_p \leq r_{SOI}$ . Other similar formulations of a ballistic gravity assist methodology may exist as informal unpublished algorithms, but are lacking a complete formal characterization and development as presented in the current research.

When implementing the asymptote corrected ballistic gravity assist algorithm in an interplanetary trajectory optimization scheme, not only is the execution time per iteration reduced compared to using flyby constraints or a periapse maneuver, but the number of iterations required for convergence are also reduced for a majority of trials. However, there are trajectory examples (usually including multiple consecutive gravity assists) where this algorithm alone cannot deliver a ballistic trajectory solution. Some of these difficult multiple gravity assist trajectory cases can be better solved by combining the advantages of the asymptote maneuver algorithm and the nonlinear flyby constraints together. Implementing both methods at the same time provides a stronger incentive to guide the optimization toward ballistic gravity assists. The asymptote maneuver contributes a penalty  $\Delta V$  to the performance index, which for some optimization routines (those that separate the performance function and the constraint functions, e.g. MATLAB's fmincon [25])

can better align the efforts of the  $\Delta V$  minimization and constraint handling. These performance characteristics are demonstrated in the mission trajectory examples presented in the next section.

## 5.4 Spacecraft mission trajectory examples with ballistic gravity assists

In this section the ballistic gravity assist trajectory examples are simulated and analyzed for the recent Juno mission and the past Galileo mission, both to Jupiter. Each interplanetary spacecraft mission is simulated in TRACT with the variety of gravity assist algorithms developed in this dissertation, as well as a variety of popular MATLAB optimization algorithms. The TRACT gravity assist algorithms modeled are the optimal powered gravity assist, the periapse powered gravity assist, the asymptote corrected ballistic gravity assist, and the traditional ballistic gravity assist nonlinear constraints. To demonstrate the convergence capability of the differing gravity assist algorithms towards the desired ballistic gravity assist trajectory goal, only the directed optimization schemes fminunc, fminsearch, and fmincon are included. These commonly used unconstrained periapse and constrained optimization schemes use gradient and gradient-free methods to provide alternative search characteristics useful for varying optimization strategies.

### 5.4.1 Juno mission trajectory with accurate initial parameter values

The Juno mission to Jupiter is a relatively simple gravity assist trajectory shown in Figure 5.4. Launched in August 2011, the Juno spacecraft departed Earth with only enough energy to reach out to an aphelion of about 2.25 AU, a little past

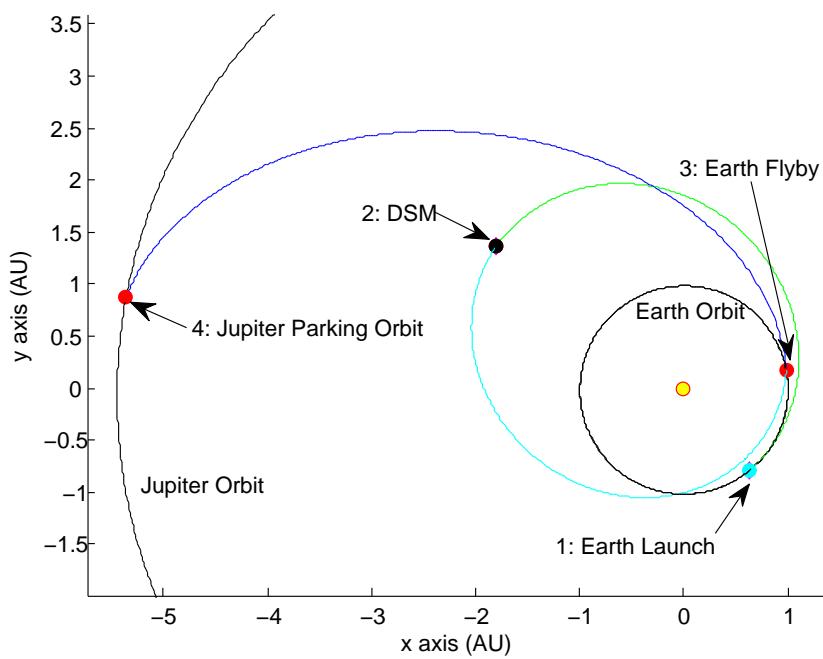


Figure 5.4: Juno mission trajectory: Launch from Earth, DSM near aphelion, Earth gravity assist, and capture into parking orbit at Jupiter

the orbit of Mars and less than half the distance to Jupiter at over 5 AU [75]. The efficient energy saving strategy of the Juno trajectory is to perform two DSMs near aphelion as a  $V_\infty$ -leveraging maneuver for the ensuing ballistic Earth gravity assist. This allows the Juno spacecraft to gain enough energy from the single Earth gravity assist to reach Jupiter for orbit insertion into a capture orbit. In the following simulations, the launch is assumed to have a  $C_3 = 31.1 \text{ km}^2/\text{s}^2$  available at an inclination of  $28.5^\circ$ ; the two close consecutive DSMs are modeled as a single DSM for simplicity; the minimum Earth flyby altitude is 500 km; and the Jupiter polar capture parking orbit is assumed with  $r_p = 1.06$  Jupiter radii, a period of 107 days, a  $90^\circ$  inclination, and a periapse insertion maneuver [75, 66]. The optimization algorithms use a  $\Delta V$  function tolerance of  $1E - 6 \text{ km/s}$ , a normalized optimization variable tolerance of  $1E - 8$ , and a normalized constraint tolerance of  $1E - 8$  (if applicable).

The first batch of simulations for the Juno mission used optimization parameters initialized close to the actual mission trajectory with a Launch date of August 5, 2011, TOFs of  $[ 393.5 \ 402.5 \ 1000 ]$  days, and DSM [X Y Z] location of  $[ -1.8 \ 1.4 \ 0 ]$  AU [76, 66, 75]. The Juno trajectories in Table 5.1 show the unconstrained solutions using fminunc’s quasi-Newton optimization algorithm. The initial trajectory performance values at the initial guess conditions (not optimized yet) are shown in the “Initial” column. The “Optimal”, “Periapse”, and “Asymptote” columns refer to cases using the optimal powered gravity assist model, the periapse powered gravity assist model, and the asymptote corrected gravity assist model respectively. The actual Juno mission DSMs combined to approximately  $\Delta V=730 \text{ m/s}$  and the capture orbit insertion maneuver at Jupiter is estimated at

|  | Initial | Optimal  | Periapse | Asymptote | Actual  |
|--|---------|----------|----------|-----------|---------|
| MJD0 (days)  | 55778.2 | 55778.06 | 55776.94 | 55777.92  | 55778.2 |
| TOF <sub>TOT</sub> (days)                                  | 1796.0  | 1785.59  | 1792.03  | 1782.56   | 1796.0  |
| $\Delta V_1 - \text{Launch} (\frac{\text{km}}{\text{s}})$  | 0       | 0        | 0.01241  | 0         | 0       |
| $\Delta V_2 - \text{DSM} (\frac{\text{km}}{\text{s}})$     | 0.77780 | 0.68283  | 0.69652  | 0.68136   | 0.730   |
| $\Delta V_3 - \text{Earth} (\frac{\text{km}}{\text{s}})$   | 0.34133 | 0.00058  | 4.87E-08 | 4.33E-09  | 0       |
| $\Delta V_4 - \text{Jupiter} (\frac{\text{km}}{\text{s}})$ | 0.42634 | 0.42656  | 0.42629  | 0.42677   | 0.480   |
| $\Delta V_{TOT} (\frac{\text{km}}{\text{s}})$              | 1.5455  | 1.1099   | 1.1352   | 1.1081    | 1.210   |
| Func Evals   | 1       | 448      | 240      | 608       | -       |
| Iterations   | 0       | 18       | 5        | 22        | -       |
| Run Time (sec)   | 0.014   | 1.716    | 1.05     | 2.152     | -       |

Table 5.1: Juno mission trajectory simulation example cases using unconstrained fminunc quasi-Newton algorithm

about 480 m/s [75] and are shown in the “Actual” column. Each of the simulations found very similar  $\Delta V$  results to the actual mission. The simulated DSM  $\Delta V$  results are slightly better due to the actual mission imposing additional operational constraints on the DSMs. With the initial guess parameters already close to their optimal values, the implemented gravity assist models each found a nearly ballistic Earth flyby, some better than others. The asymptote corrected gravity assist model outperformed the others for both minimum total  $\Delta V$  and a ballistic Earth gravity assist, while requiring slightly more iterations than the optimal powered gravity assist. Confirming the earlier discussion on the general search pattern comparisons of the asymptote transfer and the optimal powered gravity assist models, the asymptote corrected gravity assist model and the optimal powered gravity assist model found very similar solutions, but the asymptote corrected trajectory was able to better drive the Earth flyby to an approximately zero  $\Delta V$  ballistic gravity assist.

The Juno trajectory solutions using the more explorative fminsearch sim-

|                                      | Initial | Optimal  | Periapse | Asymptote | Actual  |
|--------------------------------------|---------|----------|----------|-----------|---------|
| MJD0 (days)                          | 55778.2 | 55777.46 | 55778.51 | 55778.31  | 55778.2 |
| TOF <sub>TOT</sub> (days)            | 1796.0  | 1758.66  | 1715.94  | 1721.23   | 1796.0  |
| $\Delta V_{1-Launch}(\frac{km}{s})$  | 0       | 0        | 0        | 0         | 0       |
| $\Delta V_{2-DSM}(\frac{km}{s})$     | 0.77780 | 0.66790  | 0.65988  | 0.66390   | 0.730   |
| $\Delta V_{3-Earth}(\frac{km}{s})$   | 0.34133 | 5.74E-12 | 2.82E-13 | 1.63E-07  | 0       |
| $\Delta V_{4-Jupiter}(\frac{km}{s})$ | 0.42634 | 0.42930  | 0.43766  | 0.43634   | 0.480   |
| $\Delta V_{TOT}(\frac{km}{s})$       | 1.5455  | 1.0972   | 1.0975   | 1.1002    | 1.210   |
| FuncEval                             | 1       | 1244     | 2239     | 1022      | -       |
| Iters                                | 0       | 758      | 1450     | 583       | -       |
| RunTime                              | 0.014   | 5.086    | 8.625    | 3.958     | -       |

Table 5.2: Juno mission trajectory simulation example cases using gradient-free fminsearch simplex algorithm

plex optimization algorithm are shown in Table 5.2. As common with the simplex method, more function evaluations were required for convergence. Each of the gravity assist models found very similar ballistic gravity assist solutions, all of which are slightly lower  $\Delta V$  trajectories than from the fminunc optimization runs. Interestingly, the optimal and periapse powered gravity assist trajectories had much lower gravity assist  $\Delta V$  than the asymptote corrected model, although all of these small  $\Delta V$  values are of negligible magnitudes for preliminary trajectory design.

The batch of simulations shown in Table 5.3 use fmincon’s interior-point algorithm and include the traditional ballistic gravity assist constraints model (shown in the “Constr” column), as well as applying the asymptote corrected model together with the traditional constraints at the same time (shown in the “Asy & Con” column). The actual Juno trajectory values are omitted from these tables to manage the table width, but these values can be seen in the prior tables. Each of these Earth flybys were driven to negligible  $\Delta V$  levels and can be approximated as ballistic. The

|                                      | Optimal  | Periapse | Asymp    | Constr   | Asy & Con |
|--------------------------------------|----------|----------|----------|----------|-----------|
| MJD0 (days)                          | 55775.85 | 55775.31 | 55775.21 | 55778.54 | 55776.32  |
| TOF <sub>TOT</sub> (days)            | 1735.63  | 1716.30  | 1720.94  | 1717.89  | 1695.94   |
| $\Delta V_{1-Launch}(\frac{km}{s})$  | 0        | 0        | 0        | 0        | 0         |
| $\Delta V_{2-DSM}(\frac{km}{s})$     | 0.65600  | 0.64694  | 0.64945  | 0.70823  | 0.65784   |
| $\Delta V_{3-Earth}(\frac{km}{s})$   | 5.94E-05 | 8.06E-08 | 2.02E-07 | 1.90E-11 | 4.12E-09  |
| $\Delta V_{4-Jupiter}(\frac{km}{s})$ | 0.43357  | 0.43843  | 0.43721  | 0.43823  | 0.44482   |
| $\Delta V_{TOT}(\frac{km}{s})$       | 1.0896   | 1.0854   | 1.0867   | 1.146    | 1.103     |
| FuncEval                             | 361      | 373      | 687      | 604      | 266       |
| Iters                                | 30       | 33       | 62       | 53       | 24        |
| RunTime                              | 1.861    | 1.861    | 2.877    | 4.297    | 2.183     |

Table 5.3: Juno mission trajectory simulation example cases using constrained fmincon interior-point algorithm

case using only traditional nonlinear ballistic gravity assist constraints had the most ballistic gravity assist trajectory, but at a sacrifice of higher total  $\Delta V$ . The traditional constraints method has much longer run time at only 141 function evaluations per sec compared to the other models, while the asymptote corrected model runs at a brisk 239 function evaluations and the optimal powered gravity assist at 194 evaluations. Adding the asymptote correction model to the traditional constraints drove the search to find a lower total  $\Delta V$  trajectory solution, while maintaining a satisfactorily ballistic gravity assist. Though the traditional gravity assist constraints model does not compute an associated  $\Delta V$  value during the optimization process, the post-optimization procedure computes this model's resulting gravity assist trajectory solution as a periapse powered gravity assist in order to compare with the other models.

Another powerful fmincon algorithm is the SQP (sequential quadratic programming) scheme. The fmincon simulations are executed again with the SQP

|                                      | Optimal  | Periapse | Asymp    | Constr   | Asy & Con |
|--------------------------------------|----------|----------|----------|----------|-----------|
| MJD0 (days)                          | 55774.22 | 55774.94 | 55774.04 | 55775.12 | 55775.27  |
| TOF <sub>TOT</sub> (days)            | 1705.24  | 1703.51  | 1704.53  | 1700.26  | 1696.31   |
| $\Delta V_{1-Launch}(\frac{km}{s})$  | 0        | 0        | 0        | 1.50E-08 | 0         |
| $\Delta V_{2-DSM}(\frac{km}{s})$     | 0.64480  | 0.64183  | 0.64537  | 0.64053  | 0.63942   |
| $\Delta V_{3-Earth}(\frac{km}{s})$   | 5.82E-05 | 1.19E-05 | 4.41E-05 | 1.05E-07 | 3.28E-07  |
| $\Delta V_{4-Jupiter}(\frac{km}{s})$ | 0.44211  | 0.44243  | 0.44240  | 0.44346  | 0.44478   |
| $\Delta V_{TOT}(\frac{km}{s})$       | 1.0870   | 1.0843   | 1.0878   | 1.0840   | 1.0841    |
| FuncEval                             | 429      | 425      | 633      | 729      | 399       |
| Iters                                | 31       | 31       | 51       | 61       | 28        |
| RunTime                              | 2.014    | 1.771    | 2.383    | 4.751    | 2.833     |

Table 5.4: Juno mission trajectory simulation example cases using constrained fmincon SQP algorithm

algorithm, shown in Table 5.4. The SQP algorithm is known for its excellent convergence characteristics when in the neighborhood of a minimum, which is demonstrated with each of the gravity assist models resulting in very similar solutions [46, 47]. In this batch of simulations, the traditional constraints results were slightly better than the rest by having the lowest total  $\Delta V$  and the best estimate of a ballistic gravity assist. Although, by combining the asymptote correction model with the traditional constraints, a nearly identical trajectory solution was achieved with substantially less run time and computation. This confirms the earlier discussion of melding benefits of both gravity assist models for a better outcome when used together. It is an interesting result that the optimal and periapse powered gravity assists converged upon similar solutions in slightly faster time than the asymptote corrected model, due to a nearly double number of iterations required.

All of the trajectory solutions were roughly similar and all converged upon DSM locations close to  $[ -1.7 \quad 1.5 \quad 0.0 ]$  AU. Most of the launch date and TOFs

converged close to their initial estimates: the average launch date of August 2, 2011 with a  $1-\sigma$  of 1.4 days, TOF1 = 387 days with  $1-\sigma$  = 9.4 days, TOF2 = 405 days with  $1-\sigma$  = 8.9 days, and TOF3 = 934 days with  $1-\sigma$  = 33.4 days. Of all the Juno mission trajectory simulations using initial guess trajectory values close to the optimal values (using the actual trajectory estimates), the fmincon SQP algorithm produced the most consistently good results throughout all the gravity assist models. The best performance combination was the fmincon SQP algorithm using the combined traditional ballistic gravity assist constraints along with the asymptote corrected model. The combined gravity assist models utilized in the SQP algorithm has greater enforcement of the ballistic gravity assist conditions and drives to the solution faster with dramatically less iterations, therefore with much less run time.

#### 5.4.2 Juno mission trajectory with rough initial parameter values

Another set of simulations for the Juno mission is performed with rough optimization parameters values initialized far from the actual mission trajectory. These simulations demonstrate the optimization performance of a typical rough preliminary trajectory design investigation in the process of maturing. If a mission designer were to estimate these values on their own, the guiding strategy would be in aligning the  $V_\infty$ -leveraging maneuver with proceeding Earth gravity assist. Using multiples of Earth orbital periods as baseline, the timeline from Launch to Earth flyby should be about 2 years. Splitting this in half, each leg of the trajectory before and after the DSM is estimated with a 365 day TOF. The Earth to Jupiter leg would likely be estimated with the round 1000 day value, input as as  $TOF = [ 365 \ 365 \ 1000 ]$  days. The DSM [X Y Z] location of  $[ -1.5 \ 1.5 \ 0 ]$  AU is

also a very course estimate relatively far from the optimal location.

The launch date is a very sensitive parameter which affects the timing of all other mission events. Commonly in preliminary trajectory design investigations using other tools, many separate runs of the trajectory optimization process are performed where the launch date is incrementally adjusted through a range of pre-selected dates. The launch dates increments are on the order of a few days to a fraction of a month to ensure that one of the trajectory optimization runs are initialized close to an optimal solution. Using such a fine resolution for these increments is indicative of the small convergence region of these other trajectory optimization methods. The following Juno example demonstrates TRACTs ability to converge to an optimal trajectory solution with a launch date initial guess very far from the optimal. The launch date will be initialized for May 1, 2011, over 3 months away from the actual launch date. The unoptimized Juno trajectory at these initial conditions is shown in Figure 5.5. The associated initial trajectory  $\Delta V$  maneuver values are shown in Table 5.5 with extremely high  $\Delta V$  values from the rough trajectory initial guess parameters, with a total  $\Delta V$  of over 27 km/s. The initialized gravity assist  $\Delta V$  magnitudes (measured as an asymptote transfer) are very large due to the unaligned flyby conditions.

In this Juno trajectory demonstration, the fmincon SQP optimization scheme consistently out performed the others. The trajectory results using the fmincon SQP optimization scheme and best performing gravity assist models are shown in Table 5.5. Despite the larger deviation from the actual mission values, the optimal powered gravity assist and asymptote corrected gravity assist models converged upon similar trajectory solutions as the actual Juno mission. The asymptote corrected

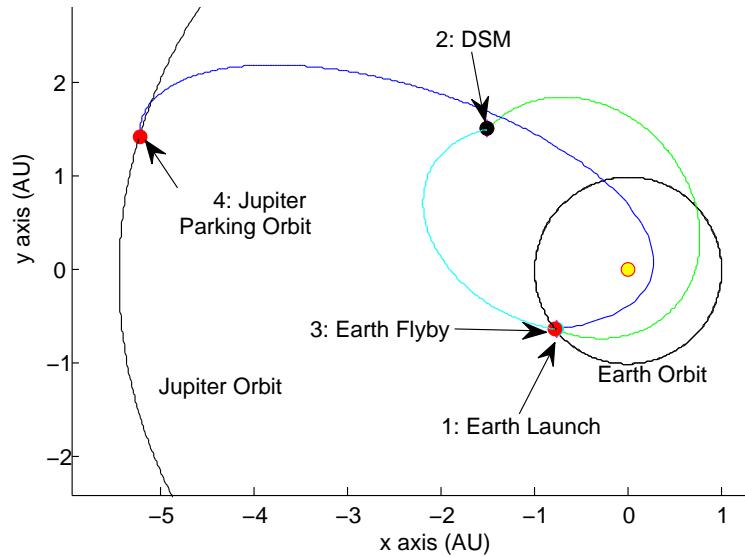


Figure 5.5: Juno mission trajectory using rough initialized guess parameters: Launch from Earth, DSM near aphelion, Earth gravity assist, and capture into parking orbit at Jupiter

|                                       | Initial  | Optimal  | Periapse | Asymptote | Actual  |
|---------------------------------------|----------|----------|----------|-----------|---------|
| MJD0 (days)                           | 55682.0  | 55774.02 | 55794.36 | 55775.40  | 55778.2 |
| TOF <sub>TOT</sub> (days)             | 1730.0   | 1710.76  | 1686.09  | 1704.34   | 1796.0  |
| $\Delta V_{1-Launch} (\frac{km}{s})$  | 7.46182  | 6.87E-05 | 0        | 0         | 0       |
| $\Delta V_{2-DSM} (\frac{km}{s})$     | 8.64404  | 0.64954  | 0.64720  | 0.64209   | 0.730   |
| $\Delta V_{3-Earth} (\frac{km}{s})$   | 10.29254 | 5.15E-04 | 5.42E-06 | 1.98E-06  | 0       |
| $\Delta V_{4-Jupiter} (\frac{km}{s})$ | 0.82778  | 0.44044  | 0.44181  | 0.44201   | 0.480   |
| $\Delta V_{TOT} (\frac{km}{s})$       | 27.226   | 1.091    | 1.089    | 1.084     | 1.210   |
| Func Evals                            | 1        | 863      | 809      | 974       | -       |
| Iterations                            | 0        | 75       | 75       | 87        | -       |
| Run Time (sec)                        | 0.014    | 4.037    | 3.613    | 4.103     | -       |

Table 5.5: Rough initialization Juno mission trajectory simulation example cases using fmincon SQP optimization scheme

flyby model yielded the lowest total  $\Delta V$  value. The resulting optimal trajectory solutions converged to launch dates very close to the actual Juno mission (with an offset of only a few days). The total flight time is similar to the actual mission, but is nearly identical to the converged flight time found in the previous Juno simulations with accurate initialized trajectory parameters. The TRACT also produced good ballistic gravity assist estimates, reducing the initial 10 km/s unaligned powered gravity assist down to a sufficiently aligned flyby with negligible  $\Delta V$ . The resulting optimal DSM  $\Delta V$  magnitude and locations are also similar to the actual Juno mission. The resulting DSM location is  $[ -1.65 \ 1.55 \ 0 ]$  AU, which is very close to the previous simulation's DSM location of  $[ -1.7 \ 1.5 \ 0 ]$  AU. The plot of the optimized Juno trajectory from rough initialized parameters is nearly indistinguishable from the trajectory plot of the previous study with accurate initialized parameters, Figure 5.4.

Considering the convergence region of TRACT's optimization capability, the same sort of incremental trajectory optimization investigation could be performed with much larger steps in launch date. Instead of increments of just a few days, TRACT enables step sizes on the order of months. This allows a range of launch dates to be analyzed much faster with much less number of steps and trajectory optimization runs.

#### **5.4.3 Galileo mission trajectory with accurate initial parameter values**

The Galileo mission to Jupiter utilized an efficient multiple ballistic gravity assist trajectory of Venus and Earth flybys to reduce the required launch energy and still reach Jupiter, Figure 5.6. The Galileo spacecraft launched on October 18,

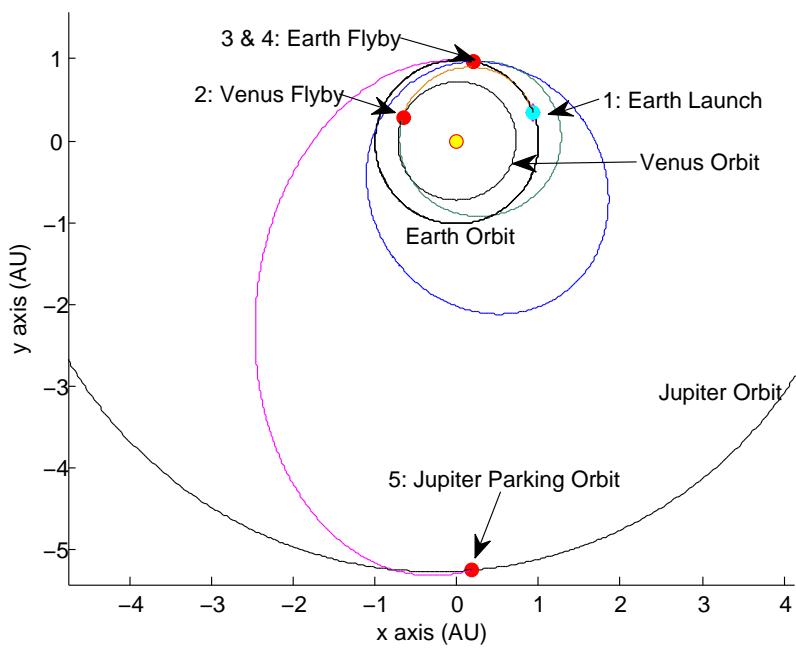


Figure 5.6: Galileo mission trajectory: Launch from Earth, Venus-Earth-Earth gravity assist, and capture into parking orbit at Jupiter

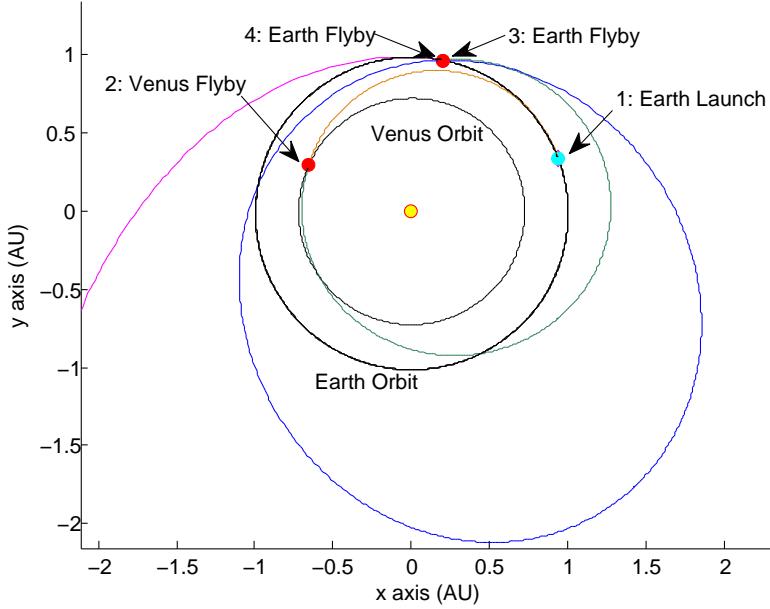


Figure 5.7: Galileo mission inner planet trajectory: Launch from Earth, Venus-Earth-Earth gravity assist

1989 with an available launch energy of only  $C_3 = 17 \text{ km}^2/\text{s}^2$  [65]. The spacecraft then completes a circuitous flyby combination of Venus-Earth-Earth gravity assists known as a VEEGA trajectory shown more clearly in Figure 5.7. After the final Earth gravity assist, the Galileo spacecraft has acquired enough orbital energy to reach Jupiter, where it inserts into a large highly eccentric capture orbit. In the following simulations, the launch is assumed at an inclination of  $28.5^\circ$ ; each Venus and Earth gravity assists have a typical minimum flyby altitude set at 300 km; and the Jupiter capture orbit has a periapse at 4 Jupiter radii and apoapse at 300 radii with a near equatorial orbit inclination equal to the arrival  $\vec{V}_\infty^-$  declination of a few degrees [65]. The trajectory leg between the first Earth flyby and the second is barely longer than 1 full revolution about the Sun (but about two years in length),

so the multi-revolution Lambert parameter is set to 1 and positive for a right branch trajectory. The optimization algorithms use a  $\Delta V$  function tolerance of  $1E - 6$  km/s, a normalized optimization variable tolerance of  $1E - 8$ , and a normalized constraint tolerance of  $1E - 8$  (if applicable).

The first simulations for the Galileo mission used optimization parameters initialized close to the actual mission trajectory with a Launch date of October 18, 1989 and TOFs of [ 115 301 731 1095 ] days [65]. The Galileo trajectories in Table 5.6 show the results for the best performing optimization algorithm and gravity assist model combination compared to the actual Galileo mission. The initial trajectory performance values at the initial guess conditions (not optimized yet) are shown in the “Initial” column. The columns of the table represent each trajectory run using various gravity assist models: asymptote corrected gravity assist model (“Asymp”), the ballistic gravity assist constraints (“Constr”), and the combination of ballistic constraints and the asymptote corrected flyby model (Asy & Con). The actual Galileo mission had ballistic gravity assists and the capture orbit insertion maneuver at Jupiter was about 630 m/s [65], as identified in the “Actual” column. The fminunc and fminsearch Galileo trajectory simulations had difficulty progressing past the initial search direction after the first iteration. This is due to the initialized trajectory having a multiple revolution trajectory leg so close to the transition between a single revolution and less than single revolution case.

The traditional ballistic flyby constraints case and the combined asymptote transfer case provide ballistic gravity assist trajectories and the minimum orbit insertion  $\Delta V$ , close to the actual orbit insertion value of 630 m/s. The combined asymptote transfer and traditional constraints strategy provided similar trajectory

|                                      | Initial | Asymp    | Constr   | Asy & Con | Actual  |
|--------------------------------------|---------|----------|----------|-----------|---------|
| MJD0 (days)                          | 47817.0 | 47840.38 | 47815.94 | 47828.01  | 47817.0 |
| TOF <sub>TOT</sub> (days)            | 2242.0  | 2354.32  | 2316.41  | 2331.15   | 2242.0  |
| $\Delta V_{1-Launch}(\frac{km}{s})$  | 0       | 0        | 0        | 0         | 0       |
| $\Delta V_{2-Venus}(\frac{km}{s})$   | 0.22127 | 0.01608  | 2.73E-11 | 2.02E-11  | 0       |
| $\Delta V_{3-Earth1}(\frac{km}{s})$  | 3.90102 | 0.00470  | 3.91E-09 | 1.86E-08  | 0       |
| $\Delta V_{4-Earth2}(\frac{km}{s})$  | 3.88953 | 0.00059  | 3.61E-09 | 2.44E-08  | 0       |
| $\Delta V_{5-Jupiter}(\frac{km}{s})$ | 0.72489 | 0.77855  | 0.74644  | 0.75894   | 0.630   |
| $\Delta V_{TOT}(\frac{km}{s})$       | 8.737   | 0.800    | 0.746    | 0.759     | 0.630   |
| FuncEval                             | 1       | 288      | 754      | 217       | -       |
| Iterations                           | 0       | 16       | 72       | 19        | -       |
| Run Time (sec)                       | 0.014   | 2.210    | 7.444    | 2.797     | -       |

Table 5.6: Galileo mission trajectory simulation example cases using fmincon SQP algorithm

results but in less than half the run time. The asymptote corrected flyby model also resulted in nearly ballistic gravity assists and a total  $\Delta V$  only about 50 m/s off from the best performing mission simulation, but with the least run time effort. The best case trajectory simulation resulted in a launch date only 2 days off from the actual launch date. The converged trajectory's total flight time is about 74 days off from the actual mission flight time, amounting to an error of only 3% of the over 6 year long Galileo mission trajectory.

#### 5.4.4 Galileo mission trajectory with rough initial parameter values

The Galileo mission trajectory is simulated again, but with rough optimization parameters values initialized far from the actual mission trajectory. These simulations demonstrate the optimization performance of a typical rough preliminary trajectory design investigation. The circuitous nature of the Galileo mission trajectory makes it much more sensitive to poor launch date initial estimates. As dis-

cussed earlier, other typical trajectory design tools require initial launch dates for complex trajectories to be accurate within a few days or a fraction of a month in order to converge upon a nearby optimal solution. These simulations will use two different initial launch dates of September 1, 1989 and November 19, 1989, more than a month offset on either side of the actual launch date October 18, 1989. If a mission designer were to estimate the TOF values, the guiding strategy would be in handling the Earth synchronous trajectory leg between the first and second Earth flyby. It was seen in the previous Galileo mission simulation study that this Earth synchronous trajectory leg is very sensitive to changes in its TOF, especially when modeling it with a multiple revolution Lambert trajectory segment. To reduce this sensitivity and allow for a more robust optimization convergence, the trajectory leg between the Earth flybys is split up by introducing a DSM in the middle.

In order to further demonstrate TRACT's effectiveness at converging poor initial guess trajectories, the midcourse maneuver guess generator option is used to automatically provide an estimate for the DSM location. The midcourse maneuver guess generator is intended to give good DSM location estimates for direct trajectory legs between different planets. But it provides relatively poor DSM location estimates when the surrounding trajectory legs begin and end at the same planet. As seen in the September launch initial guess trajectory plot, Figure 5.8, the angular position of the DSM is a reasonable estimate, but the radial location of the DSM at that angle is very poorly estimated. A different DSM location estimate is also generated automatically for the November launch, shown in Figure 5.9. Using multiples of Earth orbital periods as reference, each leg is estimated by one year for a total of two years between the Earth flybys. The other trajectory leg TOFs

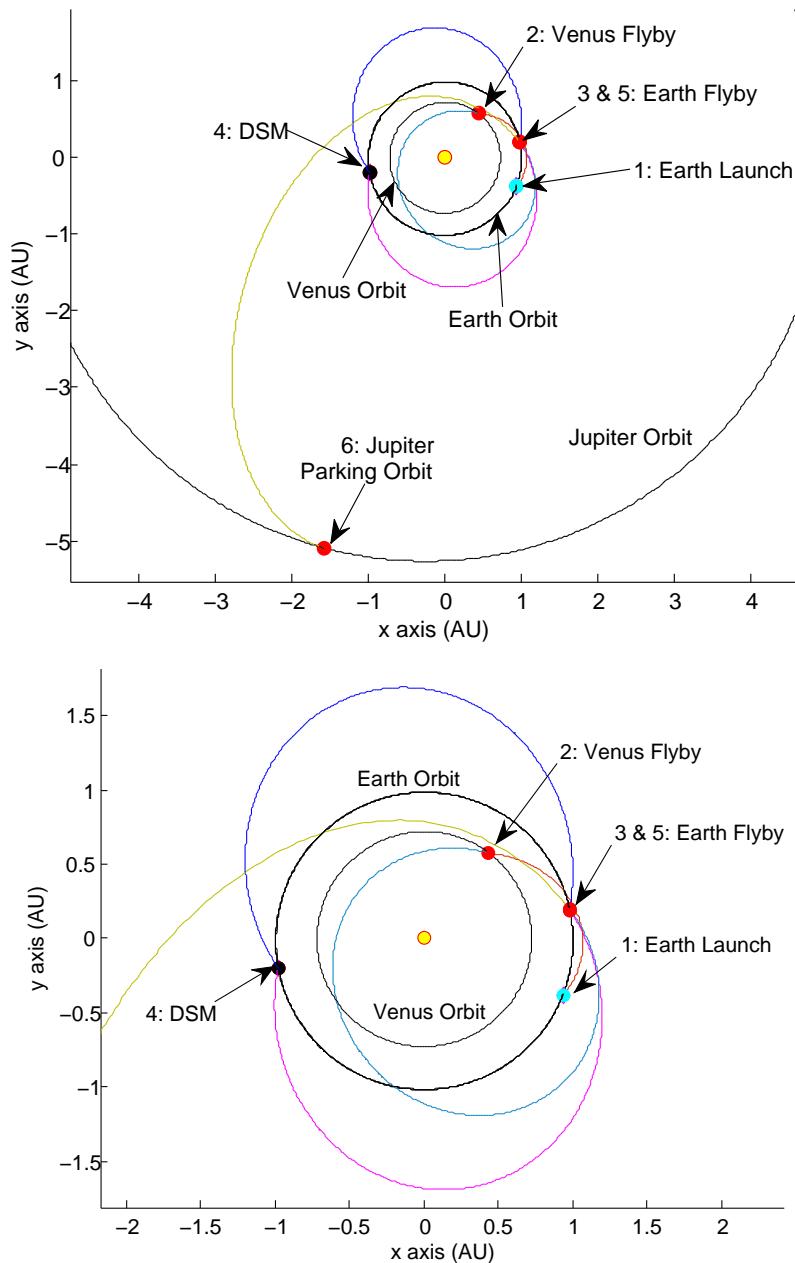


Figure 5.8: Galileo mission trajectory with rough initial guess parameters: September launch from Earth, Venus-Earth-Earth gravity assist, and capture into parking orbit at Jupiter (top), close-up inner planet trajectory view (bottom)

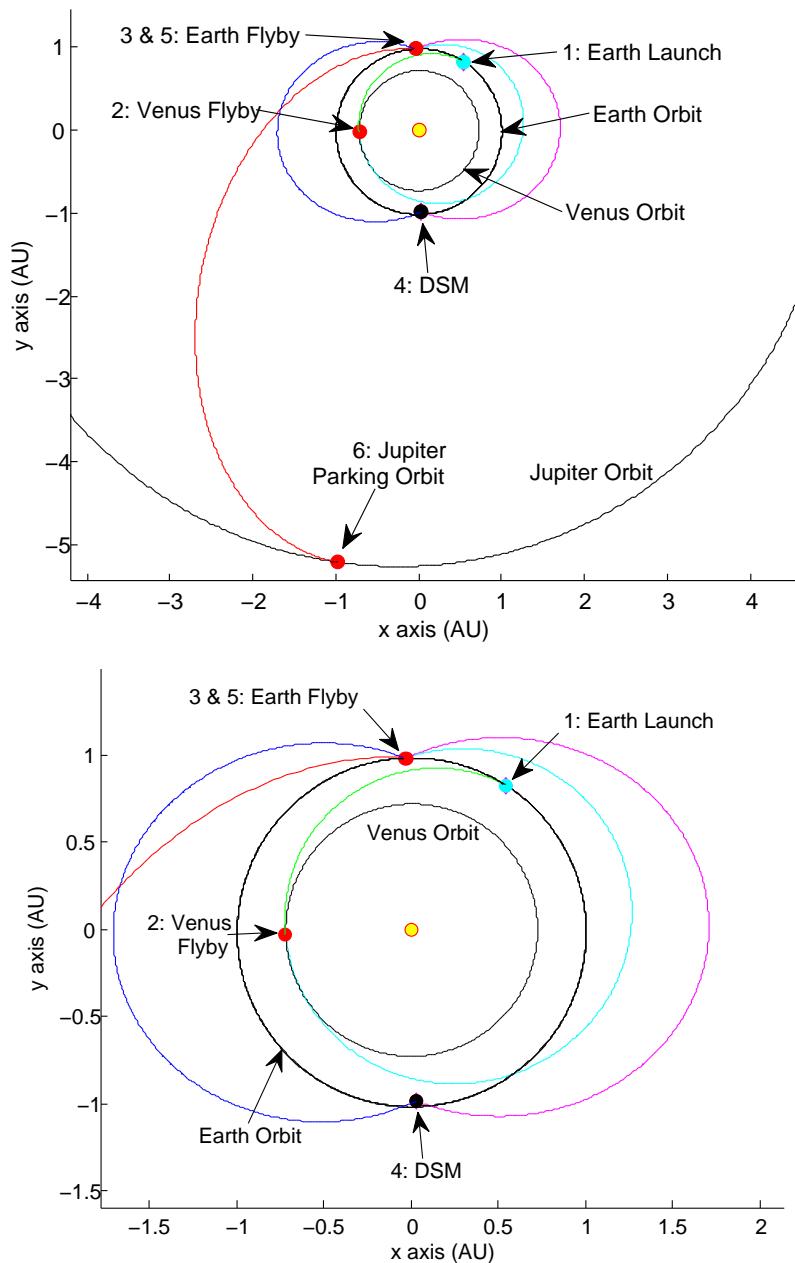


Figure 5.9: Galileo mission trajectory with rough initial guess parameters: November launch from Earth, Venus-Earth-Earth gravity assist, and capture into parking orbit at Jupiter (top), close-up inner planet trajectory view (bottom)

are estimated at rounded values to simulate the rough guesses by a mission planner, input as  $TOF = [ 100 \ 300 \ 365 \ 365 \ 1000 ]$  days. With each launch date, the Galileo trajectory optimization will use the same TOFs initial guess values.

The focus of these simulations was to use the fast and efficient asymptote corrected gravity assist model with the rough initial guess trajectory parameters. The fmincon interior-point optimization scheme performed the best for the rough initial guess values of the complex trajectory, shown in Table 5.7. The September launch date initial trajectory parameters are shown in the “Initial-Sep” column, while the resulting optimized trajectory values are shown beside, labeled “Asym-Sep.” The November launch date initial trajectory parameters are shown in the “Initial-Nov” column, while the resulting optimized trajectory values are shown beside, labeled “Asym-Nov.” Again, the actual Galileo trajectory values are shown for reference in the “Actual” column. The extremely large total  $\Delta V$  values for each rough initial trajectory are indicative of just how far from the optimal solutions these initial trajectory parameters were. Despite the poor starting points, TRACT was able to converge the trajectory to very similar solutions as those found with accurate initial guess values, which is also close to the actual mission values. Both of the different launch date cases converged upon good ballistic gravity assist estimates while also driving the DSM  $\Delta V$  to a negligible value. The optimized trajectory launch dates are just a few days from the actual mission, demonstrating the TRACT bi-level optimization capability of large and accurate optimization steps toward the optimal solution. The final optimized trajectory simulation plots for both cases are indistinguishable from the prior Galileo trajectory plots using accurate initial guesses. This successful trajectory simulation also demonstrates the use of

|                                      | Init-Sep | Asym-Sep | Init-Nov | Asym-Nov | Actual  |
|--------------------------------------|----------|----------|----------|----------|---------|
| MJD0 (days)                          | 47769.0  | 47812.29 | 47849.0  | 47811.23 | 47817.0 |
| TOF <sub>TOT</sub> (days)            | 2130.0   | 2324.19  | 2130.0   | 2328.46  | 2242.0  |
| $\Delta V_{1-Launch}(\frac{km}{s})$  | 4.09825  | 0        | 0        | 0        | 0       |
| $\Delta V_{2-Venus}(\frac{km}{s})$   | 4.28198  | 6.19E-06 | 1.02279  | 4.26E-07 | 0       |
| $\Delta V_{3-Earth1}(\frac{km}{s})$  | 22.88700 | 5.71E-07 | 18.12020 | 3.66E-07 | 0       |
| $\Delta V_{4-DSM}(\frac{km}{s})$     | 24.64652 | 4.52E-04 | 25.51214 | 8.25E-04 | -       |
| $\Delta V_{5-Earth2}(\frac{km}{s})$  | 5.33476  | 8.40E-07 | 12.07847 | 3.10E-06 | 0       |
| $\Delta V_{6-Jupiter}(\frac{km}{s})$ | 0.85425  | 0.74839  | 0.72463  | 0.74981  | 0.630   |
| $\Delta V_{TOT}(\frac{km}{s})$       | 62.103   | 0.749    | 57.458   | 0.751    | 0.630   |
| FuncEval                             | 1        | 2022     | 1        | 1871     | -       |
| Iterations                           | 0        | 176      | 0        | 158      | -       |
| Run Time (sec)                       | 0.014    | 12.628   | 0.014    | 12.203 s | -       |

Table 5.7: Galileo mission trajectory simulation rough initial guess cases using fmincon interior-point algorithm

including DSMs in a trajectory as a method of desensitizing certain optimization or trajectory dynamics and allowing a more stable convergence toward an optimal desired trajectory solution.

## 5.5 Extension of asymptote $\Delta V$ for other constraint handling

An unexpected outcome from the powered gravity assist analysis is the development of a fast algorithm for handling unpowered ballistic gravity assists. By always using the asymptote  $\Delta V_\infty$  correction maneuver as a penalty included in the performance index and assuming an asymptote maneuver powered gravity assist, the algorithm automatically handles the unpowered ballistic gravity assist conditions. The algorithm drives the interplanetary trajectory optimization to find ballistic flyby configurations, where the asymptote  $\Delta V$  maneuver reduces to zero. The following extensions of this research are preliminary studies and not fully devel-

oped, but are intended for future work. Even so, some general characterizations and comparisons can be made based on the completed research without necessarily having the current extension topic fully realized.

The periapse impulse problem can be modified to enforce the  $r_{min}$  constraint while still maintaining a collinear periapse maneuver by applying a correction  $\Delta V_\infty$  maneuver at one asymptote. Even though this configuration has two maneuvers during the flyby encounter, the intent of the algorithm is for the interplanetary trajectory optimization routine to find flyby configurations that drive this second maneuver at the asymptote to zero (or even drive both maneuvers to zero). Such an algorithm would only come into affect when the original periapse impulse solution falls below the  $r_{min}$  constraint. There are a variety of methods to determine the collinear periapse  $\Delta V$  magnitude: using the infeasible  $\Delta V$  computed at the beginning of the algorithm, using a new estimate determined from heuristics or optimal control theory, or even simply using a predetermined  $\Delta V$  magnitude. Despite the method of determining the adjusted magnitude of the periapse maneuver, the resulting hyperbolic transfer leg inherently cannot meet the required  $\vec{V}_\infty^\pm$  boundary conditions without applying a second  $\Delta V$  maneuver somewhere along the trajectory. Assuming the second maneuver occurs at the asymptote provides the advantages seen in the ballistic gravity assist algorithm in the previous section. In this manner, the gravity assist configurations that breach the  $r_{min}$  constraint are penalized with a greater total  $\Delta V$ . Also, the asymptote  $\Delta V_\infty$  correction maneuver has a sensitivity proportional to the  $\vec{V}_\infty^\pm$  configuration, which persuades an optimization scheme to reduce this penalty  $\Delta V$ , as seen in the ballistic gravity assist algorithm.

The periapse maneuver magnitude can also be monitored to have less than,

greater than, or equal to a predetermined value, and using the same methodology of applying the predetermined  $\Delta V$  tangentially, determining an appropriate  $r_p$ , and applying a correction  $\Delta V_\infty$  at an asymptote to complete the gravity assist, shown in Figure 5.10. This would effectively handle inequality and equality constraints on various parameters of the powered gravity assist problem.

Considering the optimal gravity assist problem, implementing similar  $\Delta V$  constraints has more degrees of freedom when applying the predetermined  $\Delta V$  maneuver magnitude with a specified direction and trajectory location, Figure 5.11. Though further investigation would likely produce a more efficient strategy, a direct method would be to apply the predetermined  $\Delta V$  in the same direction and maneuver location as the optimal solution that was later determined to be infeasible. Then the flyby is completed with the necessary asymptote  $\Delta V_\infty$ .

Another extension of constraint handling with an asymptote maneuver is to allow only discrete  $\Delta V$  levels, simulating solid rocket motor options. These  $\Delta V$  levels model the performance levels of various solid rocket motors available to a spacecraft mission. The powered gravity assist would be executed in a similar manner as above for either the periapse maneuver flyby method or the optimal powered flyby method. Then the original  $\Delta V$  maneuver is replaced by the closest predetermined  $\Delta V$  level available. The associated maneuver parameters (such as the maneuver direction) would be determined by previous estimates or other methods, then the remaining hyperbolic trajectory would be completed by the asymptote  $\Delta V_\infty$  correction maneuver.

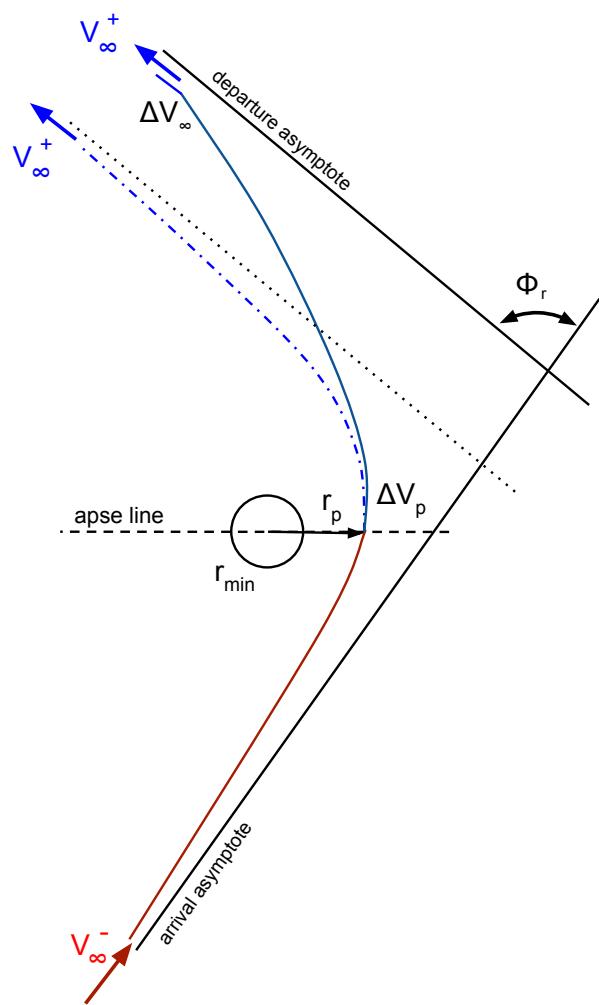


Figure 5.10: Periapse impulse powered gravity assist with  $\Delta V$  constraint applied with a penalty  $\Delta V$  maneuver, showing infeasible trajectory (dashes) and feasible trajectory (solid)

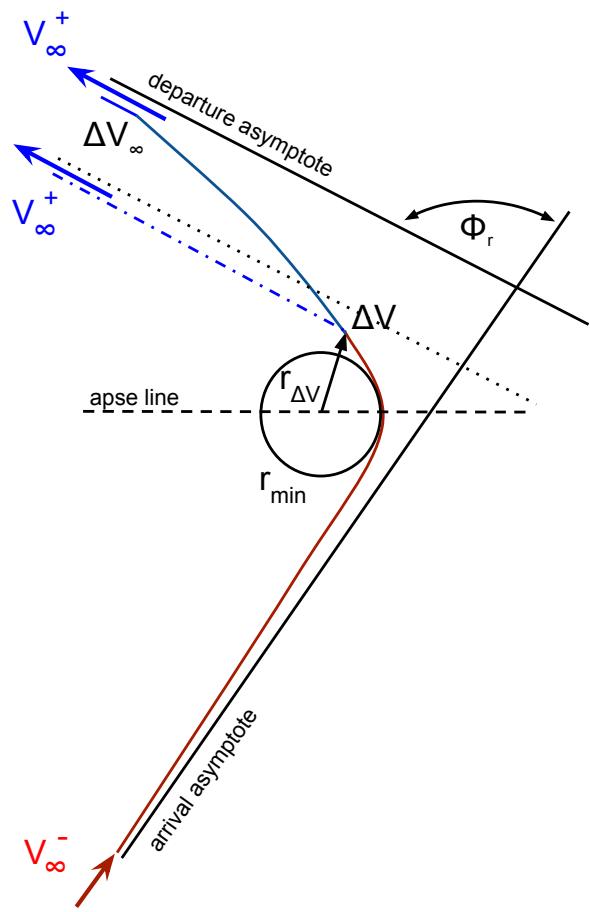


Figure 5.11: Optimal impulse powered gravity assist with maximum  $\Delta V$  constraint applied at  $r_{\Delta V}$  and a  $\Delta V_{\infty}$  correction maneuver applied at the asymptote, showing infeasible trajectory that requires too large of  $\Delta V$  at  $r_{\Delta V}$ (dashes) and feasible trajectory using an acceptable  $\Delta V$  magnitude at  $r_{\Delta V}$  (solid)

## 5.6 Chapter conclusions

The major benefits of the asymptote maneuver for constraint handling is the potential computational efficiency compared with alternative methods, as well as its tangible measurement of multiple trajectory constraints. This performance benefit was demonstrated in the Juno and Galileo trajectory examples using the asymptote corrected ballistic gravity assist model. Since the asymptote  $\Delta V$  penalty/correction maneuver applies directly to the performance index function, it can easily be transferred to most other optimization techniques, such as unconstrained optimization routines, simplex methods, genetic algorithms, etc. The asymptote maneuver models can also be applied concurrently with nonlinear constraints for a more reliable and effective constraint strategy, while improving the computational efficiency and run time compared to implementing just the nonlinear constraints. This powerful strategy was demonstrated multiple times with in the Juno and Galileo trajectory examples, especially when utilizing fmincon's SQP optimization algorithm.

Also, the reduction of optimization parameters can aid in the efficiency and effectiveness of most optimization methods chosen. By using a consistent and relatable  $\Delta V$  metric for the constraint penalty, multiple complex trajectory constraints and maneuvers can be directly compared. These methods potentially make mission planning easier and managing trajectory requirements less cumbersome. At minimum, these algorithms and methods provide a useful alternative with a different perspective for trajectory design.

# Chapter 6

## Interplanetary Spacecraft Mission Case Studies

### 6.1 Recent Interplanetary Spacecraft Missions

#### 6.1.1 New Horizons mission example

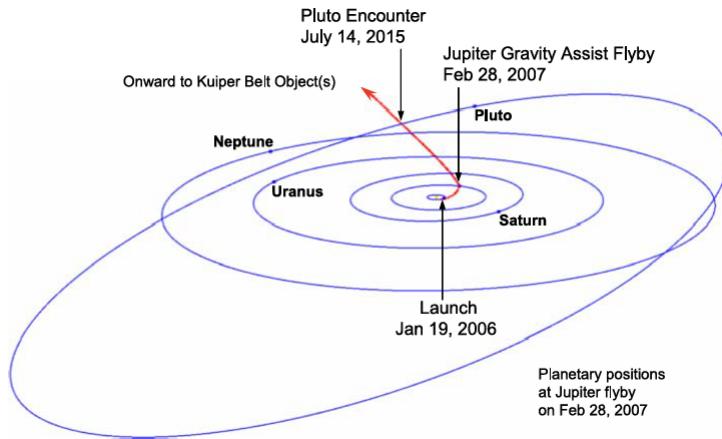


Figure 6.1: New Horizons mission trajectory: Launch from Earth, Jupiter gravity assist, and Pluto flyby [Source: Guo and Farquhar [8]]

The New Horizons mission to the Pluto system and the Kuiper belt beyond launched from Earth on January 19, 2006 with the highest Earth departure velocity to date, attaining a  $C_3 = 158 \text{ km}^2/\text{s}^2$ . The fast trajectory was aided by a Jupiter gravity assist, shortening the lengthy trip to Pluto to only 9.5 years, Figure 6.1 . The spacecraft will encounter the Pluto system with a heliocentric velocity reduced to about 14.5 km/s, as shown in Figure 6.2. After the Pluto system flyby, the New

Horizons spacecraft will continue into the Kuiper belt to study other small bodies as shown in Figure 6.3 [8].

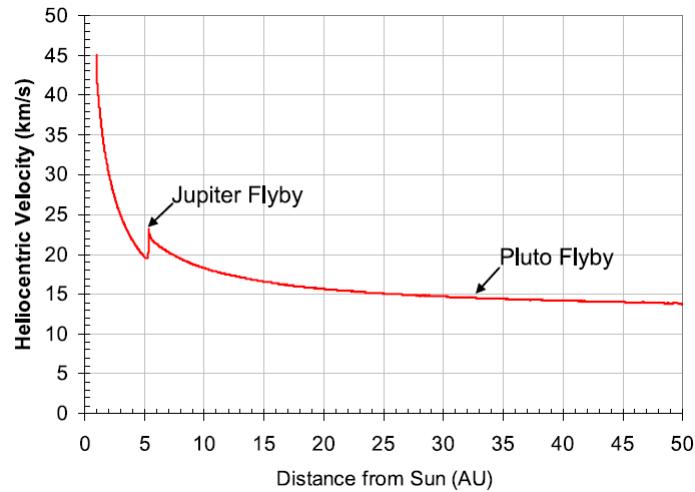


Figure 6.2: New Horizons mission heliocentric velocity plot throughout interplanetary trajectory [Source: Guo and Farquhar [8]]

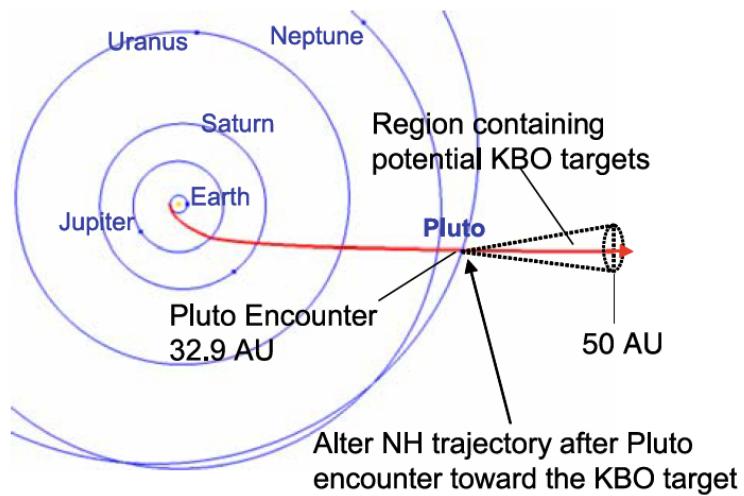


Figure 6.3: New Horizons mission trajectory to Kuiper belt post Pluto system flyby [Source: Guo and Farquhar [8]]

The New Horizons mission was simulated using optimization parameters initialized close to the actual mission trajectory with a launch date of January 19, 2006 and TOFs of [ 405 3058 ] days. In the following simulations, the launch is assumed to have a  $C_3 = 158 \text{ km}^2/\text{s}^2$  available at an inclination of  $28.5^\circ$ ; the minimum Jupiter flyby radius is maintained at 30 Jupiter radii for radiation concerns; and the Pluto flyby is modeled by hyperbolic arrival conditions with a maximum  $C_3 = 15 \text{ km}^2/\text{s}^2$  and a declination equal to that of the arrival  $\vec{V}_\infty^-$  vector. The simulation results that best matched the true interplanetary trajectory are shown in Table 6.1, where all of the presented trajectory solutions have negligible gravity assist  $\Delta V$  for a ballistic Jupiter flyby.

|  | Initial  | Simplex  | Int-Pt   | Int-Pt   | SQP      |
|--|----------|----------|----------|----------|----------|
| Flyby Model  | Asym     | Asym     | Constr   | Asym-Con | Asym     |
| MJD0 (days)  | 53754.79 | 53754.89 | 53751.45 | 53757.71 | 53756.06 |
| TOF <sub>TOT</sub> (years)                         | 9.48     | 9.55     | 9.66     | 9.55     | 9.63     |
| $\Delta V_{1-Launch}(\frac{\text{km}}{\text{s}})$  | 0.01085  | 0        | 0        | 0        | 0        |
| $\Delta V_{2-Jupiter}(\frac{\text{km}}{\text{s}})$ | 0.01005  | 1.16E-09 | 1.85E-12 | 9.46E-12 | 1.70E-07 |
| $\Delta V_{TOT}(\frac{\text{km}}{\text{s}})$       | 0.021    | 1.16E-09 | 1.85E-12 | 9.46E-12 | 1.70E-07 |
| Pluto $V_\infty$ (km/s)                            | 13.77    | 13.65    | 13.47    | 13.64    | 13.51    |
| Jupiter $r_p$ (radii)                              | 32.599   | 33.071   | 33.575   | 33.216   | 33.677   |
| Func Evals   | 1        | 151      | 107      | 133      | 150      |
| Iterations   | 0        | 76       | 12       | 16       | 15       |
| Run Time (sec)                                     | 0.014    | 0.720    | 1.239    | 1.408    | 1.215    |

Table 6.1: New Horizons mission trajectory simulation cases with best results

The table columns represent a different trajectory optimization case with the identified optimization scheme and flyby model. The optimization schemes included are fminsearch (“Simplex”), fmincon with Interior-Point algorithm (“Int-Pt”), and fmincon with SQP algorithm (“SQP”). The flyby models implemented

are the asymptote corrected gravity assist (Asym), the nonlinear ballistic flyby constraints, (Constr), and the combination of both asymptote corrected flyby model and the nonlinear constraints (Asym-Con). The actual trajectory performance values are omitted from the table to manage table width on the page. The actual MJD0 and total TOF are the same as in the “Initial” column and all of the actual  $\Delta V$  values are zero since each flyby was performed ballistically. The actual Jupiter flyby periapse radius was  $r_p = 32.25$  Jupiter radii and the actual Pluto encounter excess velocity was  $V_\infty = 13.78$  km/s .

Since the emphasis of this mission trajectory is to reach the Pluto system in a short amount of time, the best matched trajectories are considered the fminsearch with the asymptote transfer model and the fmincon interior-point scheme using the combined asymptote transfer and traditional gravity assist constraints. It is interesting to note that the fminsearch algorithm achieved such a good solution close to the initial guess in the fastest run time of only 0.72 seconds. All of these simulations produced nearly identical trajectories, but the fmincon interior-point case using the asymptote transfer model and gravity assist constraints solution trajectory is shown in Figure 6.4. All of the presented solution launch dates are very close to the actual, varying only up to about 3 days.

The ballistic gravity assist trajectory of New Horizons successfully modeled by TRACT is used as a baseline comparison for the next part of the mission example analysis. In order to demonstrate the capability of a powered gravity assist and its effect on a realistic mission trajectory, an alternate scenario is formed for the New Horizons mission. In this demonstration it is assumed that after the launch from Earth and approaching the Jupiter flyby, the mission planning and operations team

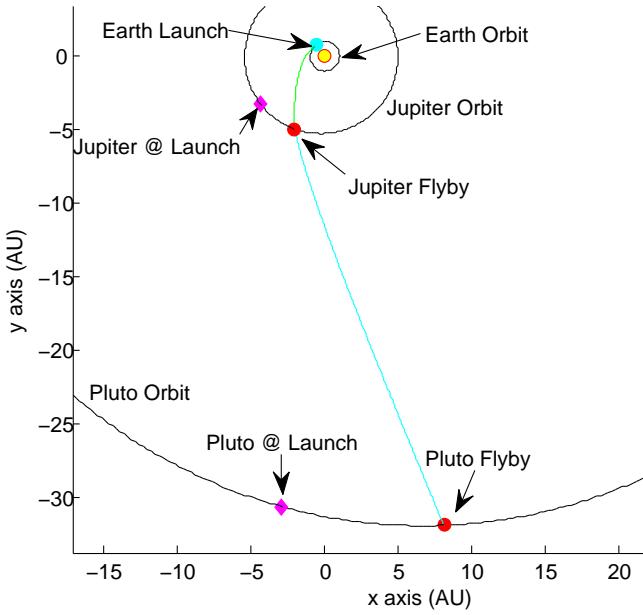


Figure 6.4: New Horizons mission TRACT simulation trajectory

have decided to implement a powered gravity assist at Jupiter to further reduce the flight time to the Pluto system. The need for this flight time reduction could be from a variety of causes, such as a failing subsystem onboard the spacecraft, newly discovered activity in the Pluto system that might be time sensitive, or unexpected mission funding problems, etc. The fmincon optimization scheme will be utilized along with the powered gravity assist models to achieve the reduced flight time. The New Horizons spacecraft has allocated the onboard propellant for a total navigation maneuver  $\Delta V = 242$  m/s, which assumes a portion of the propellant reserved for attitude control. If all of the onboard propellant were used for a single maneuver the maximum is  $\Delta V = 272$  m/s [77]. These maneuver values will be used as a maximum  $\Delta V$  for the powered gravity assist maneuver.

|  | Baseline | Allocated Nav | Max $\Delta V$ | 1 Year Saved |
|--|----------|---------------|----------------|--------------|
| Flyby Model                            | Asym-Con | Opt Flyby     | Opt Flyby      | Opt Flyby    |
| TOF <sub>TOT</sub> (years)             | 9.55     | 9.29          | 9.26           | 8.55         |
| TOF <sub>saved</sub> (months)          | -        | 3.12          | 3.44           | 12           |
| $\Delta V_2(\frac{km}{s})$             | 9.46E-12 | 0.242         | 0.272          | 1.130        |
| $v_{\Delta V}$ (deg)                   | -        | -0.85         | -0.95          | -3.83        |
| Jupiter $r_p$ (radii)                  | 33.216   | 31.639        | 31.476         | 27.091       |
| Jupiter $V_{\infty}^{-}(\frac{km}{s})$ | 18.420   | 18.443        | 18.443         | 18.443       |
| Jupiter $V_{\infty}^{+}(\frac{km}{s})$ | 18.420   | 18.722        | 18.756         | 19.760       |
| Pluto $V_{\infty}(\frac{km}{s})$       | 13.64    | 14.131        | 14.183         | 15.712       |
| Func Evals                             | 133      | 77            | 89             | 88           |
| Iterations                             | 16       | 9             | 12             | 12           |
| Run Time (sec)                         | 1.408    | 0.745         | 0.799          | 0.780        |

Table 6.2: New Horizons mission trajectory simulation cases for reduced flight time using powered gravity assist

The fmincon SQP optimization scheme was used along with the optimal powered gravity assist model (“Opt Flyby”) to quickly converge upon the constrained trajectory solution. To accommodate the scenario described, the baseline launch date was fixed with an equality constraint and the total flight time had an upper bound applied. The trajectory optimization routine was executed multiple times with manual changes to the total flight time upper bound, until the required powered gravity assist  $\Delta V$  was reduced to the prescribed levels. The results are shown in Table 6.2, comparing the baseline trajectory with the varying levels of  $\Delta V$  allowable. The smaller allowable maneuver was the  $\Delta V = 242\text{m/s}$  using only the allocated navigation propellant and when used in an optimal powered gravity assist about Jupiter, reduced the flight time to Pluto by over 3 months. This also increased the Pluto encounter  $V_{\infty}$  slightly. If the maximum available  $\Delta V = 272\text{m/s}$  was used in the Jupiter powered gravity assist, the flight time was reduced by over 3.4 months.

These slight flight time savings may be useful to the overall mission design scenario, but are restricted due to the limited onboard propellant. If the trajectory were to save 1 year off the flight time, the required powered gravity assist maneuver is  $\Delta V = 1.13\text{km/s}$ . This level of  $\Delta V$  is typically available on large spacecraft missions and therefore might be a feasible option for such missions. The interplanetary trajectories with flight time savings of about 3 months are nearly indistinguishable from the baseline, therefore the interplanetary trajectory for the 1 year reduction case is provided in Figure 6.5 for comparison.

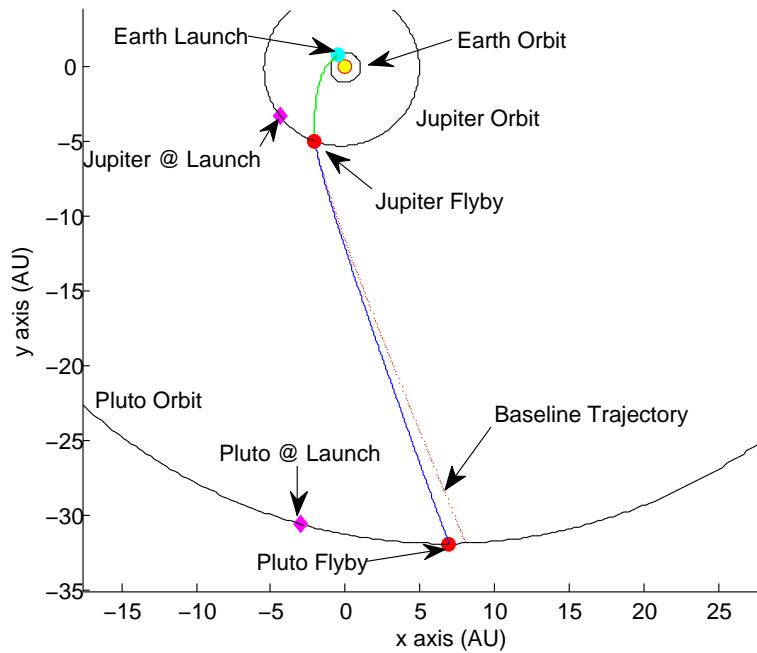


Figure 6.5: New Horizons mission trajectory comparison of 8.55 year flight time case (blue) with 9.55 year baseline trajectory (red)

### 6.1.2 MESSENGER mission example

The MESSENGER mission to the inner most planet Mercury was launched on August 3, 2004 and recently concluded the mission with the spacecraft impact on the surface of Mercury. A Mercury orbiter mission is extremely difficult due to the incredible amount of energy change required to reach Mercury with sufficiently low velocity to be able to insert into a parking orbit. The MESSENGER mission strategy was to utilize multiple gravity assists of Earth, Venus, and Mercury along with  $V_\infty$ -leveraging maneuvers to efficiently reduce its heliocentric orbital energy. The circuitous 6.6 year trajectory completes many revolutions (15.3) about the Sun as the spacecraft's orbital period is reduced with each gravity assist [9]. Simulating MESSENGER's spacecraft trajectory to Mercury demonstrates TRACT's capability to handle difficult trajectories and its adaptability with using multiple optimization schemes to progressively converge upon a near ballistic trajectory solution.

To display the MESSENGER mission trajectory, it is helpful to breakup the trajectory into 3 sections and view them using the excellent diagram excerpts from "MESSENGER Mission Design and Navigation" [9]. The MESSENGER spacecraft was launched with a  $C_3 = 16.38 \text{ km}^2/\text{s}^2$  into a 1-year Earth synchronous trajectory, where the first Earth gravity assist is executed at the next encounter. The Earth gravity assist reduced the perihelion below Venus's orbit, where the first DSM takes place  $\Delta V_{DSM1} = 315.6 \text{ m/s}$ , Figure 6.6. Next two consecutive Venus gravity assists occur, followed by DSM 2 of  $\Delta V_{DSM2} = 223.7 \text{ m/s}$  and a Mercury gravity assist, Figure 6.7. From this point on in the spacecraft trajectory, every Mercury encounter is preceded by a  $V_\infty$ -leveraging DSM, as seen in Figure 6.8. Two more DSMs and Mercury gravity assist combinations take place, where

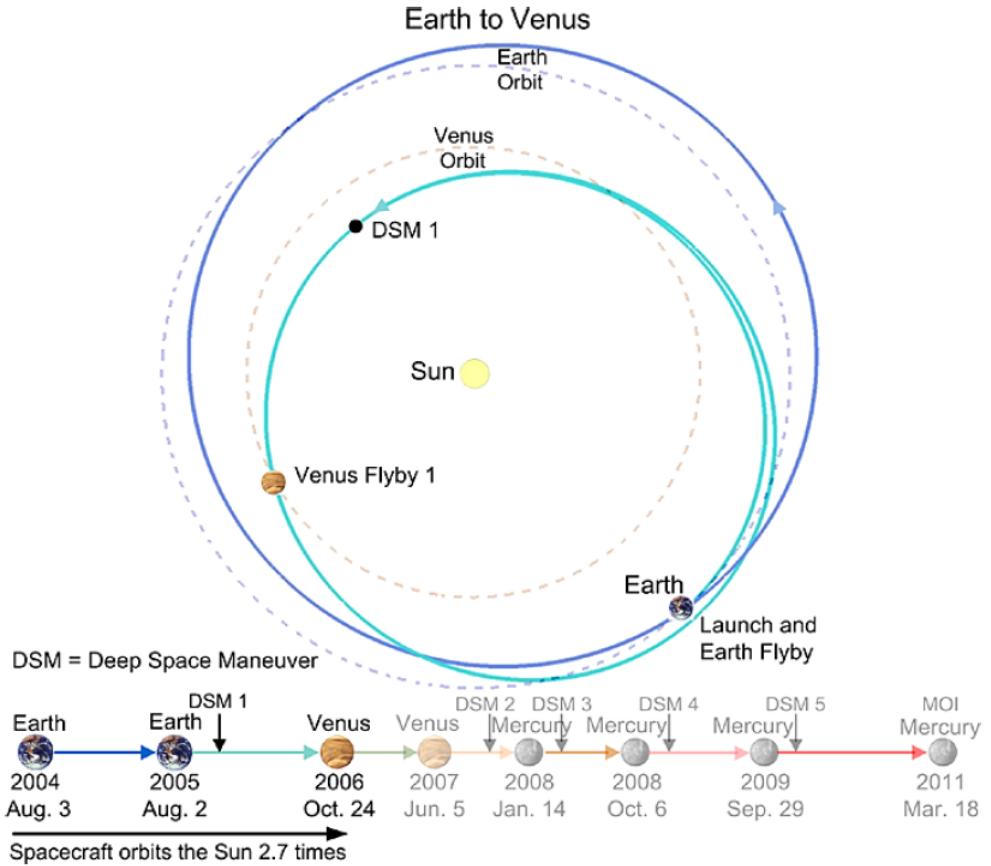


Figure 6.6: MESSENGER mission trajectory first section: Launch from Earth, Earth flyby after 1 year, DSM 1 near perihelion, and first Venus flyby [Source: McAdams et al. [9]]

$\Delta V_{DSM3} = 73.4\text{m/s}$  and  $\Delta V_{DSM4} = 241.8\text{m/s}$ . Then a final DSM of  $\Delta V_{DSM5} = 177.5\text{m/s}$  directs the MESSENGER spacecraft for Mercury orbital insertion (MOI) with  $\Delta V_{MOI} = 862\text{m/s}$ . The high propellant requirement for the MESSENGER mission totals  $\Delta V_{tot} = 2.229\text{km/s}$ , whereof  $1.032\text{km/s}$  is allocated to DSMs,  $0.115\text{km/s}$  to launch corrections,  $0.946\text{km/s}$  to MOI + orbit-corrections, and  $0.136\text{km/s}$  for contingency.

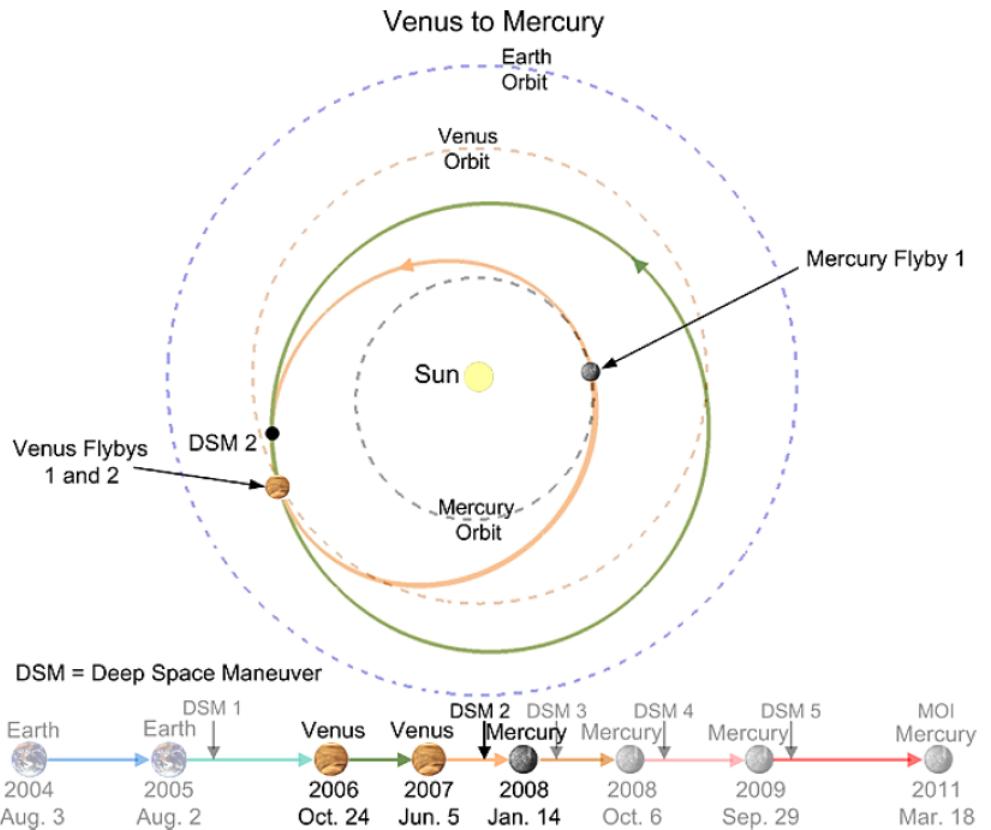


Figure 6.7: MESSENGER mission trajectory second section: After first Venus flyby, second Venus flyby after full Venus orbit (synchronous), DSM 2 performed just before completing a full solar orbit, followed by first Mercury flyby [Source: McAdams et al. [9]]

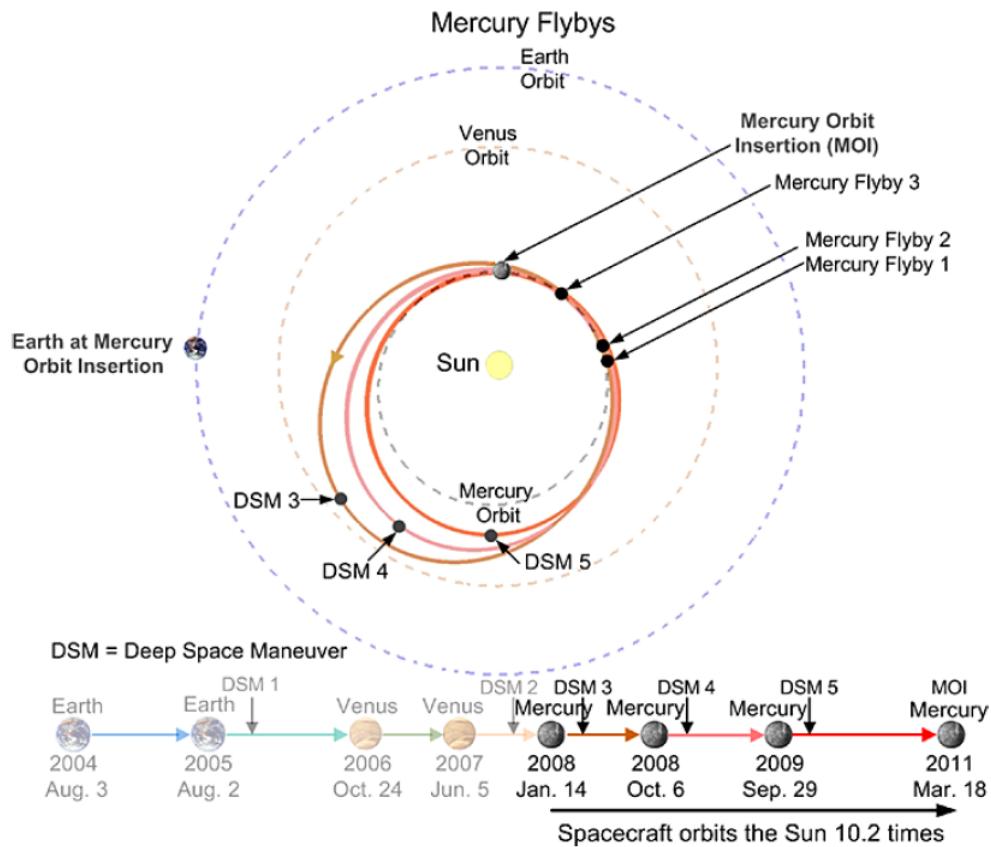


Figure 6.8: MESSENGER mission trajectory last section: After first Mercury flyby, DSM 3 occurs, then Mercury flyby 2 after about 1.5 solar orbits. DSM 4 is performed about 2.5 solar orbits before Mercury flyby 3, followed by DSM 5 and a 4.5 solar orbit waiting period before MOI. [Source: McAdams et al. [9]]

The MESSENGER trajectory is initialized in TRACT with the actual trajectory dates with a launch date of August 3, 2004 and the TOFs shown below in days. The launch is assumed to have  $C_3 = 16.4 \text{ km}^2/\text{s}^2$  at a declination of  $45^\circ$ . The MOI is assumed to occur at periapse into a 12 hour Mercury parking orbit with periapse altitude of 200 km and  $80^\circ$  inclination. The planetary flybys are assumed to have a minimum periapse radius of 1.05 radii of the encounter planet. The following initial DSM locations are estimated from the trajectory diagrams.

$$TOF = [ 364 \ 132 \ 316 \ 224.697 \ 134 \ 89 \ 63 \ 203 \ 61 \ 297 \ 61 \ 474 ] \quad (6.1)$$

$$\begin{aligned} DSM_1 &= [ -0.27 \ 0.56 \ -0.007 ] AU \\ DSM_2 &= [ -0.65 \ -0.15 \ 0.049 ] AU \\ DSM_3 &= [ -0.53 \ -0.43 \ 0.016 ] AU \\ DSM_4 &= [ -0.33 \ -0.53 \ -0.013 ] AU \\ DSM_5 &= [ -0.14 \ -0.545 \ -0.03 ] AU \end{aligned} \quad (6.2)$$

The Lambert multiple revolutions input vector associated with the TOFs trajectory segments is  $[ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ -1 \ 0 \ -2 \ 0 \ -4 ]$ , showing the right and left branch choices by positive and negative revolution integers respectively. The fmincon optimization scheme is used with the interior-point and SQP algorithms. The optimization was performed in two stages, where one optimization scheme and gravity assist model would be used until converged, then another would be implemented on where the first left off. It was found that the optimal powered gravity assist model quickly reduced the initial guess trajectory to nearly ballistic gravity assists for the entire trajectory by using the fmincon interior-point (I-P)

|  | Initial  | Stage-1  | Stage-2  | Actual   |
|--|----------|----------|----------|----------|
| Optimization   | -        | I-P      | I-P      | -        |
| Flyby Model  | Peri     | Opt      | Con      | -        |
| MJD0 (days)  | 53220.50 | 53218.40 | 53220.06 | 53220.50 |
| TOF <sub>TOT</sub> (days)                                      | 2418.70  | 2418.84  | 2417.80  | 2418.70  |
| $\Delta V_1 - \text{Launch} (\frac{\text{km}}{\text{s}})$      | 0.71944  | 0.01012  | 0.03311  | 0        |
| $\Delta V_2 - \text{Earth1} (\frac{\text{km}}{\text{s}})$      | 3.96464  | 0.01744  | 1.37E-08 | 0        |
| $\Delta V_3 - \text{DSM1} (\frac{\text{km}}{\text{s}})$        | 0.20063  | 0.16309  | 1.95E-04 | 0.3156   |
| $\Delta V_4 - \text{Venus1} (\frac{\text{km}}{\text{s}})$      | 1.89412  | 0.00097  | 1.96E-07 | 0        |
| $\Delta V_5 - \text{Venus2} (\frac{\text{km}}{\text{s}})$      | 3.04882  | 0.26861  | 1.39E-07 | 0        |
| $\Delta V_6 - \text{DSM2} (\frac{\text{km}}{\text{s}})$        | 0.91151  | 0.38555  | 0.76581  | 0.2237   |
| $\Delta V_7 - \text{Mercury1} (\frac{\text{km}}{\text{s}})$    | 0.30568  | 0.05097  | 4.77E-08 | 0        |
| $\Delta V_8 - \text{DSM3} (\frac{\text{km}}{\text{s}})$        | 0.14305  | 0.06279  | 0.05663  | 0.0734   |
| $\Delta V_9 - \text{Mercury2} (\frac{\text{km}}{\text{s}})$    | 0.24691  | 0.03911  | 1.40E-07 | 0        |
| $\Delta V_{10} - \text{DSM4} (\frac{\text{km}}{\text{s}})$     | 0.27007  | 0.23212  | 0.23905  | 0.2418   |
| $\Delta V_{11} - \text{Mercury3} (\frac{\text{km}}{\text{s}})$ | 0.69804  | 0.01628  | 3.99E-08 | 0        |
| $\Delta V_{12} - \text{DSM5} (\frac{\text{km}}{\text{s}})$     | 2.58777  | 0.23768  | 0.17910  | 0.1775   |
| $\Delta V_{13} - \text{MOI} (\frac{\text{km}}{\text{s}})$      | 1.05807  | 0.84595  | 0.83217  | 0.862    |
| $\Delta V_{TOT} (\frac{\text{km}}{\text{s}})$                  | 16.049   | 2.331    | 2.106    | 1.894    |
| FuncEval   | 1        | 2180     | 5365     | -        |
| Iterations   | 0        | 48       | 167      | -        |
| Run Time (sec)   | 0.015    | 21.167   | 55.389   | -        |

Table 6.3: MESSENGER mission trajectory simulation stages

scheme, shown in Table 6.3.. The initial trajectory is shown with gravity assists modeled by a periapse maneuver. The major difficulty with optimizing this trajectory are the many multiple revolution Lambert trajectory solutions, where too large of a step in a trajectory leg's TOF may result in the multiple revolution trajectory unachievable for the desired number of revolutions. When this occurs the Lambert targeter fails to generate a trajectory solution and the leg must be recomputed with an adjusted number of revolutions. This action can disrupt the normally smooth optimization performance. The traditional ballistic gravity assist constraints method and the asymptote transfer model were especially affected by this at the initialization point. Although the optimal powered gravity assist seemed to be much less affected by these initial Lambert disruptions.

The results from the optimal powered gravity assist model were the starting values for the next optimization trial using the traditional gravity assist constraints with fmincon and the interior-point algorithm. This easily converged the results leftover from the optimal powered gravity assist model. An interesting feature in the final trajectory solution is that the first DSM maneuver is essentially removed from the trajectory, while DSM 2 grows to compensate. The other DSM maneuvers and MOI are in reasonable agreement with the actual MESSENGER trajectory values. Although there are notable differences in the dynamics of the actual MESSENGER trajectory from the two-body linked conics used here, a similar ballistic gravity assist trajectory solution was found using TRACT and a variety of gravity assist models. The final total  $\Delta V$  from the TRACT optimization process was about 2.1 km/s while the actual mission required about 1.9 km/s, giving only a 10% error in the estimate. The resulting 2.1 km/s  $\Delta V$  is still within the allocated mission  $\Delta V$

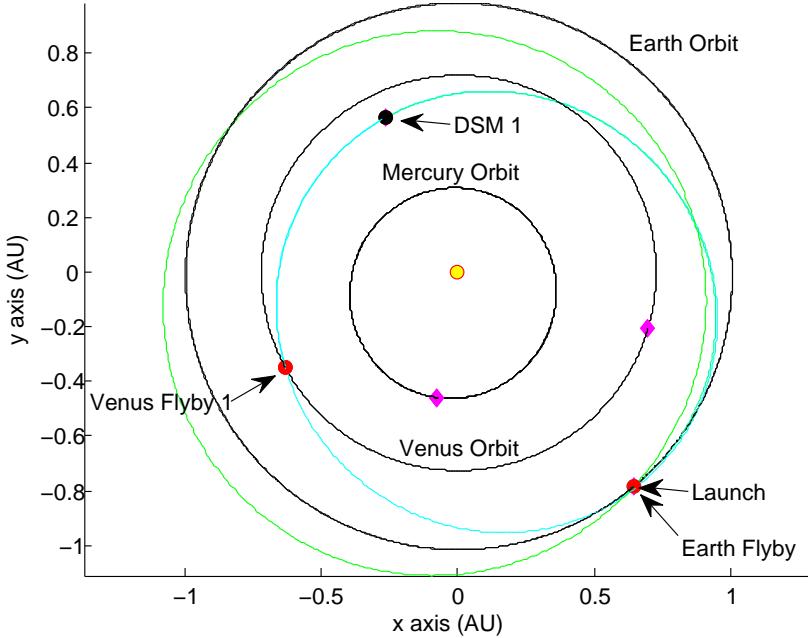


Figure 6.9: MESSENGER mission trajectory simulation first section, similar to the actual trajectory except the shift in the first leg: Launch from Earth, Earth flyby after 1 year, DSM 1 near perihelion, and first Venus flyby

budget of over 2.2 km/s. The initial guess trajectory started the optimization process with a very large 16 km/s total  $\Delta V$ . Even with relatively accurate MESSENGER initial guess trajectory data, the slight misalignments in the trajectory resulted in very large  $\Delta V$  values for many of the DSM and gravity assist maneuvers. Considering the starting trajectory performance value, the converged trajectory results reduced the total  $\Delta V$  by a factor of nearly 8. This demonstrates the robust and efficient optimization capability of TRACT.

The only major visible difference in the trajectory simulation compared to the actual MESSENGER trajectory is seen in Figure 6.9, where the Earth depar-

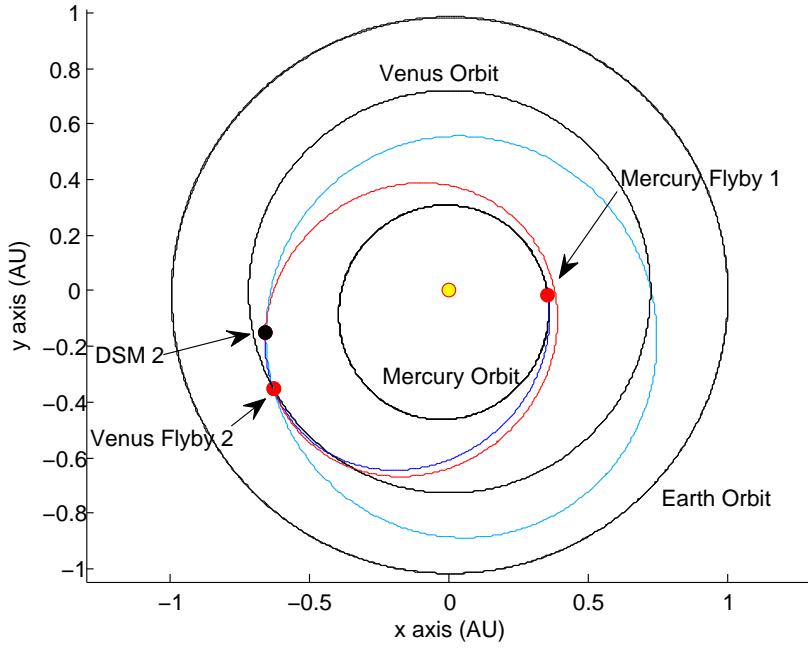


Figure 6.10: MESSENGER mission trajectory simulation second section, very similar to the actual trajectory: After first Venus flyby, second Venus flyby after full Venus orbit (synchronous), DSM 2 performed just before completing a full solar orbit, followed by first Mercury flyby

ture leg after launch travels interior to Earth's orbit and crosses outside later before returning to encounter Earth's gravity assist. The actual trajectory departs from launch exterior to Earth's orbit and crosses inside later before the Earth flyby. With some minor manual reformulation this leg can be adjusted to match that of MESSENGER and likely converge to a better solution. But the goal of this demonstration is to see where the optimization procedures naturally drive the trajectory from the initial conditions. The remaining legs of the trajectory seen in Figures 6.10 and 6.11 appear to be very similar to the actual MESSENGER mission, which correlates with the very similar resulting total  $\Delta V$  values.

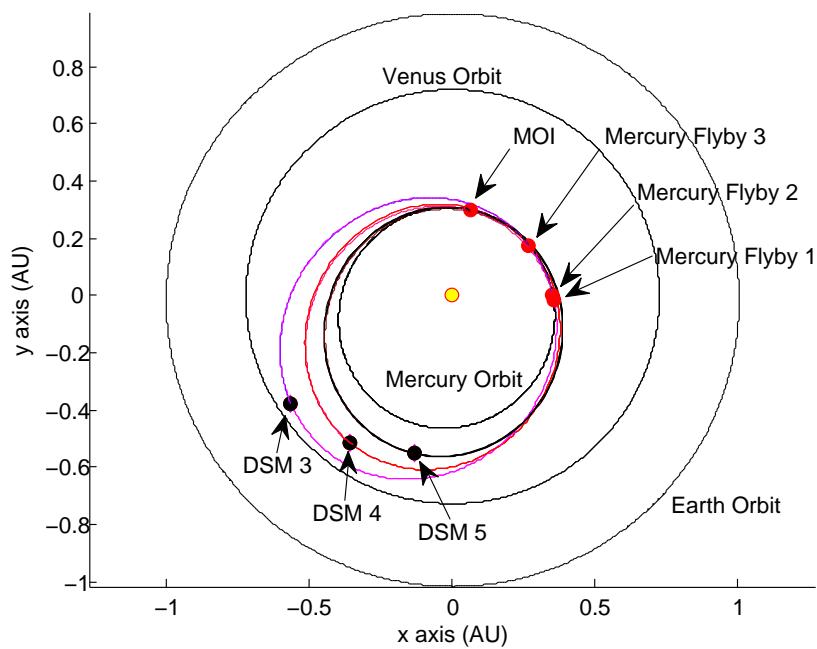


Figure 6.11: MESSENGER mission trajectory simulation last section, very similar to the actual trajectory: After first Mercury flyby, DSM 3 occurs, then Mercury flyby 2 after about 1.5 solar orbits. DSM 4 is performed about 2.5 solar orbits before Mercury flyby 3, followed by DSM 5 and a 4.5 solar orbit waiting period before MOI

### 6.1.3 Cassini mission example

The Cassini mission to Saturn was launched on October 15, 1997 and performed a series of gravity assists of a Venus, Earth, and Jupiter to efficiently reach Saturn while also reducing the launch vehicle energy requirements. The Cassini mission trajectory launched from Earth, performing a gravity assist of Venus just after perihelion to gaining energy to raise the aphelion to about 1.5 Au, shown in Figure 6.12. At aphelion the DSM  $V_\infty$ -leveraging maneuver of  $\Delta V_{DSM} = 450$  km/s is executed to lower the perihelion, enabling a more effective Venus gravity assist. Shortly after the second Venus encounter, an Earth gravity assist sends Cassini towards its last boost from a Jupiter gravity assist. Upon reaching Saturn, Cassini performs a Saturn orbital insertion (SOI) maneuver of  $\Delta V_{SOI} = 626$  km/s. In the following simulations, the launch is assumed to have a  $C_3 = 16.6 \text{ km}^2/\text{s}^2$  available at an inclination of  $28.5^\circ$ . The minimum Venus and Earth flyby altitudes are 280 and 800 km respectively, while the minimum Jupiter flyby radius is set at 30 Jupiter radii. The Saturn capture parking orbit is assumed with  $r_p = 1.3$  Saturn radii, a period of about 116 days, and a low inclination equal to the declination of the incoming  $V_\infty^-$ . It is also assume that the SOI maneuver occurs at periapse. [78, 67].

The Cassini mission trajectory simulations was performed using fmincon's SQP algorithm and the asymptote corrected gravity assist model. The actual Cassini mission values are use for the initial trajectory guess parameters, with a launch date of October 15, 1997, the flight times of  $TOF = [193 \ 220 \ 203 \ 54 \ 500 \ 1279]$  days, and a DSM location of  $[ -0.12 \ 1.57 \ 0.02 ] \text{AU}$  [78, 67]. The optimized trajectory results are provided in Table 6.4, showing the initial guess trajectory values in the “Initial” column. The first Cassini simulation was executed with the typ-

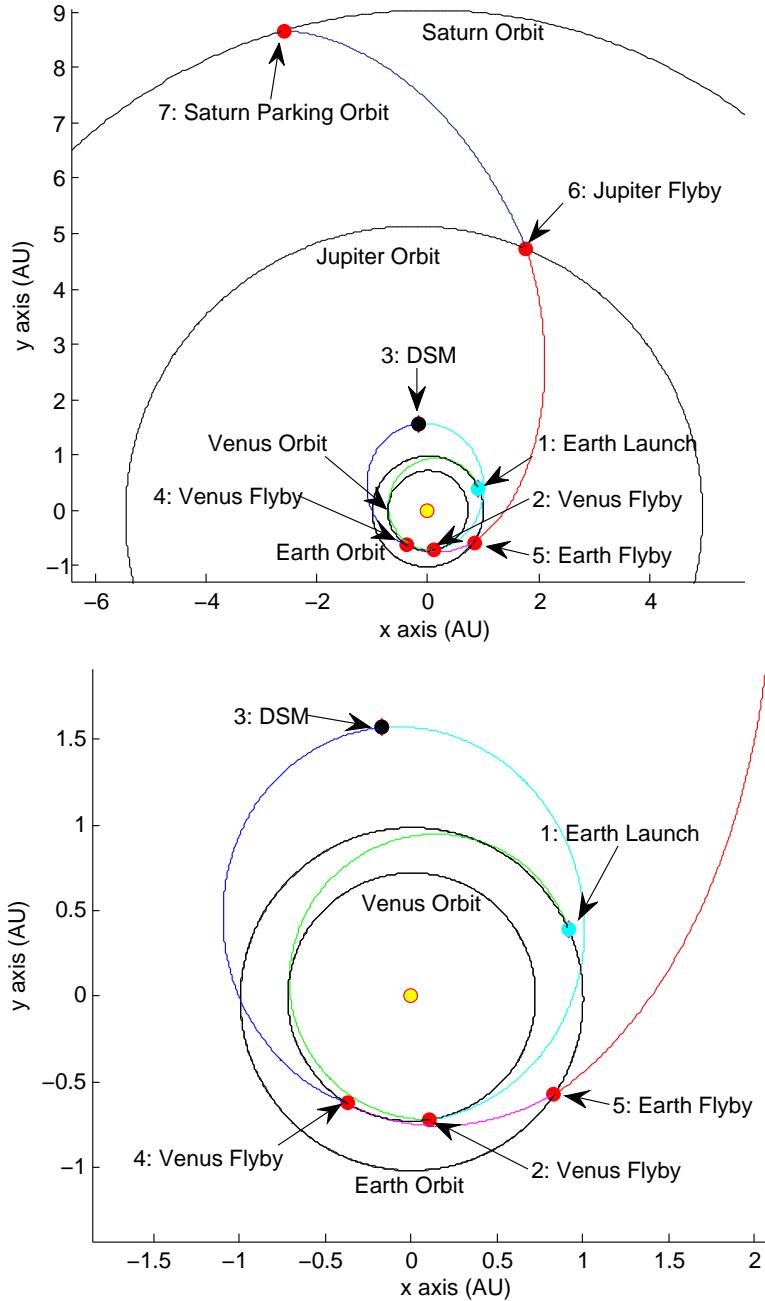


Figure 6.12: Cassini mission trajectory: Launch from Earth, followed by a Venus flyby, then a DSM maneuver at aphelion, a second Venus flyby, with an Earth flyby shortly after, and a Jupiter flyby before reaching Saturn for insertion into a highly eccentric capture orbit (top), close-up inner planet trajectory view (bottom).

|                                      | Initial  | Normalized | Fine-Tune | Actual   |
|--------------------------------------|----------|------------|-----------|----------|
| MJD0 (days)                          | 50736.36 | 50736.39   | 50737.54  | 50736.36 |
| TOF <sub>TOT</sub> (years)           | 6.71     | 11.66      | 6.72      | 6.71     |
| $\Delta V_{1-Launch}(\frac{km}{s})$  | 0        | 0          | 0         | 0        |
| $\Delta V_{2-Venus1}(\frac{km}{s})$  | 0.02835  | 3.81E-04   | 5.47E-05  | 0        |
| $\Delta V_{3-DSM}(\frac{km}{s})$     | 0.44956  | 0.37032    | 0.43740   | 0.450    |
| $\Delta V_{4-Venus2}(\frac{km}{s})$  | 0.06310  | 1.78E-08   | 2.50E-05  | 0        |
| $\Delta V_{5-Earth}(\frac{km}{s})$   | 0.18629  | 2.33E-03   | 1.18E-06  | 0        |
| $\Delta V_{6-Jupiter}(\frac{km}{s})$ | 0.17501  | 2.54E-03   | 3.27E-05  | 0        |
| $\Delta V_{7-Saturn}(\frac{km}{s})$  | 0.60122  | 0.41326    | 0.60485   | 0.626    |
| $\Delta V_{TOT}(\frac{km}{s})$       | 1.504    | 0.789      | 1.042     | 1.076    |
| FuncEval                             | 1        | 1759       | 597       | -        |
| Iterations                           | 0        | 117        | 36        | -        |
| Run Time (sec)                       | 0.014    | 12,842     | 4.566     | -        |

Table 6.4: Cassini mission trajectory simulation cases using fmincon SQP algorithm and the asymptote correction gravity assist model

ical normalized optimization parameter scheme, where each parameter is normalized by its initial guess values, as identified by the “Normalized” column. Without any constraints on flight times or dates, the optimization routine found a very interesting Cassini trajectory that actually saves about 290 m/s in total  $\Delta V$  compared to the actual Cassini mission. Although, this lower total  $\Delta V$  comes at a cost of dramatically higher total flight time of 11.7 years, compared to the actual 6.7 years. It is seen in Figure 6.13 that the inner-planet trajectory is nearly identical to the actual Cassini trajectory, but the major deviation occurs after the Jupiter flyby. In the actual Cassini mission the Jupiter flyby has a periapse at about 150 Jupiter radii, while this optimized trajectory converged upon a Jupiter encounter periapse of 48 Jupiter radii. This Jupiter gravity assist adjustment enabled a reduced  $\Delta V$  at Saturn orbital insertion.

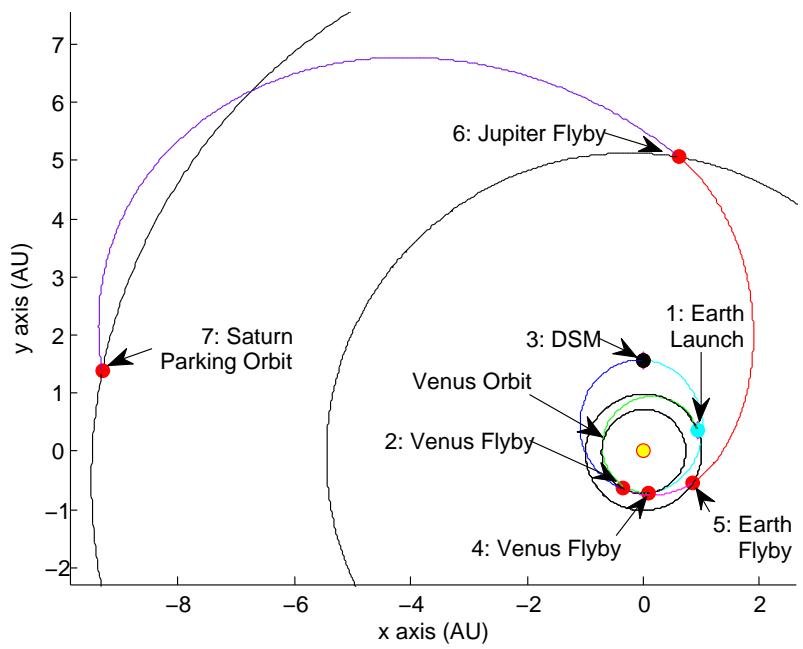


Figure 6.13: Cassini trajectory simulation depicting an 11.7 year flight time minimum  $\Delta V$  trajectory solution with 290 m/s lower total  $\Delta V$  compared to actual mission, starting from actual mission initial guess values.

The trajectory simulation was run again from the initial trajectory values but with a finer tuned scaling on the optimization parameters. This is done in order to converge upon a trajectory very similar to the actual Cassini mission and the results are shown in the “Fine-Tune” column for comparison. With the finer resolution, TRACT quickly converged upon the local minimum  $\Delta V$  solution near the initial guess values. These results demonstrate TRACT’s capability for finding an minimum total  $\Delta V$  trajectory solution. This deviation away from the actual Cassini mission values towards a lower  $\Delta V$  solution is enabled by TRACT’s bi-level optimization architecture, which allows the primary heliocentric optimization process to take large steps.

## 6.2 Student designed interplanetary missions

### 6.2.1 Orbiter mission to Jupiter’s moon Europa

A student spacecraft mission design team developed an ambitious mission to orbit Jupiter’s moon Europa. The mission required an efficient trajectory to reach Jupiter for insertion into a highly elliptical Jupiter capture orbit. TRACT was used to design the interplanetary trajectory by utilizing a similar strategy as the Juno mission with a  $V_\infty$ -leveraging DSM and Earth gravity assist to reach Jupiter quickly and efficiently, Figure 6.14. The optimized TRACT trajectory provides the mission with realistic launch dates, TOFs, and reasonably accurate estimates for the required  $\Delta V$  maneuver values. The student design team also utilized TRACT to develop a launch window analysis for the mission, showing the required  $C_3$  versus launch date to complete the mission. Including such an analysis greatly improves the maturity of the mission design. It also enables the students to make trade studies based on

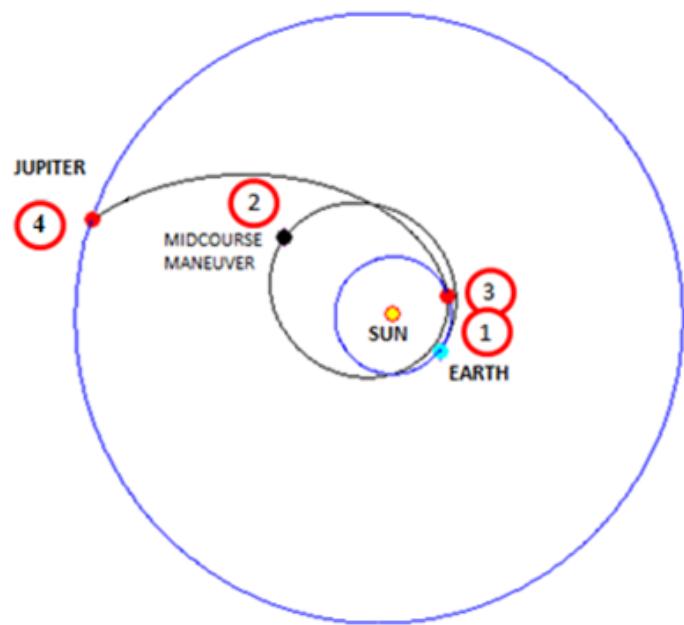


Figure 6.14: Student design team mission trajectory to Jupiter similar to Juno mission trajectory: Earth launch (1) followed by  $V_\infty$ -leveraging DSM (2) and Earth gravity assist (3) to transfer to Jupiter for insertion into capture orbit (4)

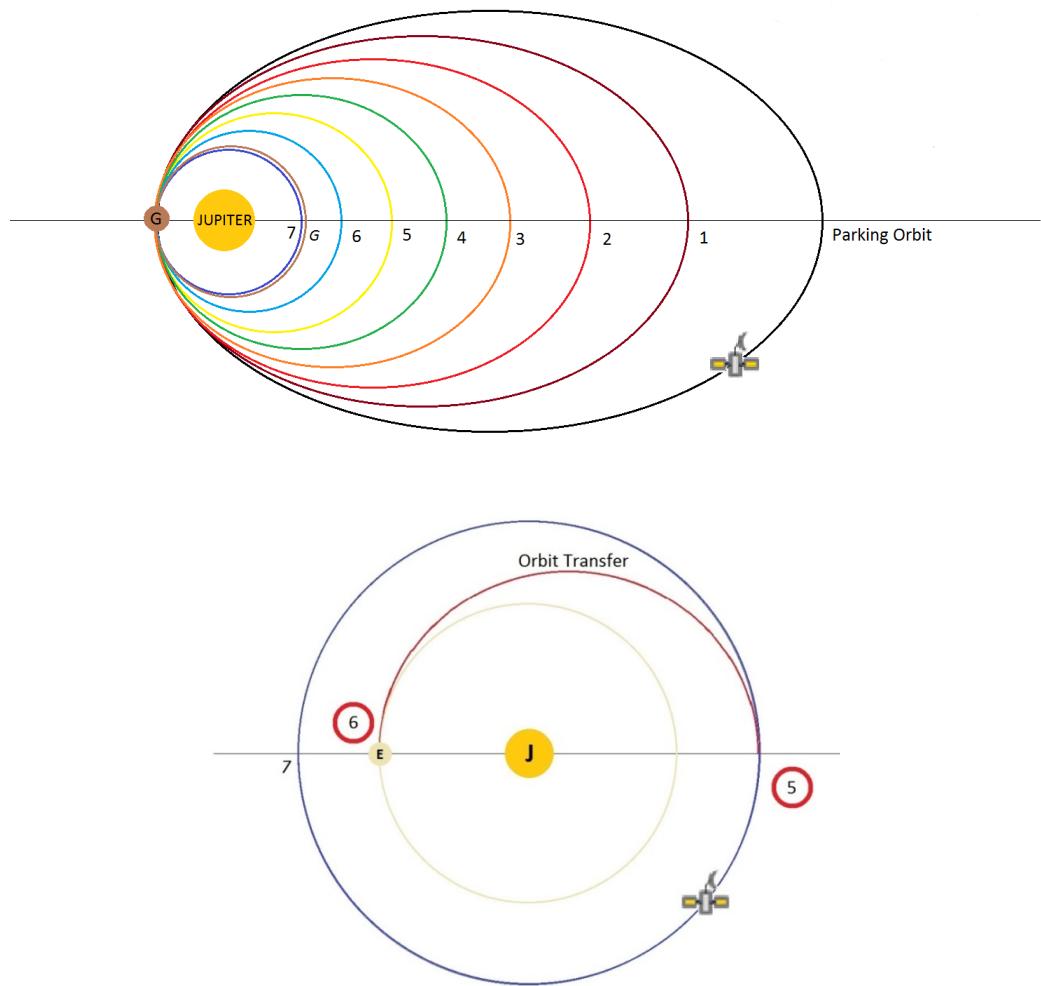


Figure 6.15: Example orbital lowering from capture orbit about Jupiter using Ganymede gravity assists shown above, followed by a DSM (5) and orbit insertion about Europa (6) shown below

simulation data and contend with realistic mission constraints.

The short learning curve for TRACT coupled with the ease of configuring a new trajectory allowed the student design team to quickly develop the interplanetary trajectory and requirements, allowing more time to enhance other facets of the mission design. This team was able to put additional effort into designing a Europa capture strategy using multiple consecutive Ganymede gravity assists followed by a transfer DSM and Europa orbit insertion maneuver, shown in Figure 6.15.

### 6.2.2 Flyby mission to dwarf planet Eris

Another student mission design team developed a flyby mission to the dwarf planet Eris. Since Eris orbits very far out in the Kuiper belt, the mission trajectory had a focus on reducing the flight time, which is very similar to the New Horizons mission to Pluto. Although the New Horizons mission reached Pluto at about 30 AU in only 9.5 years, the Eris mission needs to reach the dwarf planet at nearly 100 AU in a reasonable time frame. The primary limitation of the mission flight time was the selected ASRG (Advanced Stirling Radioisotope Generator) power supply component with a guaranteed lifetime of only 17 years.

Fortunately, the many TRACT optimization options provided a robust way to constrain the total flight time and find an efficient ballistic gravity assist trajectory that would quickly reach Eris, while maintaining a realistic launch energy. The reasonable upper limit launch energy was determined to be  $C_3 = 166 \text{ km}^2/\text{s}^2$  adopted from the New Horizons mission, which was a heavy influence on their mission and spacecraft design. The final interplanetary trajectory design is shown in Figure 6.16, with ballistic gravity assists of Jupiter and Neptune, resulting in a minimum

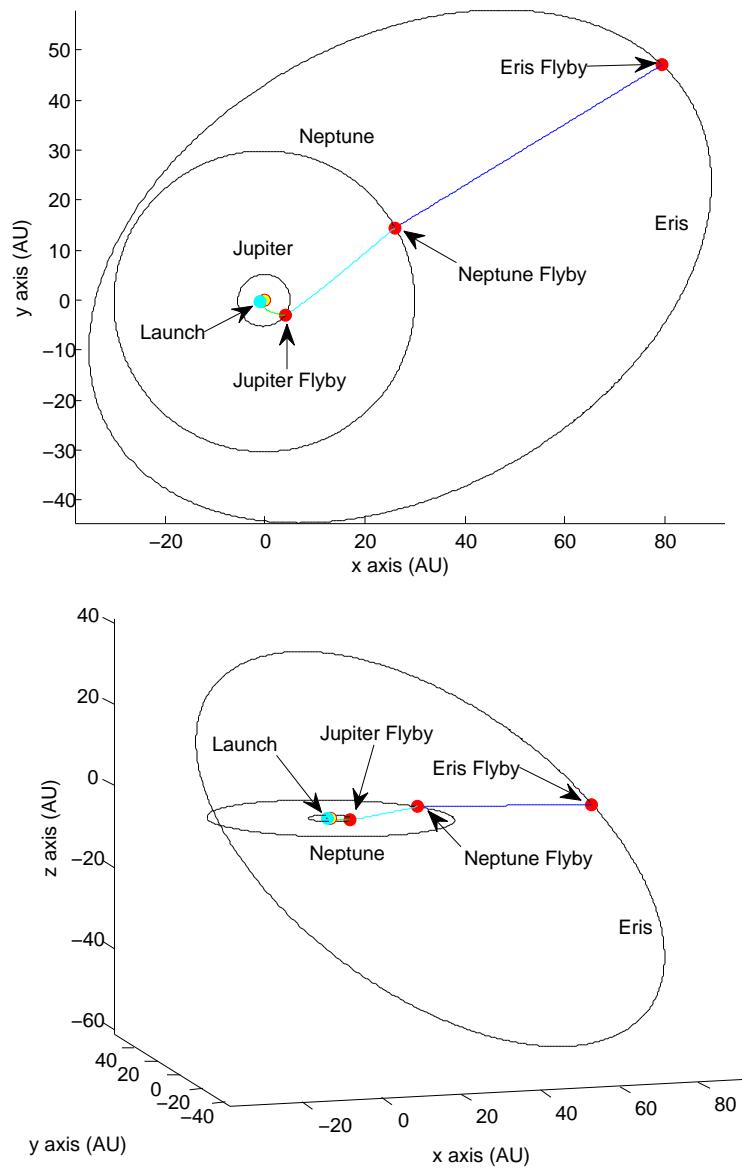


Figure 6.16: Student design team mission to dwarf planet Eris (overhead view above and out-of-plane view below): Launch from Earth followed by a Jupiter and Neptune gravity assist combination before reaching Eris in only 18 years

possible flight time of 18 years. The student design team formed a trade study analysis which resulted in the decision to postulate the development of a new version of the ASRG in development by NASA that with a longer lifespan to cover the 18 year mission and beyond. This enabling technology helps to satisfy the mission constraints, but the students had to downgrade the technology readiness level of the power system.

# **Chapter 7**

## **Conclusion**

### **7.1 Conclusions**

The research presented in this dissertation was motivated by the need for preliminary trajectory design tools and algorithms for interplanetary missions. The TRAjectory Configuration Tool (TRACT) was developed to meet the needs of novice and expert mission planners for quickly generating optimal complex preliminary interplanetary trajectories. TRACT is able to model many different mission maneuvers including various departure/arrival parking orbit options, deep space maneuvers (DSMs), launch vehicle transfers, and ballistic and powered gravity assists, etc. TRACT also includes preconfigured Excel data output and MATLAB trajectory simulations that are helpful in interpreting and presenting the mission trajectory solution.

TRACT was developed with a new bi-level trajectory optimization architecture that employs boundary value problem (BVP) algorithms to solve each heliocentric and planetocentric trajectory segments independently. The BVP methodology is the backbone of TRACT's modular architecture and is largely responsible for its success as an user friendly trajectory design tool. The bi-level architecture has been demonstrated to enable accurate large optimization steps. This feature provides a large convergence region and allows TRACT to better converge poor trajectory ini-

tial guesses to a feasible and lower  $\Delta V$  trajectory solution. TRACT utilizes a standardized parameterization of the interplanetary trajectory. This standard trajectory structure allows many optimization algorithms to be used in TRACT (specifically all of MATLAB's applicable optimization routines), including global optimization routines. TRACT is proven useful as an adaptable testbed for new trajectory BVP algorithms or optimization methods.

In order to provide TRACT with the most efficient trajectory options, the powered gravity assist problem was thoroughly researched. The literature reviewed did not present an effective optimal powered gravity assist solution method that could be realistically used in a spacecraft mission. Therefore a single-impulse powered gravity assist was developed to provide performance gains and to guarantee a mission feasible solution on every iterate. The solution space behavior for the single-impulse powered gravity assist was characterized. The performance of the new powered gravity assist algorithm was tested and verified with various flyby configuration conditions, proving its reliability and effectiveness.

The powered flyby research was also extended to developing an efficient algorithm to be used in trajectory optimization for producing ballistic gravity assists. The resulting algorithm is modeled as a powered gravity assist with the  $\Delta V$  maneuver restricted to occurring at one asymptote of the flyby at  $r_\infty$ . This asymptote correction maneuver only adjusts the asymptote  $\vec{V}_\infty$  to align the flyby. This asymptote maneuver acts as a penalty  $\Delta V$  for non-ballistic flybys, and drives the heliocentric trajectory optimization towards ballistic flyby configurations. This algorithm was proven to be an effective alternative to using the traditional nonlinear constraints for producing ballistic flybys.

The optimal powered gravity assist algorithm and the asymptote corrected gravity assist algorithm were integrated into TRACT. Extensive testing of these algorithms performance were done using heritage interplanetary mission trajectories. The asymptote correction flyby algorithm was successful in driving TRACT's trajectory optimization to converge on trajectories very similar to the actual missions. First, terminal convergence (how accurate the two-body model is compared to the actual mission) was tested using accurate trajectory initial guesses. Then, the effectiveness of TRACT as a preliminary mission planning tool was evaluated for the Galileo and Juno missions by starting with very poor initial guesses and observing the convergence. Convergence to values near those of the actual mission was found in both cases.

The optimal powered gravity assist algorithm also was proven as a useful tool for producing ballistic gravity assists. It was demonstrated as an effective trajectory maneuver option, providing the minimum  $\Delta V$  cost for a powered gravity assist mission trajectory. The use of DSMs as  $V_\infty$ -leveraging maneuvers was shown, as well as using DSMs as a method of desensitizing complex trajectory dynamics for a more stable optimization convergence to the desired trajectory solution.

The accessibility of these powered gravity assist algorithms and research as incorporated into TRACT, provides a powerful suite of trajectory design tools. A mission planner or trajectory designer using TRACT is able to effectively model a mission trajectory or make adjustments to an existing trajectory quickly. TRACT's adaptable user interface, many trajectory visualization options, and data availability facilitates an enhanced user interaction with the trajectory design process.

## 7.2 Contributions

The major contributions of this research to the areas of preliminary trajectory design and fundamental astrodynamics are listed below.

1. An interplanetary trajectory design tool (TRACT) has been designed, developed, and documented, using a new bi-level optimization formulation of two-body trajectory boundary value problems. The bi-level architecture provides efficient and robust trajectory optimization solutions with a reduced set of optimization parameters. TRACT offers enhanced accessibility and a short user learning curve, as well as providing an adaptable open source architecture in MATLAB for specialized user modifications.
2. A complete and rigorous characterization of the optimal single-impulse powered gravity assist problem is presented. The transfer maneuver solution types were organized into a convenient categorization scheme and key solution solution space features were identified for each category.
3. A new single-impulse powered gravity assist boundary value problem is formulated. The boundary value problem is then solved with an efficient and robust cubic interpolation optimization algorithm. The solution algorithm features strict enforcement of minimum and maximum flyby radius constraints.
4. The new powered assist algorithms have been integrated into the TRACT design tool.
5. The TRACT tool has been tested and verified by using it to re-create preliminary mission design trajectories for the Galileo, Cassini, Juno, New Horizons,

and MESSENGER missions. The resulting preliminary mission design trajectories successfully converge to trajectories that closely match the actual trajectory timelines, maneuvers, and propulsion requirements.

In addition, the research presents an alternative ballistic gravity assist constraint methodology and algorithm is developed by using penalty  $\Delta V$  maneuvers. The algorithm is also implemented in TRACT and analyzed with heritage mission trajectories, showing potential improvements in optimization performance.

### 7.3 Recommendations for Further Study

The research presented in this dissertation can be further extended with the following future work on powered gravity assist algorithms and trajectory design tools.

#### 7.3.1 Powered gravity assist algorithms

- Modify the periapse powered gravity assist BVP algorithm with strict  $r_{min}$  constraints using the asymptote penalty  $\Delta V$  correction maneuver strategy
- Research the different strategies for handling the powered gravity assist  $\Delta V$  magnitude constraints using the asymptote penalty  $\Delta V$  correction maneuver strategy
- Develop an efficient collinear powered gravity assist BVP algorithm utilizing hodograph analysis and relationships when applicable

- Research and characterize various interplanetary trajectory types that are enabled only by the use of powered gravity assists

### **7.3.2 TRACT modifications and improvements**

- Develop a graphical user interface as an alternative to the mission driver script interface
- Formulate the analytic gradients for improved optimization accuracy and efficiency
- Formulate  $n\pi$  Lambert transfer options with predefined plane preferences or automated plane optimization
- Incorporate additional optimization functions, such as SNOPT, differential evolution routines, etc.
- Incorporate a database of minor celestial body ephemerides, including asteroids, comets, etc.
- Incorporate additional DSM location initial guess generator options, such as for  $V_\infty$  leveraging maneuvers.
- Develop an outer-loop trajectory sequence optimization routine using TRACT as the inner-loop to determine each candidate trajectories performance

## **Appendices**

## Appendix A

### Strict $r_{min}$ constraint handling analysis for Type F flybys

The following analysis demonstrates the special occasions where the analytic Type F algorithm fails to meet the  $r_{min}$  constraint by producing a powered gravity assist trajectory with a periapse below  $r_{min}$ . This failure occurs when the flyby configuration is close to that of the maximum turning angle condition, where each leg of the powered gravity assist meets at their common periapse at  $r_{min}$ . This makes sense because these conditions have the periapse so close to the  $r_{min}$  threshold. The algorithm falters under these conditions because it is designed to determine the unconstrained periapse optimal powered gravity assist maneuver location without any consideration for the  $r_{min}$  constraint. The sorting procedure catches the obvious periapse constrained  $r_p = r_{min}$  cases by checking if  $\Phi_r > \Phi_p$ . Yet, determining if the  $\Phi_r < \Phi_p$  cases are constrained is more difficult to know without further computation. Therefore, the chosen procedure for the seemingly periapse unconstrained  $\Phi_r < \Phi_p$  cases is to first compute the analytic Type F algorithm solution and then check if the periapse is below  $r_{min}$ . For the unconstrained periapse sorted cases of  $\Phi_r < \Phi_p$  as the required turning angle  $\Phi_r$  increases towards  $\Phi_p$ , the optimal powered gravity assist has its periapse constrained by  $r_{min}$  just before reaching the transition point of  $\Phi_r = \Phi_p$ , where any greater  $\Phi_r$  and the flyby is

guaranteed to be periapse constrained as  $\Phi_r > \Phi_p$ .

The reference flyby conditions of a maximum turning angle powered flyby as defined in literature [2, 3, 4] has a turning angle  $\Phi_p$  for a common periapse maneuver at  $r_{min}$  and is used to demonstrate the conditions when a Type F  $\Phi_r < \Phi_p$  flyby case has its periapse breach the  $r_{min}$  constraint. The same investigation as above was performed for an Earth encounter but with the turning angle  $\Phi_r$  set as  $\Phi_p$  for the case of  $V_\infty^+ = 2.5\text{km/s}$ , where  $\Phi_r = \Phi_p = 144.1941^\circ$  assuming  $r_{min} = 1$  Earth radii. The plots in Figures A.1 and A.3 reveal that the unconstrained periapse single impulse solution may breach the  $r_{min}$  constraint when approaching the optimum powered flyby condition of a periapse maneuver at  $r_{min}$ . Though the required turning angle condition  $\Phi_r < \Phi_p$  suggests the flyby periapse may be unconstrained for these points (except the final point where  $\Phi_r = \Phi_p$  was defined), the actual results show that the unconstrained periapse (Type F) solution would fall below  $r_{min}$  if not constrained. The periapse constrained powered gravity assist algorithm developed in the dissertation is implemented in the following analysis to accommodate the special cases of  $\Phi_r < \Phi_p$  sorted flybys requiring their periapse to be restricted at  $r_{min}$ .

While the periapse is unconstrained, the optimal  $\Delta V$  solutions always have a lower magnitude than the periapse maneuver solutions and they grow farther apart as  $V_\infty^+$  increases farther from  $V_\infty^-$ , but the constrained periapse cases generate quite different relationships. A focused view of these periapse constrained cases is provided in Figures A.2 and A.4. For the constrained cases, the optimal impulse is driven back toward the periapse impulse magnitude as  $V_\infty^+$  approaches the value defined for the  $\Phi_p$  condition. During this phase, the impulse angle  $\Lambda_{\Delta V}$  also goes back

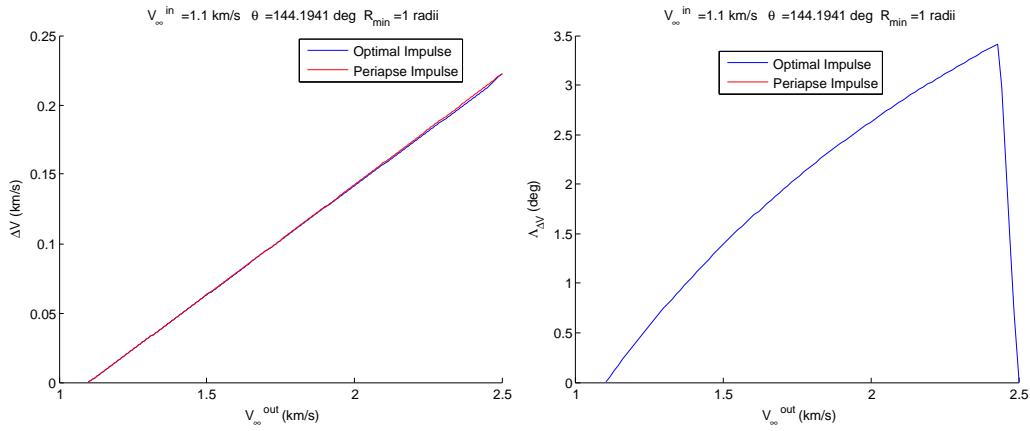


Figure A.1: Optimal feasible flyby solution  $\Delta V$  is shown to increase and slowly diverge away from periapse maneuver solutions as  $V_{\infty}^+$  increases away from  $V_{\infty}^-$  magnitude for unconstrained  $r_p$  flyby cases, then the optimal feasible maneuver  $\Delta V$  converges back to periapse maneuver solution line as flyby  $r_p$  is restricted by  $r_{min}$  (left). The associated effect on impulse angle  $\Lambda_{\Delta V}$  is shown where unconstrained flyby maneuvers were growing away from velocity alignment (collinear) as  $V_{\infty}^+$  increases away from  $V_{\infty}^-$ , then the impulse angle  $\Lambda_{\Delta V}$  quickly drops back to collinear aligned maneuvers for  $r_p$  constrained flybys approaching the maximum turning angle  $\Phi_p$  conditions (right).

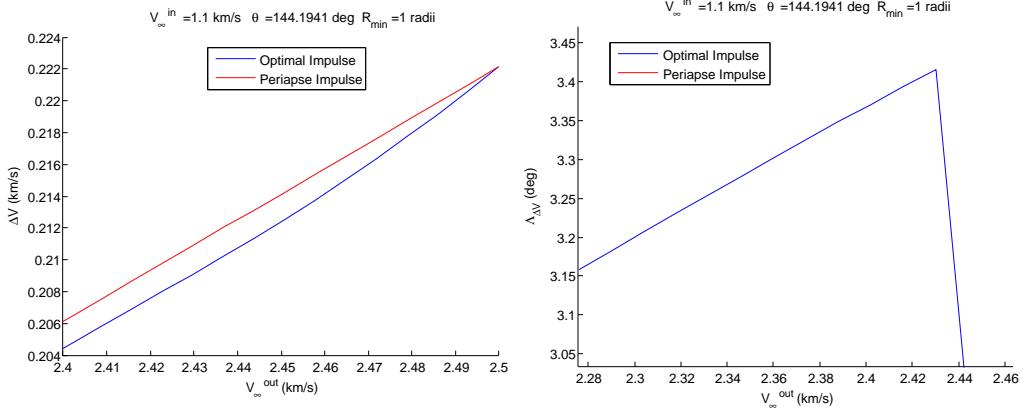


Figure A.2: Zoomed in view of optimal feasible flyby solution  $\Delta V$  converges to periapse maneuver solution line as flyby  $r_p$  is restricted by  $r_{min}$  (left). The associated effect on impulse angle  $\Lambda_{\Delta V}$  is shown to drastically change its trend of increasing away from collinear aligned maneuvers to quickly dropping back towards collinear alignment for  $r_p$  constrained flyby cases as the flyby configuration approaches the maximum turning angle  $\Phi_p$  conditions (right).

to zero as the optimum powered flyby condition approaches. Though the periapse radius  $r_p$  is restricted to  $r_{min}$  for all the constrained flybys shown in Figure A.4, the maneuver location  $r_{\Delta V}$  continues a monotonic trend with a change in slope/shape (due to the switch to constrained solution dynamics). These findings agree with Edelbaum's conclusions that a nonzero  $\Delta V$  impulse at a finite distance from the flyby body will only be tangent to the direction of motion (collinear  $\Delta V$ ) if it occurs at the minimum allowable periapse radius,  $r_{min}$ . Essentially, the necessary and sufficient condition that an optimal finite  $\Delta V$  maneuver will be collinear is the case of a periapse powered gravity assist with the common periapse at  $r_{min}$ . Therefore, the only case where the minimum  $\Delta V$  maneuver occurs at  $r_{min}$  is when the impulse is collinear with the motion and it is at periapse. These findings reinforce the concept that generally an optimal powered gravity assist maneuver is non-tangential,

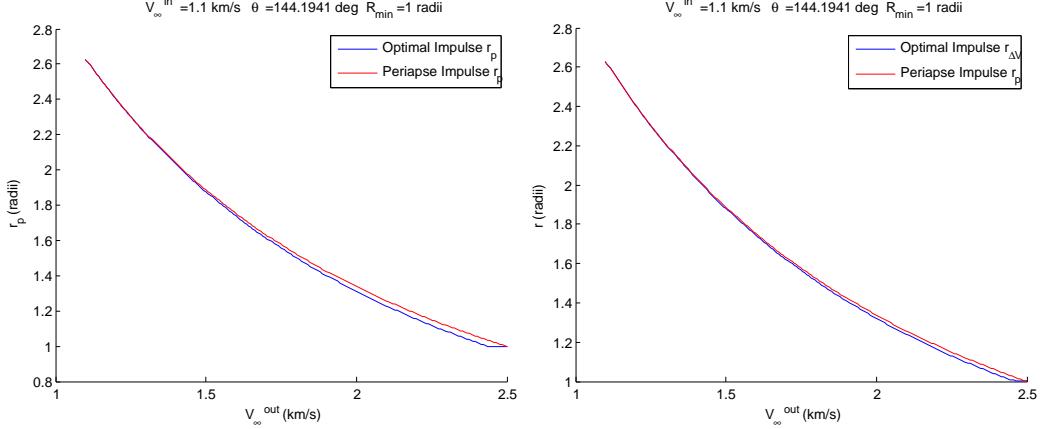


Figure A.3: Optimal flyby maneuver periapse radius is shown transitioning from an unconstrained peripase flyby to a peripase constrained flyby (left) and the associated optimal  $\Delta V$  maneuver radius transitioning to coincide with the peripase maneuver location for constrained flyby cases (right) for  $\Phi_p$  turning angle case

utilizing a portion of the  $\Delta V$  impulse to aid in torquing the trajectory for additional change in the overall flyby turning angle.

The  $r_{min}$  constrained case solutions determined in Figures A.3 and A.4 utilize the methodology and algorithms developed in the dissertation to efficiently and reliably solve the optimal maneuver location for a minimum  $\Delta V$ , while maintaining the  $r_{min}$  restriction. The  $r_{min}$  constrained optimal gravity algorithm is initialized using the infeasible  $v_{\Delta V}$  result from the unconstrained peripase analytic algorithm. The analysis of the solution space near the optimal peripase constrained flyby confirms that this initialization point is always close to the desired feasible solution, providing fast convergence.

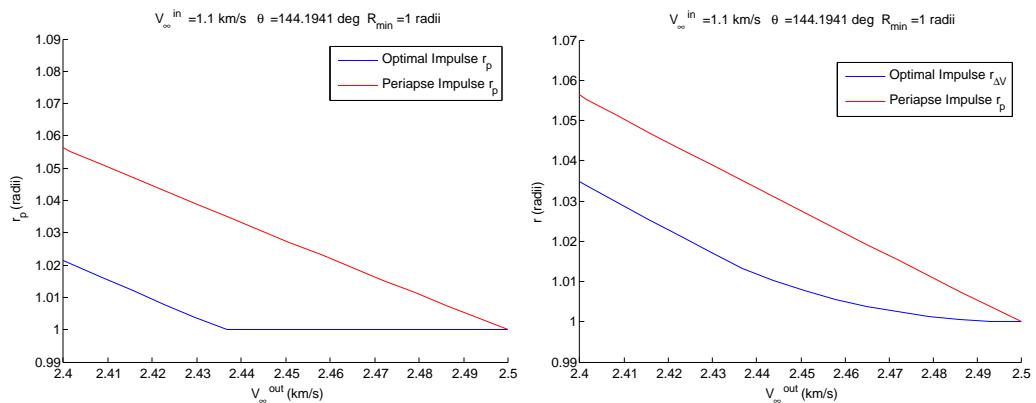


Figure A.4: Optimal maneuver periapse radius meeting  $r_{min}$  constraint (left) and the associated optimal  $\Delta V$  maneuver radius transitioning to periapse for constrained flyby cases (right): zoomed-in view

## Appendix B

### Strict $r_{SOI}$ constraint handling analysis for Type F flybys

There exists a second set of special case solutions to the single-impulse gravity assist problem where the required turning angle is less than the maximum available turning angle ( $\Phi_r < \Phi_p$ ). One of the major foundational assumptions of linked conics is that the encountered gravitational body must have an effect the trajectory. This is achieved by limiting the flyby periapse distance to a maximum at the sphere of influence  $r_{SOI}$ . Without such a restriction, extreme flyby configurations would have a periapse that diverge to unrealistic distances approaching infinity. When the unconstrained periapse Type F flyby periapse exceeds the maximum radius  $r_{SOI}$ , it is reevaluated as a constrained flyby with  $r_p = r_{SOI}$  and the optimal  $\Delta V$  maneuver location is determined. Next, a constraint is set for the maximum transfer maneuver location distance at an arbitrary radius above  $r_{SOI}$ . In the following analysis, the maximum transfer maneuver radius is set to  $r_{max} = 2r_{SOI}$ . The group of  $r_{SOI}$  constrained solutions occur at very low flyby turning angles with respect to the asymptote excess velocities.

These are extreme cases where the flyby is to be minimally affected by the presence of the encounter planet. These flyby configuration conditions could be judged by a mission planner to remove the gravity assist encounter from the tra-

jectory, although the intent of the mission planner could be to use this flyby configuration as a guide to finding a more useful flyby configuration. The later intent is assumed in the optimal single-impulse powered gravity assist algorithm developed here, so that the overall spacecraft mission trajectory optimization routine can find the best possible trajectory with the selected gravity assist encounters. It is left to the mission planner to manually remove the gravity assist encounter from the spacecraft mission trajectory if judged necessary.

An example case is analyzed to illustrate the discussion above. In the following analysis, the outgoing  $V_\infty^+$  magnitude is varied while holding the incoming  $V_\infty^- = 1$  km/s and low turning angle of  $\Phi_r = 20^\circ$  for an Earth encounter, shown in Figures B.1 - B.3. With such a low turning angle of only  $20^\circ$  for the selected  $V_\infty^-$  and  $V_\infty^+$  combinations, the optimal flyby periapse radius  $r_p$  increases until restricted at  $r_{SOI}$  as the outgoing  $V_\infty^+$  decreases towards  $V_\infty^-$ . The Earth  $r_{SOI}$  is at a distance of about 145 Earth radii, as shown by the plateau in the periapse  $r_p$  plot. The decreasing  $V_\infty^+$  magnitude results in an increasing maximum available turning angle  $\Phi_p$ , therefore requiring larger and larger associated  $r_p$  to achieve such a low  $\Phi_r$ . The optimal flyby radius plot in Figure B.2 also reflects the transition from unconstrained periapse solutions to  $r_p = r_{SOI}$  constrained solutions as the slope changes shape near  $V_\infty^+ \approx 1.55$  km/s. Then the flyby maneuver radius continues to increase as the outgoing  $V_\infty^+$  decreases towards  $V_\infty^-$ , until it is restricted by the chosen maximum flyby radius  $r_{max}$ , where  $r_{max}$  for this analysis is about 290 Earth radii, as seen in Figure B.2. The maximum transfer maneuver radius is chosen by the user to ensure that the  $\Delta V$  maneuver remains within the vicinity of the planet's gravitational influence. The cases with both periapse and transfer maneuver distance restrictions

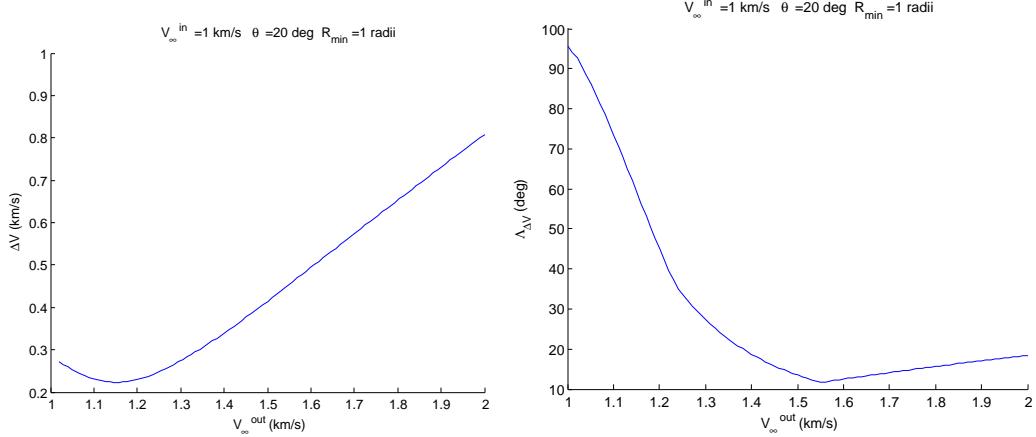


Figure B.1: Optimal feasible solution  $\Delta V$  is shown to be restricted from reducing to zero as  $V_{\infty}^+$  approaches  $V_{\infty}^-$  for  $r_{SOI}$  and  $r_{max}$  upper bound constrained cases (left). The associated effects on the impulse angle  $\Lambda_{\Delta V}$  is shown, as  $V_{\infty}^+$  approaches  $V_{\infty}^-$  the unconstrained case impulse was becoming more aligned with the velocity, but abruptly begins increasing away from collinear conditions for constrained cases (right)

continue to relegate the impulse maneuver at  $r_{max}$  as the outgoing  $V_{\infty}^+$  decreases towards  $V_{\infty}^-$ . It is worth noting that the trough seen in the  $\Delta V$  solutions example in Figure B.1 occurs due to these restrictions. If the restrictions were lifted, such that the flyby periapse and maneuver location were free to grow, the  $\Delta V$  solutions plot for the same low turning angle flyby would always be driven to zero with  $V_{\infty}^+$  decreasing to  $V_{\infty}^-$ , as described in prior unconstrained periapse analysis.

Due to the concave nature of the  $r_{SOI}$  and  $r_{max}$  constrained flyby solution space, a fully restricted flyby will have the minimum  $\Delta V$  solution at either boundary, periapse or  $r_{max}$ , displayed in Figure B.3. All low  $\Phi_r$  optimal flyby maneuver locations are first constrained at  $r_{max}$  as  $V_{\infty}^+$  decreases toward  $V_{\infty}^-$ , but often as  $V_{\infty}^+$  approaches even closer to  $V_{\infty}^-$  the optimal maneuver location abruptly switches to

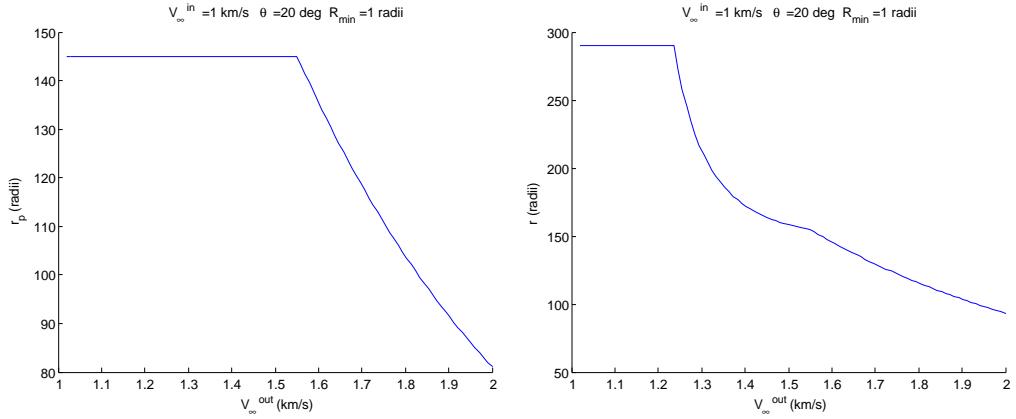


Figure B.2: Optimal feasible solution periapse is shown to increase as  $V_{\infty}^+$  approaches  $V_{\infty}^-$ , until restricted at  $r_{SOI}$  (left). The  $\Delta V$  maneuver radius has slow near-linear increase as  $V_{\infty}^+$  approaches  $V_{\infty}^-$  for unconstrained  $r_p$  flyby cases, then it shows a nonlinear increase for constrained  $r_p$  flybys until the maneuver radius is restricted at  $r_{max}$  (right)

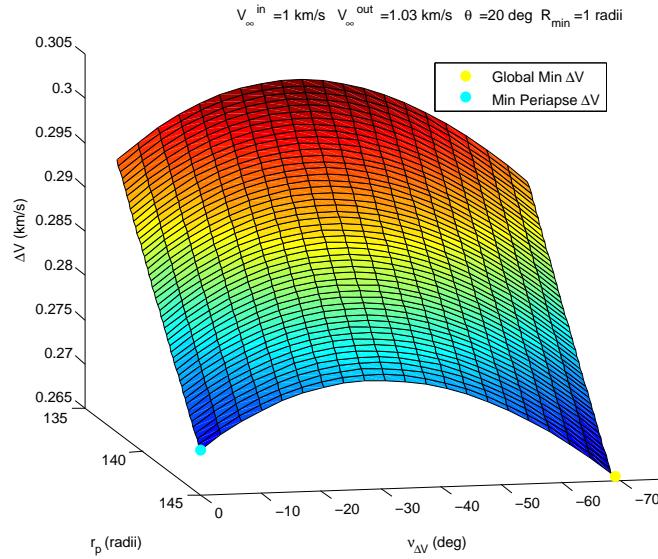


Figure B.3: Optimal feasible solution space example for  $r_{SOI}$  and  $r_{max}$  upper bound constrained cases, displaying the concave solution space characteristic of these special types of flyby configurations. The optimal maneuver location is always found at  $r_{max}$  boundary or at periapse, where  $r_p = r_{SOI}$

a periapse maneuver at  $r_{SOI}$ . With  $V_\infty^-$  and  $V_\infty^+$  closing in, the  $\Delta V$  magnitude at the periapse maneuver boundary decreases to and below the  $\Delta V$  value at  $r_{max}$  boundary, switching the optimal location from the  $r_{max}$  boundary to the  $r_{SOI}$  periapse boundary, shown in Figures B.4, B.5, and B.6. The optimal  $\Delta V$  plot is continuous through the switching of maneuver locations, though there may be a change in slope due to the difference in constrained dynamics, Figure B.4. With any further decrease of  $V_\infty^+$  approaching  $V_\infty^-$ , the optimal maneuver location remains at periapse. Other fully restricted flyby solution cases have the maneuver remain at  $r_{max}$  and never switch to the periapse boundary. The example solution space shown in Figure B.6 also demonstrates multiple local minima at the boundary cases, periapse and  $r_{max}$ , even though the particular periapse  $\Delta V$  solution was the overall minimum. When implementing an algorithm for this particular special case solution, it is recommended to check the boundary solutions in addition to the result of any chosen gradient or non-gradient based interior solution method to ensure that the global minimum solution is found for the boundary value problem.

The set of candidate single-impulse gravity assist solutions for the  $\Phi_r < \Phi_p$  flyby types are completely determined with the unconstrained periapse Type F powered flyby and the two special cases using constrained periapse powered flyby solution methods. It can easily be seen that the neighborhood around the constrained cases would benefit by the multiple impulse transfer solutions: Type  $F_\infty$ , Type  $RF_\infty$ , and Type PNP. For the two-impulse transfers, the total  $\Delta V$  is reduced by using the impulse at infinity (or at least  $r_{SOI}$ ) to adjust the periapse location for a more efficient impulse in the vicinity of the planet. The same concept is true for the Type PNP transfers, but usually for more extreme circumstances. Because these multiple-

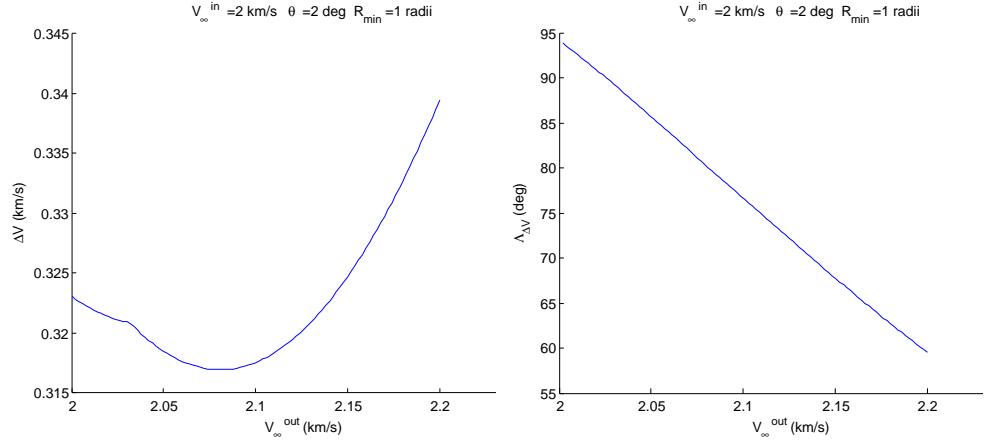


Figure B.4: Optimal feasible solution  $\Delta V$  for  $r_{SOI}$  and  $r_{max}$  upper bound constrained cases, showing the change in the  $\Delta V$  solution line dynamics as the maneuver location switches from the  $r_{max}$  boundary to periapse at the lower  $V_{\infty}^+$  region presented (left). The associated effects on the impulse angle  $\Lambda_{\Delta V}$  is displayed with imperceptible changes during the maneuver location switch (right)

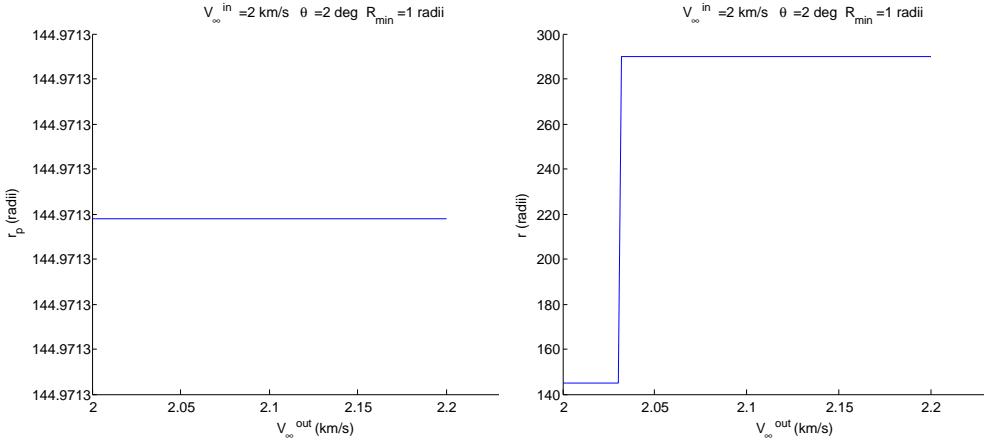


Figure B.5: Optimal solution perihelion radius is shown to be maintained at  $r_{SOI}$  for all flyby configurations presented (left). The optimal  $\Delta V$  maneuver location radius is clearly shown to transition from the  $r_{max}$  boundary to periapse at  $r_{SOI}$  as  $V_{\infty}^+$  approaches  $V_{\infty}^-$  for these extreme flyby conditions (right).

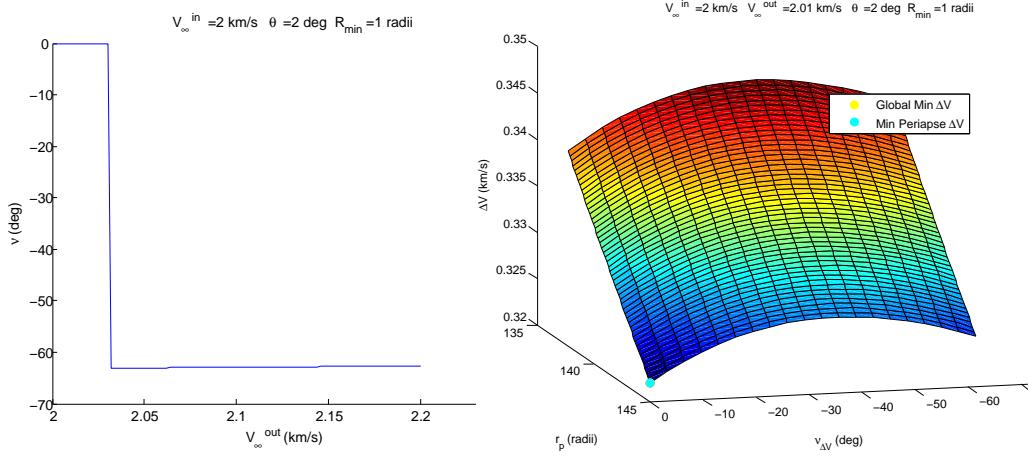


Figure B.6: Optimal maneuver location true anomaly  $v$  is clearly shown to transition from the  $r_{max}$  boundary to periapse as  $V_\infty^+$  approaches  $V_\infty^-$  for these extreme flyby conditions (left). The associated concave solution space example shows the periapse boundary with lower  $\Delta V$  magnitudes and therefore the optimal maneuver location (right).

impulse solution case have lower total  $\Delta V$  magnitudes compared to the constrained single-impulse powered flyby cases, the single-impulse powered flyby problem was never characterized for these particular conditions in the literature reviewed by the author. Therefore, the characterization of these constrained periapse powered gravity assists of the type  $\Phi_r < \Phi_p$  and the developed solution methods are considered new and part of the major contributions of the dissertation.

## Appendix C

### Hyperbolic terminal velocity BVP hodograph analysis

The general hyperbolic terminal velocity problem as presented by Battin determines the two endpoint velocities  $\vec{V}_1$  and  $\vec{V}_2$  required for a hyperbolic conic transfer between known fixed locations  $\vec{r}_1$  and  $\vec{r}_2$ , similar to the Lambert problem. The solution is achieved with orbital relations developed through hodograph analysis using a velocity vector diagram. The diagram excerpt from Battin [10] is provided in Figure C.1. The relative velocity magnitude and angle relationships are referenced by the chord axis defined along the horizontal. Some of the key relationships involved in building the hodograph diagram are presented in Equation C.1. Although not identified in Battin's problem description, a key feature found of the hodograph velocity diagram is that the common chordal velocity magnitude and direction at the endpoints is depicted by segment BA along the chord axis. In the diagram Further detail of this particular problem's derivation can be found in Battin [10].

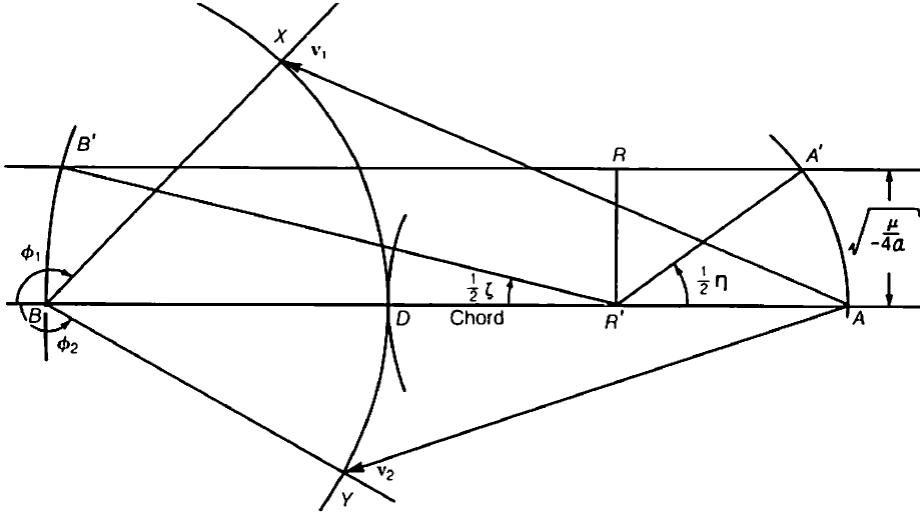


Figure C.1: General hyperbolic terminal velocity vector hodograph diagram  
[Source: Battin [10]]

$$R'R = V_\infty/2$$

$$RA' = \sqrt{\mu / (r_1 + r_2 + C)}$$

$$RB' = \sqrt{\mu / (r_1 + r_2 - C)} \quad (\text{C.1})$$

$$B'R' = BR'$$

$$BA = V_C$$

When the endpoint  $\vec{r}_2$  is pushed to infinity  $r_2 \rightarrow \infty$ , the result is the boundary case of the hyperbolic terminal velocity problem described in Section 2.3.2 and utilized in the constrained periapse powered gravity assist BVP. In addition, the chord direction  $\vec{C}$  and endpoint radial direction  $\vec{r}_2$  are driven to be along the  $\vec{V}_\infty$  asymptote direction. The hodograph velocity vector diagram was constructed to this  $V_\infty$  asymptote condition for further analysis, Figure C.2. The goal of the analysis is to derive the chordal component  $V_C$  of the transfer velocity in terms of the maneuver

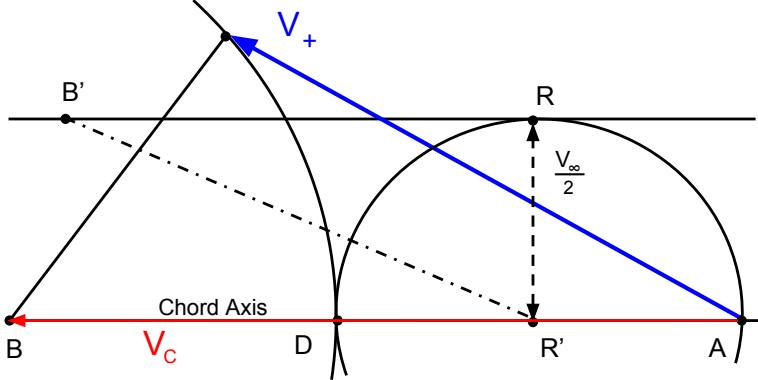


Figure C.2: Hyperbolic terminal velocity vector hodograph diagram for transfer to  $V_\infty$  asymptote

location,  $r$  or  $v$ . In the limit  $r_2 \rightarrow \infty$ , the chord length is also driven to infinity  $C \rightarrow \infty$ , and the relative velocity relationships were adjusted to support the  $V_\infty$  endpoint conditions. To aid in the derivation, the semiperimeter  $S$  of the transfer geometry is defined as  $S = (r_1 + r_2 + C)/2$ . The semiperimeter has a useful relationship with  $r_2$  in the limit, where  $\lim S/r_2 = 1$ . The equation for line segment  $RB'$  above must be adjusted to accommodate the diverging  $r_2$  and  $C$ , Equation C.2.

$$\begin{aligned}
 RB' &= \sqrt{\mu / (r_1 + r_2 - C)} \\
 &= \sqrt{\frac{2\mu S}{(r_1 + r_2 - C)(r_1 + r_2 + C)}} \\
 &= \sqrt{\frac{\mu S}{r_1 r_2 (1 + \cos \theta_\infty)}}
 \end{aligned} \tag{C.2}$$

where the chord length can be defined as

$$C^2 = r_1^2 + r_2^2 - 2r_1 r_2 \cos \theta_\infty.$$

Then substituting the limit  $S/r_2 = 1$ , the equation for  $RB'$  in terms of the  $V_\infty$

asymptote is found in Equation C.3. Also, line segments  $R'D$ ,  $R'A$ , and  $R'R$  are all driven equal to  $V_\infty/2$ . Knowing line segments  $RB'$  and  $R'R$ , the triangular relation yields  $B'R'$ , which then determines the chordal velocity component Equation C.4.

$$R'R = V_\infty/2$$

$$R'D = R'A = V_\infty/2$$

$$RB' = \sqrt{\frac{\mu}{r(1 + \cos \theta_\infty)}} \quad (\text{C.3})$$

$$B'R' = BR'$$

$$BA = V_C$$

$$B'R' = \sqrt{\frac{\mu}{r(1 + \cos \theta_\infty)} + \frac{V_\infty^2}{4}} \quad (\text{C.4})$$

$$V_C = \sqrt{\frac{\mu}{r(1 + \cos \theta_\infty)} + \frac{V_\infty^2}{4}} + \frac{V_\infty}{2}$$

## Appendix D

### GTOC 7 Contribution

The international Global Trajectory Optimization Competition (GTOC) has recently been held on a roughly annual basis, where a spacecraft mission problem is presented and students, academics, professionals, and other optimization enthusiasts engage the optimization challenge to find the best solution. The GTOC 7 competition comprised of a mothership and three probe spacecraft launching from Earth towards the asteroid belt. The mothership is allotted a limited supply of impulsive propulsion, while each of the probes have a limited supply of low thrust propulsion. The optimization challenge is to rendezvous with the most asteroids while meeting the required minimum stay time of 30 days at each asteroid. The mothership's initial mass is 18000 kg of which 12000 kg is propellant with an efficient chemical engine  $I_{sp} = 900$ s. Each probe's initial mass is 2000 kg of which 1200 kg is propellant with an efficient low thrust electric propulsion engine of  $I_{sp} = 3000$ s. Then each probe must rendezvous with the mothership. As a member of the University of Texas at Austin team for GTOC 7, the author utilized TRACT to aid in achieving a final optimal trajectory solution. The capabilities of TRACT were perfectly suited to contribute to this problem for optimizing and constraining the mothership trajectory to link up with the probe trajectories [44]. The final mothership trajectory is shown in Figure D.1.

The trajectory is a launch from Earth, followed by a DSM before reaching the first asteroid to drop off Probe 1. Then probe 2 and probe 3 are released in succession. After all the probes are released the second DSM is performed to reach the first probe rendezvous and pickup. The second probe is returns and rendezvous and finally the third probe is picked up by the mothership. There were many timing constraints applied in order to exactly match up the low thrust probe trajectories with the impulsive mothership trajectory. Also, there was a great need in reducing the required  $\Delta V$  as much as possible, hence the necessary inclusion of the DSMs. The trajectory required 3 DSMs primarily to accommodate plane changes between the probe rendezvous. With this mothership trajectory, the probes were able to reach 35 asteroids with a final combined probe mass of 2493 kg (93kg of remaining propellant), earning the University of Texas at Austin team a 3rd place ranking for GTOC 7. The 1st place trajectory solution only achieved one more asteroid rendezvous at 36 asteroids and a final combined probe mass of 2450 kg [44].

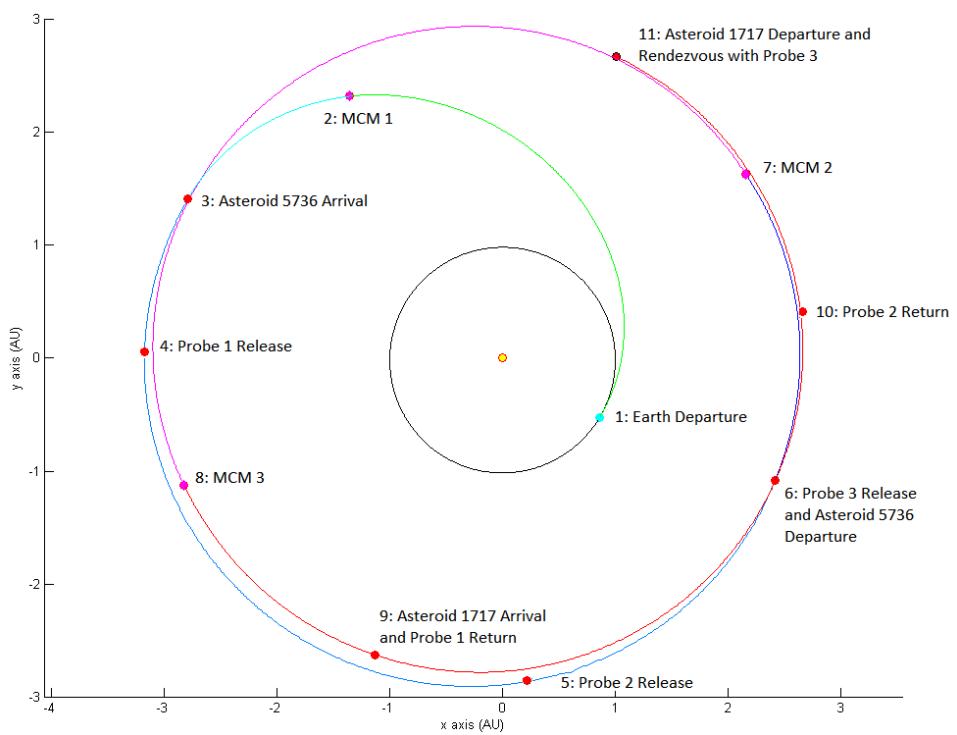


Figure D.1: GTOC 7 mothership trajectory formulated and optimized with TRACT

## Appendix E

### TRACT optimization schemes

The current MATLAB optimization functions easily integrated with TRACT are available in two packages: the Optimization Toolbox and the Global Optimization Toolbox. The Optimization Toolbox is typically included in most MATLAB offerings, while the Global Optimization Toolbox is usually not included, but it can be purchased separately for less than \$30. These optimization functions and their capabilities are outlined below according to their toolbox.

Optimization Toolbox:

**fminunc:** Unconstrained optimization routine using gradient based quasi-Newton or trust-region algorithms. The routine requires an initial guess for the optimization parameter values.

**fsolve:** Unconstrained nonlinear systems of equations solver employed to reduce each node's  $\Delta V$  to zero with gradient based trust-region or levenberg- marquardt algorithms. The routine requires an initial guess for the optimization parameter values.

**fminsearch:** Unconstrained optimization routine using the derivative-free Nelder-Mead simplex algorithm. The routine requires an initial guess for the optimization parameter values.

**fmincon:** Constrained optimization routine using interior-point, trust-region, sequential quadratic programming (SQP), or active-set, algorithms. The routine handles linear inequality and equality constraints directly on the optimization variables, as well as handling nonlinear inequality and equality constraints in a separate function. The routine requires an initial guess for the optimization parameter values.

**lsqnonlin:** Constrained nonlinear least-squares optimization routine using trust-region and levenberg-marquardt algorithms. The routine handles lower and upper bound constraints on the optimization variables. The routine requires an initial guess for the optimization parameter values.

Global optimization Toolbox:

**patternsearch:** Constrained global optimization routine using derivative-free automated pattern search algorithms. The routine handles linear inequality and equality constraints directly on the optimization variables, as well as handling nonlinear inequality and equality constraints in a separate function. The routine requires an initial guess for the optimization parameter values.

**simannealbnd:** Constrained global optimization routine using a stochastic annealing algorithm. The routine handles lower and upper bound constraints on the optimization variables. The routine requires an initial guess for the optimization parameter values.

**ga:** Constrained global optimization routine using a stochastic genetic algorithm. The routine handles linear inequality and equality constraints directly on the

optimization variables, as well as handling nonlinear inequality and equality constraints in a separate function. Integer value constraints can also be employed to require any specified optimization variables to take integer values. The routine does not require an initial guess values for the optimization parameters.

**particleswarm:** Constrained global optimization routine using a stochastic particle swarm algorithm. The routine handles lower and upper bound constraints on the optimization variables. The routine does not require an initial guess values for the optimization parameters.

The traditional optimization function choices are found in the Optimization Toolbox and are easily executed by the user. The user can also adjust a variety common optimization settings for these algorithms without difficulty understanding them and their impact. While the optimization functions found in the Global Optimization Toolbox are quite powerful and can easily be run by the user, the many optimization options for adjusting and tuning the algorithms require a slight learning curve. With some time spent accessing MATLAB's detailed online documentation, even these more complex global optimization functions can be executed with confidence. Many of these MATLAB optimization functions offer automated parallel computing options to aid the computation speed.

The gradient based functions offer strong convergence characteristics and constraint handling (when available). The constraints can be set to strict or loose management during the optimization process, either strictly avoiding infeasible solutions or allowing access to better feasible regions by passing through infeasible

ble regions. The Nelder-Mead simplex method of fminsearch provides a directed derivative-free search, that offers more explorative search characteristics while driving toward a minimum in general region about the initialized guess point.

The global optimization functions are by default derivative-free, where some have automated search algorithms similar to the simplex method and others use stochastic techniques. The strength of these global optimization functions are how they can explore many more regions of the solution space and potentially find the global minimum solution, while also satisfying the applied constraints. Many of the global optimization functions can be coupled with more directed algorithms (including fminunc, fmincon, fminsearch) in order to improve the final solution's convergence and accuracy.

## Appendix F

### TRACT constraints

#### F.1 Linear constraints:

The commonly used heliocentric trajectory constraints that apply to the optimization variables are outlined below, where each can have upper bound, lower bound, and equality constraints applied.

**TOF<sub>total</sub>:** The summed total time of flight

**TOF:** Each individual heliocentric trajectory leg time of flight

**JD:** Each individual trajectory node Julian date

**DSM[x,y,z]:** Each individual DSM location [x,y,z] (mean ecliptic and mean equinox of J2000)

#### F.2 Nonlinear constraints

The commonly used nonlinear trajectory constraints are outlined below, where each can have upper bound, lower bound, and equality constraints applied, unless otherwise noted.

**DSM[r]:** Each individual DSM location's radius magnitude:

$$r_{DSM} = \sqrt{x^2 + y^2 + z^2}$$

**$\Delta V$ :** Each individual trajectory node  $\Delta V$  magnitude

**$V_\infty^-$ :** Each individual trajectory node incoming excess velocity  $V_\infty^-$  magnitude

**$V_\infty^+$ :** Each individual trajectory node exiting excess velocity  $V_\infty^+$  magnitude

**Match- $\tilde{V}_\infty^\pm$ :** Each individual trajectory node excess velocity vectors are aligned

using equality constraint ( $V_\infty^- = V_\infty^+$ ) and inequality constraint ( $\Phi_r < \Phi_p$ )

## Appendix G

### TRACT Mission Driver options

#### Optimization parameters

**optimization** (scalar)

- 0:** No optimization performed (Fixed parameters)
- 1:** Optimize parameters using MATLAB fminunc
- 2:** Optimize parameters using MATLAB fminsearch (simplex)
- 3:** Optimize parameters using MATLAB fmincon
- 5:** Optimize parameters using MATLAB fsolve
- 6:** Optimize parameters using MATLAB lsqnonlin
- 7:** Optimize parameters using MATLAB fminimax
- 8-9:** –Reserved for other non MATLAB deterministic optimization routines
- 10:** Global optimize given parameters using MATLAB patternsearch
- 11:** Global optimize given parameters using MATLAB simannealbnd
- 12:** Global optimize without initial parameters using MATLAB ga (genetic algorithm)
- 13:** Global optimize without initial parameters using MATLAB particleswarm

**optnode** (vector): Weight factor for each node's  $\Delta V$  maneuver

**optscale** (scalar): Specifies how the optimization parameters are to be normalized

**0:** No scaling, variables are iterated in natural units

**1:** Scale variables by initial guess values

**2:** Scale times by initial total TOF, DSMs scaled as in (1)

**3:** Scale times by average TOF, DSMs scaled as in (1)

**optr** (scalar): Specifies if the periapsis radius of all gravity assists are varied or fixed in optimization procedure (either all are varied or all are fixed)

**0:** Periapsis radius is fixed at user-defined value

**1:** Periapsis radius is varied by optimization process

## Plotting parameters

**plotting** (vector): Specifies the preconfigured plots to be generated (whether all or just some are plotted)

**0:** Heliocentric trajectory with planets' orbits

**1:** Heliocentric trajectory with planets' orbits and positions at date of first node

**2:** Heliocentric trajectory with planets' orbits and each planet's SOI, North pole, and angular momentum vector direction of planetary orbits/encounters

**9:** Heliocentric trajectory movie with planets

**10:** Departure trajectories at nodes

**20:** Arrival trajectories at nodes

**30:** Flyby trajectories at nodes

### General trajectory configuration parameters

**npl** (vector): Planets encountered at each node

**0:** Deep Space Maneuver/ User-Defined Orbit

**1:** Mercury

**2:** Venus

**3:** Earth

**4:** Mars

**5:** Jupiter

**6:** Saturn

**7:** Uranus

**8:** Neptune

**9:** Pluto (dwarf planet)

**event** (vector): Mission event at each node

**10:** Departure from a parking orbit about planet with  $i$ ,  $\Omega$ , and  $\omega$  as optimization variables

**11:** Departure from a parking orbit about planet with  $\Omega$  and  $\omega$  as optimization variables

**12:** Departure from a parking orbit about planet with fixed orbit and optimized maneuver true anomaly  $v_{\Delta V}$

- 13:** Departure from a parking orbit about planet with completely fixed orbit position (no optimization)
- 14:** Departure from a launch vehicle trajectory with  $\Omega$  and  $\omega$  as optimization variables
- 17:** Depart user defined body (with negligible gravity) or heliocentric orbit
- 20:** Arrival to a parking orbit about planet with  $i$ ,  $\Omega$ , and  $\omega$  as optimization variables
- 21:** Arrival to a parking orbit about planet with  $\Omega$  and  $\omega$  as optimization variables
- 22:** Arrival to a parking orbit about planet with fixed orbit and optimized maneuver true anomaly  $v_{\Delta V}$
- 23:** Arrival to a parking orbit about planet with completely fixed orbit position (no optimization)
- 24:** Arrival to entry trajectory with  $\Omega$  and  $\omega$  as optimization variables
- 27:** Rendezvous with user defined body (with negligible gravity) or heliocentric orbit
- 30:** Unpowered ballistic gravity assist (no  $\Delta V$  computed, must use non-linear flyby constraints)
- 31:** Asymptote corrected gravity assist ( $\Delta V$  location restricted to  $\vec{V}_\infty^-$  or  $\vec{V}_\infty^+$  asymptote)
- 33:** Optimal powered gravity assist (optimizes  $\Delta V$  location  $v_{\Delta V}$  during flyby)

**35:** Periapse powered gravity assist (optimizes periapse radius  $r_p$ ,  $\Delta V$  location restricted to periapse)

**40:** Midcourse maneuver (DSM) with initial location guess provided by program

**41:** Space burn (DSM) with initial location guess provided by user

**JD0** (scalar): Initial Julian date (can also provide yr, mon, d, h, min, s for conversion to Julian date)

**tof** (vector): Time of flight in days for each trajectory leg

**grade** (vector): Flight direction of trajectory leg

**0:** Prograde (similar direction as planets)

**1:** Retrograde

**rev** (vector): Number of orbital revolutions of trajectory leg

**0:** Less than a full orbital revolution

**±1,2,3...** (positive or negative integer): Multiples of full orbital revolutions. Sign  $\pm$  of integer identifies the left branch or right branch multi-rev Lambert trajectory solutions

**branch** (scalar): Identify if using particular multi-rev branch configuration or to automatically test all permutations

**0:** Use fixed multi-rev branch indicated by the **rev** parameter

**1:** Automatically determine and run all branch permutations

**branchopt** (scalar): Specify the number of iterations to execute for each permutation run

**0:** Fully optimize each permutation case

**1,12,101 ...** (integer) number of iterations to execute before stopping  
each permutation run

### Departure/arrival parameters for parking orbits

**coep** (array): Classical orbital elements of parking orbit at particular node

$[a \ e \ i \ \Omega \ \omega \ v]$

### Launch/entry parameters

**altp** (vector): Altitude of periapsis (km) for particular node

**ilv** (vector): Inclination of launch or entry trajectory for particular node

**vinfavmag** (vector): Hyperbolic velocity at infinity available from launch  
vehicle or entry

**dvopt** (vector): Defines if excess  $V_\infty$  is detrimental ( $\Delta V$ ) or acceptable (no  
 $\Delta V$ )

**0:** Required  $V_\infty =$ available  $V_\infty$

**1:** Required  $V_\infty \leq$ available  $V_\infty$

### Gravity assist parameters

**rfb** (vector): Minimum flyby radius magnitude (in radii of particular planet)  
at node

### DSM parameters

**revig** (vector): Number of orbital revolutions completed by the leg segments between closest preceding celestial body node and closest celestial body node following

**0:** Less than a full orbital revolution

**1,2,3...** (positive integer): Multiples of full orbital revolutions

**rnb** (array): Space burn heliocentric location in Cartesian coordinates (AU) at particular node

### User-defined body/heliocentric orbit parameters

**coenp** (array): Classical orbital elements of parking orbit and corresponding Julian date at particular node

[ $a \ e \ i \ \Omega \ \omega \ v \ JD$ ]

## Bibliography

- [1] D. A. Vallado, *Fundamentals of Astrodynamics and Applications, Second Edition.* El Segundo, CA: Microcosm Press, 2001.
- [2] J. M. Walton, C. Marchal, and R. D. Culp, “Optimum Transfer Between Hyperbolic Asymptotes With Turn Angle Less than the Maximum Natural Turn Angle,” *The Journal of the Astronautical Sciences*, vol. 19, no. 3, pp. 205–231, November 1971.
- [3] J. M. Walton, “Optimum Transfer between Hyperbolic Asymptotes about a Planet of Finite Size,” Ph.D. dissertation, The University of Colorado, May 1973.
- [4] J. M. Walton, C. Marchal, and R. D. Culp, “Synthesis of the Types of Optimal Transfers between Hyperbolic Asymptotes,” *AIAA Journal*, vol. 13, no. 8, pp. 980–988, August 1975.
- [5] H. D. Curtis, *Orbital Mechanics for Engineering Students, Second Edition.* New York, NY: Elsevier, 2010.
- [6] “The juno mission to jupiter - a pre-launch update,” NASA Jet Propulsion Laboratory, Tech. Rep., March 2011.
- [7] F. W. Gobetz, “Optimum Transfers between Hyperbolic Asymptotes,” *AIAA Journal*, vol. 1, no. 9, pp. 2034–2041, September 1963.

- [8] Y. Guo and R. W. Farquhar, “New horizons mission design,” *Space Science Reviews*, no. 140, pp. 49–74, 2008.
- [9] J. V. McAdams, R. W. Farquhar, A. H. Taylor, and B. G. Williams, “Messenger mission design and navigation,” *Space Science Reviews*, no. 131, pp. 219–246, 2007.
- [10] R. H. Battin, *An Introduction to the Mathematics and Methods of Astrodynamics, Revised Edition.* Reston, VA: AIAA Education Series, 1999.
- [11] J. R. Wertz, D. F. Everett, and J. J. Puchell, *Space Mission Engineering: The New SMAD.* Hawthorne, CA: Microcosm Press, 2011.
- [12] C. Ocampo and J. P. Munoz, “Variational Equations for a Generalized Spacecraft Trajectory Model,” *Journal of Guidance, Control, and Dynamics*, vol. 33, no. 5, pp. 1615–1622, 2010.
- [13] B. A. Conway, *Spacecraft Trajectory Optimization.* Cambridge, NY: Cambridge University Press, 2010.
- [14] R. R. Bate, D. D. Mueller, and J. E. White, *Fundamentals of Astrodynamics.* Mineola, NY: Dover Publications, 1971.
- [15] P. K. Seidelmann, Ed., *Explanatory Supplement to the Astronomical Almanac.* Mill Valley, CA: University Science Books, 1992.
- [16] N. Arora, R. P. Russell, N. Strange, and D. Ottesen, “Partials Derivatives of the Solution to the Lambert Boundary Value Problem,” *Journal of Guidance, Control, and Dynamics*, TBD 2015.

- [17] E. Barrabes, G. Gomez, and J. Rodriguez-Canabal, “Advanced topic in astrodynamics: Notes for the gravitational assisted trajectories lectures,” Barcelona, Spain, July 2004.
- [18] M. J. Brennan, “Patched Conic Interplanetary Trajectory Design Tool,” Master’s thesis, The University of Texas at Austin, December 2011.
- [19] W. Folkner, “Planetary and lunar ephemerides (de405),” Jet Propulsion Laboratory Solar System Dynamics Website,[http://ssd.jpl.nasa.gov/txt/aprx\\_pos\\_planets.pdf](http://ssd.jpl.nasa.gov/txt/aprx_pos_planets.pdf), [cited 3 March 2015].
- [20] C. F. Walker, “Matlab solar system ephemeris toolbox, version 1.0 (r12),” Open Channel Foundation,<http://www.openchannelfoundation.org/projects/>, Kennedy Space Center, FL, July 2006, [cited 3 March 2015].
- [21] E. M. Standish, “Keplerian elements for approximate positions of the major planets,” Jet Propulsion Laboratory Solar System Dynamics Website,<http://ssd.jpl.nasa.gov/?ephemerides>, [cited 3 March 2015].
- [22] R. P. S. Oldenhuis, “Robust solver for Lambert’s orbital-boundary value problem,” MATLAB File Exchange, <http://www.mathworks.com/help/matlab>, April 2011, [cited 3 March 2015].
- [23] D. Izzo, “Revisiting Lambert’s Problem,” *Celestial Mechanics and Dynamical Astronomy*, vol. 121, no. 1, pp. 1–15, 2015.
- [24] R. H. Gooding, “A procedure for the solution of Lambert’s orbital boundary-value problem,” *Celestial Mechanics and Dynamical Astronomy*, vol. 48, no. 2, pp. 145–165, June 1990.

- [25] MATLAB, “R2013b documentation,” MathWorks, <http://www.mathworks.com/help/matlab>, January 2015, [cited 3 March 2015].
- [26] J. A. Sims, J. M. Longuski, and A. Staugler, “V-infinity Leveraging for Interplanetary Missions: Multiple-revolution Orbit Techniques,” *Journal of Guidance, Control, and Dynamics*, vol. 20, no. 3, pp. 409–415, May 1997.
- [27] G. R. M. Vasile, M. Ceriotti, “Global trajectory optimisation: Can we prune the solution space when considering deep space manoeuvres?” Department of Aerospace Engineering, University of Glasgow, UK, Tech. Rep., January 2008.
- [28] J. T. Olympio and D. Izzo, “Designing Optimal Multi-gravity-assist Trajectories with Free Number of Impulses,” in *Proceedings of the International Symposium on Space Flights Dynamics*, Toulouse, France, September 2009.
- [29] N. P. Tedford and J. R. R. A. Martins, “On the Common Structure of MDO Problems: A Comparison of Architectures,” in *Proceedings of the AIAA Multi-disciplinary Analysis and Optimization Conference*, Portsmouth, VA, September 2006.
- [30] N. P. Tedford and J. Martins, “Benchmarking multidisciplinary design optimization algorithms,” *Optimization and Engineering*, vol. 11, no. 1, pp. 159–183, 2006.
- [31] J. Englander, “Automated Trajectory Planning for Multiple-Flyby Interplanetary Missions,” Ph.D. dissertation, The University of Illinois at Urbana-Champaign, 2013.

- [32] J. A. Englander, B. A. Conway, and T. Williams, “Automated Interplanetary Trajectory Planning ,” in *Proceedings of the AAS/AIAA Astrodynamics Specialist Conference*, Minneapolis, MN, August 2012, pp. 4517–4537.
- [33] J. A. Englander, B. Conway, and T. Williams, “Automated Mission Planning via Evolutionary Algorithms,” *Journal of Guidance, Control, and Dynamics*, vol. 35, no. 6, pp. 1878–1887, December 2012.
- [34] D. V. Lantukh, “Preliminary Design of Spacecraft Trajectories for Mission to Outer Planets and Small Bodies,” Ph.D. dissertation, The University of Texas at Austin, August 2015.
- [35] D. V. Lantukh, R. P. Russell, and S. Campagnola, “V-Infinity Leveraging Boundary-Value Problem and Application in Spacecraft Trajectory Design,” *Journal of Spacecraft and Rockets*, vol. 52, no. 3, pp. 697–699, May 2015.
- [36] D. V. Lantukh and R. P. Russell, “Automated Inclusion of n-pi Transfers in Gravity-Assist Flyby Tour Design,” in *Proceedings of the AAS/AIAA Astrodynamics Specialist Conference*, Minneapolis, MN, August 2012.
- [37] R. D. Sugar and A. R. Stubberud, “Computational Aspects of Multilevel Trajectory Optimization,” in *Proceedings of the IEEE Conference on Decision and Control*, New Orleans, LA, December 1972.
- [38] S. HongXin, Z. JianPing, P. QiBo, L. HaiYang, and L. JiuTian, “Multi-objective interplanetary trajectory optimization combining low-thrust propulsion and gravity-assist maneuvers,” *Science China Technological Sciences*, vol. 55, no. 3, pp. 841–847, 2012.

- [39] D. Izzo and C. Bombardelli, “Mulitple gravity assists,” ESA - Global Trajectory Optimization Problems Database, <http://www.esa.int/gsp/ACT/inf/projects/gtop/gtop.html>, April 2007, [cited 21 June 2015].
- [40] R. P. S. Oldenhuis, “Skipping stone - an interplanetary space mission design tool,” MATLAB File Exchange, <http://www.mathworks.com/help/matlab>, Nov. 2010, [cited 21 June 2015].
- [41] D. Eagle, “Gravity assist trajectory design and analysis,” MATLAB File Exchange, <http://www.mathworks.com/help/matlab>, December 2014, [cited 21 June 2015].
- [42] J. L. Horsewood, “Mission analysis environment for heliocentric high-thrust missions, mathematical formulation,” SpaceFlightSolutions:Software documentation of MAnE, Hendersonville, NC, July 2011.
- [43] F. Santilli, “Astrotik 1.0,” MATLAB File Exchange, <http://www.mathworks.com/help/matlab>, April 2010, [cited 21 June 2015].
- [44] L. Casalino and G. Colasurdo, “Problem description for the 7th global trajectory optimization competition,” GTOC Portal, [http://sophia.estec.esa.int/gtoc\\_portal/?page\\_id=515](http://sophia.estec.esa.int/gtoc_portal/?page_id=515), [cited 29 June 2015].
- [45] S. C. Chapra and R. P. Canale, *Numerical Methods for Engineers*, 7th ed. New York, NY: McGraw-Hill Education, 2015.
- [46] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed. New York, NY: Springer, 2006.

- [47] V. DeMiguel and W. Murray, “A local convergence analysis of bilevel decomposition algorithms,” *Optimization and Engineering*, vol. 7, no. 2, pp. 99–133, 2006.
- [48] V. DeMiguel and F. J. Nogales, “On the relationship between bilevel decomposition algorithms and direct interior-point methods,” Universidad Carlos III, Departamento de Estadística y Econometría, Tech. Rep., April 2004.
- [49] M. Ceriotti and M. Vasile, “Automated Multigravity Assist Trajectory Planning with a Modified Ant Colony Algorithm,” *Journal of Aerospace Computing, Information, and Communication*, vol. 7, no. 9, pp. 261–293, September 2010.
- [50] A. Gad and O. Abdelkhalik, “Hidden genes genetic algorithm for multi-gravity-assist trajectories optimization,” *Journal of Spacecraft and Rockets*, vol. 48, pp. 629–641, July 2011.
- [51] J. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright, “Convergence Properties of the Nelder-Mead Simplex Method in Low Dimensions,” *SIAM Journal of Optimization*, vol. 9, pp. 112–147, 1998.
- [52] A. F. B. A. Prado, “A study of powered swing-by,” Brazil, June 1995.
- [53] M. Ceriotti, “Global optimisation of multiple gravity assist trajectories,” Ph.D. dissertation, University of Glasgow, July 2010.
- [54] T. N. Edelbaum, “Optimal space trajectories,” Analytical Mechanics Associates, Inc., Tech. Rep., December 1969, [cited 3 March 2015].

- [55] F. W. Gobetz and J. R. Doll, “A Survey of Impulsive Trajectories,” *AIAA Journal*, vol. 7, no. 5, pp. 801–834, May 1969.
- [56] D. F. Lawden, *Optimal Trajectories for Space Navigation*. London: Butterworths, 1963.
- [57] L. Casalino, G. Colasurdo, and D. Pastrone, “Simple strategy for powered swingby,” *Journal of Guidance, Control, and Dynamics*, vol. 22, no. 1, pp. 156–159, 1999.
- [58] A. F. B. A. Prado and G. de Felipe, “An analytical study of the powered swingby to perform orbital maneuvers,” *Advances in Space Research*, vol. 40, no. 1, pp. 102–112, 2007.
- [59] A. F. Silva, A. F. B. A. Prado, and O. C. Winter, “Optimal impulsive control in a powered Swing-By,” in *Proceedings of the AIAA Guidance, Navigation, and Control (GNC) Conference*, Boston, MA, August 2013.
- [60] E. W. Weisstein, “Halley’s method,” MathWorld: Wolfram Web Resource, <http://mathworld.wolfram.com/HalleyMethod.html>, [cited 9 May 2015].
- [61] M. Ovchinnikov, S. Trofimov, and M. Shirobokov, “Method of virtual trajectories for the design of gravity assisted missions,” *Cosmic Research*, vol. 51, no. 6, pp. 439–451, 2013.
- [62] M. Vasile and P. D. Pascale, “Preliminary design of multiple gravity-assist trajectories,” *Journal of Spacecraft and Rockets*, vol. 43, no. 4, pp. 794–805, July 2006.

- [63] A. F. B. A. Prado, “Powered Swingby,” *Journal of Guidance, Control, and Dynamics*, vol. 19, no. 5, pp. 1142–1147, 1996.
- [64] J. Martins, P. Sturdza, and J. J. Alonso, “The Complex-Step Derivative Approximation,” *ACM Transactions on Mathematical Software*, vol. 29, no. 3, pp. 245–262, 2003.
- [65] L. A. D’Amario, L. E. Bright, and A. A. Wolf, “Galileo trajectory design,” *Space Science Reviews*, no. 60, pp. 23–78, 1992.
- [66] T. D. Kowalkowski, J. R. Johannesen, and T. Lam, “Launch Period Development for the Juno Mission to Jupiter,” in *Proceedings of the AIAA/AAS Specialist Conference*, Honolulu, HI, August 2008.
- [67] “Cassini prime mission plan,” NASA Jet Propulsion Laboratory, Tech. Rep., July 2008.
- [68] J. A. Dunne and E. Burgess, “The voyage of mariner 10,” Jet Propulsion Laboratory, Tech. Rep., 1978.
- [69] C. E. Kohlhase and P. A. Penzo, “Voyager mission description,” *Space Science Reviews*, no. 21, pp. 77–101, November 1977.
- [70] K. P. Wenzel, R. G. Marsden, D. E. Page, and E. J. Smith, “The ulysses mission,” *Astronomy and Astrophysics Supplement Series*, no. 92, pp. 207–219, January 1992.
- [71] E. Lancaster and R. Blanchard, “A unified form of lambert’s theorem,” NASA Technical Report TN D-5368, NASA, Tech. Rep., 1969, [cited 3 March 2015].

- [72] S. Zimmer and C. Ocampo, “Use of Analytical Gradients to Calculate Optimal Gravity-Assist Trajectories,” *Journal of Guidance, Control, and Dynamics*, vol. 28, no. 2, pp. 324–332, April 2005.
- [73] R. P. Russell, “Global Search and Optimization for Free-Return Earth-Mars Cyclers,” Ph.D. dissertation, The University of Texas at Austin, August 2004.
- [74] D. Izzo and C. Bombardelli, “Mulitple gravity assists with deep space maneuvers,” ESA - Global Trajectory Optimization Problems Database, <http://www.esa.int/gsp/ACT/inf/projects/gtop/gtop.html>, March 2007, [cited 21 June 2015].
- [75] P. Blau, “Juno mission and trajectory design,” Spaceflight 101, <http://www.spaceflight101.com/juno-mission-design-timeline-trajectories.html>.
- [76] S. Kayali, W. McAlpine, H. Becker, and L. Scheick, “Juno Radiation Design and Implementation,” in *Proceedings of the IEEE Aerospace Conference*, Big Sky, MT, March 2012.
- [77] G. H. Fountain, D. Y. Kusnierkiewicz, C. B. Hersman, T. S. Herder, T. B. Coughlin, W. C. Gibson, D. A. Clancy, C. C. DeBoy, T. A. Hill, J. D. Kinnison *et al.*, “The new horizons spacecraft,” Southwest Research Institute, Tech. Rep., April 2009.
- [78] T. D. Goodson, D. L. Gray, Y. Hahn, and F. Peralta, “Cassini maneuver experience: Finishing inner cruise,” Jet Propulsion Laboratory, Tech. Rep., 2000.

## Vita

Martin James Brennan developed a passion for Science and Mathematics throughout his upbringing in the Ocean Springs school system. He graduated from Mississippi State University (MSU) with a B.S. double major in Aerospace Engineering and Physics and a minor in Mathematics in December 2008. While at MSU he met the love of his life, Holly. He soon after began his graduate career in Aerospace Engineering with a focus in Orbital Mechanics at the University of Texas at Austin, earning an M.S. in 2011. Throughout his graduate career he was privileged to be an Assistant Instructor for the Spacecraft Mission Design course series with Dr. Fowler, supporting his research financially and motivationally. He gained priceless work experience Co-Oping with United Space Alliance at NASA's Marshall Space Flight Center and interning at NASA's Jet Propulsion Laboratory, where he will begin a full-time position Fall 2015. He enjoys exploring nature with his dogs Penny Lane and Luna Lovegood, tinkering with projects, and loving life with his wife Holly.

Permanent address: marty.brennan@utexas.edu

This dissertation was typeset with  $\text{\LaTeX}^{\dagger}$  by the author.

---

<sup>†</sup> $\text{\LaTeX}$  is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's  $\text{T}_{\text{E}}\text{X}$  Program.