The Dissertation Committee for Demyan Vasilyevich Lantukh
certifies that this is the approved version of the following dissertation:

Preliminary Design of Spacecraft Trajectories for Missions to Outer Planets and Small Bodies

Committee:

Ryan P. Russell, Supervisor

Wallace Fowler

Srinivas Bettadpur

Yanping Guo

Stephen Broschart

Preliminary Design of Spacecraft Trajectories for Missions to Outer Planets and Small Bodies

by

Demyan Vasilyevich Lantukh, B.S.A.E., M.S.A.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

Dedicated to Gbenga Yomi Afolabi.

With just seven months of friendship, his joy for living, his willingness to trust, and his passion for the transforming message of Jesus Christ have left an indelible mark on my life. His life – lived to the full as Jesus promised in John 10:10 – has left ripples that will meet G in eternity, where he went before us July 17, 2012.

Acknowledgments

My thanks first and foremost go to God: my Creator and my Savior Jesus Christ, who has given me a promise of unconditional love and a life of purpose – and provided everything I have needed not only for earning a PhD but also for truly living.

I also thank Ryan Russell, who saw potential in me and encouraged and inspired me from when I was an undergraduate through the end of my studies. He continues to provide a model for the pursuit of excellence as a professor and as a human being.

I am grateful to my various collaborators and coworkers: to Yanping Guo and Uday Shankar and others at the Applied Physics Lab for the *Explore* project; to Steve Broschart for teaching me asteroid proximity operations and hosting me at the Jet Propulsion Lab; to the rest of my dissertation committee; to my colleagues formerly and currently at the University of Texas – notably Vivek, Etienne, Sonia, Ricardo, and Nick – for sharing in the struggle of the PhD program and also in more than a few miscellaneous adventures.

Outside the academic sphere, I am thankful for the friendship and support offered by my spiritual family over the last few years: the UT Navigators provided help that I needed to keep me on track in the degree program and in life. To properly thank everybody there and in the GT Navigators and from OME'14 would require a whole appendix – though I cannot fail to mention "the elements" by name: Devin, Mickey, Matt, and Kevin. In addition to these, I am especially indebted to JT, David, Chris, Nick, Bethany, Abbi, Lauren, Rachel, Zach, Jozabad, and Jackson from the UT Navigators.

Additionally, the unwavering confidence of my family and their counsel and encouragement have been critical in this phase of my life as they have been in every phase. I am thankful to Dad & Mom, to Vasily & Katie, Vadim & Diana, Michael & Victoria, James & Liliya, and to Alex.

Though my thanks here focus on those with greatest prolonged impact over the last few years, I am humbled and grateful to remember that there are many more who have invested in me and contributed to my life from my time at Georgia Tech – most notably Sterling, Donnie, and Jason – and from even before that as well.

Finally, on top of the academic and personal support, financial support for the research and the PhD have been crucial to the completion of this dissertation. I thank the NASA Space Technology Research Fellowship and particularly Claudia Meyer there for continued interest and the bulk of the financial support. I am also grateful to APL for providing finances in conjunction with the main and follow-on *Explore* contracts. In conclusion, I thank the University of Texas Cockrell School of Engineering and the W.M. Keck Foundation for graciously providing me the W.M. Keck Foundation Endowed Graduate Fellowship in Engineering.

Preliminary Design of Spacecraft Trajectories for Missions to Outer Planets and Small Bodies

Demyan Vasilyevich Lantukh, Ph.D.
The University of Texas at Austin, 2015

Supervisor: Ryan P. Russell

Multiple gravity assist (MGA) spacecraft trajectories can be difficult to find, an intractable problem to solve completely. However, these trajectories have enormous benefits for missions to challenging destinations such as outer planets and primitive bodies. Techniques are presented to aid in solving this problem with a global search tool and additional investigation into one particular proximity operations option is discussed.

Explore is a global grid-search MGA trajectory pathsolving tool. An efficient sequential tree search eliminates v_∞ discontinuities and prunes trajectories. Performance indices may be applied to further prune the search, with multiple objectives handled by allowing these indices to change between trajectory segments and by pruning with a Pareto-optimality ranking. The MGA search is extended to include deep space maneuvers (DSM), v_∞ leveraging transfers (VILT) and low-thrust (LT) transfers. In addition, rendezvous or $n\pi$ sequences can patch the transfers together, enabling automatic augmentation of the MGA sequence.

Details of VILT segments and $n\pi$ sequences are presented: A boundary-value problem (BVP) VILT formulation using a one-dimensional root-solve enables inclusion of an efficient class of maneuvers with runtime comparable to solving ballistic transfers. Importantly, the BVP VILT also allows the calculation of velocity-aligned apsidal maneuvers (VAM), including inter-body transfers and orbit insertion maneuvers. A method for automated inclusion of $n\pi$ transfers such as resonant returns and back-flip trajectories is introduced: a BVP is posed on the v_∞ sphere and solved with one or more $n\pi$ transfers – which may additionally fulfill specified science objectives. The $n\pi$ sequence BVP is implemented within the broader search, combining $n\pi$ and other transfers in the same trajectory.

To aid proximity operations around small bodies, analytical methods are used to investigate stability regions in the presence of significant solar radiation pressure (SRP) and body oblateness perturbations. The interactions of these perturbations allow for heliotropic orbits, a stable family of low-altitude orbits investigated in detail. A novel constrained double-averaging technique analytically determines inclined heliotropic orbits. This type of knowledge is uniquely valuable for small body missions where SRP and irregular body shape are very important and where target selection is often a part of the mission design.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Constraints from the required orbital energy and from the necessary navigation tolerances make certain celestial bodies more challenging to visit. Advances in spacecraft propulsion, in launch vehicles, in trajectory design methods, and in navigation capabilities have led to an expansion of the types of missions proposed and flown and in the destinations which are achievable for an acceptable cost.

In this chapter, multiple gravity assist (MGA) trajectories are introduced, including a history of the use of gravity assists. An overview of the historical and state of the art methods for calculating such trajectories follows as context for the method which is described in Chapter 2. These MGA trajectories are particularly useful for missions to asteroids and to the outer planets and also for moon tours around the outer planets. Note that Appendix A provides a list of acronyms and notation used throughout.

Two particular trajectory types are also introduced in preparation for detailed discussion later in this dissertation. An efficient class of maneuvers is introduced via v_∞ leveraging transfers (VILT). Basic background is provided which is then expanded upon in Chapter 3 where a particular method for the inclusion of VILTs in a broad trajectory search is introduced. Resonant or $n\pi$ transfers are also introduced in this chapter then further detailed and applied in Chapter 4

A brief introduction to proximity operations around asteroids follows, focusing on the orbital perturbations experienced by a spacecraft operating in the vicinity of a small body. These challenges introduce the context and motivation for the analytic and numerical studies of a class of orbits around asteroids presented in Chapter 5.

1.1 Multiple gravity assist trajectories

Gravity assist flybys are essentially the intentional use of gravitational perturbations from minor bodies along a trajectory. Although perturbations to comets (and spacecraft) caused by planetary bodies had been known and observed, their systematic implementation as a means to benefit spacecraft missions was enabled by a mathematical framework which later developed into the patched conics model [1]. Further study has expanded the understanding and mathematical modeling of gravity assist flybys with the aim of easier inclusion in trajectory design [2, 3]. Modern astrodynamics texts regularly include the basic principles of gravity assists and MGA trajectories [4, 5, 6].

1.1.1 History of gravity assist missions

The application of gravity assist flybys has enabled missions which would have otherwise required too much ΔV. A gravity assist (or flyby or swing-by) uses a close passage to a massive body to exchange momentum between the flyby body and spacecraft, potentially significantly altering the spacecraft trajectory while altering the trajectory of the much more massive body by a negligible amount. Gravity assist flybys were first used by Mariner 10, which used a gravity assist at Venus to pass by Mercury [7], and Pioneer 11, which flew by Jupiter on the way to a flyby of Saturn [8]. After navigation of a gravity assist was demonstrated, gravity assists became a critical part of many space missions; Table 1.1 gives some key details on some selected flown and planned space missions which utilize gravity assists. Table 1.1 notably does not include any missions which did not successfully navigate any gravity assists or which only used the Moon as a gravity assist body.

Beginning with the Galileo mission to Jupiter, MGA trajectories have been used to create moon tours which use repeated flybys of one or more natural satellites of a planet to shape the spacecraft trajectory and achieve the desired science objectives. The final tour trajectory chosen for Galileo came after a series of studies [35, 36, 37, 38]; its calculation was the first major implementation of specialized computational aids for MGA trajectories [13]. The Cassini-Huygens mission implemented a moon tour at Saturn [19] which has been so successful at conserving spacecraft consumables and also effective at delivering valuable science that it has been extended twice like Galileo, but with the second extension lasting seven years [21]. Such an ambitious extension indicates the improvement in tools available to determine MGA trajectories. The inclusion of targeted low-altitude passes of small moons – namely because of interest in Enceladus – also shows the improvement of MGA trajectory methods since such small bodies provide little bending ability for effective gravity assists.

In addition to the many programs of record shown in Table 1.1, many other studies of potential missions have been conducted which include MGA trajectories to the outer planets [39, 40] as well as to Pluto and beyond [41, 42]. New moon tours at Jupiter [43, 44, 45] and Neptune [40] have also been studied, with a particular recent focus on missions to study Europa [46, 47, 48]. Similar methods have also been applied to gain the benefits of gravity assists

Table 1.1: Selected flown and planned space missions utilizing one or more gravity assist flybys.[a]

Mission	Destination(s)	Launch/ Start	Sequence	Ref.
Mariner 10	Venus/Mercury	1973	EVMe	[7]
Pioneer 11	Jupiter/Saturn	1973	EJS	[8]
Voyager 1	Jupiter/Saturn/Titan	1977	JSTi	[9]
Voyager 2	”grand tour”	1977	JSUN	[9]
Vega 1 & 2	Venus/Halley	1984	EVHa	[10]
Giotto	Halley/Grigg-Skjellerup	1985	EHaEGr	[11]
Galileo (IP)	Jupiter	1989	EVEEJ	[12]
Galileo (moon tour)	Jupiter	1995	{12 flybys}	[13]
Galileo (GEM tour)	Jupiter	1997	{+17 flybys}	[14]
Galileo (GMM tour)	Jupiter	2000	{+4 flybys}	[15]
Ulysses	Sun	1990	EJSun	[16]
NEAR-Shoemaker	Eros	1996	EEEros	[17]
Cassini-Huygens (IP)	Saturn/Titan	1997	EVVEJS	[18]
Cassini (moon tour)	Saturn	2004	{54 flybys}	[19]
Cassini (Equinox)	Saturn	2008	{+35 flybys}	[20]
Cassini (Solstice)	Saturn	2010	{+7 years tour}	[21]
Stardust/NExT	Wild-2/Tempel-1	1999	EEWiETe	[22]
Hayabusa	Itokawa	2003	EEIt-ItE	[23]
Rosetta	Churyumov-Gerasimenko	2004	EEMEECh	[24]
MESSENGER	Mercury	2004	EEVVMe^{3}Me	[25]
Deep Impact/EPOXI	Tempel-1/Hartley-2	2005	ETeEEEHar	[26]
New Horizons	Pluto	2006	EJP	[27]
Dawn	Vesta/Ceres	2007	EMVe-VeCe	[28]
Juno	Jupiter	2011	EEJ	[29]
Hayabusa2	1999 JU3	2014	EEJu-JuE	[30]
OSIRIS-REx	Bennu	2016	EEBe-BeE	[31]
BepiColombo	Mercury	2017	EEVVMe^{5}Me	[32]
Solar Probe+	Sun	2018	EV^{7}Sun	[33]
JUICE (IP)	Jupiter	2022	EVEEJ	[34]
JUICE (moon tour)	Jupiter	2030	{27 flybys}	[34]

[a] Mission information updated using online resources as needed: nasa.gov, esa.int, jaxa.jp, [Accessed April 4, 2015]

for deflecting near-earth asteroids [49] and for getting to Mars via one-way transfers [50] or cycler trajectories [51] as well as for satellite-aided capture at Jupiter [52, 53].

1.1.2 Methods for solving multiple gravity assist trajectories

With all the advantages of MGA trajectories, they are also difficult to calculate because the solution space can involve many continuous parameters with nonlinear constraints and many local optima, as investigated by [54, 55, 56]. As noted by these authors as well as those from [57], the models used for the MGA trajectory and the inclusion of more trajectory options such as maneuvers and low-thrust arcs affects the solution space and complicates the search for trajectories.

A variety of methods have been proposed to solve for MGA trajectories, with continuing active research in the topic, as described in the following sections. This introduction to methods is focused on global search (GS) and global optimization (GO) methods; there also exist local search [58] and local optimization methods for MGA trajectories [59] and spacecraft trajectories in general [60]. In addition, commercially available software and black-box optimizers are available for the local problem, as described by [61], and often applied to optimize a trajectory found by a global search tool. By contrast, global MGA trajectory search is generally conducted with specialized methods or by heavily modifying general GO codes.

1.1.2.1 Global search versus global optimization

A GS seeks to map a solution space within a set of constraints. A GO is aimed at locating global minimizers of a performance index. When there are multiple objectives and a single performance index cannot be defined, the Pareto front of the solution space is often sought by a GO method. In the multiobjective case, a GS and GO can provide similar end results, though a search will generally provide more solutions. Thorough and helpful introductions to MGA global optimization can be found in [57, 62].

An optimizer generally returns superior solutions to a search, though a properly performed search will have one or more solutions in the vicinity of the minimizers, which can then be found by a local optimization method. As a result, many GS and GO techniques also implement a local optimization portion. Finding the local minimizers is extra computational effort, but optimizing reduces the number of solutions within one basin of attraction, which can reduce total compute time for the search as a whole. In addition, the minimizers are the solutions of interest, so GO provides a more accurate view of the Pareto front than GS methods without a local optimization.

Promising solutions returned from a GS are almost always optimized in some way later (e.g. the work of [59] applied to refine original trajectory searches for Galileo [13]), but this optimization is only performed on relatively few individual solutions. As a result of this postprocessing, GS and GO may be functionally interchangeable for well-posed multiobjective problems, so a clear distinction between the methods is not maintained in this chapter except where differences between GS and GO are being noted.

1.1.2.2 Pathfinding versus pathsolving

Solution methods fall into three categories based on which part of the MGA trajectory problem they aim to solve. One subproblem of the MGA trajectory problem consists of determining a sequence of events where countable sets of categorical variables combine to outline a sequence. Such variables can include the number of gravity assists and the order of bodies in the gravity assist sequence. This sequence subproblem is called *pathfinding* or more generally *planning* [57]. The pathfinding problem results in a combinatorially large number of options for all but the simplest problems – and pathfinding is complicated by the need to solve for each of these paths to determine feasibility and/or optimality. Solving for particular solutions of a given sequence is *pathsolving* or *scheduling* and generally involves choosing values for continuous or mixed integer free parameters to meet constraints and/or minimize an objective function. Solution methods may focus on pathfinding, pathsolving, or be applicable to both. Techniques designed to handle both pathfinding and pathsolving may be called *hybrid* methods and may or may not be a combination of distinct GO methods. Such hybrid methods may separate the pathfinding and pathsolving into two subproblems or may be handle both as a single problem.

Global solution methods for MGA trajectories may be systematic or stochastic or a mixture of both types. Each approach has advantages and disadvantages; the following sections lay out the different methods, but details of implementations are often unpublished or vague. Because many people engaged in calculating MGA and other complex spacecraft trajectories are practitioners and interested in potentially flying the results of their search and optimization tools, the results are often published with only a cursory mention (or no mention at all) of the methodology used to discover them. Figure 1.1 breaks down some of the types of methods and different references which utilize them. Additional insight is gathered by personal correspondence regarding unpublished methods and prototype tools [63, 64]. Institution of the Global Trajectory Optimisation Competition (GTOC) which has been occurring every year or two since 2005 [65] has helped encourage dialog on different techniques and their effectiveness. However, GTOC problems are often solved by one-off solution methods which lean heavily on intuition and may not be more broadly applicable to other spacecraft trajectory problems.

1.1.2.3 Graphical methods

The earliest MGA design techniques were the combination of intuition and mathematical formulas and intensive manual processes. As the number of gravity assists increased, graphical approaches to laying out the solution space were introduced. Planning the Galileo moon tour options introduced some graphical methods for visualizing the mission as a whole and for visualizing what orbits are achievable by one or more gravity assists [37]. These graphical methods are systematic in nature but generally work in a reduced number of dimensions and level of fidelity compared to the systematic automatic pathsolving methods discussed in the next section.

Isoline graphs are one graphical-analytic tool; the graphs plot contours against two variables which fully define an orbit so that every point on the graph represents an orbit of given size and shape but arbitrary phasing [5]. These isoline graphs have been further refined for MGA trajectories as Tisserand graphs [66]. Tisserand graphs are limited to MGA trajectory pathfinding because they do not provide any information about phasing and so require another method to do the pathsolving for a given sequence. There is also a limit to how much information may be effectively portrayed and processed at once and the method for gravity assists is not suitable for eccentric body orbits or non-coplanar orbits [66]. Nevertheless, Tisserand graphs have been refined to include various information in a standardized way and to incorporate addi-

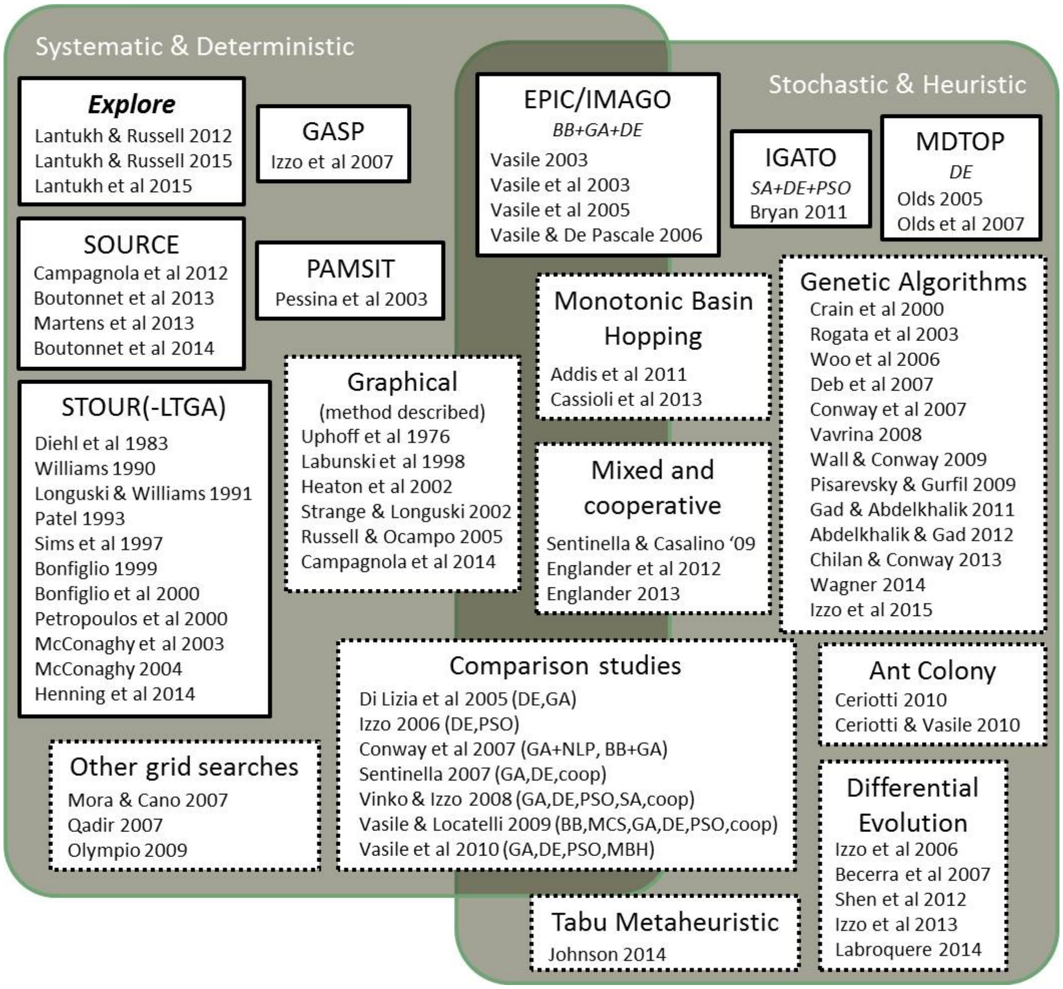


Figure 1.1: Several global search and optimization methods applied to MGA trajectories, along with associated references. Boxes with solid edges are named tools and boxes with dotted edges categorize the remaining references.

tional types of transfers [67] and recently extended beyond traditional patched conics models to Tisserand-leveraging transfers [68, 48].

Another graphical and numerical method for both pathfinding and pathsolving is the v_∞ sphere [5, 3]. This method for casting the problem has been used to study moon tours [69, 40] where it is particularly adept at mapping resonant transfers. Chapter 4, which presents automated methods for pathfinding and pathsolving $n\pi$ subsequences, utilizes v_∞ spheres and provides more details on the v_∞ sphere.

Graphical methods have continued to be used as part of the trajectory search process even as other more automated methods have been developed [66, 43]. Recent MGA search results have continued to effectively utilize graphical methods as part of the process [46, 48]. Graphical methods provide the designer insight into the entire design space, albeit in a limited number of dimensions, and allow significant reduction in needed pathsolving efforts where solving the problem globally might otherwise be computationally intractable.

1.1.2.4 Systematic methods

Systematic methods use grid or tree search techniques to predictably explore the design space, or at least the portions of it expected to harbor desirable solutions [70, 71]. These methods, also called deterministic approaches, are advantageous because they can be exhaustive and can be proven to converge globally and map the entire space. However, if the methods require discretization they are highly sensitive to resolution. Additionally, the time required to fully map the solution space of complex MGA trajectories is generally computationally prohibitive [54, 55, 72].

One of the earliest and most used MGA search tools is STOUR (Satellite Tour Design Program) originally developed at JPL for the Galileo mission [13]. This tool functions by sequentially progressing through a trajectory one leg at a time, where a leg is the trajectory between two body encounters. Along the way, the mission analyst makes decisions about the path and which trajectory to keep following. This process was subsequently automated [73, 74] and the tool has been improved to include aerogravity assists (AGA) [75, 76, 77], powered flybys [78], v_∞ leveraging maneuvers [39], and approximate low thrust transfers [79, 80]. STOUR has been the pathsolving component of several published trajectory searches, where the pathfinding was generally conducted with graphical tools [41, 39, 43].

Other deterministic methods for pathsolving have also been introduced, such as AGA tours using PAMSIT [81] and independent search algorithms similar to STOUR [82, 46] or based on linear programming [83]. Another simple though potentially inefficient approach is to apply a sophisticated local optimizer over a suitable grid of initial conditions [84]. A more recent tool, SOURCE, seeks to overcome the runtime limitations of previous systematic pathsolving methods by precomputing trajectory phases and applying fast vectorized filters to trajectory segments before any optimization is performed [71]. These innovations, along with specific separation of different types of solutions to preserve variety, has led to competitive results in several trajectory searches [40, 85, 86].

Deterministic algorithms have also been applied to aiding the search process without necessarily performing the pathsolving. These pruning or branch and bound (BB) methods include stand-alone methods like GASP (Gravity Assist Space Pruning) [70] and its follow-up pruning studies [87, 72, 88] which can be applied to limit the search bounds of a separate GO algorithm. Pruning may also be incorporated into the main search tool, as with SOURCE [71]. Some stochastic search tools – described in the next section – also utilize deterministic branching or pruning strategies [56, 89, 57, 45]. *Explore*, the MGA pathsolving tool described in Chapter 2 and in [90], is an automated deterministic method which performs the pathsolving using a grid search while also progressively pruning the search space.

1.1.2.5 Stochastic methods

A variety of stochastic methods have been applied to the MGA search problem as well, beginning with the application of genetic algorithms (GA) [91]. Stochastic methods use probability and a set of rules to decide where to search next within the design space, potentially providing them some adaptability and better scaling with number of parameters than grid search methods. However, such methods often rely on heuristics with their own tuning parameters and there is no general proof of exhaustiveness or convergence to a global optimum with stochastic methods [57, 92]. Both deterministic and stochastic methods may rely on heuristics, but stochastic methods specifically utilize one or more random variables in the search process such that the search itself and its results are dependent on random numbers (or, in practice, the seed of the pseudorandom number generator).

Several different stochastic methods have been applied to the search for MGA trajectories, as shown in Figure 1.1. Stochastic methods have often been paired with one or more deterministic or other stochastic methods to achieve faster convergence or more global coverage [93, 94]. Other types of challenging spacecraft trajectories have also been approached using stochastic methods (e.g. [95] use of GA) but the work described in this dissertation focuses specifically on MGA applications.

Even within a single stochastic method there are various types of each method. For example, a GA may be multi-objective [95] or variable population size [96] or niching [91] or a combination of the above [97]. There is a wide variety of refinements of methods and of variations on a given theme. This occurs because stochastic methods rely on heuristics which can be tuned or altered to achieve different results [93]. While tuning can help specialize a method to a particular problem – such as generating the best known solution to the challenging GTOC4 problem [92] – the tuning is a challenging and time-consuming process.

One way to work around tuning for a specific problem is to run multiple stochastic methods in parallel, using a cooperative method that can involve several populations with different constraints, tuning, or even different optimization methods [98, 99, 100]. Cooperative combinations have been shown to improve results even when the individual methods are not necessarily optimally tuned to the problem. Improvements in evolutionary methods for trajectory optimization have recently been showcased by providing one of the best-known solutions to the GTOC6 problem, albeit not in time for the competition itself [45, 97]. Nevertheless this method was recognized for automatically producing results competitive with those a human could produce, earning a Humies gold medal in genetic programming in 2013 [101].

Dividing the GS and GO methodologies between stochastic and systematic types is an imperfect distinction because many solution methods involve both a deterministic and a stochastic component, hence the overlapping area in Figure 1.1. As mentioned above, deterministic BB and pruning is applied to reduce the search space for some stochastic methods. In addition, Monotonic

Basin Hopping (MBH) assumes a local optimization level to locate the graph of minimizers, so MBH is usually paired with a standard deterministic local optimizer [102, 62, 103]. Several studies have paired a GA with a local optimizer in a two-level approach to find many local minima [104, 105, 106]. This pairing of a GA or other GO method with a local optimizer is often applied for LT MGA trajectories [107, 108, 109, 110]. In such cases, each optimization can take significant computation time and attempting an exhaustive grid or BB search is even less tractable than the ballistic or impulsive ΔV cases.

1.2 Parameter reduction of complex trajectories

One common theme for speeding up the search for and optimization of trajectories is to cleverly design the routines at the innermost loops which are called the most. The current section introduces the idea of reducing the number of parameters which must be decided or searched over and so reducing search time. The reduction is accomplished by combining the appropriate set of assumptions with a solution algorithm. Introductions to the trajectory types follow, with algorithms presented in their respective chapters.

1.2.1 Models and assumptions

For the methods and implementations developed in the current dissertation, the following general assumptions are used throughout unless otherwise stated: two-body dynamics govern the spacecraft motion about the primary body; the maneuvers are instantaneous, impulsive, occur at an apse, and are aligned with the spacecraft velocity; and instantaneous zero-radius sphere of influence gravity-assist flybys occur at the minor body locations. All the stated approximations are consistent with several previous studies [2, 56, 73].

Although these preliminary model assumptions generally work better for interplanetary trajectories than intermoon trajectories, they are also applicable to high-energy moon tours.

1.2.2 v_∞ leveraging transfers

The ability to include instantaneous maneuvers efficiently in a search is valuable, and many of the pathsolving methods introduced above aim to achieve this end. By specifying a particular type of maneuver, it is possible to greatly reduce the search space and so the computation time required to locate maneuvers. Choosing v_∞ leveraging transfers (VILTs) as the subset of maneuvers is advantageous as these transfers have been demonstrated to enable reduced launch mass, ΔV, or flight time.

Beginning with the ΔV-Earth gravity assist (ΔV-EGA), various studies and papers have laid the theoretical framework for VILTs and shown their utility [111, 112, 113]. VILT techniques have been extended to more general interplanetary transfers, including example preliminary design cases [112, 114, 115]. The same techniques also apply to moon tours [116, 117, 118]. These methods extend beyond theory and have been implemented for several flight missions, including Cassini, NEAR, Juno, and MESSENGER [18, 17, 119, 25].

Implementing a VILT method effectively in the conceptual design stage requires both making the correct set of assumptions and also creating an implementation that is fast and robust. Previous approaches to this problem have looked at both choosing the sequence of flyby bodies (the pathfinding problem with VILTs) and the method of implementing the VILT itself [112, 113, 114, 115, 116, 117, 118, 18]. The VILT method formulation is the focus of the Chapter 3: the assumptions, the inputs/outputs, the function behavior, and some example applications. The inclusion of VILTs into the search process is accomplished by formulating them as similarly as possible to the Lambert problem. The motivation for such a formulation is to be compatible with existing broad multi-flyby trajectory search techniques that are fundamentally based on Lambert solutions [3, 2, 56, 73]. Chapter 2 presents just such an algorithm, implemented in the search tool *Explore*. This similarity to the traditional Lambert problem enables the same search algorithm to handle both the Lambert and VILT methods, so both ballistic and VILT segments can be included in the search in any combination. In effect, the presented VILT formulation provides a straight-forward and efficient mechanism to include maneuvers in broad trajectory searches or any other application where computation time is important.

Similar to the Lambert solution, the VILT method developed is a function primarily of boundary position vectors and connecting flight time. Therefore, the assumed circular orbit of the flyby body of previous VILT formulations is not required and ephemeris or arbitrary locations can be used for the bodies. The formulation developed also handles both tangent and non-tangent body encounters for the VILTs, as defined in the Types of VILTs subsection in Chapter 3.

Furthermore, since the VILT method is decoupled from the bodies, a VILT-like transfer that originates and ends at different flyby bodies is possible, which is termed an interbody VILT (IB-VILT). Although the IB-VILT has been used by both Cassini and MESSENGER, the connection of these maneuvers to VILTs has only recently been explored systematically [120] using the method described in detail in this dissertation. Chapter 3, where a specific solution method is proposed and explored, further highlights the advantages of choosing this class of maneuvers since the method extends beyond traditional VILTs to more general velocity-aligned apsidal maneuvers (VAMs).

1.2.3 $n\pi$ transfers

The next category of trajectories to be incorporated into the MGA GS are $n\pi$ transfers. Free return coasting trajectories whose transfer angle is an integer multiple of π ($n\pi$ transfers) present unique advantages and challenges in the context of the pathfinding and pathsolving problems because of the extra degrees of freedom associated with the planes of the transfers [3, 121, 122, 123]. While this makes an $n\pi$ transfer extremely useful for overcoming phasing and flyby altitude constraints and enabling interesting science orbits, selecting the correct degree of freedom (DOF) is not trivial, especially in the case of broad searches where this selection must be repeated many times [123, 124, 114, 125, 126, 127].

This process of choosing the correct DOF is further complicated when combining multiple $n\pi$ transfers in a sequence, where each transfer has its own DOF and the selection of these DOFs is dependent of each other. Despite these complications, the resulting trajectories have helped enable difficult missions like Galileo, Cassini and MESSENGER [13, 19, 25]. A fast and robust way to include sequences of $n\pi$ transfers and determine the corresponding DOFs is the motivation of presented method, detailed in Chapter 4. The implementations of $n\pi$ sequences within STOUR [13] SOURCE [71] follow similar principles but do not include automated resonance-hopping like the one detailed in this dissertation, a method first presented in 2012 [128].

Other published [73] and commercially available methods [129] can also include $n\pi$ transfers in the trajectory search process, but they currently require some knowledge about the sequence DOFs or the number and type of transfers in a sequence, or they impose other limits such as the number of transfers. Increasing the flexibility of the trajectory search by lifting these restrictions is a primary motivation of the $n\pi$ sequence methods presented in Chapter 4.

In addition to the assumptions made in the broad search, each segment of the spacecraft trajectory that is an $n\pi$ transfer is a ballistic conic section with the same encounter body at both ends. Also, the gravity assist flybys are instantaneous changes in spacecraft velocity by rotations of the v_∞ vector. These zero-radius patched conics assumptions are consistent with previously published methods and studies, as indicated above [66, 56, 73].

The flybys themselves are constrained by the maximum angle that the v_∞ vector can be rotated by one flyby: δ_{MAX}. For the zero-radius patched conics approximation, δ is determined as a function of the periapse distance of the flyby, allowing the maximum turn angle (δ_{MAX}) to be specified by the minimum allowed altitude of the flyby. This relation, a simple substitution of equations for a hyperbolic trajectory from [4], is given in Equation (1.1).

$$\delta = 2\sin^{-1}\left(\frac{\mu_B}{\mu_B + r_P v_\infty^2}\right) \tag{1.1}$$

In general, the segments in the search can encounter the same body at both ends of the segment or two different flyby bodies. When the encounter body is the same, the trajectory is described as a ballistic return or a free return. Free returns can be subdivided into $n\pi$ transfers and generic returns, where generic returns are all non-$n\pi$ transfers. In this generic return case, the transfer plane is well defined by the position of the flyby bodies at the initial and final times, so a Lambert solution can lead to a connecting transfer. On the other hand, the plane of an $n\pi$ transfer is not defined, so a different approach is required [3, 121].

Because $n\pi$ transfers can provide inclined trajectories with respect to the flyby body and because their regular, repeating encounter locations can enable a kind of loitering at a body, their inclusion in the GS described Chapter 2 is a primary motivation for the development of the method. Including $n\pi$ transfers requires a means of choosing the transfer plane. Because the flyby v_∞ is often known from previous segments of the search or targeted by mission constraints, the problem of choosing the plane is simplified by considering a given v_∞ at a particular flyby body. If both bounding v_∞ vectors are known (in the context of a tour search algorithm), the $n\pi$ transfer problem can be reduced to a boundary value problem on the v_∞ sphere (see Chapter 4 for more details). This approach also allows the inclusion of a sequence of successive $n\pi$ transfers whose DOFs are optimized together.

All $n\pi$ transfers can be categorized as even $n\pi$ and odd $n\pi$ transfers based on whether the transfer angle is an even or odd multiple of π, respectively. For the purposes of this discussion, transfer angle always refers to the angle that the flyby body sweeps out in its orbit between the two encounters of a transfer. Even $n\pi$ transfers (often called resonant or full-rev transfers) encounter the body in the same location at both ends of the transfer. Identical r at both encounters also results in identical v_∞ at both encounters [3]. Odd $n\pi$ transfers are an extension of the back-flip transfer (π-transfer) into multiple spacecraft revolutions and also are referred to as half-rev transfers. For non-circular body orbits, half-rev transfers with $r_1 \neq r_2$, have different spacecraft v_∞ at each encounter [3]. Examples of an even $n\pi$ transfer and an odd $n\pi$ transfer are shown in Figure 1.2.

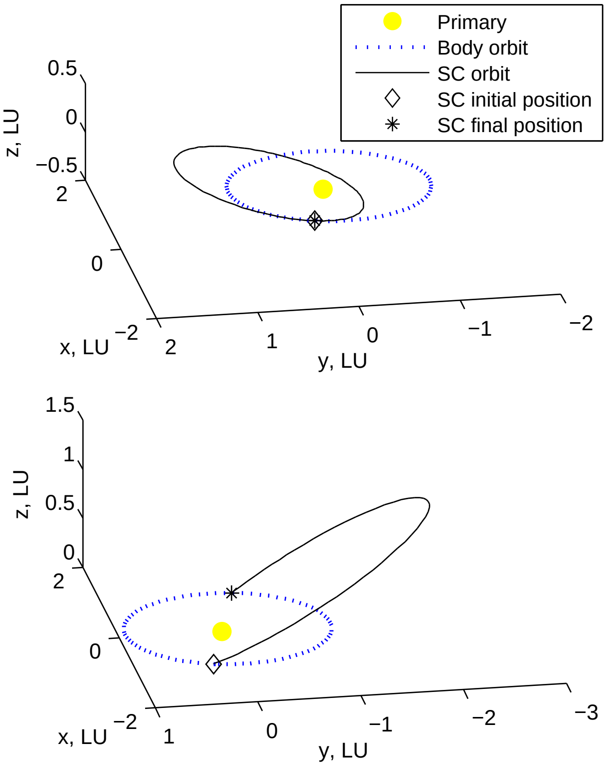


Figure 1.2: Examples of an even $n\pi$ transfer (top, $n = 2$) and an odd $n\pi$ transfer (bottom, $n = 1$)

1.3 Primitive body proximity operations

Orbital dynamics are particularly challenging for missions to small bodies, not only because of the ΔV and timing of reaching the body, but also because perturbations lead to non-Keplerian proximity orbits at these bodies. Solar Radiation Pressure (SRP) and irregular central body gravity distribution are often two of the most significant perturbations to spacecraft orbital dynamics in close proximity to small primitive bodies [130]. Several solutions for stable spacecraft orbits have been developed for orbit altitudes where one of these two is the dominant perturbation. When SRP is dominant, then the terminator, equatorial Sun-frozen, and quasi-terminator orbits may be possible [131, 132, 133]. When body oblateness is dominant, then Sun-synchronous and precessing orbits provide potential solutions of interest [6, ch. 11]. When an irregular gravity field is the dominant perturbation, frozen orbits and body-fixed periodic orbits are of particular interest [134, 135].

1.3.1 Introduction to Sun-frozen and heliotropic orbits

Where SRP and irregular central body gravity distribution are comparable perturbations and primary drivers of orbital motion, both need to be taken into account to find orbit families. Sun-frozen orbits and heliotropic orbits are two categories of orbits which can exist stably in this dynamic regime by taking advantage of SRP and body oblateness perturbations. These two categories of orbits and especially heliotropic orbits are the focus of presented investigations – other perturbations, orbit families, and dynamical regimes around small bodies are rich areas of research as well but are outside the scope of this dissertation. These two categories overlap: Heliotropic orbits in the equatorial plane are Sun-frozen if the body has no obliquity.

In general, a frozen-eccentricity orbit requires that the eccentricity vector be constant in time [6, p. 802]. By analogy, a Sun-frozen orbit requires that the eccentricity vector is constant with respect to the Sun line. Inclined (3D) heliotropic orbits are, by definition, Sun-frozen in an averaged sense; However, in the 3D case, the instantaneous eccentricity vector circulates, thus preventing 3D heliotropic orbits from technically being Sun-frozen.

In the context of strong SRP and body oblateness perturbations heliotropic orbits are particularly attractive for low-altitude science orbits. Heliotropic orbits are a class of orbits which maintain long orbit lifetimes by combining these perturbations to keep the apoapsis pointed towards the Sun on average, as shown in Figure 1.3. The name "heliotropic" comes from botany, where the term was coined in 1832 and describes the tendency of plant stems, leaves, and flowers to bend toward the Sun [136, p. 36]. There are also antiheliotropic orbits with periapsis in the direction of the Sun [137, 138], but these orbits are not explored further in the analyses presented within the current dissertation.

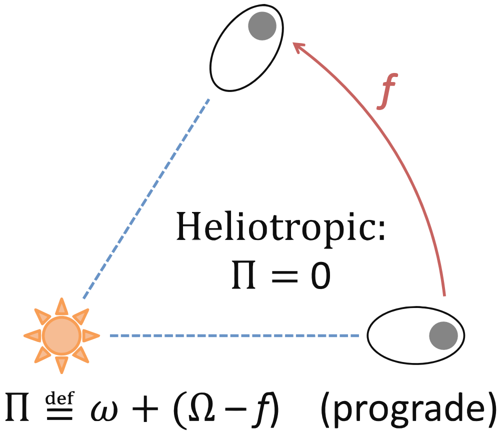

f
Heliotropic:
Π = 0
$\Pi \overset{\text{def}}{=} \omega + (\Omega - f)$ (prograde)

Figure 1.3: A heliotropic orbit uses SRP and zonal gravity perturbations to keep its apoapsis pointed towards the Sun on average. See Chapter 5 for definitions of angles and additional discussion of heliotropic orbits.

Heliotropic orbits are part of an orbit class that was discovered in the context of studying the dynamics of orbiting planetary dust, with the term coined for orbits in the study of Saturnian ring dynamics [139, 140]. Heliotropic orbits were identified as Sun-frozen orbits in the equatorial dynamics of circumplanetary dust, with some study of the behavior of inclined orbits as well [137, 141, 142]. The resulting orbits were then proposed for high area-to-mass ratio (HAMR) spacecraft orbiting Earth, necessitating improved methods to investigate out-of-plane and non-zero obliquity characteristics [143, 138, 144]. Past studies have only included the second-degree zonal gravity harmonic (J_2) in calculating heliotropic orbits [137, 141, 142, 143, 138], with a recent extension of these orbits to non-equatorial cases and application to small-body missions [145] as also described in this dissertation. Using only SRP and J_2 works well for finding heliotropic orbits at planets and large moons and asteroids, but most small bodies have sufficiently non-spherical shape that higher degree zonal terms contribute significantly to the oblateness perturbation.

When considering the effect of non-spherical body perturbations, the even zonal gravity harmonics are particularly important because they dominate the secular change of the orbital elements [146]. The secular effects of zonal harmonics are determined by averaging the corresponding terms of the disturbing potential, first over one period of the spacecraft mean anomaly and then over one period of the argument of periapsis [147, 148]. The averaging leaves only the even terms to impact the secular change of the elements [149]. The resulting disturbing potentials are applied to the Lagrange Planetary Equations (LPE) to determine the averaged dynamics of the orbital elements [150].

1.3.2 Applying heliotropic orbits to asteroids

As explained further in Chapter 5, heliotropic orbits require body obliquity to be near zero or 180°. They also require a generally oblate spheroidal shape for the body, so heliotropic orbits are not applicable at all primitive bodies.

Although in general primitive bodies span the full range of shapes and obliquities, the Yarkovsky-O'Keefe-Radzievskii-Paddack effect (YORP) torque

[151, 152] drives a disproportionate number of smaller bodies to have spin vectors closer to the ecliptic ±z-axis [153]. These statistics, along with the shapes of about half of known asteroids being mostly oblate [154], lead to the preliminary estimate that roughly 25% of small asteroids may meet the assumptions used here in development of the heliotropic orbits [145]. No systematic method has been applied to find maximum acceptable deviations from the assumptions, but specific application to Bennu parameters is demonstrated in Chapter 5 and for Earth parameters in [143, 138, 144].

Bennu is particularly conducive to supporting heliotropic orbits because the asteroid is approximately an oblate spheroid and has a pole approximately 88 deg from the ecliptic plane [155]. Radar and lightcurve data for Bennu have been used to estimate the shape and corresponding constant-density gravity model as well as the gravitational parameter [155, 156]. In addition, Bennu is scientifically interesting, as evidenced by its selection as the primary target for the OSIRIS-REx sample return mission.

Heliotropic orbits are investigated in depth in Chapter 5. Averaging techniques and the Lagrange Planetary Equations (LPE) are used together to isolate secular effects of the perturbations [147, 148, 149, 146, 150]. First, a singly-averaged Sun-frozen orbit analysis is performed. This analysis identifies several orbit families, including the equatorial heliotropic orbits. The inclusion of higher degree zonal gravity terms and the search for heliotropic orbits out of the equatorial plane are accomplished by preforming a constrained second average on the SRP disturbing potential. In addition, preliminary studies are conducted on the effect of body gravity uncertainty and the robustness of the heliotropic orbits to a complex gravity field. An estimate of Bennu's gravity field is used for this purpose.

1.4 Summary of contributions and dissertation structure

A variety of topics have been introduced in the current chapter to set up for the detailed descriptions provided in the following chapters. Publications resulting from and planned for the material presented within this dissertation are summarized in Appendix B, which also connects the publications to the different chapters. Some of this dissertation – notably work relating to GTOC6 and certain examples – is not planned for any separate publications.

Chapter 2 focuses on the MGA global search, presenting an effective algorithm which is implemented as *Explore*, a GS pathsolving tool. The search algorithm is a cascade of boundary value subproblems with similarities to existing methods. This problem decomposition enables the incorporation of different trajectory types and patch point types.

- Original pruning strategies are discussed beyond what has been published before, including the pruning conditions and when in the search process they are applied. Although the time required to exhaustively solve a given problem is generally unknown, performance index based pruning is introduced to limit the search space to that which fits within computational resources. Additionally, multi-objective pruning using Pareto ranking of solution candidates is introduced as a practical way to judiciously choose a subset of the space to search.

- An efficient implementation is described, including details of the memory structure and the flow of the search process that have not been provided in other literature.

These important details meet some of the challenges associated with a GS, helping keep the run time tractable for presented problems of interest.

Chapter 3 presents an original and unique boundary value formulation for solving VILTs. Setting it apart from other methods in the literature, the presented method is independent of the encounter body orbit parameters and can be applied more broadly as a means to calculate VAMs.

- The mathematical basis of the VILT BVP and an implementation algorithm which can solve the VILT with a similar solution structure and run time to solving the multi-revolution Lambert problem are provided. This solution algorithm is characterized and several VILT cases are examined.
- The presented VILT method is applied in an MGA GS (*Explore*) and trajectory searches including VILTs in traditional and novel (VAM) roles are presented. Specifically, the effects of using ephemeris locations, the effects of applying an inter-body VILT, and the use of VAMs for planetary capture and LT initial guesses are discussed.

Chapter 4 develops an original and unique method for automatically determining $n\pi$ sequences by solving a boundary value problem on the v_∞ sphere. This extension of past v_∞ sphere applications provides an automated method for performing both pathfinding and pathsolving on the v_∞ sphere.

- The BVP is defined by a novel gap calculator that allows the search to proceed to the next leg of the trajectory and determine the second boundary condition.
- Two methods for solving the BVP are introduced, implemented, and discussed: A new two-level approach that combines an exhaustive pathfind-

ing method with fast feasibility pathsolving using box flyby altitude constraints is developed in detail. An alternative hybrid stochastic pathfinding/pathsolving sphere-walking method – which was used to solve the GTOC6 problem – is also described; in this method a random variable and a weighted performance index are used to take each step along the sphere, with MS providing multiple options to choose between for each set of boundary conditions.

The focus of Chapter 4 is on applications to an MGA GS. Examples of solutions of the BVP are presented along with examples of application within *Explore*.

Chapter 5 considers Sun-frozen and heliotropic orbits to asteroid orbiter missions.

- As part of the first published application of heliotropic orbits to small body orbiters, orbit characteristics and limits are explored, including the derivation of an inclination limit for heliotropic orbits.
- Particular focus is placed on extending heliotropic orbit theory to 3D orbits and to orbits utilizing high degree zonal harmonics, two aspects of these orbits that are new in the literature.
- The analytical theory of 3D heliotropic orbits is enabled by an innovative application of constrained averaging to the SRP disturbing potential.
- Applications focus on orbits around a model of Bennu (the target for the planned mission OSIRIS-REx), including preliminary uncertainty analysis calculating the probability of existence of heliotropic orbits at the Bennu model given the uncertainty of the gravity and SRP parameters.

Chapter 6 provides a summary of the dissertation with a focus on applicability of the methods and implications of the results. Considerations for future work are also discussed.

Chapter 2

Trajectory search algorithm

A pathsolving tool called *Explore* is introduced in this chapter[1]. The systematic grid search and pruning algorithm at the core of *Explore* is described, with some qualitative comparison to other methods mentioned in Chapter 1. Pruning methods implemented in the search are described, including multi-objective Pareto ranking. The flexibility of the pathsolving algorithm is demonstrated by the inclusion of different trajectory and patch point types. Comparative performance is given for the different trajectory types included in a broad global search (GS). The presented pathsolving algorithm has some unique characteristics, but the detailed implementation description and the discussion on pruning strategies with multiple objectives are the primary contributions of this chapter to the state of the art in MGA GS.

2.1 Posing the pathsolving problem

The pathsolving problem involves the implementation of a mission concept to the initial trajectory design, answering the basic questions of when the

[1] The presented tool has also been published in:

- Demyan Lantukh and Ryan P. Russell. "Multi-Objective Search for Multiple Gravity Assist Trajectories," AAS/AIAA Astrodynamics Specialist Conference, 9-13 August 2015, Vail, CO

Ryan P. Russell contributed proof-of-concept and the ballistic Lambert solver used in *Explore* as well as development of various modules of the tool not presented here. He also provided general development guidance and management.

spacecraft will go where, how it will get there, and what the cost (e.g. ΔV) will be. As described in Chapter 1, the pathfinding problem determines the sequence of events in a trajectory and outlines the spacecraft path categorically. The pathsolving problem then attempts to find a trajectory which corresponds to this set of categorical parameter choices, possibly with the added requirement that it be a locally or globally optimal trajectory. In the general case, exhaustive pathsolving is intractable, but specific problems can be solved at sufficient resolution. In such cases, reduced algorithm run time and number of parameters to discretize enables the investigation of larger and more complex solution spaces. In this chapter, a systematic GS method for pathsolving is developed in detail with specific focus on addressing run time, problem scaling, and dimensionality issues associated with its implementation.

The complexity of the problem and the infinite options motivate software tools to aid in the pathsolving problem. In the current chapter, the focus is on the "suggested-sequence" pathsolving problem, where the pathsolving algorithm may be allowed to change categorical variables like path length within certain specified parameters or limitations. The software (here *Explore* specifically) receives as input a desired sequence of important events such as flybys and maneuvers in addition to all relevant start, end, and path constraints. Figure 2.1 illustrates the concept of the suggested-sequence pathsolving tool.

The sequence provided for pathsolving is an ordered list of nodes (B_i) and legs (L_i) connecting them, up to a specified number of legs (l). Equation (2.1) shows this sequence definition with the nodes defined by their inbound (I) and outbound (O) times as needed. Note that the definition of a node by two times allows more flexibility in node types and makes the presented GS implementation distinct from previously published algorithms such

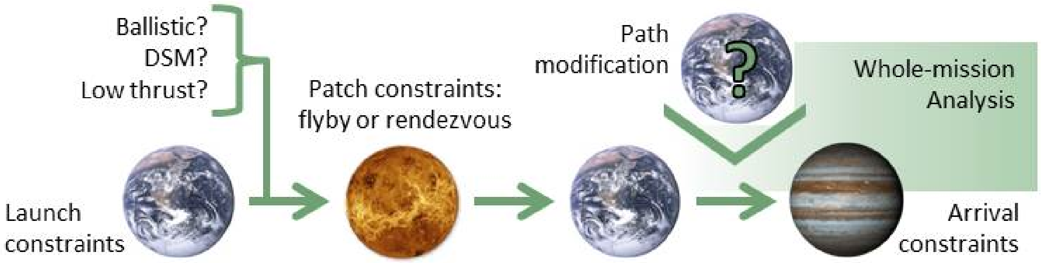


Figure 2.1: Schematic of how a mission concept – turned into a specific sequence – is passed to the suggested-sequence pathsolving tool to determine appropriate details of implementation. The suggested sequence is EVEJ.

as those given by [73, 74, 71, 83, 82, 46].

$$B_1(t_1^O) \to L_1 \to B_2(t_2^I, t_2^O) \to L_2 \to B_3(t_3^I, t_3^O) \cdots \to L_l \to B_{l+1}(t_{l+1}^I) \quad (2.1)$$

Table 2.1 outlines the definition of the scheduling problem for a given sequence. This high-level definition is implemented in *Explore*, where it is also extended for other kinds of pruning, leg types, and node types as described below.

Each node is associated with an ephemeris object b (nominally a celestial body), so a given time allows the state of b to be determined. The set of all bodies which may be used in a sequence is $\{b\}$, each with its corresponding parameters. Given an inbound and outbound time for each node, the legs are bounded by known body states and each leg is a boundary value subproblem. The first and last nodes in a sequence have only one associated time and $\mathbf{v}_\infty$ but all other nodes have two times and a pair of $\mathbf{v}_\infty^I$ and $\mathbf{v}_\infty^O$, defining each of these nodes as a boundary value subproblem as well.[2] In addition, each leg has particular options and pruning conditions which apply to it, and each

[2] *Explore* also allows the user to optionally specify $\mathbf{v}_\infty^I$ for B_1; the first node is then solved like other nodes.

Table 2.1: Definition of MGA pathsolving as a node scheduling problem including some feasibility constraints

GIVEN:

Problem definition: m_0 (initial mass), l (number of legs),
$b_i,\ R_i\ \forall\ i = 1, 2, 3, \cdots, l+1$

System definition: μ_{Pr}, $\{b\}$: μ_{body}, r_{body}, ephemeris $\forall\ b \in \{b\}$

Constraint definition: box (*MIN* and *MAX* defined for each parameter)
Sequence: $TOF_{mission}$, $\Delta V_{mission}$
Nodes:
$v^O_{\infty i},\ t^O_i,\ \Delta V^O_i\ \forall\ i = 1, 2, 3, \cdots, l$
$v^I_{\infty i},\ t^I_i,\ \Delta V^I_i\ \forall\ i = 2, 3, 4, \cdots, l+1$
Δt_i (dwell time) $\forall\ i = 2, 3, 4, \cdots, l$
$h_i \forall\ i \in \{\text{flyby}\}$
Legs: $TOF_i,\ \Delta V_i,\ N_i\ \forall\ i = 1, 2, 3, \cdots, l$

FIND:

ALL set(s):
$t^O_i\ \forall\ i = 1, 2, 3, \cdots, l$
$t^I_i\ \forall\ i = 2, 3, 4, \cdots, l+1$
Additional information needed define $L_i\ \forall\ i = 1, 2, 3, \cdots, l$:
e.g. ballistic transfers: N_i, type (short or long period)
Additional information needed define $B_i\ \forall\ i = 1, 2, 3, \cdots, l+1$
Other outputs of interest: *e.g.* $\mathbf{v}^I_\infty\ \forall\ i = 2, 3, 4, \cdots, l+1$

SUBJECT TO:

All defined box constraints

node has particular options and pruning conditions which apply to it. These conditions may apply to all nodes (or legs) of a particular type (e.g. all flybys), all nodes at a particular body b (e.g. at Venus), or to a specific element in the ordered set (e.g. B_3 or L_2).

Figure 2.1 provides an example sequence where EVEJ is the set of nodes ($B_1 - B_4$) of a three-leg trajectory. This set of nodes and the applicable timing constraints are input by the user. This sequence is one of multiple possible ways to perform a mission concept (e.g. interplanetary trajectory to Jupiter).

Segments are patched together by nodes with patch constraints; in the EVEJ gravity assists make the logical patch constraints, but a different mission may perform multiple-body rendezvous and have rendezvous and stay times as a patch constraint. The $n\pi$ sequence algorithm developed in Chapter 4 is also a patch constraint, albeit a complicated one, with respect to the broader MGA GS. In the case shown in Figure 2.1 the designer is leaving the trajectory type free, allowing the search to investigate ballistic, VILT, DSM, and LT options. Leaving all options free to consider may make a search intractable, but one advantage of a systematic method is the option to do so.

The sequence provided is called *suggested* because, ideally, the software tool should be able to calculate and evaluate modifications such as adding maneuvers or removing flybys within the set parameters. The current algorithm described here and implemented in *Explore* is a step towards that end, with the ability to efficiently calculate a given sequence of flybys. Methods described in the Chapter 3 add the ability to include maneuvers and some capability to suggest removal of a maneuver by driving ΔV near zero. The $n\pi$ sequence method in Chapter 4 includes the addition of flybys via multiple linked resonant transfers. There is currently no implemented capability to remove a flyby from the suggested sequence list.

2.2 Sequential path search algorithm

The presented search algorithm – also described in [90] – decomposes the problem into a cascade of subproblems where each subproblem connects two encounters in the MGA sequence, similar to the problem decomposition done by [74, 70, 157] with favorable results. As described above, legs and nodes alternate as different BVPs in this cascade of subproblems. For the case of ballistic transfers such a sequential decomposition can be pruned in polynomial time, though the exhaustive GS itself is still NP-hard but on a smaller domain [70].

The presented GS method has the following key features to aid in efficient computation of trajectories:

1. Breadth-first implementation enables straightforward low-level parallelization and solution evaluation against comparable candidate trajectories as well as efficient, scalable memory handling

2. Discretization of body orbits with a grid search and discontinuity corrector algorithm for matching flyby constraints keeps only feasible trajectories and allows pruning conditions based on physical quantities with straightforward interpretations

3. Feed-forward search simplifies the search process and reduces total computations by only allowing change on one leg of the trajectory at a time

4. Cascade of BVPs allows straightforward implementation of different trajectory types without major modifications to the GS algorithm

5. Search tree pruning based both on hard constraints and on solution desirability counter the exponential nature of the search process

Note that points 2 and 4, and to some extent point 3, fall in the heritage of [73, 74, 78, 81] for MGA GS. The specifics of memory handling on a modern computer for problem scalability, however, are new to this dissertation. The detailed discussion of pruning with multiple objectives and multiple levels of pruning within the search – details which have proven extremely helpful in use

– are also new to the literature. These beneficial characteristics come out of the algorithm described as follows:

2.2.1 Discretization

The search process begins by defining a set of nodes the spacecraft will visit and the possible range of times at which the spacecraft may be at each node. Each node must have a means of defining the state of its corresponding ephemeris object within that range of times. For simplicity, the nodes are all assumed to be natural bodies within the Solar System with defined ephemerides, but the concept extends to rendezvous with spacecraft or single-point or path equality constraints, as described later.

Each of these nodes is discretized in time along its path according to user-specified resolution parameters R_i. For celestial bodies, the number of discretization points per body orbit revolution (n) is a convenient metric for resolution that spans across different instances of a search. In this case, discretization is done with equal steps in time; for highly eccentric bodies a different discretization may be preferable, but using time keeps ephemeris calculations relatively fast and simple. Using an anomaly for discretization was considered as an alternative and would make a useful future extension. However, using time allows a straightforward implementation of TOF box constraints and avoids root-solving for the anomalies which correspond to particular times. Figure 2.2 shows a notional search for an EVEJ trajectory from a particular epoch. For visualization purposes the grid discretization is much coarser than an actual search would use. The first leg of the search, Earth to Venus, is also shown in Figure 2.2 and discussed below.

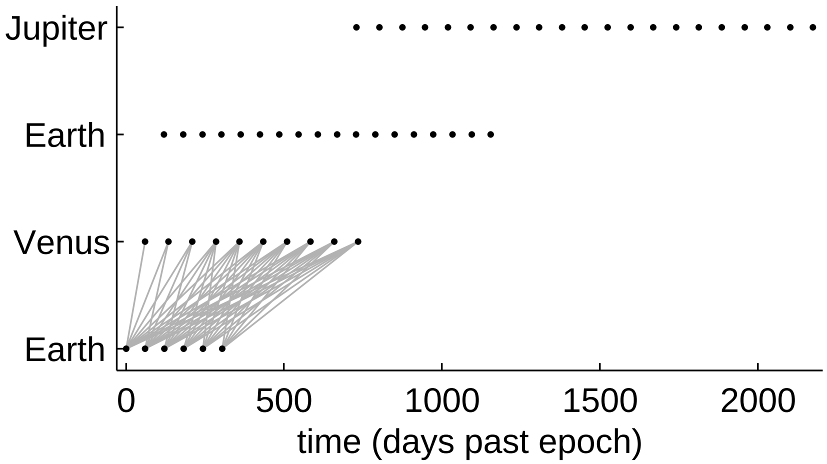


Figure 2.2: Notional EVEJ search: discretized grid and first leg all-to-all search

2.2.2 First Leg: all-to-all grid search

The first leg of the trajectory L_1 is calculated by attempting to connect each discretized point from B_1 to each discretized point from B_2 which occurs later in time and satisfies all other user-specified constraints which can be evaluated (e.g. flight time). This attempt to connect the first two nodes is termed the all-to-all grid search. Apart from the application of constraints, number of computations for the first leg grows with the square of discretization point density, so properly constraining the search can provide great benefits in computation time. Further constraints can be applied once the leg is evaluated, and these constraints will help reduce the computation time of subsequent legs. Additional details on constraint implementation are discussed in the section devoted to that topic. Figure 2.2 shows the resulting connection points for the first leg of the example EVEJ sequence without any additional constraints. As mentioned before, discretization resolution has intentionally been set low in Figure 2.2 so that the individual discretization points are clearly visible.

Since knowing the discretization points on B_1 and B_2 fully defines the set of BVP for L_1, the solution process for these BVPs is readily parallelizable. A parallel ballistic Lambert solver which utilizes the Graphics Processor Unit (GPU) has been implemented by colleagues and incorporated into *Explore*. Parallel computation for solving the set of BVPs at each leg or node can also be implemented using multiple CPUs as an area for future work. There is also an opportunity for parallelization based on problem subdivision in memory as described in the section below devoted to memory management.

For the initial implementation of the algorithm, two nodes are connected by a ballistic two-body transfer between a discretization point on each of the two nodes. Since each discretization point has associated time and position data, the resulting connection is a well-defined Lambert problem. To help make the search more applicable to a broad range of spacecraft missions, a multi-revolution Lambert solver using universal variables is implemented, based on the method provided by [4].

2.2.3 n^{th} Leg, part 1: one-to-all grid search

The first leg of the trajectory requires special treatment in the search algorithm, but every following leg is treated identically. Each leg begins with a set of discrete valid points on the preceding node, where each valid point is connected to the first node by a candidate trajectory encompassing all intermediate nodes.

A grid search is implemented which attempts to connect each valid point from the preceding node to each discretized point on the current node which is later in time and satisfies all constraints which can be evaluated. The number of search computations scales with product of the number of preceding

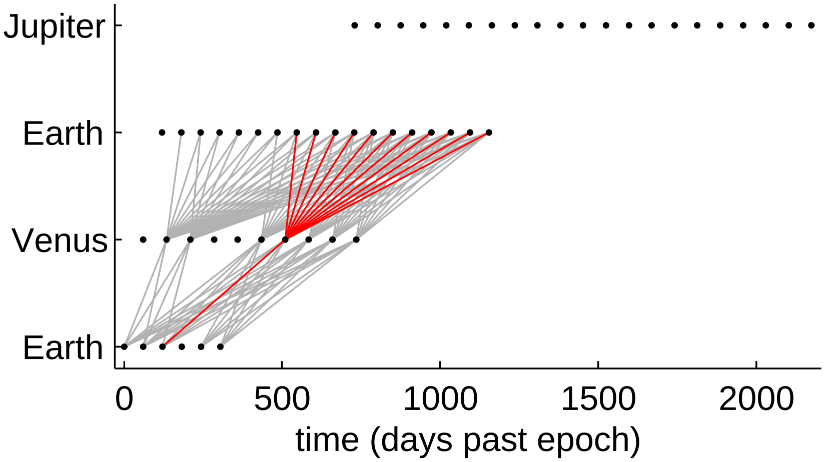


Figure 2.3: Notional EVEJ search: continuing the search for the Venus-Earth leg from all points still valid at Venus. One particular EV trajectory with its VE branches is highlighted.

node points and number of discretization points on the current nodes. As a result, pruning the search on any applicable constraints is extremely helpful for reducing the downstream search time. Figure 2.3 shows the calculation of L_2 (Venus-Earth) leg of the example EVEJ sequence. One candidate L_1 trajectory and the resulting fan of candidate L_2 trajectories stemming from it are highlighted to illustrate the search process. In Figure 2.3, some discretization points at Venus are not considered in calculating L_2 because they have been pruned out – that is, they are not part of any valid candidate trajectories in the search going forward.

2.2.4 n^{th} Leg, part 2: patch constraint root solve

With a grid search from the preceding to current node complete, the resulting trajectories are not necessarily valid because positions align at the patch point but there may be a patching condition at the previous node. When there is no dwell time at the preceding node, an instantaneous maneuver may be used to patch the two legs of the trajectory or a zero sphere of influence gravity assist flyby may be used. A powered flyby combines both of these into a single patch point.

In the case of a ballistic flyby, two conditions need to be met: $v_{\infty}^{I} = v_{\infty}^{O}$ for the flyby and the turn angle of the flyby must be achievable with an allowed flyby altitude. With a particular set of trajectories from a single preceding node like that highlighted in Figure 2.3, none of the discretization points on the current node (e.g. second Earth in Figure 2.3) may satisfy the flyby conditions at the preceding node (Venus in Figure 2.3), but there still may exist one or more valid points between discretization points.

The equality v_{∞} constraint is met by root-finding for a valid trajectory. This root-finding procedure is designed to alter only one leg of the candidate trajectory, leaving unchanged all preceding legs. Equation (2.2) provides the specific equation zeroed when calculating L_i:

$$\Delta v_{\infty,i} = v_{\infty,i}^{O}(t_{i+1}^{I}) - v_{\infty,i}^{I} = 0 \tag{2.2}$$

In Equation (2.2), $v_{\infty,i}^{I}$ is determined by solving L_{i-1} and is fixed for each candidate trajectory since t_i^O is fixed at this point in the search process. Choosing to keep t_i^O fixed when meeting the equality v_{∞} constraint keeps both the root-finding process and candidate solution storage relatively simple. The only remaining free parameter for a given trajectory type (e.g. ballistic, $N = 1$, short-period transfer) is t_{i+1}^{I}; as it varies, so does the $v_{\infty,i}^{O}$ to reach the corresponding state at B_{i+1} with the given trajectory type.

Figure 2.4 shows the Δv_{∞} function in Equation (2.2) for a Venus-Earth leg of the EVEJ trajectory; only the t_{i+1}^{I} and corresponding arrival location of

Earth are being varied along the Δv_∞ function. As Figure 2.4 shows, there may be more than one solution for a given starting node and given trajectory type. The required t^I_{i+1} (or equivalently TOF) to satisfy Equation (2.2) are each bounded and an initial guess for each set of bounds is generated by comparing the Δv_∞ of comparable trajectories: if there is a sign change in Δv_∞ of two trajectories from the same preceding node point to adjacent discretization points on the current node, then those two trajectories bound a root-finding search for a trajectory which meets the v_∞ constraint in Equation (2.2). Two trajectories must be otherwise identical to bound a root-finding search: for the ballistic case they must have identical number of revolutions and be the same family of solutions (short or long period).

The high points in Δv_∞ seen in Figure 2.4 correspond to the $n\pi$ transfers, here seen in the interplanetary (Venus-Earth) case instead of the traditional single-body case. The two steep spikes correspond to odd $n\pi$ transfers, which have a high Δv_∞ because small differences in body planes lead the ballistic connection to occur near polar inclination. Small correction maneuvers near these ballistic solutions can lead to much lower inclination transfers and generate interesting solutions [71]. Although these trajectories can be found with *Explore* using the optimized DSM transfer (described below), the description of these trajectories is beyond the scope of the current investigation. The other two high Δv_∞ points in Figure 2.4 (which only have one steep side) correspond to even $n\pi$ transfers. The high Δv_∞ in these cases comes from an unfavorable geometry associated with limiting all shown trajectories to a fixed number of orbit revolutions. The resulting ballistic connections at and somewhat beyond the even $n\pi$ transfer point require much higher flight path angle at the departure body (here Venus) and a corresponding higher v_∞.

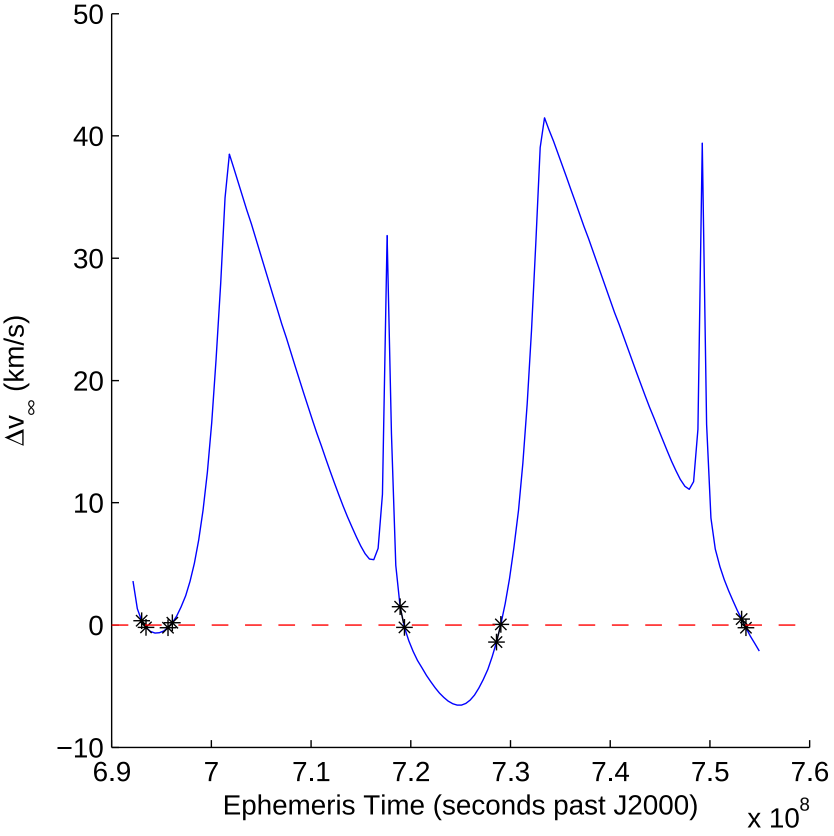


Figure 2.4: Example of the Δv_∞ function defined in Equation (2.2). Black stars show discretization points which fall on both sides of the zero crossing; these pairs of points bound the root-finding which removes the v_∞ discontinuity.

In all cases, the steep Δv_∞ areas can cause solutions to be missed if body discretization is not sufficiently high.

Neighboring grid points where Δv_∞ crosses zero are identified with stars in Figure 2.4. The identification of neighboring grid points can be computationally expensive if performing an all-to-all pairwise comparison ($O(n^2)$ algorithm) of candidate trajectories. Alternatively, this comparison can take significant memory if solutions are stored in a structure by trajectory type to make comparison faster. If the order of solutions in memory is unknown, the comparison remains $O(n^2)$ but with n greatly reduced by only comparing like trajectories. Knowledge of solution order in the array can be used to check each candidate solution only against its neighbors in memory, reducing run time to $O(n)$ but requiring a solution array or helper array of solution indices with (maximum possible solutions) × (maximum possible solution types) elements. An effective compromise of the two comparison methods was found by taking advantage of solution order and using a helper array of integers to keep track of the last solution index of each trajectory type that has been checked. The result is a single $O(n)$ pass of the solution list with each candidate solution first compared against the solution of like type referenced in the helper array. After the comparison, the candidate solution takes the place in the helper array of the solution it was compared against.

Once the bounding pairs of trajectories are identified, root-finding is performed on each pair. In *Explore* a bounded secant method[3] is the primary method used for this process. Internal to the root-finding routine is a Lambert problem solution, which is itself iterative. Although satisfying this flyby

[3] A customized variant of root.f available from `http://netlib.org/napack/root.f` (accessed September 24, 2010) is utilized in *Explore*

constraint is relatively expensive computationally, it the ability to use flyby parameters instead of Cartesian states in a subsequent trajectory optimization can provide run time and convergence benefits for the optimizer [59]. An additional advantage of satisfying the flyby constraint is the ability to find ballistic solutions with a relatively course discretization. The alternative – leaving a v_∞ discontinuity at the flyby point to be optimized later – also requires the user to provide a maximum size for these discontinuities and to estimate their effect on any ΔV constraints since they are expected to be reduced to near zero once optimized.

Once the v_∞ constraint is met, the flyby altitude constraint can be checked for the resulting trajectory. Since the process of meeting the v_∞ constraint is computationally intensive a heuristic pre-pruning can be performed: the pair of solutions bounding the root-finding problem can be checked against the altitude constraint, v_∞ magnitude constraint, and other pruning conditions: if neither of the bounding solutions meets the same constraint and the points are close enough that the solution space and constraint functions can be approximated as linear between them, then the root-finding can be skipped entirely since the resulting solution with matching v_∞ will also not meet the constraint. This pruning idea is suggested as a heuristic only; in practice it is difficult to establish how close trajectories need to be for the solution space between them to be approximately linear. The required discretization can change with different problem and trajectory types.

Completing the v_∞ matching for a leg of the trajectory effectively creates a new grid at the current node as shown in Figure 2.5. This new grid enables valid candidate solutions leading up to it. Once the trajectories are valid, they can be pruned according to any applicable leg or node constraints.

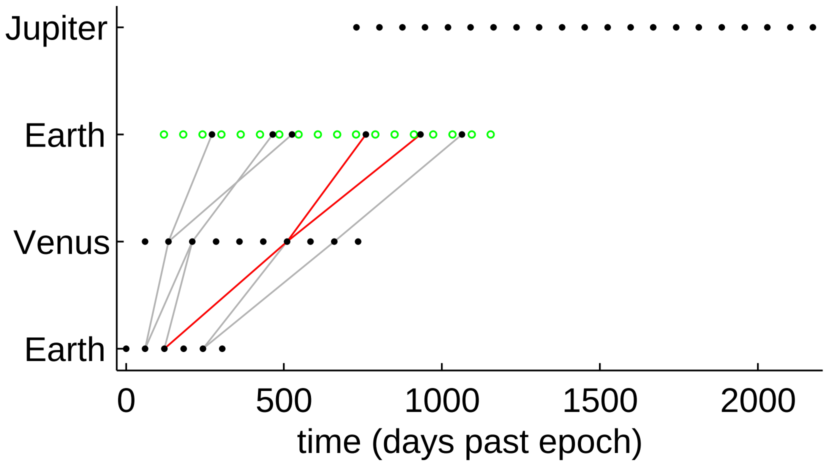


Figure 2.5: Notional EVEJ search: the search to the second Earth encounter after v_∞ matching at the Venus gravity assist. One particular trajectory with its branches is highlighted.

As Figure 2.5 shows conceptually, performing the v_∞ matching greatly reduces the valid candidate trajectories and so reduces the search space for the following leg.

2.3 Search tree pruning

As mentioned in the search description above, pruning out solutions which will not yield favorable candidate solutions is a critical part of the process. If computing resources and sufficient runtime exist, an unconstrained trajectory search can map the solution space and constraints can be applied on the resulting solutions to filter them to the best candidates. However, for all but the simplest trajectories, some kind of pruning needs to be performed during the trajectory search to keep it computationally tractable in a reasonable time.

Two types of pruning conditions are implemented. First, feasibility constraints set boundaries outside which solutions are not considered. These constraints are defined by the designer, so satisfying them does not necessarily correspond to a flyable trajectory, but the constraints do have physical significance (e.g. flyby altitude, v_∞ magnitude, etc.). The second type of pruning conditions are performance-index based conditions. In these conditions the designer specifies the desirable direction of a particular quantity of interest and how the search space should be pruned when there are many trajectory options. Figure 2.6 provides an overview of the search process for each trajectory leg with the pruning included. The following sections describe in more detail these two types of pruning conditions.

2.3.1 Feasibility constraint pruning

Feasibility constraint pruning removes trajectories from consideration when they violate a given constraint. For example, a designer may specify that launch $C_3 \leq 50\ km^2/s^2$ so any solutions that do not meet this constraint are immediately pruned out. In the presented feed-forward approach, eliminating candidate solutions early in the search process with effective pruning leads to significant reductions in trajectory computations and so in search runtime. Note from Figure 2.6 that feasibility pruning is done before discontinuity handling to reduce computations needed for this matching and then this pruning is done again on the valid continuous trajectories.

Alternatively, a designer can specify that a subsurface flyby is allowed even though this is physically infeasible. The freedom to violate physical constraints allows the designer to broaden the search in the hope that a higher resolution search or an optimizer can later tweak the outputs to make the

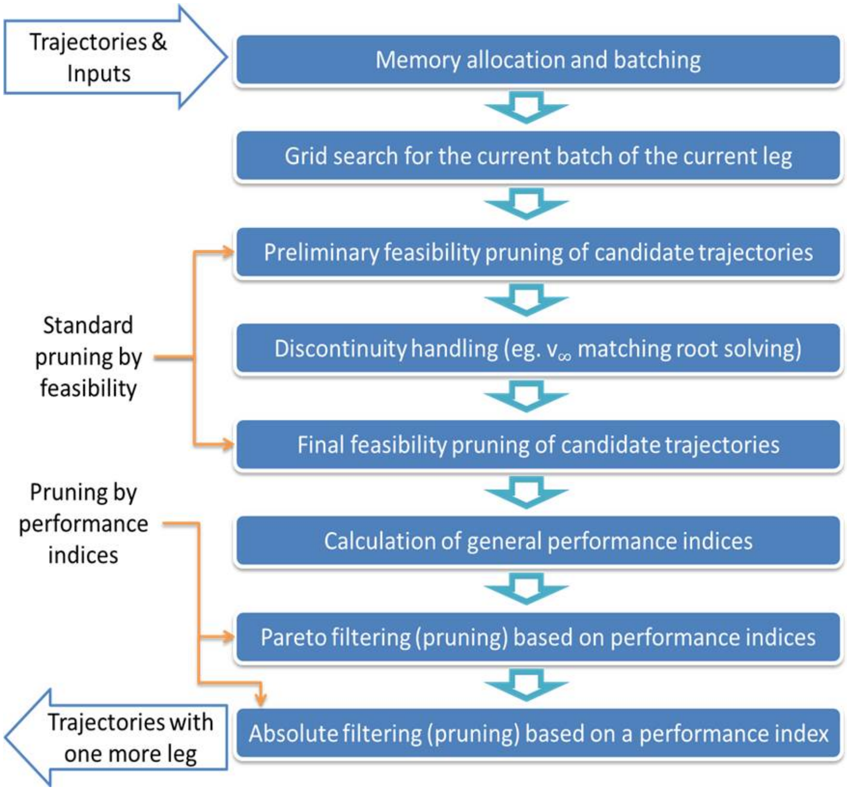

Trajectories & Inputs
Memory allocation and batching
Grid search for the current batch of the current leg
Preliminary feasibility pruning of candidate trajectories
Standard pruning by feasibility
Discontinuity handling (eg. v_∞ matching root solving)
Final feasibility pruning of candidate trajectories
Pruning by performance indices
Calculation of general performance indices
Pareto filtering (pruning) based on performance indices
Trajectories with one more leg
Absolute filtering (pruning) based on a performance index

Figure 2.6: Flowchart for the calculation of one memory batch of a trajectory leg in *Explore*. Note the application of various pruning techniques: effectively pruning the search space can be critical to keeping the search tractable. More detailed descriptions of the search process are provided in Algorithms 2.1-2.2.

resulting trajectories feasible. This relaxation of physical constraints can also be helpful when no trajectories are found which are feasible so that the designer can get an idea of how far from feasibility the trajectories are and which physical constraints they violate. Similarly, keeping track of which constraints are active (e.g. how many trajectories are pruned by each constraint) can provide valuable feedback about the solution space.

In this view, constraints can be used as "tuning parameters" for the search and can provide significant speedups when implemented well. One advantage of tuning a search with these types of values is their physical significance: the values for constraints and the value for body discretization required to find a particular trajectory reflects on the physical characteristics of that trajectory neighborhood. For example, if a particular search yields many more solutions in one family than in another, the family with more solutions is likely more robust to changes in initial conditions or constraints. Alternatively, a subsurface ballistic flyby can indicate that a powered flyby or deep space maneuver is needed to make the trajectory feasible and the ΔV of the powered flyby can be estimated using the difference in achievable and required bend angles.

2.3.2 Performance-index based Pareto pruning

In optimization, performance indices are used as the target for optimization, with the trajectory (or other free parameters) varied to minimize (or maximize) the given indices. In the context of the trajectory search algorithm presented, the trajectory is not changed once calculated and so the performance indices do not provide this kind of feedback. Instead, the indices are figures of merit which are useful for comparing similar trajectories. One way to use performance indices is to keep up to a user specified maximum (n_{MAX}) number of solutions; if the number of solutions exceeds n_{MAX} then only the best n_{MAX} solutions according to a performance index are kept and the remaining solutions are pruned out of the search. This type of pruning is termed *absolute filtering* because a hard cutoff is used to keep a fixed number of trajectories.

A single performance index can be an aggregate of different quantities such as launch C_3 and spacecraft ΔV, but it is more useful to be able to compare trajectories without the need to aggregate and provide relative weights. Pareto optimality provides one means for comparison of different solutions across multiple performance indices in a scale-invariant way, eliminating the need for user-specified weights on performance quantities [95]. A solution is said to be Pareto optimal if, when it is compared with a set of candidate solutions, it cannot be made strictly better in any performance index without being made strictly worse in another performance index [158].

The basic principle of Pareto optimality is applied recursively to determine a Pareto rank (k) as shown in Figure 2.7 and described below. Note that Figure 2.7 highlights the eleven nondominated solutions with distinct letters; the corresponding trajectories are shown in Figure 2.8.

Pareto sorting separates solutions into two categories: dominated solutions and non-dominated solutions. Non-dominated solutions have no other solutions which are better in any of the active performance indices. When Pareto sorting is applied to the whole solution space, the resulting non-dominated solutions are assigned Pareto rank $k = 1$. The remaining (dominated) solutions are then sorted again by the same process and the non-dominated solutions in this set are given $k = 2$. The process can be repeated with the dominated

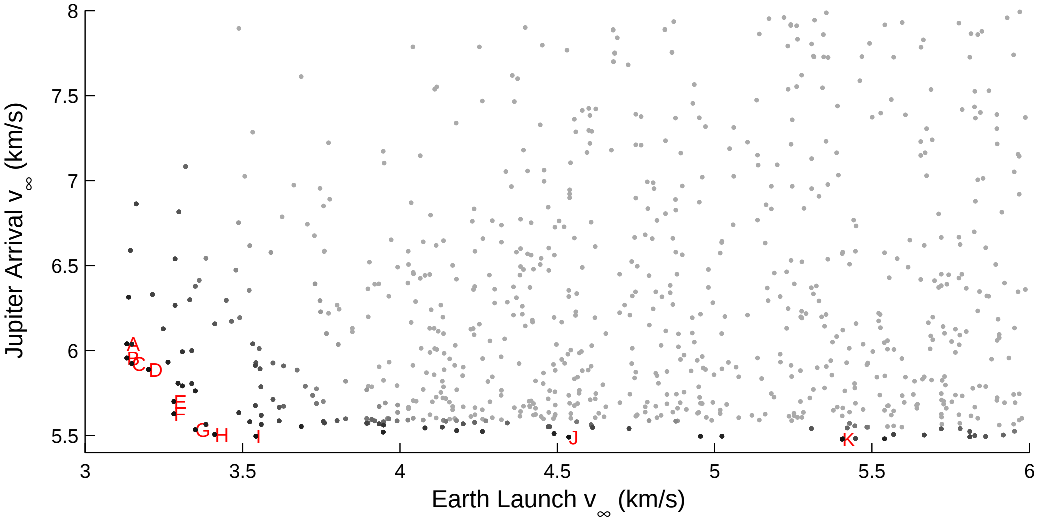


Figure 2.7: Pareto front of arrival v_∞ vs launch v_∞ along an EVEJ trajectory search with the flexibility to include extra resonant flybys. The darker solutions have lower Pareto rank (closer to the front), with all solutions of rank ≥ 10 being the same shade of light grey. The eleven Pareto optimal (rank $= 1$) solutions are marked by distinct letters with the corresponding trajectories shown in Figure 2.8.

solutions until all solutions are ranked or a user-specified maximum rank is reached, as shown in Figure 2.7 (where all solutions with rank of 10 or more have the same shade).

A user-specified number of Pareto ranks may be then kept for continuing the search and all remaining solutions pruned out. Since there is no way to know in advance how many solutions Pareto ranks will actually entail (the number of solutions in each rank is heavily dependent on the distribution), it is difficult to know what Pareto rank will not stifle the search but still provide enough of a limitation of the search space to keep computation time low. As shown in Figure 2.6, successively applying the Pareto-based pruning and then absolute filtering based on a single performance index can help provide the best of both types of performance-index based pruning since the n_{MAX} solutions kept will have lowest Pareto ranks within the solution space.

As with many forms of search tree pruning, the pruning by performance index is performed before the entire trajectory is evaluated, so a solution that has $k = 2$ on a particular leg of the search may have $k = 1$ or $k = 100$ when considering the whole trajectory or may have no corresponding solutions at all, having become infeasible at a later leg of the trajectory search. The choice of performance indices also greatly affects the effectiveness of pruning. For example, arrival or final v_∞ is not defined until the last leg of the trajectory so it is not useful for pruning until the entire trajectory is calculated. As another example, accumulated ΔV, being monotonically nondecreasing, is a useful comparison metric at any leg. As a result of these difficulties with pruning, choosing performance indices and how much of the solution space to search is often an art guided by experience. Appendix C provides a solution to the GTOC6 problem that was found using *Explore* and the methods described in this chapter, including carefully chosen performance indices which varied leg-to-leg along the different phases of the trajectory.

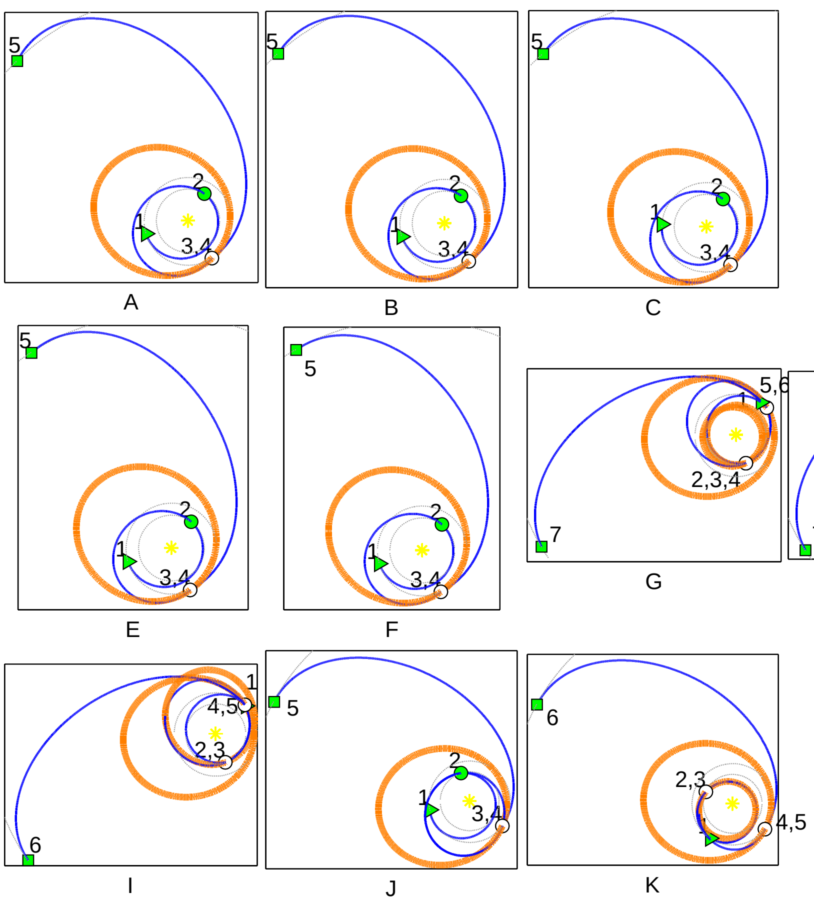


Figure 2.8: Example trajectories from the Pareto front of the EVEJ search shown in Figure 2.7, identified by their corresponding letters. Gravity assists are numbered sequentially, showing the inclusion of additional resonant flybys: Trajectories A–F, J follow the sequence EVEEJ; trajectories G–H follow the sequence EVVVEEJ, and trajectories I and K use the sequence EVVEEJ. Resonant transfers are shown with thicker lines.

2.4 Including non-ballistic legs

The trajectory search algorithm has been described assuming that discretization points on adjacent nodes are connected with ballistic two-body transfers. It is also possible to use a different method to connect these points, including low-thrust arcs or impulsive maneuvers. Any method to connect the different nodes needs to solve the boundary value problem inherent in the Lambert problem: connecting two positions with a specified TOF. The method may introduce a velocity discontinuity, as the Lambert problem does, but then a patching constraint must be solved in an additional step.

One of the powerful advantages of the presented algorithm for search and pruning is that it does not limit how the nodes may be connected as long as the quantities used for pruning can be determined. Some customization is required for each type of trajectory implemented to connect nodes, but unlike methods which rely on the structure of the problem, the presented algorithm and architecture do not significantly limit what those trajectory types may be or how they are connected at the nodes. One example of the benefits of such flexibility is shown in Figure 2.8; in this figure, example trajectories that used resonant flybys to augment the suggested EVEJ sequence are shown. These eleven trajectories are the approximate Pareto front within the constraints of their search, as shown in Figure 2.7.

Specifically, the following types of trajectory legs or transfers have been incorporated into the developed trajectory search tool. Most of these have been developed or implemented by colleagues, but all required system integration by the author.

- Ballistic: multi-revolution Lambert algorithm using a variation of universal variables – including hyperbolic, parabolic, and retrograde trajectories

- Parallelized ballistic: multi-revolution Lambert algorithm implemented on a GPU for faster computation taking advantage of the parallelizable nature of the search algorithm

- VILT: v_∞-leveraging based method that enables general velocity-aligned apsidal maneuvers (VAM)

- Optimized impulsive maneuver: local gradient-based optimizer with stochastic multi-start initialization to find multiple local minima. Can be constrained to place maneuver inside a specific time range within the leg

- Low-thrust: shape-based multi-revolution technique for feasible suboptimal low-thrust trajectories with fast computation times

2.5 Including other node types

The different trajectory legs may be patched with different types of nodes. The nodes themselves may be planets, moons, spacecraft, or anything which can be defined by an ephemeris, including arbitrary points in space, as long as a patched conics assumption remains valid. Node patching can be accomplished in multiple ways, with implemented methods including:

- Flyby (nominal case): instantaneous patched-conics ballistic gravity assist enforcing v_∞ matching and altitude bounds

- Rendezvous: instantaneous rendezvous and departure maneuvers with a dwell time in between for staying at an intermediate body. Multiple models are available for determining the ΔV required based on the concept of operations at the body

- $n\pi$ flyby sequence: addition of $n\pi$ transfers and required flybys, filling in time gaps, taking the trajectory out of the plane, and adding flybys as needed to satisfy turn angle constraints. An internal DOF optimizer can be used accomplish a specified goal with the $n\pi$ sequence; such as minimizing the maximum turn angle of n flybys, or mapping the body with periapse passages, as was done for GTOC6.

The two following chapters are devoted to derivation and demonstration of the VAM/VILT method and the $n\pi$ sequence patching node.

2.6 Search performance

The multi-leg trajectory search is intended to be efficient and scalable to different types and classes of missions, including interplanetary and moon tour missions which operate on very different time scales. The architecture is also intended to scale to multiple types of transfers and different patching conditions. Several of these different transfers and patching conditions have been implemented within the search architecture described, with timing performance summarized in Table 2.2. This table is focused on run times. Both run time and solution quality are highly dependent on the problem chosen and the constraints used, but run time can be accurately reflected even if no solutions generated are feasible.

Search parameters for the different cases described are given in Table 2.3. The body locations are provided by the NAIF SPICE database de421

Table 2.2: Comparison of trajectory searches with different trajectory types. Note that adding $n\pi$ sequences does change the MGA sequence. Empty spaces indicate cases which are not applicable or were not tested.

Trajectories	EM		EVEJ	
considered	run time (norm.)	# solns.	run time (norm.)	# solns.
Ballistic	1.00	1810	11.86	1
Ballistic+VILT	1.830	2814	101.36	1
Ballistic+LT	56.10	17200	2147.74	1
Ballistic+VILT (nontangent)	58.52	20105		
Ballistic+DSM (optimized)	228.69	7810		
Ballistic+$n\pi$ Seq.			53.43	208
Ballistic+VILT+$n\pi$ Seq.			558.99	235

available online. The search architecture is also scalable in terms of the computer used; in this case searches are conducted on a desktop workstation but timing is provided in relative units assuming only single-core processing. The same *Explore* software has also been used on more powerful servers and much less powerful laptops. With the provided parameters on the machine used for the test, a single unit of normalized search time in Table 2.2 is 1.56 seconds.

2.7 *Explore* architecture and implementation details

The algorithm presented in this chapter is coded in Fortran 95 and compiled with Intel Fortran as the tool *Explore*. The software tool provides a variety of inputs, outputs, diagnostics, and utilities to the main algorithms it hosts: The v_∞ matching (one-dimensional root-finding) is conducted with a bounded secant method[4] which has been found to work very well for the problems *Explore* encounters. There are also other zeroth and first order

[4] A customized variant of root.f available from `http://netlib.org/napack/root.f` (accessed September 24, 2010)

Table 2.3: Parameters for the searches timed in Table 2.2

Parameter	EM	EVEJ
Launch window	01/01/2018-12/31/2021	01/01/2020-12/31/2021
max TOF (days)	730.5	2920
max launch v_∞ (km/s)	6	6
max revs. per leg	2	2
max resonant loiter (revs per leg)		2
max mission ΔV (km/s)	2	2
max ΔV per leg	2	1
discretization		
nontangent VILT (deg)	5	
Earth (ppr)	55	36
Mars (ppr)	65	
Venus (ppr)		30
Jupiter (ppr)		60

root-finders and minimizers within the code as well as basic astrodynamics calculators and interoperability with the SPICE toolkit. One of the major implementation challenges has been interoperability of the different parts within *Explore* as the tool complexity has increased, requiring a balance between writing fast code and code that is easy to maintainable.

2.7.1 Architecture and algorithm

Figure 2.9 shows a basic architectural diagram of *Explore*, which also includes a helpful user interface developed in MATLAB. Algorithms 2.1-2.2 spell out the specific steps and loops in the pathsolving process: Algorithm 2.2 is the memory batch algorithm used within Algorithm 2.1 where the solution space subdivided and solved in batches that each fit into available memory. In these algorithms, s is used to indicate a solution counter, where the subscript distinguishes the various solution counters. A indicates an array of data for candidate trajectories: A_{prev} stores pertinent data of the candidate trajectories generated by pathsolving to the preceding node (the candidate trajectories themselves are stored in unformatted binary files which are referenced by A_{prev} as described below), A_{input} defines the set of BVPs to be solved and A_{output} contains data pertaining to the solutions of the BVPs.

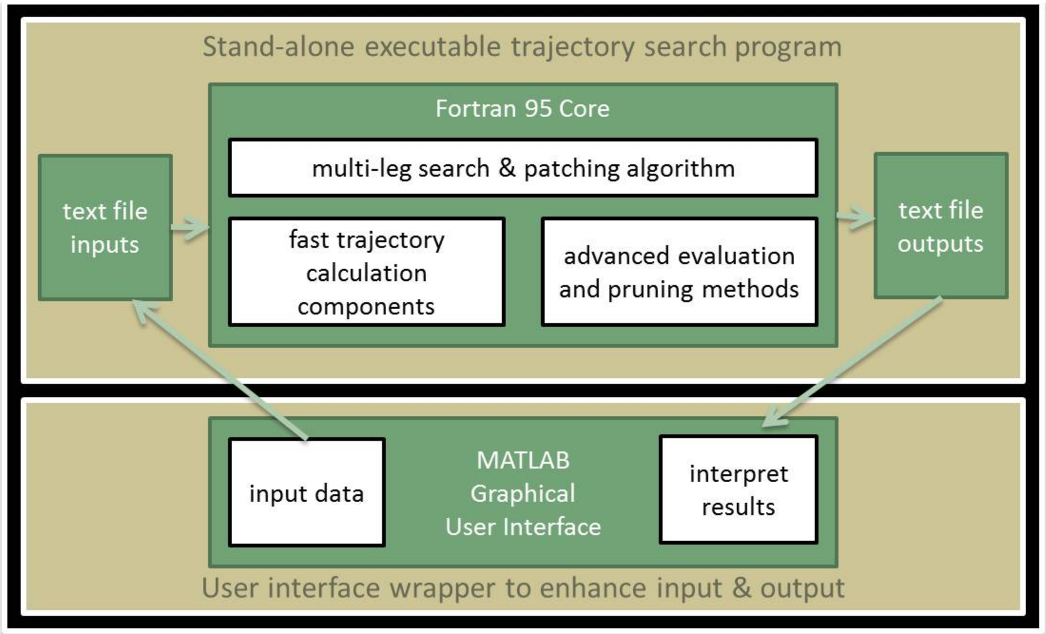

Stand-alone executable trajectory search program
Fortran 95 Core
multi-leg search & patching algorithm
text file inputs
fast trajectory calculation components
advanced evaluation and pruning methods
text file outputs
input data
MATLAB Graphical User Interface
interpret results
User interface wrapper to enhance input & output

Figure 2.9: Summary of the architecture and components of *Explore*, the pathsolving tool presented in this dissertation.

2.7.2 Memory management: problem subdivision

Some of the other challenges with implementing *Explore* have been storing and tracking solutions and managing memory use. Solutions are stored in structures to help with maintainability, but the number of solutions investigated at one time can be as high as tens or hundreds of millions, so special care needs to be taken with how solutions are handled.

Algorithm 2.1 Process for pathsolving the suggested-path scheduling problem outlined in Table 2.1 as implemented in *Explore* using innovative pruning and available RAM

```
Read input files
Interpret inputs and initialize search
Create list of discretized encounter body states at $B_1$
$s_{valid}$ = number of discretized points in $B_1$ list
Save $B_1$ list to corresponding file as valid candidate solutions
$L = 1$
while $L \leq l$ do
    Solve leg $L$ using Algorithm 2.2 to subdivide solution space to available memory, returning $n_{batch}$
    if $n_{batch} > 1$ then Redo performance index pruning for all solutions:
        Load all valid solutions to node $B_{L+1}$
        Rank and prune valid solutions by Pareto optimality
        Prune to specified $n_{MAX}$ solutions (absolute filtering)
        Save remaining solutions to file for node $B_{L+1}$, overwriting previous file
    end if
    $s_{valid}$ = Valid candidate solutions to node $B_{L+1}$ saved in file
    $L = L + 1$
end while
Create output files
for $i = 1 \cdots s_{valid}$ do
    Load solution $i$ from files starting with $B_{l+1}$ and following linked list to $B_1$
    Recalculate solution $i$ to regenerate data not stored in files
    Write solution $i$ data to output files
end for
```

Algorithm 2.2 Memory batch loop for solving leg L within Algorithm 2.1

```
Input: $s_{valid}$ from calling Algorithm 2.1
$s_{start} = 1$; $s_{end} = s_{valid}$; $s_{done} = 0$; $n_{batch} = 0$
while $s_{done} < s_{valid}$ do
    $n_{batch} = n_{batch} + 1$
    while $A_{input}$ and $A_{output}$ unallocated do
        Load solutions $s_{start}$ to $s_{end}$ to previous-node array $A_{prev}$
        if $n_{batch} == 1$ then
            Create list of discretized states at $B_{L+1}$, bounded using $A_{prev}$
        end if
        Estimate memory needed for solving all branches from $A_{prev}$
        if Sufficient RAM or $s_{end} - s_{start} = 1$ then
            Allocate input array $A_{input}$, output array $A_{output}$ and helper arrays
        else
            $s_{end} = s_{end} \times \{\text{memory reduction factor}\}$
        end if
    end while
    Generate input set list and store in $A_{input}$
      • Zero-dwell time cases for flybys
      • $n\pi$ dwell cases for $n\pi$ sequences (if included in search)
      • Discretized dwell cases for rendezvous (if included in search)
    Solve BVPs in $A_{input}$ using desired methods, storing solutions in $A_{output}$
      • Ballistic Lambert solver
      • VILT BVP solver
      • General single $\Delta V$ transfer optimizer
      • LT BVP shape-based solver
    if $L > 1$ OR $\mathbf{v}_\infty^I$ for $B_1$ provided by user then
        Split solutions into solver bin sets
          • No $v_\infty$ discontinuity or rendezvous: solutions valid
          • Other solutions binned by zero-crossing pairs of like type
        Pre-prune solutions by feasibility before $v_\infty$ continuity root solve (heuristic)
        Root-solve to enforce $v_\infty$ matching at $B_L$ on valid solution pairs
        Prune by feasibility of leg (transfer) data for valid solutions
        Apply $n\pi$ sequence solver to $n\pi$ dwell solutions (enforce patching at node $B_L$)
        Prune by feasibility of patching node $B_L$: e.g. flyby $h$, rendezvous $\Delta V$, etc.
    else
        Prune by feasibility of leg (transfer): e.g. VILT $\Delta V$, LT thrust, etc.
    end if
    Calculate performance indices of remaining valid solutions
    Rank and prune remaining valid solutions by Pareto optimality
    Prune to specified $n_{MAX}$ solutions by one performance index (absolute filtering)
    Save remaining valid solutions to file for node $B_{L+1}$ (Append if file already exists)
    Deallocate $A_{input}$, $A_{output}$, and helper arrays
    $s_{done} = s_{end}$; $s_{end} = (s_{end} - s_{start}) \times \{\text{memory increase factor}\}$; $s_{start} = s_{done} + 1$
end while
Output: $n_{batch}$ to calling Algorithm 2.1
```

In the interest of speed, all data which are currently being manipulated are stored in random access memory (RAM). This reduces latency associated with reading from or writing to the hard drive, which is only accessed at the beginning and end of each leg calculation, and then using unformatted binary files for improved read and write speed. However, the RAM available is generally much more limited than hard drive space. To circumvent this limitation, *Explore* is built to be flexible with how much RAM it uses, subdividing the solution space to fit part of it in RAM as explained in Algorithm 2.2. The algorithm dynamically allocates the amount of RAM it needs or the amount available (or set by the user), whichever is smaller. When the entire leg of the trajectory being calculated cannot fit into RAM, the leg is split into batches by subsets of start time on the leg (that is, the preceding node). None of the search space is lost in this batch processing, but underestimating the RAM requirements for a memory batch can lead to solutions being discarded from lack of storage for them. Memory can be allocated to the maximum possible number solutions to avoid the risk of discarding solutions, but this strategy leads to the allocation of much more memory than needed in most practical problems.

Alternatively, the solution subdivision into batches can be used to allocate the different batches to different CPUs for parallel computation. Although there would still be a sequential bottleneck in that each leg would need to be fully evaluated before progressing to the next leg, this method for potentially speeding up computation is worth further study.

This subdivision also keeps hard drive writes low because the batch can be evaluated entirely to completion and written to the hard drive before the next batch of the same leg is set up to be solved. The batches are therefore treated as independent except for performance index pruning: Performance indices are calculated in each batch and performance index pruning is performed within each batch. If more than one memory batch was necessary, performance index pruning is done again once all the solutions for the leg are evaluated. This two-stage pruning can be done without any loss of solutions from the batching process because the Pareto rank of a solution can only increase when it is compared against solutions of other batches. Similarly, the ordinal ranking of a solution for absolute filtering can only increase once the solution is compared against all batches.

2.7.3 Memory management: solution representation and storage

One way memory requirements are reduced is by storing less data for each solution. At the end of the search the solutions are recalculated from this reduced set of data as shown in Algorithm 2.1, but since the number of final solutions is relatively small compared to the investigated candidate solutions, this process does not add significantly to the run time of a search.

Solution data for each candidate trajectory are stored in “derived-type” vectors. What data are stored within a derived type is carefully chosen so that the search process runs smoothly and solutions are reproducible, but beyond that aim, extra data are not stored but rather recalculated at the end of the search process and printed for output. For example, the single ΔV vector of an optimized DSM is stored because it is calculated with a stochastic MS but the maximum thrust along a LT arc is not stored in memory but is recalculated and printed in the output files. One major exception is made in storing body states – although these can be repeatably reproduced from the time alone, the full body state is stored to aid in run time. Care has also been taken in the number of bytes used for logicals and integers so as to reduce the memory footprint for storing each solution.

In terms of storage, each trajectory is a linked list of these derived-type memory structures as shown below. Bracketed terms are each one level of the list and the j's in parentheses are the reference number that the next element uses to link to the previous one:

$$\{N_1\}(j_1) \leftarrow \{L_1 \rightarrow N_2\}(j_2) \leftarrow \cdots \leftarrow \{L_{n-2} \rightarrow N_{n-1}\}(j_{n-1}) \leftarrow \{\{L_{n-1} \rightarrow N_n\}\}(j_n)$$

Only the list element at the end, here the n^{th} node referenced by j_n, is stored in RAM. In practice this has to be a special list element and is implemented in *Explore* as a three-level reverse tree structure to reduce duplication of data in RAM. Algorithm 2.2 indicates when each of these three arrays is created and filled:

$$A_{prev} \leftarrow A_{input} \leftarrow A_{output}$$

This structure is a reverse tree because, as the arrows show, the leaves know from which branch they originate, but the branches do not have access to the leaves which stem from them.

All other elements of the trajectory linked list are identical, single-level structures with data pertaining to their node and preceding leg, where applicable. This method of storage also provides implicit backwards pruning; if a particular trajectory does not continue in the search, its past elements are still stored on the hard drive but there are no longer any list elements pointing to them from the current node. As a result, when the trajectories are read for output, the incomplete trajectories will not be included because they no longer have a pointer to them.

Chapter 3

V-Infinity Leveraging Transfers and Velocity-Aligned Apsidal Maneuvers

The boundary value v_∞ leveraging transfer formulation is presented in this chapter[1]. First, VILTs are described in general, including types of transfers, basic assumptions, and input parameters used for their definition. This general discussion is followed by presentation of the specific VILT formulation under consideration, including an introduction to the method, the derivation of the time-of-flight (TOF) equation, and discussion of determining outputs, i.e. the solve-for parameters of the transfer solution. Next the function behavior of TOF is investigated, including the bounds, observed function shapes, and possible number of solutions. An algorithm for the presented VILT method is given, along with various notes on implementation.

Several different kinds of examples follow. First, four examples are investigated for tangent VILTs to demonstrate some of the solution families.

[1] This chapter contains information also published in:
- Demyan Lantukh, Ryan P. Russell, and Stefano Campagnola. “The V-Infinity Leveraging Boundary Value Problem and Application in Spacecraft Trajectory Design,” Journal of Spacecraft and Rockets, Volume 52, Issue 3, pp 697–710, 2015. DOI: 10.2514/1.A32918
- Demyan Lantukh, Ryan P. Russell, and Stefano Campagnola. “Automated Inclusion of v-infinity Leveraging Maneuvers in Gravity-Assist Flyby Tour Design,” AIAA/AAS Astrodynamics Specialist Conference, 13-16 August 2012, Minneapolis, MN

Ryan P. Russell provided initial problem formulation and implementation as well as definition of function bounds. Stefano Campagnola contributed to initial formulation and independent verification.

Next, two example sequences to Jupiter are presented, including the use of an inter-body VILT (IBVILT). Two flyby-aided capture examples at Jupiter are presented, one which requires nontangent VILTs to solve and a second which uses the VILT method to place the Jupiter Orbit Insertion (JOI) maneuver. The last example demonstrates the use of the VILT method as an initial guess for LT transfers as part of a solution to the GTOC6 problem.

3.1 v_∞ leveraging in a trajectory search

In the context of pathsolving or a broad trajectory search, formulating a v_∞ leveraging transfer (VILT) as a boundary value problem provides some distinct advantages. The resulting transfer is broader than a VILT, enabling the calculation of a velocity-aligned apsidal maneuver (VAM) which can occur in instances not generally classified as VILTs – instances such as inter-body transfers and orbit insertion maneuvers.

Since the form of the problem parallels the ballistic Lambert problem, inclusion in Lambert-based pathsolving algorithms is straightforward. The boundary value formulation adds a single continuous degree of freedom (DOF) and a few binary and countable DOFs, helping reduce the dimensional increases normally associated with maneuver inclusion. A method for determining VAMs (and the subset of VILTs) is developed and demonstrated in the remainder of the current chapter.

3.2 Models and assumptions for v_∞ leveraging

VILTs employ a specialized class of maneuver whose purpose is to efficiently change v_∞ at a flyby body, providing advantages for gravity assist

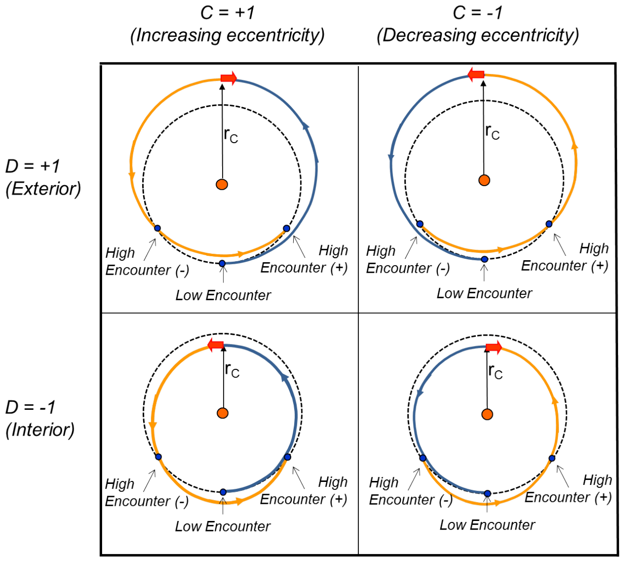


Figure 3.1: The different types of VILTs

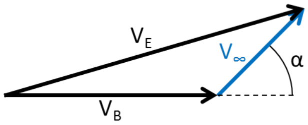


Figure 3.2: Pump angle (α)

or orbit insertion. The v_∞ leveraging maneuver (VILM) is performed near an apse to accomplish this favorable trade-off, usually as part of a VILT sequence. Figure 3.1 shows four types of VILTs based on the maneuver location and direction as described below. If the minor body has a circular orbit, then the VILM increases v_∞ when the maneuver increases spacecraft orbit eccentricity and decreases v_∞ when the maneuver decreases spacecraft orbit eccentricity [116]. The traditional advantage of VILTs is their ability to change v_∞ with a relatively small maneuver at or near the leveraging apse, an apse opposite the flyby: the change in v_∞ can be an order of magnitude greater than the ΔV [117, 18].

3.2.1 Types of VILTs

There are several different ways to describe or characterize VILTs: generally, one continuous parameter, three integers, and three binary descriptors are sufficient. For example, consider an interior, tangent $d2{:}3_1^+$ VILT, which fully describes the transfer. Here *tangent* is a special case of the continuous parameter, which is zero by the definition used in the current thesis, as given below. The three integers relate the VILT to orbit resonances, where an $N{:}M_K$ VILT performs approximately N revolutions in the time it takes the flyby body to perform approximately M revolutions, with $0 \leq K < N$ indicating the number of complete spacecraft orbit revolutions before the VILM. The example $2{:}3_1$ VILT therefore has the spacecraft completing approximately two revolutions in the same time that the flyby body completes approximately three revolutions, with the maneuver happening on the second spacecraft revolution.

The first binary descriptor for VILTs indicates the location of the leveraging apse: when the VILM is at spacecraft apoapse (periapse), the VILT is termed *exterior* (*interior*). Whether a VILT is exterior or interior specifies the domain (D) of the leveraging apse, were $D = 1$ and $D = -1$ for exterior and interior VILTs, respectively. This convention assumes a circular minor body orbit, an assumption this study does not enforce, but the terminology is maintained for simplicity and consistency with literature. The second binary descriptor, represented by the d in $d2{:}3_1^+$, deals with the direction of the transfer. One convenient way to describe direction is its effect on the spacecraft eccentricity, with one encounter designated the *Low Encounter* and the other the *High Encounter* according to eccentricity as shown in Fig. 3.1. The direction, or change (C), of the VILT is then either *increasing* ($C = 1$) or *decreasing* ($C = -1$) spacecraft eccentricity (and v_∞ for the case of circular body orbit or tangent VILTs at one body). The third descriptor, represented by the $+$ in the superscript of $d2{:}3_1^+$, indicates the location of the high encounter flyby of the VILT with respect to the non-leveraging (vacant) apse, called *Minus* or *Plus*, as illustrated in Fig. 3.1. This description of relative geometry extends to multiple-revolution VILTs as well. However, since the spacecraft position at the beginning and end of the VILT are inputs to the method developed here, whether the VILT is plus or minus is implicitly specified in the inputs. The four resulting combinations of VILTs are illustrated in Fig. 3.1.

The continuous parameter is the one remaining DOF once all of the assumptions are applied for a VILT. Because historically VILTs have been classified into two subsets by this free parameter, it is convenient to define a fourth binary descriptor that a VILT can be either tangent or non-tangent. In this study we take the liberty of defining a *tangent* VILT as a VILT where the low encounter occurs at a spacecraft apse, and any VILT that does not satisfy this condition is *non-tangent*. In the case of a circular flyby body orbit, the low encounter of a tangent VILT has a v_∞ tangent to the minor body velocity, providing consistency with the traditional definition of a tangent VILT in literature. If a VILT is non-tangent then the value of the continuous parameter must be specified to fully define the VILT, with a definition of this free input parameter used in the current study given below. Tangent VILTs have the advantage of being easier to compute and having a simpler design space because they have one fewer free parameter, but they can be limiting in certain problems, such as those which use flyby bodies with small masses [117, 18]. The nomenclature described in the current work seeks to retain consistency with the literature while also being extensible to non-circular body orbits and general VAMs

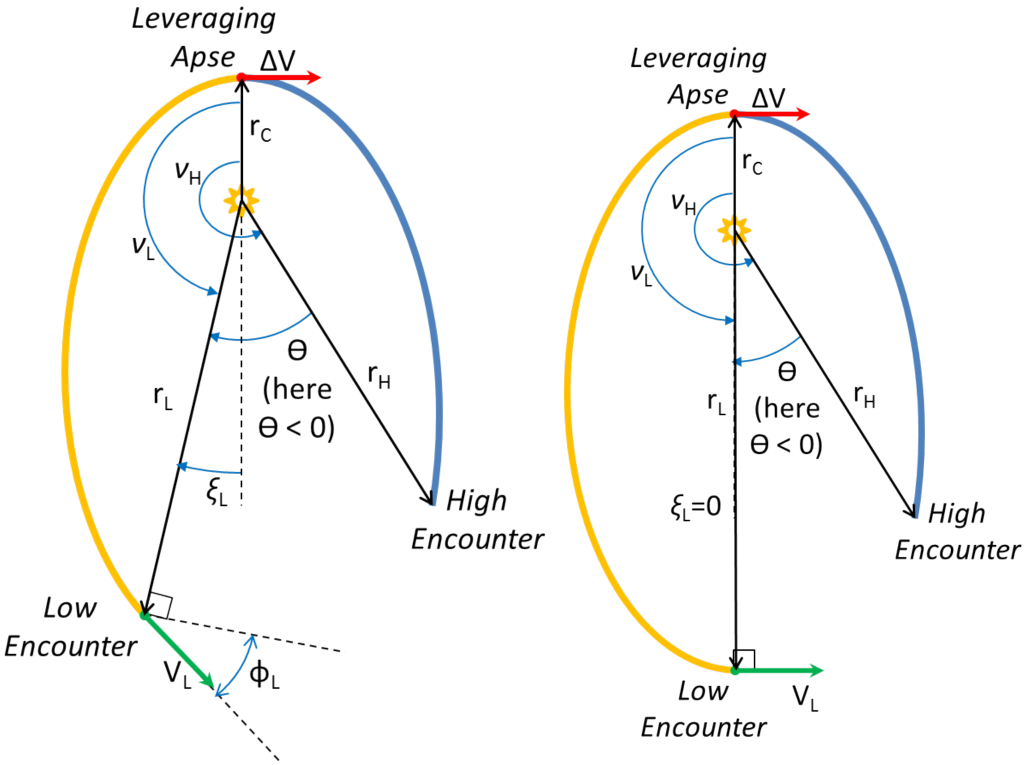
3.2.2 Assumptions

Many previous studies and the current work assume a velocity-aligned maneuver at the spacecraft orbit apse, a good approximation of the true optimal solution [112, 117]. This assumption provides a way to patch the two ballistic segments of the VILT with only one free dimension parameter, where a positive ΔV aligns with the local spacecraft velocity and a negative ΔV aligns directly opposite the local spacecraft velocity.

Many previous methods rely on two assumptions that are abandoned here: First, there are no restrictions on the body orbit or even that the spacecraft return to the same body, whereas other methods often require both flybys to be at the same body with a circular orbit. Also, there is no need to know v_∞ at either encounter. As a result, the inputs match those of the traditional Lambert boundary value problem – two position vectors and TOF between them – along with some additional integer parameters to take care of the VILM type and a single remaining free parameter that disappears in the case of tangent VILTs.

3.2.3 Free Input Parameter for Non-tangent VILTs

The remaining free parameter for non-tangent VILTs is the result of an under-constrained general problem formulation. The most natural free parameters would relate to one of the two encounters. The angle between the low encounter and the vacant apse is used because of its simple relationship to true anomaly and straightforward implementation. Choosing v_∞ as a free parameter is avoided because there is no clear a priori relationship between efficiency of the VILT and v_∞, but using v_∞ may have some advantages in the context of the trajectory search. Another natural choice would be spacecraft



(a)

(b)

Figure 3.3: Nontangent (a) and tangent (b) VILTs are calculated by determining the t_E of the two constituent ballistic arcs of each VILT and solving for the r_C which provides the desired total TOF. These examples are decreasing ($C = -1$), interior ($D = -1$) VILTs.

Φ, which has the advantage of being easily related to the body Φ. This latter choice provides a different method to extend the concept of tangent and non-tangent VILTs to non-circular body orbits since having a pump angle of 0 or π radians maximizes or minimizes, respectively, the central body velocity resulting from a flyby. The definition for pump angle is illustrated in Fig. 3.2. However, relating to the true anomaly results in a significantly simpler and better-behaved function that is easier to implement than the Φ method.

Therefore, a modification of the true anomaly is used: the angle to the vacant apse (the non-leveraging apse), as shown in Fig. 3.3. This angle has the advantage of being easy to relate to the true anomaly and leads to a relatively simple function space, as shown in the Function Behavior Investigation section below. A value of zero for this angle at the low encounter ($\xi_L = 0$) results in a tangent VILT. If the body orbit is circular or the encounter occurs at the body orbit apse, then the spacecraft Φ is also 0 or π radians when $\xi_L = 0$.

3.3 Boundary-Value VILT Formulation the Parallels the Lambert Problem

The VILT is composed of two coplanar ballistic arcs connecting the encounters and the VILM, as shown in Fig. 3.3. The initial and final positions and the TOF are given, together with the free parameter (ξ_L). For a given VILT type, these parameters are sufficient to compute the initial and final velocities and the VILM. To ease analysis, the leveraging apse radius r_C is introduced as the solve-for parameter and determined using a phasing constraint equation, as explained in the next section.

Once in the plane of the transfer, the VILT formulation is defined entirely in terms of scalar parameters and gives the TOF by splitting the VILT into its two ballistic arcs and calculating the time of flight of each of those arcs (t_E). Input position vectors are transformed into the plane of the VILT where the transfer lies in the plane of the vectors, except when θ is a multiple of π radians. In such cases, the transfer plane is undefined, as with the Lambert algorithm. These resonant and odd-$n\pi$ return cases are not explored in this study. The outputs of the VILT calculation can similarly be transformed into vectors in the original reference frame.

3.3.1 Time-of-Flight Equation

The time of flight is separated into the time for each ballistic arc as shown in Fig. 3.3 with the maneuver assumed to be instantaneous. This section develops the time of flight for each ballistic arc as a function of leveraging apse distance r_C as well as the boundary conditions and VILT type. The subscripts for the low and high arcs (L and H respectively) are replaced by the subscript E where the equations are to be applied to both arcs or encounters separately. Using D and taking advantage of the knowledge that one end of the arc must be the spacecraft periapse or apoapse, the transfer parameters can be defined by simultaneously solving Eqs. (3.1) and (3.2) to get Eqs. (3.4) and (3.5). All Eqs. (3.1) through (3.15), except Eqs. (3.3) through (3.5), are common in astrodynamics texts or simple combinations thereof [4].

$$r = \frac{a\,(1-e^2)}{1+e\cos\nu} \tag{3.1}$$

$$r_C = a(1+D\,e) \tag{3.2}$$

$$\nu_E = \xi_E + \frac{\pi}{2}(1-D) \tag{3.3}$$

$$e_E = \frac{r_C - r_E}{r_E\cos|\nu_E| + D\,r_C} \tag{3.4}$$

$$a_E = \frac{r_C\,(r_E\cos|\nu_E| + D\,r_C)}{r_E\cos|\nu_E| + D\;(2r_C - r_E)} \tag{3.5}$$

Provided the arc semi-major axis and eccentricity, the spacecraft eccentric anomaly at the encounter is determined, and the Kepler equation defines time since periapse. Valid true anomaly values can be either positive or negative, but the absolute value is used so that the correct eccentric anomaly is calculated in Eq. (3.6), with the sign of the true anomaly accounted for by the geometric parameter O in Eq. (3.10), which is defined below. The absolute value operation on ν_E in Eqs. (3.4) and (3.5) is not necessary because it does not affect the cosine but the absolute value is maintained throughout for consistency.

$$E_E = 2\tan^{-1}\left[\tan\left(\frac{|\nu_E|}{2}\right)\sqrt{\frac{1-e_E}{1+e_E}}\right] \tag{3.6}$$

$$n_E^* = \sqrt{\frac{\mu}{a_E^3}} \tag{3.7}$$

$$T_E = \frac{2\pi}{n_E^*} \tag{3.8}$$

$$t_{PE} = \frac{E_E - e_E \sin E_E}{n_E^*} \tag{3.9}$$

The number of full orbit revolutions in a ballistic arc is K_E. N is the total approximate revolutions of the VILT so $N = K_L + K_H$, but the number of actual complete spacecraft revolutions may be either N or $N+1$ depending on the values of O_L and O_H. For $O = 1$, the spacecraft passes through at least $K + 1/2$ revolutions and for $O = -1$ the spacecraft passes through fewer than $K + 1/2$ but more than K full revolutions in the arc. This information is used to determine t_E for each arc and these times are combined into TOF.

$$t_E = t_{PE}\, D\, O_E + T_E\left[K_E + \frac{1}{2} - \frac{O_E}{4}(D-1)\right] \tag{3.10}$$

$$TOF = t_L + t_H \tag{3.11}$$

Because the true anomaly is specified only at the low encounter through the free parameter ξ_L, applying the above method requires determining the true anomaly at the high encounter.

$$\nu_H^* = \nu_L + \theta \tag{3.12}$$

The parameter θ is the angle from the initial spacecraft position vector before the VILT to the final spacecraft position vector after the VILT, as shown in

Fig. 3.3, and is readily determined from the vector inputs. This equation comes from the fact that the two transfers of the VILT share the leveraging apse, which is also called the common apse. Here ν_H^* may not necessarily be in the correct domain and ν_H should be set to the equivalent of ν_H^* in the domain $[-\pi, \pi]$, which is also the domain of ν_L.

3.3.2 Determining Outputs

Equation (3.11) provides a relation of TOF as function of (r_C, ξ_L) and other known parameters. For a choice of the free parameter ξ_L, Eq. (3.11) can be numerically solved for r_C, as explained in the Implementation section. Once r_C is known, the other dependent variables are computed with Eqs. (3.4)-(3.12). In addition, the velocity at both encounters and right before and after the maneuver is computed using the energy equation, Eq. (3.13).

$$V = \sqrt{\mu \frac{2a - r}{a\,r}} \tag{3.13}$$

Subscripts are left off of Eqs. (3.13) through (3.15) because they are general in nature and can be applied to any transfer. Maneuver size and direction is calculated from the velocity magnitudes on the low and high end of r_C since the direction C is also given. With this information and flight path angle of the spacecraft at both encounters as given in Eq. (3.15), the outputs can be converted to vectors in the original reference frame.

$$p = a\,(1 - e^2) \tag{3.14}$$

$$\Phi = \cos^{-1} \frac{\sqrt{p\,\mu}}{r\,V} \tag{3.15}$$

3.4 TOF Function Behavior Investigation

The boundary value VILT is calculated by setting up a one-dimensional root-solve based on Eq. (3.11) and solving $G = TOF - TOF^* = 0$, given with explicit dependencies in Eq. (3.16). This function is investigated to determine the number of possible roots, convexity, singularities, and any other characteristics which would affect implementation.

$$G_{(\mu,D,N,\theta,TOF^*,r_L,\xi_L,K_L,O_L,r_H,O_H)}(r_C) = t_{L\,(\mu,D,r_L,\xi_L,K_L,O_L)}(r_C) + t_{H\,(\mu,D,N,\theta,r_H,O_H)}(r_C) - TOF^* \tag{3.16}$$

Specifically, an investigation is performed on the mapping from r_C to $TOF(r_C)$ from Eq. (3.16), with all other inputs systematically varied as constants for any one case. Both r_C and $TOF(r_C)$ are physical quantities that are only considered valid when they are real, positive, and finite.

3.4.1 Bounds

The function domain can be divided into two sub-domains: exterior and interior VILTs. When calculating an interior VILT, $0 < r_C \leq r_E$ and for an exterior VILT, $r_E \leq r_C < \infty$, although a more reasonable but still large upper bound is used in implementation. If the body orbit is circular, then $r_H = r_L$ and the separation of the two sub-domains is clear. Otherwise, when r_C falls between r_L and r_H the VILT is not well defined even though a VAM can still exist. In this case, one of the primary purposes of the maneuver is to change the spacecraft energy, but gravity assists are much more efficient for this task; it is unlikely an optimal trajectory would employ such a maneuver except for targeting or as a trajectory correction. For this reason, the solutions where $r_L < r_C < r_H$ or $r_L > r_C > r_H$ have been left out of the current study of VILTs, where it effectively becomes a maneuver exclusion zone as shown

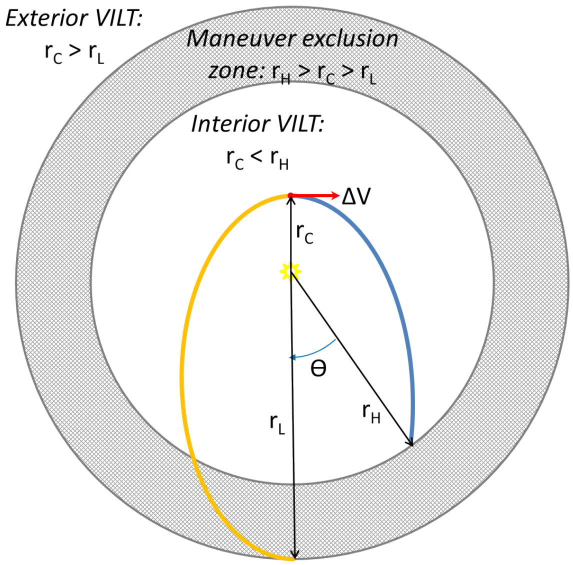


Figure 3.4: The annulus in which the leveraging apse is excluded because optimal trajectories are not expected to have maneuvers within it, shown for the case described in Fig. 3.3.

in Fig. 3.4. This exclusion zone can be quite large for highly eccentric bodies and for inter-body transfers, but maneuvers in this region can be included by accounting for the fact that one ballistic arc is part of an interior VILT while the other is part of an exterior VILT.

For interior VILTs, Eq. (3.11) can have a singularity and an associated region of the function domain which is invalid. The function becomes invalid when $r_C \leq r_{C,LB}$ where $r_{C,LB}$ is the maximum of the singularity function given in Eq. (3.18) evaluated at the low and high encounters.

$$r_{C,LB} = \max(r_{C,singular}|_L, r_{C,singular}|_H) \tag{3.17}$$

$$r_{C,singular} = \frac{r_E}{2}(1 - D \cos|\nu_E|) \tag{3.18}$$

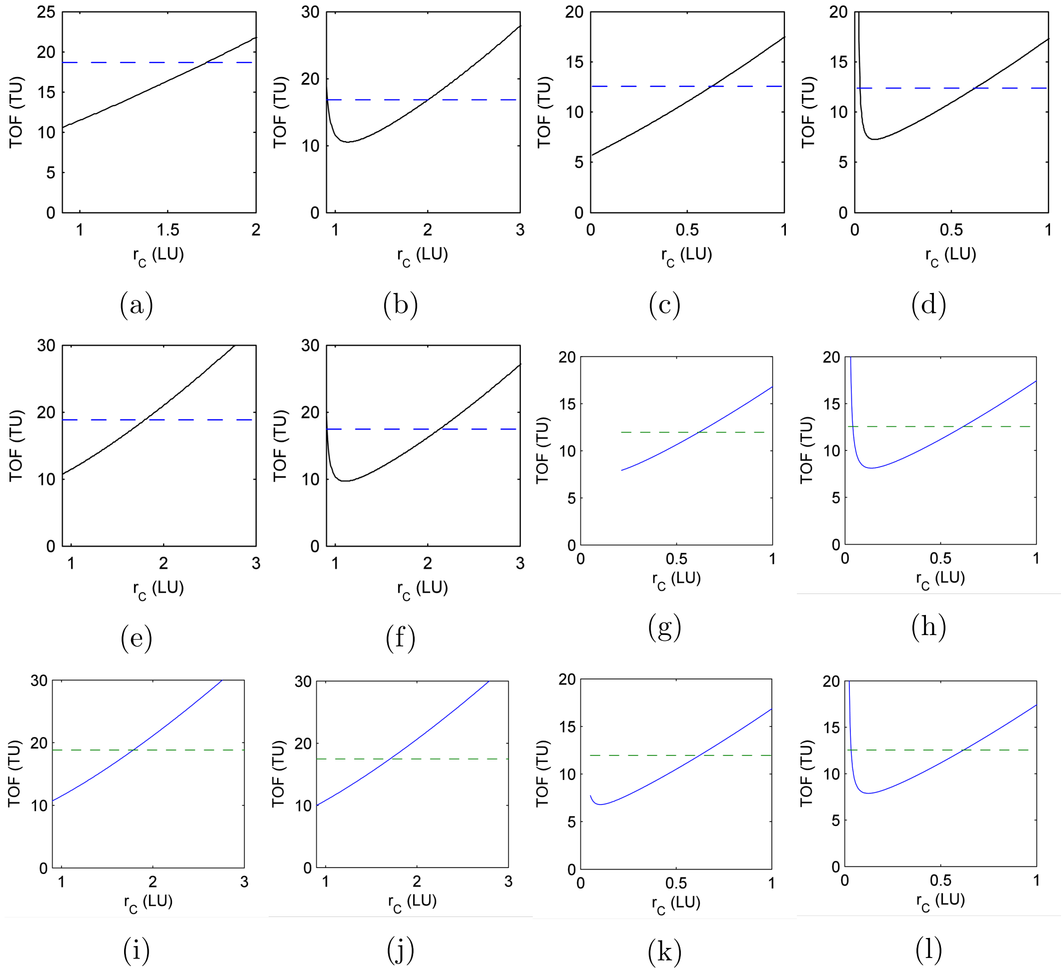
For practical implementation, the bounds on which the function is evaluated are offset in the direction of the valid domain by a small parameter.

3.4.2 Observed Function Behaviors

The behavior for the function $TOF(r_C)$ is investigated by systematically varying the other parameters and evaluating TOF as the dependent variable and r_C as the independent variable. This TOF is essentially the function that the root solver evaluates, with an offset by the target TOF so that the zero corresponds to the target. All investigations and the conclusions from them are based on numerical studies.

Figure 3.5 shows the basic function shapes observed for $TOF(r_C)$, with the horizontal dashed line indicating a target TOF. Table 3.1 provides the inputs used to generate Figure 3.5. Distances and times are given in normalized length and time units. Figures 3.5a and 3.5b show exterior VILTs with one solution and two solutions for the target TOF, respectively. Figures 3.5c and 3.5d show interior VILTs with one and two solutions respectively. Figure 3.5 shows cases where $\xi_L = 0$ (Figs. 3.5a-3.5d) and also shows that the same type of function shapes result in cases where $\xi_L \neq 0$ (Figs. 3.5e-3.5l). Among the nontangent ($\xi_L \neq 0$) cases, Figs. 3.5i-3.5l repeat Figs. 3.5e-3.5h except with the opposite sign of ξ_L. VILTs have been observed to have one of two distinct function behaviors: either TOF increases monotonically with increasing r_C and there is at most one solution, or TOF has one minimum and there are one or two solutions (no cases with more than two solutions were encountered). Cases can be found where the second derivative $\mathrm{d}^2(t_E)/\mathrm{d}(r_C)^2 < 0$, but no complete VILT input set evaluated was found to have more than two possible solutions.

Whether the solution space is monotonic or not is a function of the



(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j) (k) (l)

Figure 3.5: Examples of the different function shapes for different VILTs, with r_C as the independent variable. $\xi_L = 0$ for parts a-d and $\xi_L \neq 0$ for parts e-l.

Table 3.1: Inputs for example $TOF(r_C)$ functions ($\mu = 1$, normalized units, angles in radians)‘

Figure	D	C	N	K	S	r_L	r_H	θ	TOF	ξ_L
a	1	1	2	1	1	0.9000	0.9018	6.0751	18.680	0.0000
b	1	1	2	1	2	0.9000	1.0502	4.1016	16.839	0.0000
c	-1	1	3	2	1	0.9000	0.9000	6.2600	12.548	0.0000
d	-1	1	3	1	1	0.9000	0.9026	6.0290	12.359	0.0000
e	1	1	2	1	1	0.9000	0.9002	6.2446	18.818	-0.7854
f	1	1	2	1	2	0.9000	0.9924	4.6883	17.455	-0.7854
g	-1	1	3	3	1	0.9000	0.9217	5.5466	11.597	-0.2618
h	-1	1	3	1	1	0.9000	0.9000	6.2677	12.554	-0.2618
i	1	1	2	1	1	0.9000	0.9002	6.2446	18.818	0.7854
j	1	1	2	1	1	0.9000	0.9924	4.6883	17.455	0.7854
k	-1	1	3	3	1	0.9000	0.9217	5.5466	11.597	0.2618
l	-1	1	3	1	1	0.9000	0.9000	6.2677	12.554	0.2618

other inputs and is determined by evaluating the derivative $\mathrm{d}(TOF)/\mathrm{d}(r_C)$ at the bounds of the solution space. For the monotonic cases evaluating TOF at the lower bound allows the existence of a solution to be verified. In the other cases (similar to the multi-revolution Lambert problem function space) the minimum of $TOF(r_C)$ must be found to determine whether or not a solution exists for a desired TOF, and the solution number (S) is specified to determine whether the lower or higher value of r_C is desired. In order to distinguish the two solutions, first the minimum of $TOF(r_C)$ is found: in this case, a one-dimensional root-solve of the first derivative is employed, but a direct, derivative-free minimizer is used instead if that root solve fails to converge in a reasonable number of iterations. A derivative-free approach is occasionally found to be preferable for both finding the minimum and for solving for the desired TOF because the slope of $TOF(r_C)$ can be exceedingly steep as the function approaches the singularity (e.g. the left side of Fig. 3.5d). Such steep changes are known to cause problems for gradient-based root solvers. When the function $TOF(r_C)$ has a minimum in the sub-domain of interest, then cases with zero, one, or two solutions have been observed. The algorithm as implemented below for determining the VILT is designed to account for the possibility of up to two solutions.

3.5 Implementation

The implementation of the VILT solver is designed to be as close as possible to a plug-and-play addition for any outer loop search that relies on the classic Lambert problem as the inner loop such as the algorithm described in the previous chapter. However, a few key differences require special handling in the context of trajectory search software. First, additional inputs are needed to specify the type of VILT which are not needed for a Lambert solution. Also, maneuver magnitude and direction are desirable as outputs in addition to the spacecraft velocities. Taking these differences into account, a basic search algorithm that patches ballistic Lambert legs together can also be used to patch VILTs into a trajectory sequence, as implemented in this dissertation.

The algorithm presented in Algorithm 3.1 is already cast in the plane of the transfer. As mentioned previously, there may be cases where a derivative-free minimizer or root-solver may be preferred. Algorithm 3.2 presents the numerical root-finding of the TOF to find r_C, as called for within Algorithm 3.1.

3.6 Tangent VILT Design Space Exploration

Several examples cases are considered next, beginning with simple single-transfer cases. A coplanar model is used for the minor bodies in the current

Algorithm 3.1 Algorithm used to solve for VILT in a way that enables the use of inputs and outputs that parallel the traditional Lambert boundary value problem.

```
Input: $r_L, r_H, TOF^*, \xi_L, \mu, \theta, N, K, D, C, S$
  Determine $\nu_L$ from Eq. (3.3) and $\nu_H$ from Eq. (3.12) and $O_L$, $O_H$ from geometry using $C$ and $\theta$.
  Bound the domain of interest
        case $(D = 1)$: $r_{C,LB} = \max(r_L, r_H)$, $r_{C,UB} = \mathrm{U}$, where U is sufficiently large
        case $(D = -1)$: Determine $r_{C,LB}$ from Eq. (3.17), $r_{C,UB} = \min(r_L, r_H)$
  Determine the number of solutions $(F)$ and adjust bounds as needed for $S$
        if $\mathrm{d}(TOF)/\mathrm{d}(r_C)|_{r_{C,LB}} \geq 0$ then
            if $TOF|_{r_{C,LB}} > TOF^*$ then $F = 0$, EXIT
            else $F = 1$
            end if
        else if $\mathrm{d}(TOF)/\mathrm{d}(r_C)|_{r_{C,UB}} > 0$ then
            Minimize Eq. (3.11) between $r_{C,LB}$ and $r_{C,UB}$ to find $r_{C,min}$
            if $TOF|_{r_{C,min}} > TOF^*$ then $F = 0$, EXIT
            else if $TOF|_{r_{C,min}} = TOF^*$ then $F = 1$, $r_C = r_{C,min}$ is the solution
            else $F = 2$
                Adjust the bounds to encapsulate the desired solution
                    case $(S = 1)$: $r_{C,UB} = r_{C,min}$
                    case $(S = 2)$: $r_{C,LB} = r_{C,min}$
            end if
        else
            ERROR: this case should not occur unless $F > 2$, which has not been observed, or U is not sufficiently large
        end if
  Pick an initial guess $r_C^{(0)}$ such that $r_{C,LB} < r_C^{(0)} < r_{C,UB}$
  Numerically solve for $r_C$ using Algorithm 3.2
Output: $r_C$, $F$ (with resulting $V_L$, $V_H$, $\Phi_L$, and $\Phi_H$)
```

Algorithm 3.2 Numerically solving for r_C of a VILT with given TOF.

```
Input: $r_C^{(0)}$,$TOF^*$,$r_L$,$r_H$,$\nu_L$,$\nu_H$,$O_L$,$O_H$,$K_L$,$K_H$,$D$,$\mu$
  function $t_E(r_C,r_E,\nu_E,O_E,K_E,D,\mu)$
      Sequentially apply Equations (3.4)-(3.10)
  end function
  $i = 0$
  repeat
      Calculate $G$ for the iteration $i$ from Eqn. (3.16) using
          $t_L = t_E(r_C^{(i)},r_L,\nu_L,O_L,K_L,D,\mu)$
          $t_H = t_E(r_C^{(i)},r_H,\nu_H,O_H,K_H,D,\mu)$
      $i \leftarrow i + 1$
      Update $r_C^{(i)}$ using a suitable root-finding technique.
          Method for the current study: Newton-Raphson with trust region
                bounded by $r_{C,LB}$ and $r_{C,UB}$ determined in Algorithm 3.1
          $\mathrm{d}(t_E)/\mathrm{d}(r_C)$ determined by differentiation of Eq. (3.10) using the
                symbolic manipulator software, Maple
  until $G <$ tolerance
  $r_C = r_C^{(i)}$
Output: $r_C$
```

section, but additional examples with ephemeris locations are shown in Appendix D and discussed in Ref. [159]. Following sections will also utilize non-coplanar and ephemeris models. In the cases presented in the current section $\xi_L = 0$, therefore VILTs are tangent.

Simple one-transfer cases are generated by choosing one launch time and solving for many values of TOF from that time, leading to families of VILTs. Because the bodies are defined by their orbital elements, the initial and final body position vectors are entirely determined by the launch time and TOF, making this sweep slightly different from parameter sweeps often used to characterize ballistic solutions where the position vectors are fixed. Five distinct examples are presented, with all bodies assumed in circular orbits except Mercury; orbital elements for the bodies are given in Table 3.2, with two different initial true anomalies used in Mercury examples. The parameters used for the trajectory search for each case are given in Table 3.3. Design space figures for all five presented cases show VILTs as black points and ballistic solutions as gray points. VILT families are labeled by their associated resonance as described previously, but the subscript for the maneuver revolution is left off because it is always zero for these examples. Ballistic families of solutions are labeled by the number of complete spacecraft revolutions and which transfer solution (short-period or long-period) if there are more than zero revolutions. For example, a two-revolution long-period ballistic family is labeled 2_L.

3.6.1 Studying the ΔV-EGA

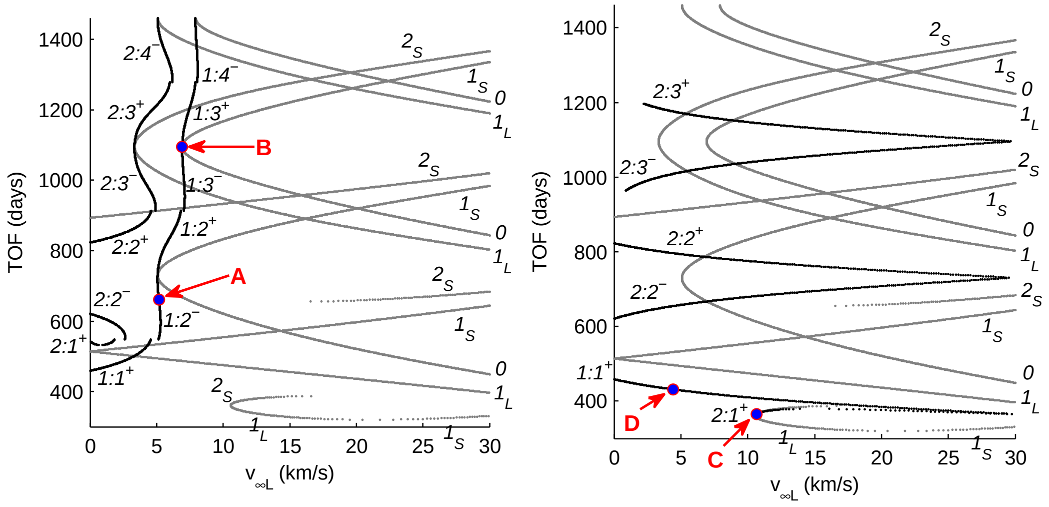
The original VILT proposed – a classic example for study – is the ΔV Earth gravity assist (ΔV-EGA) where Earth is the encounter body at both

Table 3.2: Orbital elements for bodies used by examples of the VILT design space

	Earth	**Mercury**	**Mars**	**Venus**
Semimajor axis (AU)	1.0000	0.38710	1.5237	0.72333
Eccentricity	0	0.20563	0	0
Inclination (rad)	0	0	0	0
Argument of periapsis (rad)	0	0	0	0
Longitude of the Asc. Node (rad)	0	0	0	0
Mean Anomaly (rad)	0	0, $\pi/2$	π	0
The epoch of the orbits is at launch ($t = 0$)				

Table 3.3: Constraints for the five example cases of the VILT design space

	Earth-Earth	Mercury-Mercury (both cases)	Earth-Venus
Max Revolutions	2	2	1
Resolution (points per body rev)	1000	1000	1000
Max TOF (Earth days)	1461	264	674



(a)

(b)

Figure 3.6: *TOF* of exterior (a) and interior (b) ΔV-EGA solutions, as well as ballistic transfers.

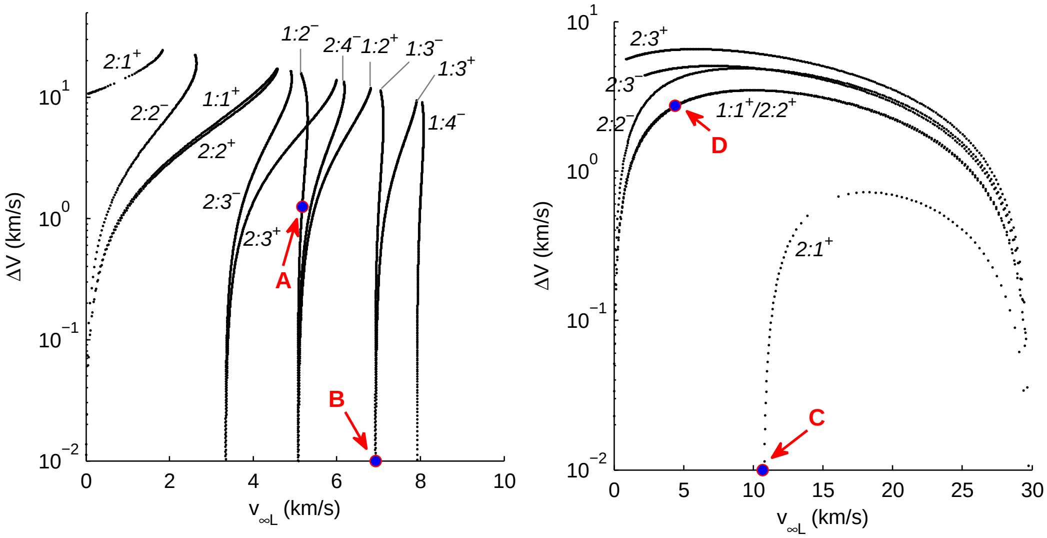
ends of the VILT and the first encounter is often spacecraft launch [111, 113, 114]. Among other interplanetary missions, ΔV-EGAs have been flown by NEAR and Juno [119, 25]. Trajectories may also use a flyby with a different body instead of a maneuver to accomplish the same result as a VILT; for example, the Venus flyby at the start of many trajectories to the outer planets could be replaced with a maneuver, making the first part of the sequence a ΔV-EGA. Figures 3.6 and 3.7 present the ballistic and VILT (ΔV-EGA) solutions for an Earth-Earth transfer. This Earth-Earth case satisfies the constraints in Table 3.3, with Earth's orbit propagated as Keplerian with the orbital elements provided in Table 3.2. Because the sweep is discretized, the resolution is given in sample points per body revolution (ppr) in Table 3.3.

Figures 3.6-3.7 show each ballistic transfer and VILT as a point, where collections of points are families labeled by resonance or by number of revolutions. The points within a family are often closely spaced and appear as a continuous line; in fact, each family would be truly continuous if the problem were not discretized. The VILT solutions center on resonances as shown in Fig. 3.6, with the resonant trajectories occurring between the plus and minus families of each resonance. These resonant VILT solutions also intersect ballistic solution families and so have maneuver sizes which approach zero, as shown in Fig. 3.7. In such cases, the ballistic Lambert solution happens to satisfy the tangent departure constraint that the VILM solver explicitly enforces. The resonant intersections happen for exterior VILT families when $N < M$ as seen in Fig. 3.6a and for interior VILT families when $N > M$ as seen in Fig. 3.6b. Note that N for VILTs is, by convention, defined differently than for ballistic transfers; N gives the number of complete spacecraft revolutions for ballistic transfers but the nearest (associated) resonance for VILTs. As a result, the resonant solutions in the ballistic families occur between the N_S and $(N-1)_L$ families. For the VILT families, associated resonances are determined from knowing N and computing the nearest M from TOF and the body period while also respecting the continuity of the families. The associated resonances are labeled on the figures.

For VILTs at a body in a circular orbit, each increasing VILT has a corresponding decreasing VILT which shares most of the same inputs and outputs. The outputs of the corresponding VILTs share the same values of $v_{\infty L}$ and $v_{\infty H}$, except that $v_{\infty L}$ corresponds to the initial v_∞ for the increasing VILT but the final v_∞ for the decreasing VILT. The only difference in the inputs is $K_{dec} = N - (1 + K_{inc})$. Because of this correspondence in VILT families, all VILTs shown for the ΔV-EGA are increasing VILTs, with the independent variable in Figs. 3.6 and 3.7 being $v_{\infty L}$.

The solution number S for VILTs is analogous to the distinction between short-period and long-period ballistic transfers. The existence of both $S = 1$ and $S = 2$ solutions happens whenever a single VILT family is not monotonic in TOF vs. $v_{\infty L}$ such as the 2:1$^+$ exterior VILT family in Fig. 3.6a and the 2:1$^+$ interior VILT family in Fig. 3.6b. When a ballistic family of solutions with the same spacecraft revolutions is not monotonic in Fig. 3.6, the extremum in TOF is the transition between the long-period and short-period solutions for that revolution number; the VILT S value performs a similar function of distinguishing two solutions of the same input set. The transition between $S = 1$ and $S = 2$ families along a given resonance is also visible in Fig. 3.7 as a gap in that resonance with a lower density of solutions near the gap. The lower density of solutions near the transition results from the discretization of the inputs, and the discretization resolution is chosen so that the families of VILT solutions are still visible in this sparser region. The $S = 1$ solutions occupy the higher $v_{\infty L}$ portion of the family in the exterior VILTs and the lower $v_{\infty L}$ portion of the family in the interior VILTs. In addition, all the VILT families with $N \leq M$ in Figs. 3.6b and 3.7b are $S = 2$ solutions whose corresponding $S = 1$ solutions are degenerate $v_\infty = 0$ cases that shadow the body orbit (not shown in Figs. 3.6 or 3.7). Figure 3.7 also shows that for a given $v_{\infty L}$, if both the plus and minus families exist for a particular resonance, then the plus family will have the lower ΔV.

Trajectories for some of the solutions are shown in Fig. 3.8, with the letter of the figure part used to show where that particular solution falls in the search space figures. For example, Fig. 3.8A is the trajectory pointed out by the annotation A in Figs. 3.6a and 3.7a. Note that the maneuver arrows in Fig. 3.8 are sized for visibility and do not correspond to the maneuver magni-



(a)

(b)

Figure 3.7: ΔV of exterior (a) and interior (b) ΔV-EGA solutions.

tudes. Figures 3.8B and 3.8C give examples of resonant trajectories where the maneuver magnitude approaches zero for an external and an internal VILT respectively. Figs. 3.8A-3.8D shows four examples of ΔV-EGA solutions as traditional tangent VILTs; they are used for comparison in the following sections that investigate VILTs at eccentric bodies and between different bodies.

3.6.2 Mercury-Mercury VILTs: the effect of eccentricity

Repeating the same type of trajectory search with Mercury allows an investigation of VILTs when applied to a body with significant eccentricity. Trajectory search constraints are given in Table 3.3 and the orbital elements for Mercury are given in Table 3.2. The VILT trajectory families are shown in Figs. 3.9 and 3.10 for the initial condition at Mercury $\nu = 0$ and in Figs. 3.11 and 3.12 for the initial condition at Mercury $\nu = \pi/2$. In these figures, part

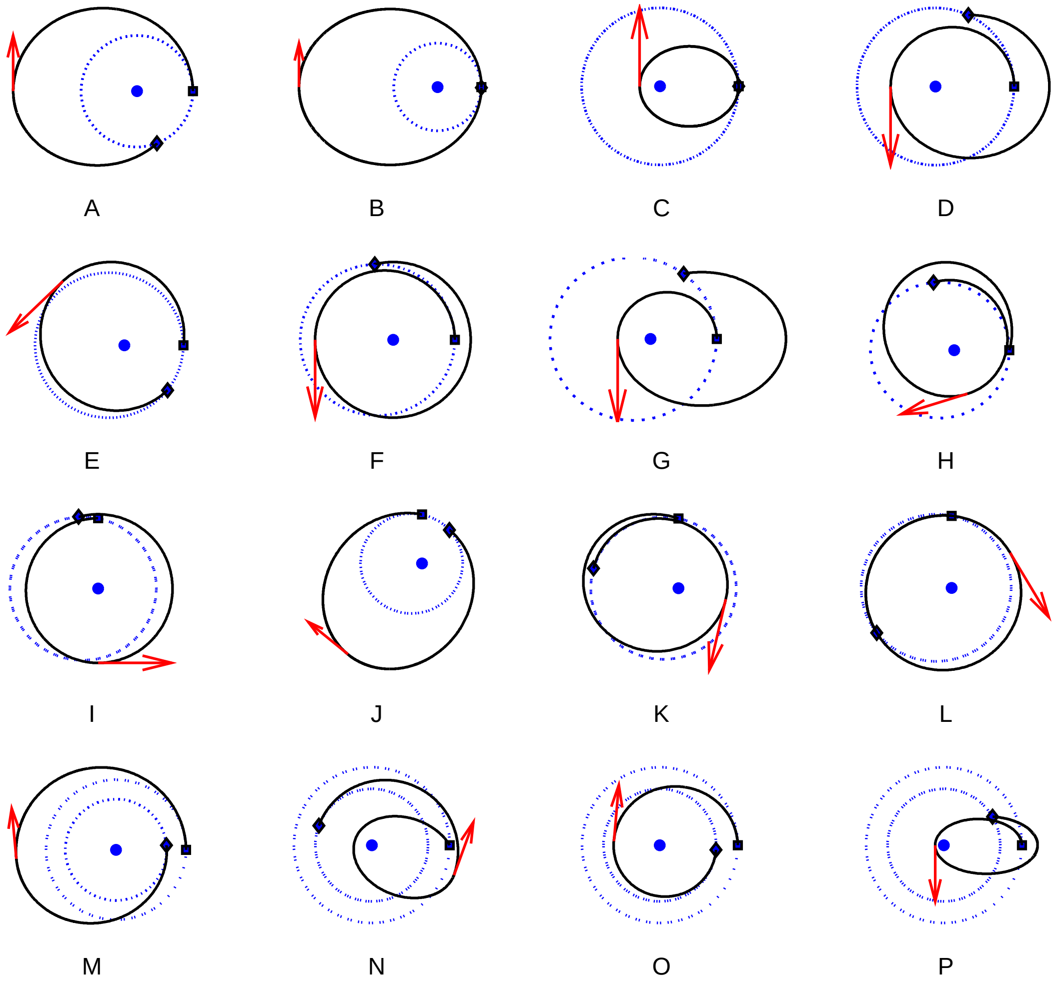
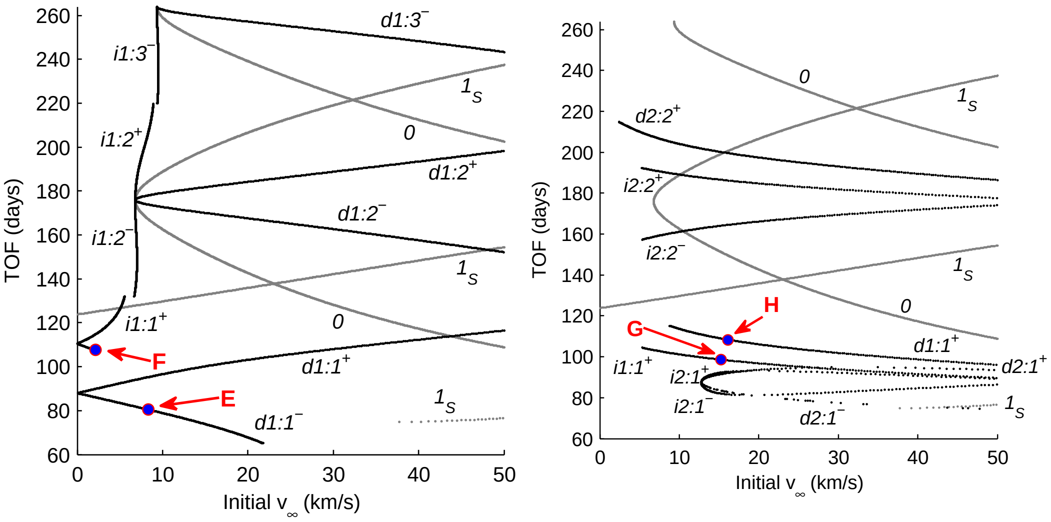


Figure 3.8: Some example VILTs from the Earth-Earth (A-D), Mercury-Mercury (E-L), and Earth-Venus (M-P) cases. The VILTs progress from an initial encounter (square) through a maneuver (arrow) to another encounter (diamond).



(a)

(b)

Figure 3.9: TOF of exterior (a) and interior (b) Mercury-Mercury VILTs and ballistic transfers from launch $\nu = 0$.

a only shows VILTs with one spacecraft revolution (1:M resonances) in order to keep the figures more readable. The other major difference with data visualization as compared to the ΔV-EGA is that the independent variable is initial (launch) v_∞, which does not necessarily correspond to $v_{\infty L}$ as in the ΔV-EGA examples. Since the second encounter of each VILT in the family occurs at a different Mercury true anomaly and Mercury's orbit is not circular, the corresponding families of increasing and decreasing VILTs do not overlap like they do in the circular orbit case. One very visible result of the change in independent variable is the much greater domain of v_∞ in Fig. 3.9 for decreasing VILTs, for which initial v_∞ corresponds to $v_{\infty H}$.

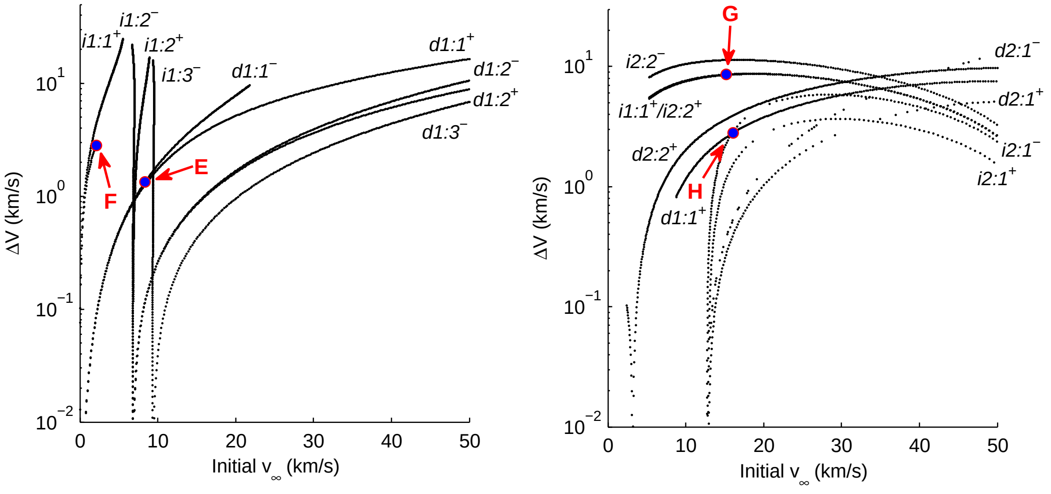
Having an eccentric body orbit allows exterior VILTs for which r_C falls inside the orbit or interior VILTs where r_C falls outside the body orbit. One result of this kind of geometry can be seen by comparing Figs. 3.8D and 3.8F:

both show 1:1$^+$ transfers and are geometrically similar but Fig. 3.8D is an interior VILT at Earth and Fig. 3.8F is an exterior VILT at Mercury. Although the maneuver location in Fig. 3.8F is clearly inside Mercury's orbit, Mercury's eccentricity makes the Sun distance at the maneuver location (r_C) greater than the Sun distance at either of the two Mercury encounters, so the transfer is an exterior VILT. Another effect of body orbit eccentricity is that the minimum launch v_∞ ($v_{\infty L}$ for the increasing VILTs) in a VILT family does not necessarily occur at one end of the family as is the case for the circular body orbit, as seen in Fig. 3.6a, but rather it may occur at a point in the interior of the family. For example, the $i1$:1$^+$ family has a minimum launch $v_\infty \approx 0$ at $TOF \approx 110$ days, corresponding to the case where the VILT shadows the body orbit, but this point is neither the minimum nor maximum TOF possible for the $i1$:1$^+$ VILT family.

Since the initial condition is not at apoapsis of the body orbit there are possible values of r_C inside the body orbit which are still exterior VILTs like Fig. 3.8F. Reaching such orbits requires a deviation from the body orbit and so a nonzero $v_{\infty L}$ and ΔV are required, as shown in Fig. 3.10a. Similarly, eccentricity of the body orbit and the location of the initial condition at body periapsis also prevent the interior VILT families in Figs. 3.9b and 3.10b from extending to very low initial v_∞.

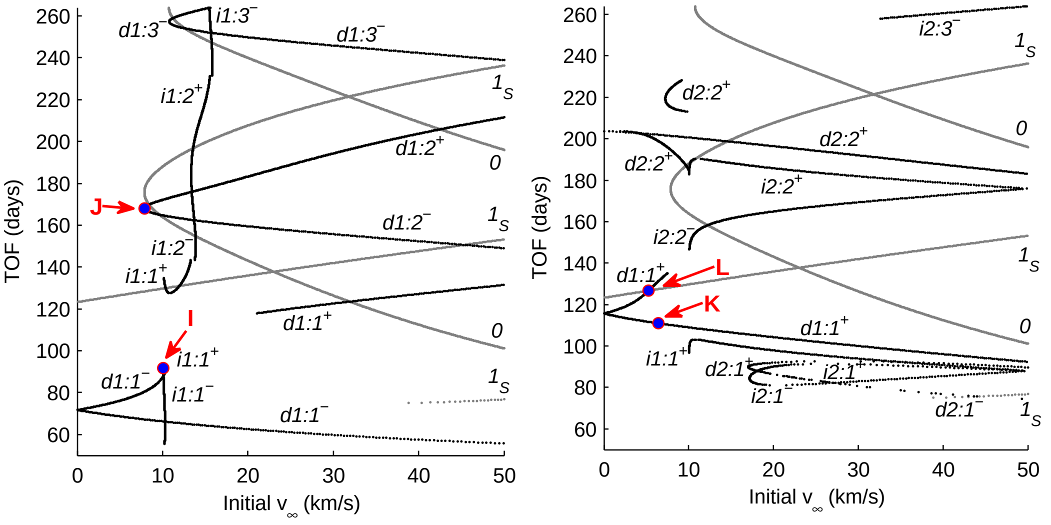
An eccentric body orbit also allows there to be VILTs which approach nonresonant ballistic transfers, as shown in Fig. 3.10b by the dip to zero ΔV along the $d2$:2$^+$ family. There are additional families of solutions where the entire family shadows the body and has $\Delta V = 0$ and $v_\infty = 0$ but these degenerate solutions are left out of the presented results.

Figures 3.11 and 3.12 demonstrate how changing the initial condition



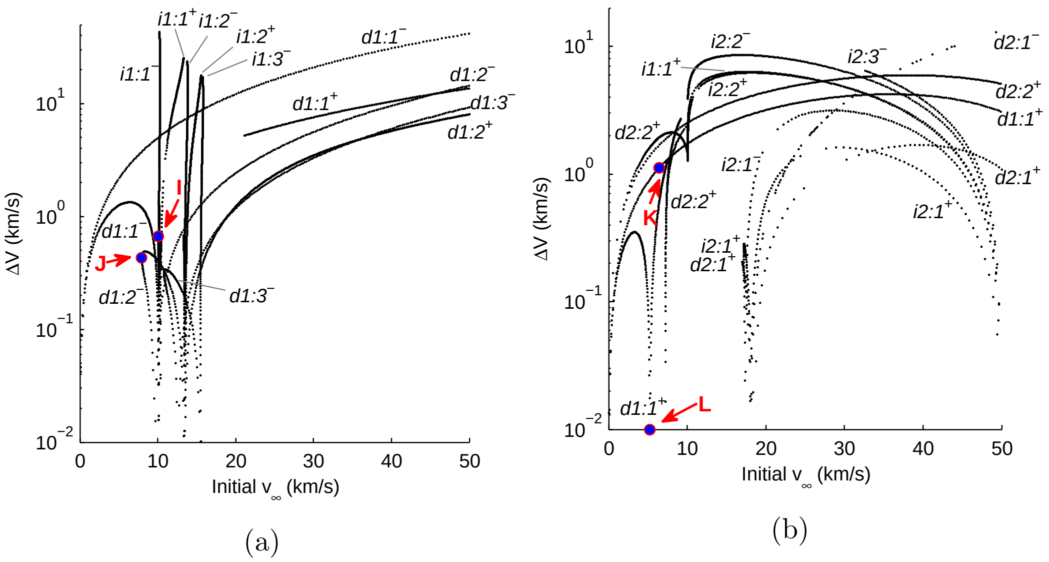
(a) (b)

Figure 3.10: ΔV of exterior (a) and interior (b) Mercury-Mercury VILTs from launch $\nu = 0$.



(a) (b)

Figure 3.11: TOF of exterior (a) and interior (b) Mercury-Mercury VILTs and ballistic transfers from launch $\nu = \pi/2$.



(a)

(b)

Figure 3.12: ΔV of exterior (a) and interior (b) Mercury-Mercury VILTs from launch $\nu = \pi/2$.

along the eccentric body orbit can affect the VILT families as compared to those shown in Figs. 3.9 and 3.10. Figures 3.8I-3.8L show example cases for Mercury $\nu = \pi/2$. Figure 3.8L is similar to Fig. 3.8F but shows the opposite effect: where the leveraging apse falls outside the body orbit for an interior VILT. This example again highlights that the body orbits are for reference only while the presented boundary value VILT method depends on TOF and the two given position vectors only.

Changing the true anomaly of the initial condition shifts all of the families of ballistic and VILT solutions. Since the body velocity is lower at $\nu = \pi/2$, each particular resonance requires a higher initial v_∞. In addition, the minimum v_∞ VILTs along resonance-crossing families are not located at the resonances as they were in the circular body orbit and $\nu = 0$ cases. Figure 3.8J is an example of a VILT which has the minimum initial v_∞ in its family but which is not the resonant solution. As with the previous cases, the resonant VILT solutions occur at the intersection of increasing and decreasing VILTs of the same resonance in Fig. 3.11. These resonant intersections do approach a maneuver size of zero, as shown in Fig. 3.12. However, these resonant VILT solutions do not occur at the same initial v_∞ as the corresponding ballistic resonance the way they did in the circular body orbit and $\nu = 0$ cases. This difference arises from the use of tangent VILTs which require that one of the two encounters be at a spacecraft apse.

Figure 3.13 illustrates how, for a given body velocity, there exists a range of v_∞ which can reach the velocity associated with a specific resonance, and each v_∞ in the valid range corresponds to a particular pump angle, with the minimum v_∞ for the resonance occurring at $\alpha = 0$. The tangent VILT constraint requires the spacecraft velocity to be perpendicular to the position vector at the low encounter, but along the eccentric body orbit the body velocity is not perpendicular to the position vector except at the body orbit apsides. As a result the tangent VILT constraint requires a nonzero pump angle at the low encounter which leads to a nonminimum resonance v_∞. In the circular body orbit case, the convergence of VILT and ballistic families to the same resonant v_∞ requires (from geometry) that this v_∞ be the minimum of the possible range of resonant v_∞.

In the previous examples whenever there were both $S = 1$ and $S = 2$ VILTs associated with a particular resonance, the different S values existed in non-overlapping domains of v_∞. However, there are two families of interior $d2{:}2^+$ VILTs (Figs. 3.11b and 3.12b) which exist within an overlapping range of initial v_∞, showing that this overlap is possible when the body orbit is eccentric. Figure 3.8L is an example of a point where a VILT family intersects

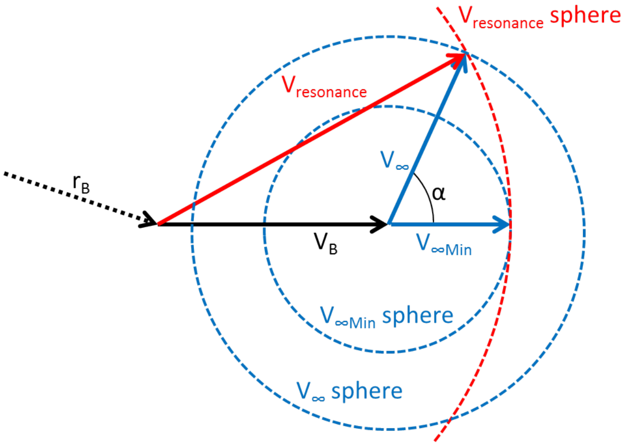


Figure 3.13: Different v_∞ magnitudes that can give the same resonant velocity, with the minimum v_∞ occurring when $\alpha = 0$.

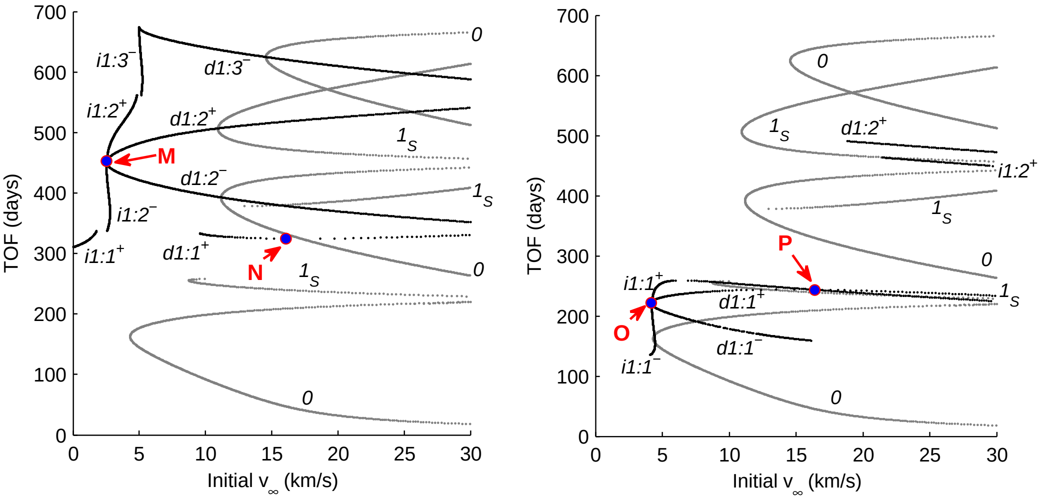
a nonresonant ballistic family, another possible situation with an eccentric body orbit.

3.6.3 Earth-Venus Transfer Example: Bielliptic transfers as a subset of interbody leveraging

As the presented VILT boundary value method is capable of inter-body leveraging, the next example is an Earth-Venus transfer. Relative geometry of the two bodies is important, so for simplicity the launch occurs at opposition in this case, with body orbital elements provided in Table 3.2 and search constraints in Table 3.3. Unlike the case where a VILT returns to the same body it departed, there is no clear definition of resonances for IB-VILTs. However, since in this search the launch time is fixed, resonances can conveniently be defined relative to the arrival body, Venus, maintaining resonance notation for the IB-VILTs.

Figure 3.14 shows that increasing and decreasing VILT families converge when Venus at arrival is in opposition with where the Earth was at launch time. This relative geometry leads to the expected and interesting result that the convergence of increasing and decreasing exterior VILT families corresponds to the classic bi-elliptic transfer case, an example of which is given in Fig. 3.8M. This convergence implies that both the encounter at Earth and the encounter at Venus occur at a true anomaly of zero or π radians and so they are tangent to their respective planet velocities for planets in circular orbits. Although bi-elliptic transfers are traditionally defined with the maneuver (leveraging) apse exterior to the initial and final positions of the transfer, there is an equivalent geometry for interior IB-VILTs, shown by Fig. 3.8O. The bi-elliptic solutions have low initial v_∞ in their families but non-negligible maneuver magnitudes, as seen in Fig. 3.15.

As seen in previous cases, the transitions between the different values of S are visible as extrema in Fig. 3.14 and as reduced solution density near the transition in Figs. 3.14-3.16. Figure 3.8N gives an example trajectory near one of these transitions. Additionally, similar to the eccentric single-body VILT examples, there are intersections with nonresonant ballistic families of solutions, occurring in exterior decreasing VILTs and interior increasing VILTs as shown in Fig. 3.15. The fact that ballistic families of transfers connect to the families of VILT solutions, showing some ballistic solutions to be zero-maneuver subsets of VILT families, has been documented before for circular-orbit, single-body VILTs [117, 18]. However, the VILT families extend into much lower ranges of initial v_∞ than the ballistic families, demonstrating one of the potential advantages of the general formulation VILT solutions. In this particular case, the ΔV of VILTs is prohibitively expensive anywhere far from the resonances, but for an intermoon case the same families of solution exist



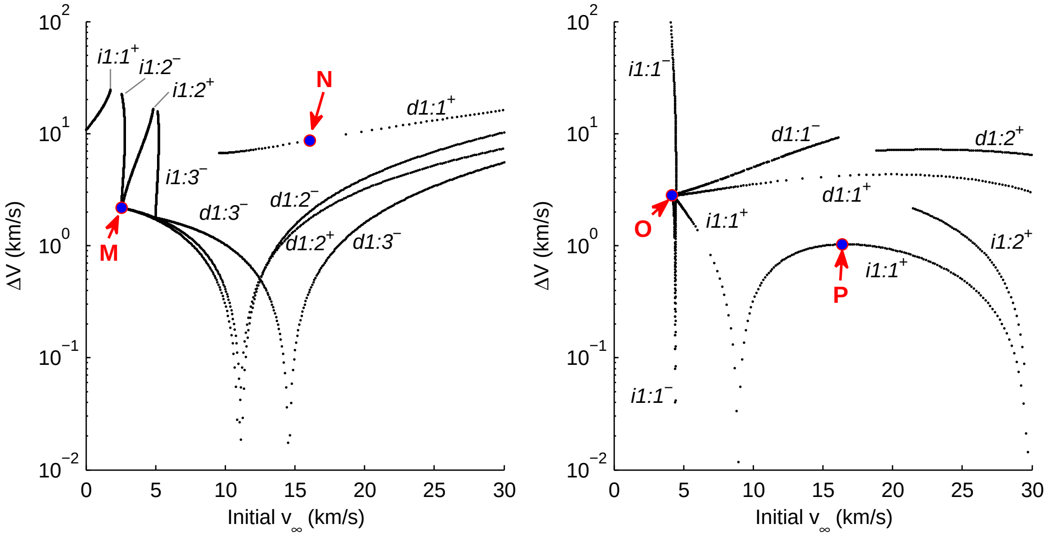
(a)

(b)

Figure 3.14: TOF of ballistic as well as exterior (a) and interior (b) IB-VILT Earth-Venus transfers.

with much lower ΔV.

Figure 3.16 shows how the IB-VILTs change v_∞, with the bi-elliptic Figs. 3.8M and 3.8O occurring at the minima of their families in $v_{\infty H}$. Each bi-elliptic VILT actually appears twice in Fig. 3.16, once between the increasing families and once between the decreasing families of its resonance. $v_{\infty L}$ and $v_{\infty H}$ are on opposite ends of the transfer between these two families, so while the intersections of the increasing and decreasing families overlap in Fig. 3.14, they are distinct points in Fig. 3.16. Similar maps of the change in v_∞ from single-body circular-orbit VILTs have been published before [116]. It is also noteworthy that $v_{\infty L}$ can approach zero, a case in which one half of the VILT shadows one of the body orbits and the VILM provides all of the required change in energy to transfer to the other body.



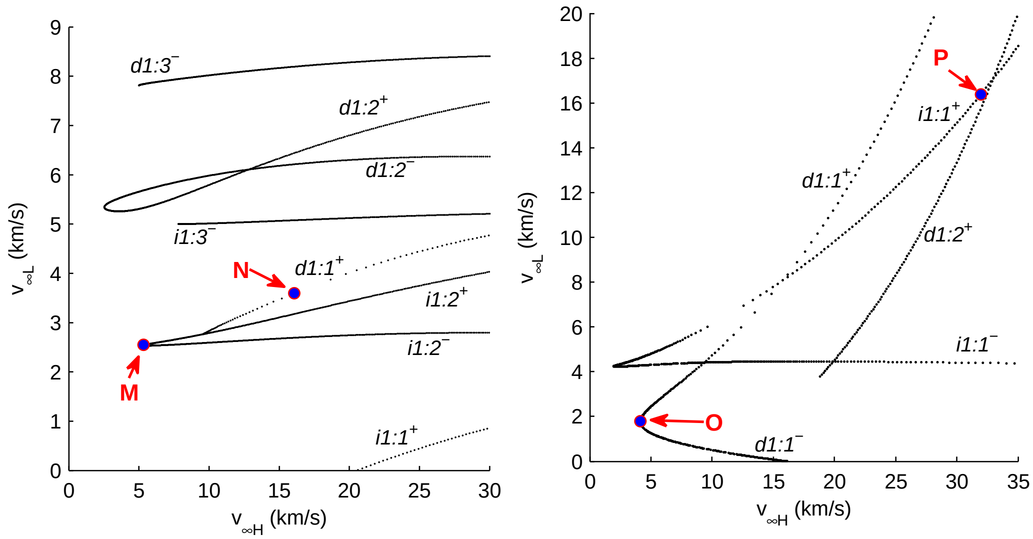
(a) (b)

Figure 3.15: ΔV of exterior (a) and interior (b) IB-VILT Earth-Venus transfers.

3.7 VILTs within an MGA sequence: some trajectories to Jupiter

The VILT formulation presented is incorporated into the MGA search algorithm described in Chapter 2 to investigate more complex trajectory options. A broad search of the path Earth-Earth-Jupiter using ephemeris states[2] shows how the expected ΔV-EGA trajectories appear, with launch dates given in Fig. 3.17 and resulting arrival v_∞ for different launch energies in Fig. 3.18a and corresponding ΔV in Fig. 3.18b. The search is constrained to closely match one of the searches done by [39] with maximum arrival v_∞ of 8 km/s, maximum spacecraft maneuvers of 1.25 km/s, and maximum mission time of

[2] JPL Database: de405.bsp Available online: ftp://ssd.jpl.nasa.gov/pub/eph/ [Accessed 10 January 2012]



(a)

(b)

Figure 3.16: v_∞ mapping of exterior (a) and interior (b) IB-VILT Earth-Venus transfers.

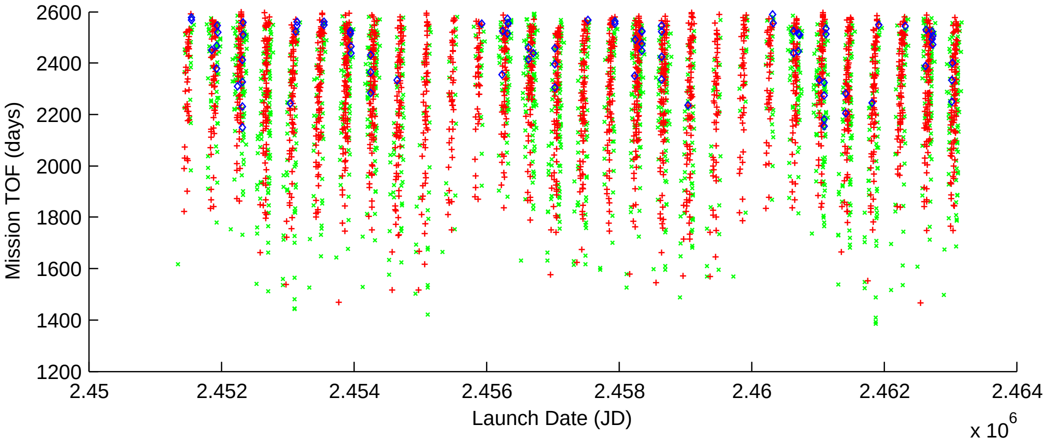


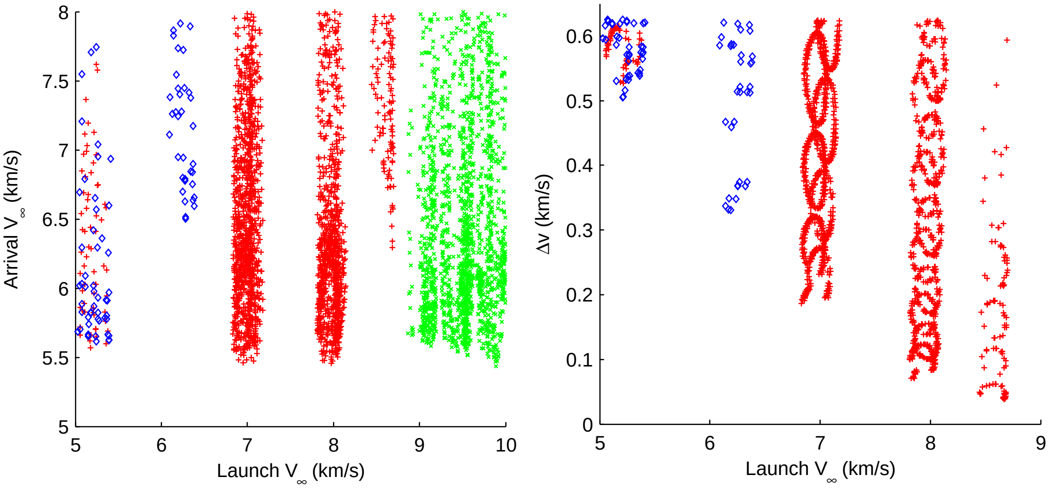
Figure 3.17: Opportunities from 1999 to 2032 for an Earth-Earth-Jupiter path, including VILT cases

Table 3.4: Symbol and color scheme for FIGURES

Case	Shape	Color
Ballistic	×	Green
VILT – 1 rev., exterior	+	Red
VILT – 2 revs., exterior	◊	Blue

2600 days (7.1 years). Additional constraints include maximum Earth launch v_∞ of 10 km/s and, for just the first leg, a maximum of five complete spacecraft revolutions and a five year time of flight. Ballistic cases of the path Earth-Earth-Jupiter, which are not truly ΔV-EGAs, are included in the trajectory search for comparison. Discretization resolutions for this search are set to 55 and 75 points per orbit period (ppr) for Earth and Jupiter respectively. Table 3.4 provides the marker and color scheme used in Fig. 3.17-3.20, where the solutions are distinguished by their first leg: Earth-Earth in Fig. 3.17-3.18; Earth-Mars in Figs. 3.19-3.20.

Figure 3.18 shows how the VILT enables significantly reduced launch



(a) (b)

Figure 3.18: Characteristics of trajectories on the path Earth-Earth-Jupiter, including VILTs

energies for the same arrival energies even for a reasonable maneuver size. The solutions fall into distinct VILT families, or rather groups of families. This example trajectory search demonstrates that the formulation presented can be used in a broad search using ephemeris states. With sufficient discretization on the search, multiple solutions per family are found, providing a global view of the options available for this sequence and constraints.

The sequence Earth-Mars-Jupiter is also investigated by [39] and they find at least one promising ballistic solution in the time range searched. This search is repeated, with the same constraints as the ΔV-EGA case described above, but now with the ability to perform IB-VILTs to expand the design space. The discretization at Mars is 65 ppr. This ability to expand the design space by considering VILT-like transfers between bodies is one of the primary contributions of this work.

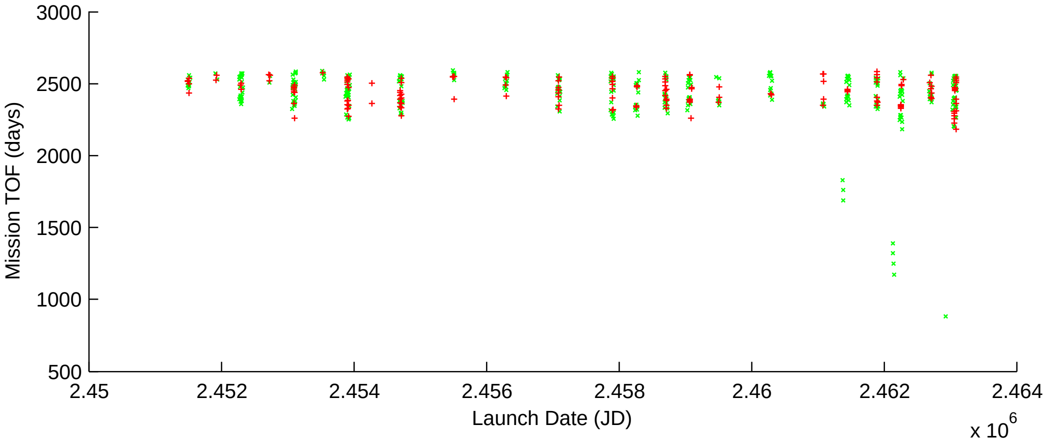


Figure 3.19: Opportunities from 1999 to 2032 for an Earth-Mars-Jupiter path, including IB-VILT cases

Figures 3.19 and 3.20 show launch opportunities and energies for launch and arrival for the period 1999 to 2032, with VILM solutions generally lining up in timing with ballistic solutions, but with lower launch and arrival energies. Although the VILT families are not as clearly distinguishable in Fig. 3.20 as in the Earth-Earth-Jupiter case, it is still clear the VILTs provide lower arrival as well as launch v_∞, generally at the expense of longer TOF.

Figure 3.21 shows one example of a trajectory found directly in this search, without further optimizing, that demonstrates a balance of good characteristics described in Table 3.5.

3.8 A case for Nontangent VILTs

For many cases tangent VILTs are preferred to nontangent VILTs because they tend to be more efficient; However, there are cases where other constraints make nontangent VILTs preferable or required. Leveraging using low-mass moons is one situation where non-tangent VILTs can be mission

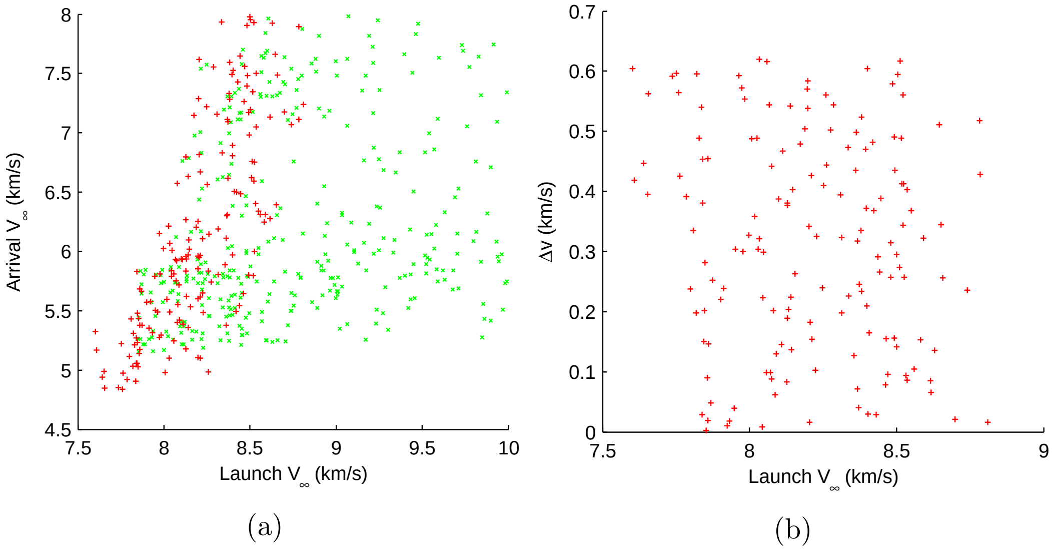


Figure 3.20: Characteristics of trajectories on the path Earth-Mars-Jupiter, including IB-VILTs

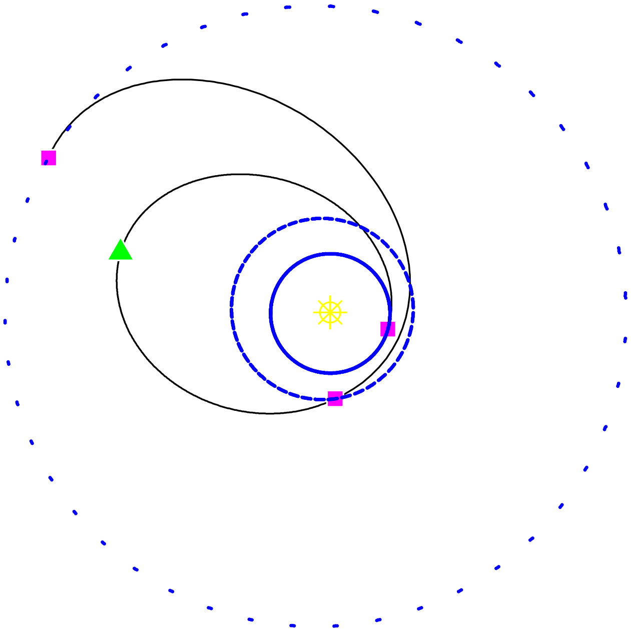


Figure 3.21: An example trajectory that utilizes an IB-VILT (Triangle indicates VILM). Trajectory characteristics are given in Table 3.5

Table 3.5: Characteristics for the Earth-Mars-Jupiter trajectory in Fig. 3.21.

Parameter	Value	
Launch Date	6 Sept.	2020
Launch v_∞	7.654	km/s
Maneuver Time	649.0	days after launch
DSM ΔV	0.396	km/s
Flyby Time	1238	days after launch
Flyby Altitude	158.6	km
Arrival v_∞	4.989	km/s
Total Time of Flight	2471	days

enabling [116, 118]. Another such case is when v_∞ is relatively high and approach geometry is tightly constrained, such as a flyby-assisted capture at Jupiter. An illustrative example follows: Beginning with a double-flyby capture, a VILT is appended after capture to reduce v_∞ at Ganymede. This case is based on the 6th Global Trajectory Optimization Competition (GTOC6) problem,[3] but with the orbital elements of the moons and their gravitational parameters provided in Table 3.7. The full sequence of encounters and maneuvers is then ∞-Io-Ganymede-VILM-Ganymede, where ∞ is a point sufficiently distant so as to approximate Jupiter's sphere of influence. The VILM in this sequence is also a Perijove Raise Maneuver (PRM) common in many Jupiter capture scenarios, except that solar radiation pressure is not accounted for. The ΔV would need to be adjusted accordingly if this important perturbation were taken into account. This sequence is shown in Fig. 3.22, with the VILT indicated by the dashed line. Table 3.6 provides details on the VILT in Fig. 3.22.

[3] Data available online at http://sophia.estec.esa.int/gtoc_portal/wp-content/uploads/2012/11/gtoc6_problem_stmt-2.pdf [retrieved 15 Feb. 2015]

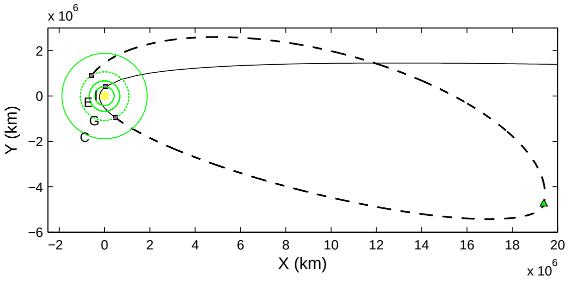


Figure 3.22: Flyby-assisted capture trajectory at Jupiter followed by a nontangent VILT (dashed line), with moons labeled by their first letter.

Table 3.6: Parameters for the VILT in Fig. 3.22.

Parameter	Value	
$v_{\infty 1}$	14.1617	km/s
$v_{\infty 2}$	6.16534	km/s
TOF	108.534	days
ΔV	0.401698	km/s
ξ_L	-0.761599	rad
efficiency	19.9064	

In this particular case, there are no tangent VILTs that can generate the same sequence. Having a high initial v_∞ limits the possible feasible trajectories in two ways: First, higher v_∞ reduces the available turn angle from a flyby at a given altitude. Second, sufficiently high v_∞ can make ranges of pump angle infeasible because the trajectories which correspond to them would lead to escape or impact with the primary. Because these trajectories represent the extremes of orbital energy possible at a given v_∞, the regions of pump angle near $\alpha = 0$ and $\alpha = \pi$ radians, which correspond to tangent VILTs for circular body orbits, are first to become infeasible at high v_∞. As a result, having the extra free parameter to include non-tangent VILTs allows for solutions that, otherwise, simply do not exist. On the other hand, if tangent VILTs are expected to be possible and most efficient, then the search can be reduced by setting the continuous parameter $\xi_L = 0$.

Table 3.7: Orbital elements and parameters for Jovian moons used in the capture trajectory in Fig. 3.22

Parameter	**Io**	**Ganymede**
Semimajor axis (km)	422030	1070590
Eccentricity	0	0
Inclination (radians)	0	0
Arg. of periapsis (radians)	0	0
Long. of the Asc. Node (radians)	0	0
Mean Anomaly (radians)	4.279603	2.215313
Epoch (Modified Julian Date)	58849	58849
Gravitational Param. (km3/s2)	5959.92	9887.83

3.9 VAMs for Orbit Insertion and Flyby-Assisted Capture

The boundary-value VILT method can not only place VILMs in a flyby sequence as Perijove Raise Maneuvers (PRM) as in Fig. 3.22 but also the Jupiter Orbit Insertion Maneuvers (JOI). Figure 3.23 shows just such an example, where the sequence ∞-Io-JOI-PRM-Ganymede is calculated using ephemeris states[4] and patched conics trajectories. Again ∞ here is a point near Jupiter's sphere of influence (SOI), distant from the planet. Table 3.8 provides important trajectory details for this example, setting the spacecraft up for a tour to follow. The maneuvers in this trajectory are not optimized, but are the result of a grid search, so using a local optimizer to refine the trajectory should produce a superior trajectory in terms of ΔV or arrival v_∞ or both.

In this case, the JOI is determined by the VILT method, but is more

[4] JPL Database: jup230l.bsp Available online: ftp://ssd.jpl.nasa.gov/pub/eph/ [Accessed 10 January 2012]

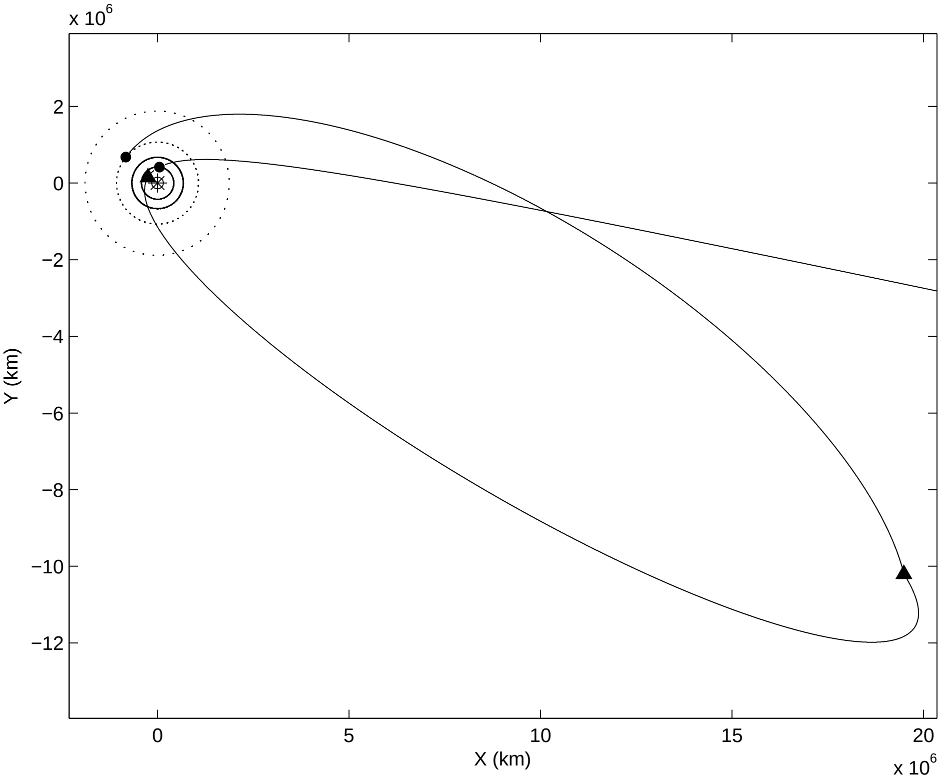


Figure 3.23: Flyby-aided capture at Jupiter with JOI placed by the VILT routine and PRM location found by grid search. This is one trajectory from the Pareto front in Fig. 3.24. Circles show encounters with the Jovian moons and triangles indicate maneuver locations. Sequence details are provided in Table 3.8

Table 3.8: Details of the flyby-assisted Jovian capture trajectory in Fig. 3.23

Event	Time from JOI (days)	dist. to center of Jupiter (km)	v_∞ (km/s)	Flyby alt. (km)	ΔV (km/s)
Start at ∞ (SOI)	-83.87	4.8200e7	5.806		
Io Gravity Assist	-0.1712	4.2185e5	14.05	358.24	
JOI ΔV	0.00	2.9546e5			0.452
PRM ΔV	159.95	2.1999e7			0.342
Ganymede Arrival	260.71	1.0732e6	4.271		

accurately a VAM since the maneuver itself is not intended to leverage v_∞. The PRM is a node in the trajectory search algorithm and is placed by a grid search. Although this significantly slows computation, placing the PRM as a node in the search is required by the limits of the search architecture and the fact that the VILT method presented requires that there be only one maneuver between two nodes. The tradeoff between v_∞ at Ganymede and ΔV for this sequence is shown in Fig. 3.24 with the example marked, again showing the utility of the presented methods: Not only do these methods enable an analysis of the trade space at this portion of the trajectory, but they can also be used calculate and continue the tour from whole families of solutions.

The same method could equivalently be used to find a sequence like ∞-GA-JOI-GA-tour, where ”GA” is a gravity assist and ”tour” is the rest of the moon tour which follows the capture. This kind of sequence would remove the grid search for the PRM, but such a sequence is undesirable because unless SRP is accounted for in the search, a DSM will be required on the first orbit and raising the perijove with a ΔV significantly reduces radiation dose experienced by the spacecraft as compared to using gravity assists to both reduce orbital energy and raise perijove. Another possible sequence which would make the most advantage of the VILT method is ∞-GA-JOI-GA-PRM-GA-tour; with a gravity assist between the two maneuvers, both the JOI and PRM can be found using two instances of the described VILT method. This sequence is operationally undesirable because the gravity assist following the JOI happens very shortly after the insertion maneuver and would magnify the maneuver dispersion errors. This gravity assist would also have limited effectiveness because the flyby altitude would have to be high to minimize chance of impact after the uncertainty introduced by the JOI.

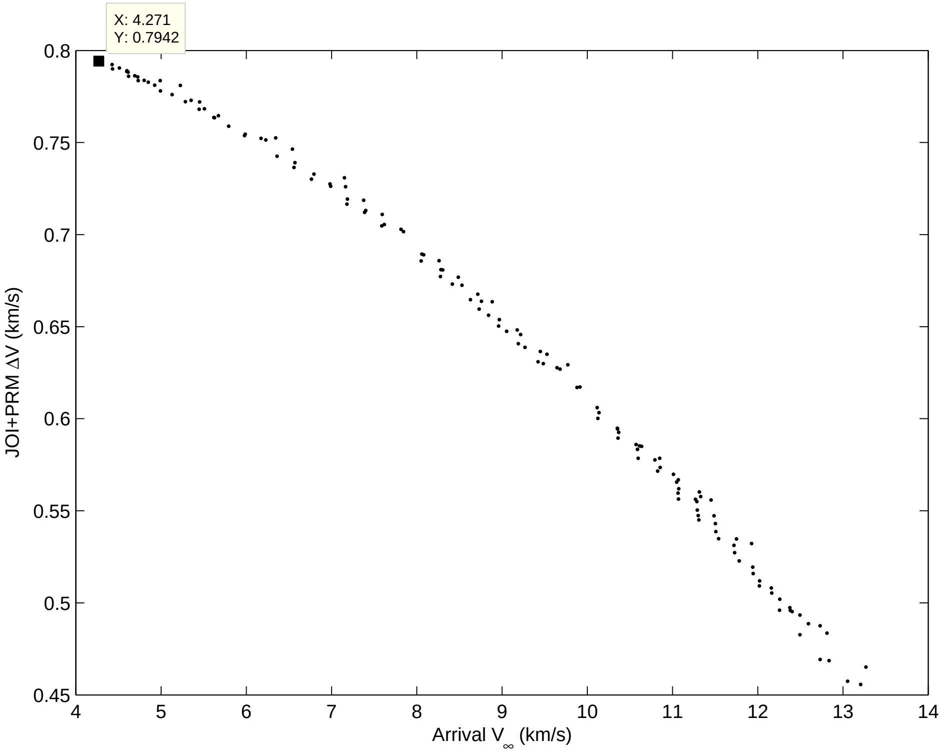


Figure 3.24: Example Jupiter capture trade space for sequence ∞-Io-JOI-PRM-Ganymede determined using the presented search algorithm and VILT method. Figure 3.23 shows the highlighted trajectory.

3.10 VAMs as initial guesses for low-thrust transfers

Since calculating low thrust (LT) trajectories is generally much more time-consuming than ballistic transfers of VILT transfers using the boundary value formulation, there are cases where an impulsive VAM can be used as a proxy and initial guess for a LT transfer.

In particular, exterior VILTs with high eccentricities have been successfully used as initial guesses for LT transfers in the pump-down phase of the GTOC6 trajectory, where the final result is an LT-VILT that greatly reduces mission time needed to reduce the spacecraft energy from the initial capture orbit to fast tours of the inner Jovian moons. The maneuver placement at apoapsis and high eccentricity allows time for the LT engine to build up the ΔV estimated by the VILT. For the GTOC6 case, the entire mission is generated by a single run of the search tool, so the conversion to LT is done to enforce the same encounter constraints as well as minimize the ΔV so that the remainder of the mission is not recalculated after the conversion. By limiting the impulsive ΔV based on the flight time of the transfer, only trajectories which were likely to convert to LT were kept in the search. Figure 3.25 shows the capture and pump-down sequence with LT VILTs in a Sims-Flanagan approximation; the initial guess for this trajectory was generated using a single impulsive VAM between each pair of flybys. The highlighted sequence incorporating capture and the LT VILTs is ∞-Io-Ganymede-LTVILT-Ganymede-LTVILT-Ganymede-LTVILT-Ganymede.

Additional information on this solution to GTOC6 is provided in Appendix C.

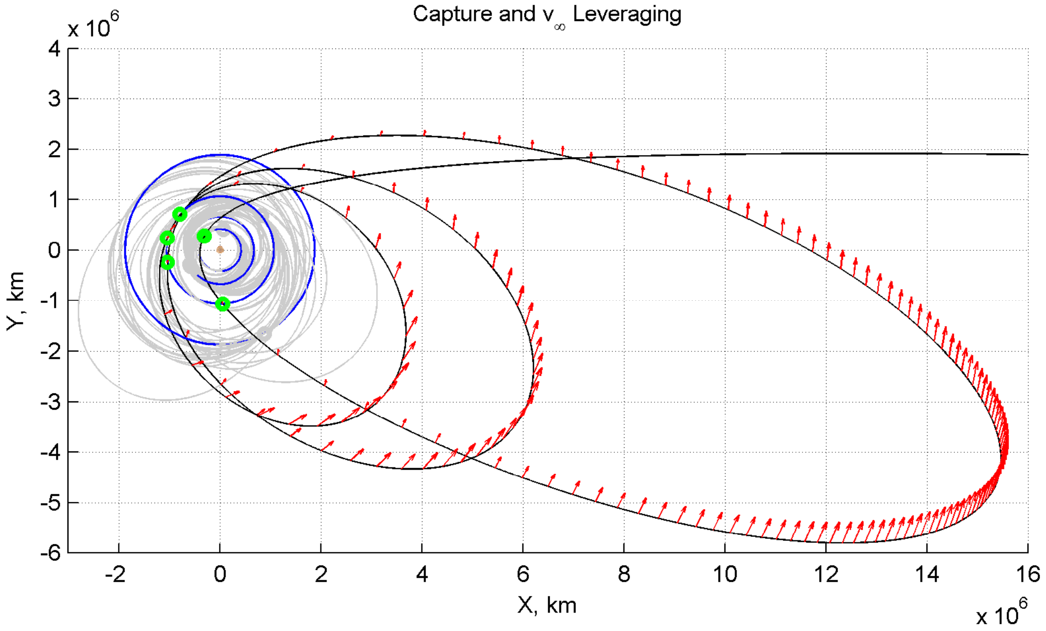


Figure 3.25: Low thrust VILTs for a Jovian pump-down as part of a solution to GTOC6. Impulsive VILTs were used as initial guesses for the LT optimizer.

Chapter 4

Automated Inclusion of $n\pi$ Sequences in the Global MGA Trajectory Search

Defined in Chapter 1, the special class of spacecraft transfers known as $n\pi$ transfers are further detailed and investigated in this chapter[1]. An automated method for performing both pathfinding and pathsolving for a sequence of $n\pi$ transfers is presented. A gap-fill algorithm enables the $n\pi$ sequence to be posed as a BVP, greatly simplifying its solution. The problem is cast on the v_∞ sphere to compactly present the solution options.

4.1 The v_∞ Sphere

The $n\pi$ sequence problem is computed and visualized using the v_∞ sphere [3, 121], which represents the locus of possible spacecraft velocity vectors preceding or following a gravity assist flyby as shown in Figure 4.1. Free return trajectories that are $n\pi$ transfers appear as intersections of velocity spheres or velocity circles with this v_∞ sphere (depending on whether they are even $n\pi$ or odd $n\pi$ transfers, respectively), as shown in Figure 4.1.

[1] This chapter contains information also published in:

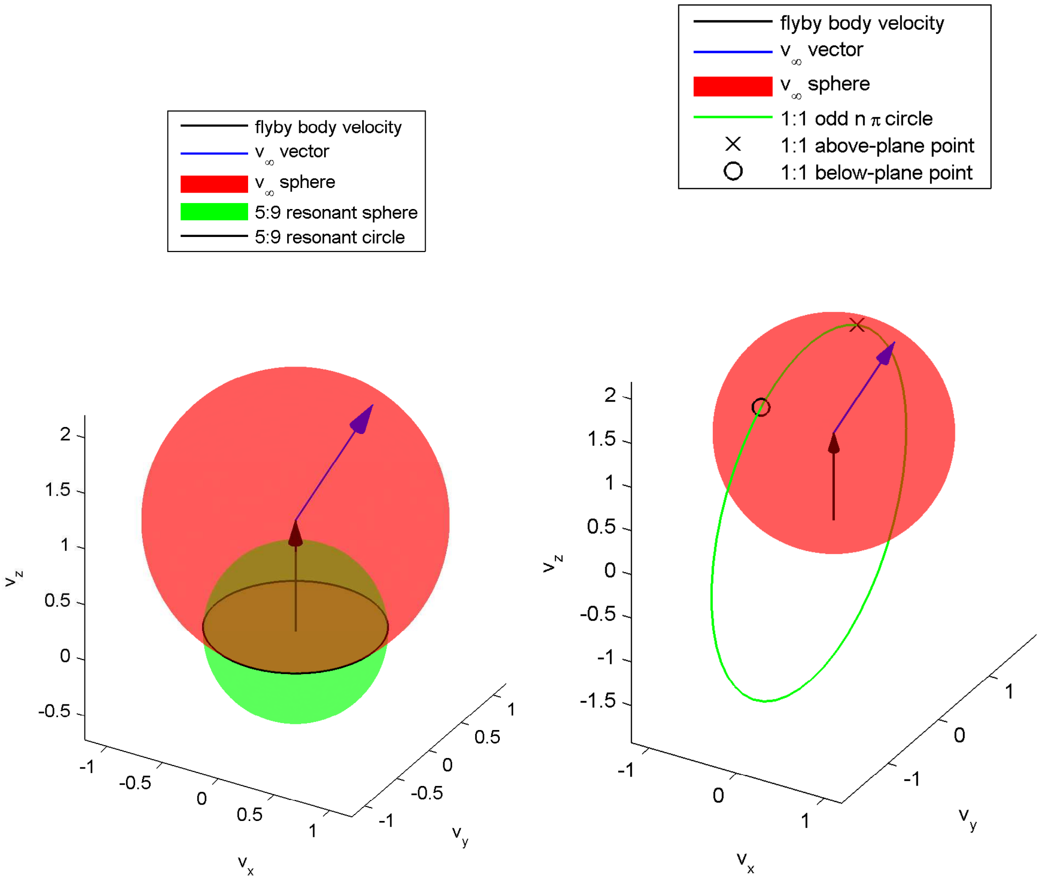
- Demyan Lantukh and Ryan P. Russell. “Automated Inclusion of n-pi Transfers in Gravity-Assist Flyby Tour Design,” *In* Advances in the Astronautical Sciences, volume 143, 2012

Ryan P. Russell developed the two BVP pathfinding/pathsolving algorithms on the v_∞ sphere described herein. He also provided implementations to be integrated into *Explore*.

Each of these intersecting spheres or circles corresponds to a particular body-to-spacecraft resonance (M:N), with the full-rev 5:9 resonance shown by the black circle in Figure 4.1a. In this figure, the smaller sphere represents all velocities which return to the body after a 5:9 resonance, so the intersection of the two spheres meets both the resonance and the v_∞ constraints. Similarly, the large circle in Figure 4.1b shows all possible velocities which would return to encounter the body after a 3π transfer angle using a 1:1 pseudoresonance, so the two intersections of this circle with the v_∞ sphere form the possible transfers (above and below plane) that meet the v_∞ constraint for the given resonance.

The intersections between the velocity surface of the resonance and the v_∞ sphere are therefore circles and pairs of points on the v_∞ sphere: they represent the feasible $n\pi$ transfers to consider for a given spacecraft v_∞. Note that for each resonance considered, the free-return solutions only exist for a range of v_∞. Details for determining v_∞ ranges and computing the resonant intersections can be found in [3] .

An example v_∞ sphere with views from different angles is given in Figure 4.2. This sphere, and specifically from the view shown in the bottom right of Figure 4.2 is reiterated in Figure 4.3a: Figure 4.3 gives two example v_∞ spheres showing the locations of several different $n\pi$ transfers with their associated body to spacecraft (M:N) resonances labeled. The N of the resonance for odd $n\pi$ transfers can be either positive or negative, indicating that the transfer is the long period or short period Lambert solution, respectively. When $N = 0$ the transfer has less than one full spacecraft revolution and there is only one possible solution, so the sign of N is not significant. The parameters used to generate Figure 4.3b are $\{\mu_{Pr}, \mu_B, a_B, e_B, i_B, \Omega_B, \omega_B, \nu_{BI}\} =$



(a) (b)

Figure 4.1: Example $n\pi$ elements from the intersections with a v_∞ sphere (v_∞ = 1.2 LU/TU): (a) 5:9 resonant even $n\pi$ sphere (b) the 1:1 odd $n\pi$ circle

$\{1.0, 7.0 \times 10^{-9}, 1.0, 0.1, 0.0, \pi/2, \pi/4, 4\pi/3\}$, with part (a) using the same conditions except $e_B = 0$. Here the orbital elements follow the definitions provided in Appendix A, and ν_{BI} is the body true anomaly at the time of the incoming v_∞^I. In Figure 4.3, the horizontal lines are even $n\pi$ resonances viewed from their edge and the x's and o's are pairs of odd $n\pi$ transfers, with each pair

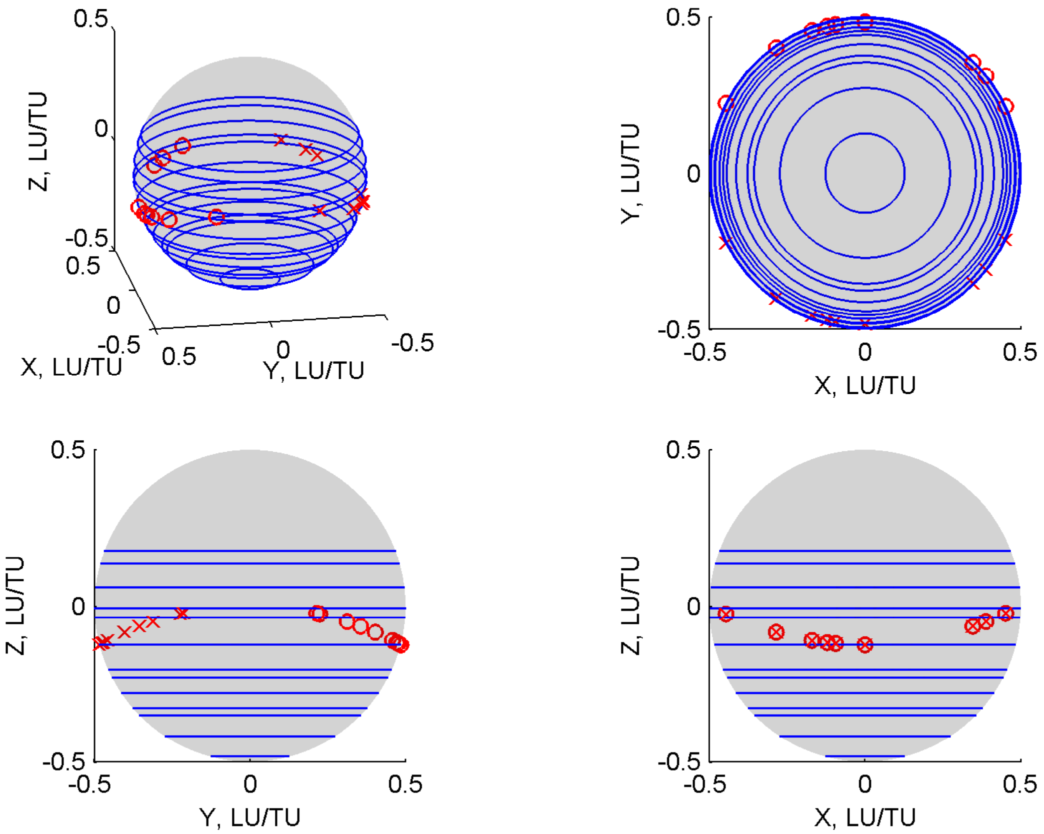
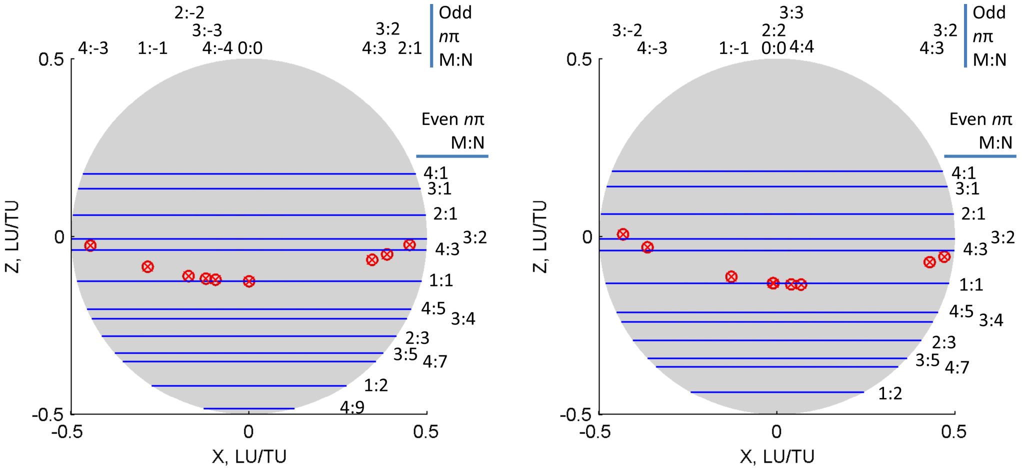


Figure 4.2: An example v_∞ sphere with even $n\pi$ transfers (circles) and odd $n\pi$ transfers (x's and o's) mapped onto it, shown from different angles to capture the three-dimensional structure of the v_∞ sphere

overlaid in the given view. The axes in Figure 4.2 and Figure 4.3 and for all v_∞ spheres presented are oriented such that Z parallels the instantaneous body velocity, Y parallels the body orbit angular momentum, and $X = Y \times Z$. In this coordinate system, the even $n\pi$ returns are circles of constant latitude on the sphere. Similarly, the odd $n\pi$ transfers are pairs of points which reflect each other over the $X-Z$ plane leading to opposing longitudes on the sphere.

In addition to the $n\pi$ transfers, all generic free-return trajectories can



(a)

(b)

Figure 4.3: v_∞ spheres with even $n\pi$ transfers (lines) and odd $n\pi$ transfers (x's and o's) mapped onto them and labeled with their respective resonances. The difference between the two spheres is the body eccentricity.

also be mapped to the v_∞ sphere since every point on the sphere represents a particular velocity vector and therefore a particular trajectory. More specifically, all transfers which return to the same body map to two points on the v_∞ sphere, one for the departure v_∞ vector and the other for the arrival v_∞ vector. In the case of even $n\pi$ transfers, these two points coincide. In the odd $n\pi$ case with a circular flyby body orbit, the return vector is the opposite point in the pair as the departure point [121].

The differences between Figure 4.3a and Figure 4.3b come from the body eccentricity effects. A non-zero eccentricity usually leads to differences in spacecraft v_∞ at the two encounters of an odd $n\pi$ transfer. Therefore, two v_∞ spheres must be considered in the eccentric body odd $n\pi$ case, which is also true for generic returns at a flyby body with a non-circular orbit. The implications of this effect are discussed in more detail below.

4.2 The v_∞ Pathfinding Problem

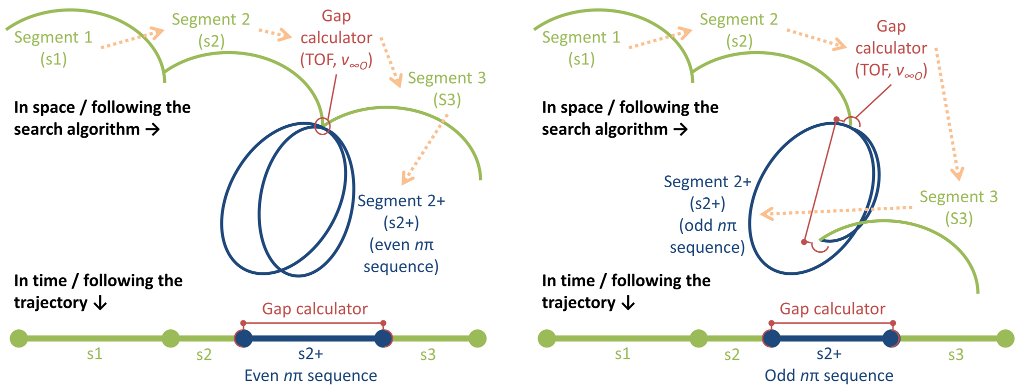
The goal is to automatically choose a sequence of $n\pi$ transfers that effectively navigates from one given inbound $\mathbf{v}_\infty$ vector ($\mathbf{v}_\infty^I$) to another given outbound $\mathbf{v}_\infty$ vector ($\mathbf{v}_\infty^O$) in the context of a MGA trajectory GS or GO. As described in Chapter 2, the trajectory search is a sequence of Lambert problems. The $n\pi$ transfers are then treated as gaps in time, or loiters, allowing existing algorithms and software like *Explore* to handle the non-$n\pi$ Lambert problems.

Using time gaps is illustrated in a simplified manner in Figure 4.4. Since the time gaps can be well defined before the DOFs for the $n\pi$ transfers are selected, the Lambert problems following the $n\pi$ transfers can be evaluated first, allowing both boundary $\mathbf{v}_\infty$ vectors to be available for the $n\pi$ sequence pathfinding and pathsolving problems on the v_∞ sphere. Having both boundary values available aids in developing a strategy for choosing DOFs and helps keep the problem numerically tractable, as described in the Choosing the Degrees of Freedom section.

Pathfinding on the v_∞ sphere is accomplished with these steps, which are each described more fully below:

1. Create a gap (discontinuity) in the trajectory search to be filled by the $n\pi$ sequence

2. Solve the trajectory segment following the $n\pi$ sequence, including v_∞ matching so that v_∞^O from the gap calculator matches v_∞ of the transfer following the $n\pi$ sequence. Now a BVP is defined on the gap: $\mathbf{v}_\infty^I$ on the inbound v_∞ sphere and $\mathbf{v}_\infty^O$ on the outbound v_∞ sphere.



(a) (b)

Figure 4.4: Qualitative description of how non-chronological calculation in the search algorithm enables the implementation of the pathfinding problem as a boundary value problem for both odd (a) and even (b) $n\pi$ sequences.

3. Populate the v_∞ sphere(s) on each bound of the gap with $n\pi$ transfers associated with the bounding v_∞ value(s)

4. Pathfinding: Enumerate potential sequences (paths) connecting $\mathbf{v}_\infty^I$ to $\mathbf{v}_\infty^O$.

5. Pathsolving: Select any degrees of freedom and evaluate sequence feasibility

Even and odd $n\pi$ cases have some distinct differences and the odd $n\pi$ cases are more difficult to evaluate because the transfer can change the v_∞ if the body eccentricity is non-zero. Also, evaluating the TOF associated with the odd $n\pi$ case is more difficult than the even $n\pi$ case. Because of these complications, odd $n\pi$ sequences are limited to one odd $n\pi$ transfer, with $n\pi$ sequences defined below. Even $n\pi$ sequences are limited to only have even $n\pi$ transfers. These limitations could be lifted with little change to the algorithm but would require significant implementation efforts to properly account for all the trajectory possibilities.

4.2.1 Defining the Problem

Define a *transfer* as the ballistic coast between two flyby body encounters and define a *sequence* as a set of transfers (at least one) for which the boundary conditions ($\mathbf{v}_\infty$ vectors) are prescribed. Note that the $\mathbf{v}_\infty$ in between transfers within one sequence are not prescribed: determining these $\mathbf{v}_\infty$ is one goal of the presented method. A sequence is therefore an ordered series of *elements*, where the first and last are nominally prescribed $\mathbf{v}_\infty$ vectors and there are $j-1$ transfers in between as depicted in Eqn. (4.1):

$$\mathbf{v}_{\infty\mathbf{I}} \xrightarrow[\delta_1]{} \epsilon_1 \xrightarrow[\delta_2]{} \epsilon_2 \xrightarrow[\delta_3]{} \cdots \xrightarrow[\delta_{j-1}]{} \epsilon_{j-1} \xrightarrow[\delta_j]{} \mathbf{v}_{\infty\mathbf{O}} \tag{4.1}$$

In this description, each arrow represents a gravity-assist flyby that navigates along the v_∞ sphere. Then there are a total of j gravity assists and $j-1$ transfers in a sequence. For an even $n\pi$ sequence, j falls in the range $2 \leq j \leq M_S + 1$. For an odd $n\pi$ sequence, j falls in the range $2 \leq j \leq 2M_S + 2$. With the assumption of no more than one odd $n\pi$ transfer, the maximum j corresponds to M_S single-rev 2π transfers and one half-rev π transfer. Note that in general j is not specified, so the number of elements in a sequence is also to be selected in the pathfinding process.

Each of the elements (ϵ_i) in the sequence is an $n\pi$ transfer with a particular resonance M_i:N_i. The set of possible even $n\pi$ elements on the v_∞ sphere is defined as $\{En\pi\}$ and each even $n\pi$ element has a DOF: longitude (λ) on the v_∞ sphere. Also define the set of possible odd $n\pi$ elements on the v_∞ sphere as $\{On\pi\}$. The two possible discreet types of odd $n\pi$ elements for a given M:N are above-plane (a) and below-plane (b). These types are treated as separate elements of $\{On\pi\}$, so $\{\exists\, M{:}Na \in \{On\pi\}\} \iff \{\exists\, M{:}Nb \in \{On\pi\}\}$. Define the set of available elements on the v_∞ sphere as $\{n\pi\}$, so that:

$$\begin{gathered}\text{If } \{\text{odd } n\pi \text{ sequence}\}\\ \text{Then } \{\{n\pi\} = \{En\pi\} \cup \{On\pi\}\}\\ \text{Else } \{\{n\pi\} = \{En\pi\}\}\end{gathered}$$

With the definitions provided, the pathfinding problem is laid out in Table 4.1

Table 4.1: Definition of the $n\pi$ sequence pathfinding/pathsolving problem on the v_∞ sphere

GIVEN:
Problem definition: $\mathbf{v}_\infty^I$, $\mathbf{v}_\infty^O$,* M_S, $N_{S,MAX}$, ν_{BI}
System definition: μ_{Pr}, a_B, e_B, i_B, Ω_B, ω_B
Constraint definition: M_{MAX}, N_{MAX}, δ_{MAX}, δ_{MIN}
FIND:
ANY or ALL set(s):
$\epsilon_i \in \{n\pi\}\ \forall\ i = 1, 2, 3, \cdots, j-1$
$\lambda_i\ \forall\ \epsilon_i \in \{En\pi\}$
j
SUBJECT TO:
$\left.\begin{array}{r} M_{MAX} \geq M_i \\ N_{MAX} \geq N_i \end{array}\right\}\ \forall\ i = 1, 2, 3, \cdots, j-1$
$\left.\begin{array}{r} \delta_{MAX} \leq \delta_i \\ \delta_{MIN} \geq \delta_i \end{array}\right\}\ \forall\ i = 1, 2, 3, \cdots, j$
$M_S = \sum_{i=1}^{j-1} M_i$
$N_{S,MAX} \geq \sum_{i=1}^{j-1} N_i$

*$|\mathbf{v}_{\infty\mathbf{O}}| = v_{\infty O}$ not an independent input, but depends on the transfers in sequence, as explained in the text

4.2.2 Creating the gap and determining the second boundary condition

The magnitude of $\mathbf{v}_\infty^O$ is not independent: it is a function of the transfers used in the $n\pi$ sequence. The direction of $\mathbf{v}_\infty^O$, however, is an independent input to the algorithm. This direction is determined by completely solving (i.e. root-solving for encounter times such that all incoming and outgoing flyby v_∞ values match) the segment of the trajectory following the $n\pi$ sequence, making $\mathbf{v}_\infty^O$ a valid and fixed part of the MGA trajectory being calculated *before* the $n\pi$ sequence path finding problem is solved. In order for the segment following the $n\pi$ sequence to be calculated, the TOF of the $n\pi$ sequence (denoted t_S) and v_∞^O must be known beforehand. A gap-filling routine provides the necessary information and the different cases to solve for the next segment in the trajectory (labeled Segment 3 in Figure 4.4).

For even $n\pi$ sequences, the gap is straightforward: the magnitude of v_∞ does not change ($v_{\infty O} = v_{\infty I}$). Also, the t_S is a multiple of the body orbit period, as indicated in Eqn. (4.2). Table 4.2 provides an algorithm for calculating the gap.

$$t_S = 2\pi M_S \sqrt{\frac{a_B^3}{\mu_{pr}}} \tag{4.2}$$

Although even $n\pi$ transfers have the same v_∞ at both ends of a sequence, odd $n\pi$ transfers can change the v_∞ at the flyby body when the body orbit is non-circular. Although this change can be useful, possibly providing a v_∞ leveraging effect without a maneuver, it complicates the process of navigating the v_∞ sphere. This complication results from the fact that when $r_1 \neq r_2$ and $e_B \neq 0$ then the v_∞ after the odd $n\pi$ return is not only different from v_∞ before the transfer, but each different odd $n\pi$ transfer (different M:N) changes v_∞ by a different amount.

Table 4.2: Gap calculator algorithm which allows the MGA search to proceed before fully solving the $n\pi$ sequence. The search proceeding on to the next segment provides the second boundary value for solving the $n\pi$ sequence.

INPUTS		**OUTPUTS**	
Problem def:	$\mathbf{v}_\infty^I$, M_S, ν_{BI}	Seq. time:	t_S (∀ branches)
System def:	μ_{Pr}, a_B, e_B, i_B, Ω_B, ω_B	Outbound:	$\mathbf{r_O}$, $\mathbf{v_{BO}}$, $v_{\infty O}$ (∀ branches)
Constraint def:	M_{MAX}, N_{MAX}	# branches:	p

```
IF even $n\pi$ sequence: **Even $n\pi$ Gap Calculator**
1 COMPUTE sequence time ($t_S$) from Eqn. (4.2)
2 COMPUTE initial body state ($\mathbf{r_I}$,$\mathbf{v_{BI}}$) from given body orbital elements
3 COMPUTE body state after sequence: $\mathbf{r_O} = \mathbf{r_I}$,$\mathbf{v_{BO}} = \mathbf{v_{BI}}$
4 COMPUTE outbound $v_\infty$: $v_{\infty O} = v_{\infty I}$
5 SPECIFY number of branches: $p = 1$
```

```
ELSE IF odd $n\pi$ sequence: **Odd $n\pi$ Gap Calculator**
1. FOR EACH $M = 1 \ldots M_{MAX}$
  1.1. COMPUTE sequence time ($t_S$) from Eqn. (4.5)
    1.1.1. The intermediate variable $t_{HR}$ is determined by solving Eqns. (4.3) and (4.4)
2. COMPUTE initial body state ($\mathbf{r_I}$,$\mathbf{v_{BI}}$) from given body orbital elements
3. COMPUTE body state after sequence ($\mathbf{r_O}$,$\mathbf{v_{BO}}$) by advancing the body true anomaly
   by $n\pi$ and using the $f$ and $g$ functions or an orbital element transformation
4. POPULATE list of feasible odd $n\pi$ transfers defined by $M$:$\pm N(a/b)$
  4.1. PERMUTE a list of eligible resonances that meet the $M$ and $N$ constraints
    4.1.1. The upper bound of $N$ may be lower than the given $N_{MAX}$. This upper bound
           can be found from the multi-revolution Lambert algorithm used to solve for $v_\infty$
           below
  4.2. FOR EACH eligible resonance $M$:$N$
    4.2.1. SOLVE multi-revolution Lambert problem with inputs $\{\mathbf{r_I}, \mathbf{r_O}, t_S, N\}$ to determine $a$
    4.2.2. IF at least one solution exists
      4.2.2.1. ADD Solution(s) to a list of Lambert-feasible solutions: $M$:$\pm N$
  4.3. FOR EACH Lambert-feasible solution $M$:$\pm N$
    4.3.1. COMPUTE odd $n\pi$ transfer latitude and longitude on the inbound $v_\infty$ sphere
           for the pair of odd $n\pi$ points, using equations in [3]
    4.3.2. IF the points exist on the inbound $v_\infty$ sphere
      4.3.2.1. ADD both above and below plane solutions to the feasible transfer list
5. COMPUTE $p =$ number of feasible odd $n\pi$ transfers
6. FOR EACH feasible odd $n\pi$ transfer
  6.1. COMPUTE spacecraft velocity at start of transfer: $\mathbf{v_1} = \mathbf{v_{BI}} + \mathbf{v_{\infty 1}}$; and the end of
       transfer ($\mathbf{v_2}$) using the equations 17 and 22 respectively in [160].
  6.2. COMPUTE spacecraft $\mathbf{v_{\infty 2}}$ at end of transfer: $\mathbf{v_{\infty 2}} = \mathbf{v_2} - \mathbf{v_{BO}}$
  6.3. COMPUTE sequence outbound $v_\infty$: $v_{\infty O} = |\mathbf{v_{\infty 2}}|$
7. STORE inbound $v_\infty$ sphere information to prevent recalculation in sequence algorithm
```

For odd $n\pi$ sequences, the gap is a change both in time and in position, as illustrated in the lower part of Figure 1.2 and qualitatively in Figure 4.4b. The final position for an odd $n\pi$ sequence is on the opposite side of the body orbit from the initial position, corresponding to a change in true anomaly of π radians (assuming only one odd $n\pi$ transfer per sequence). Knowing the structure of the odd $n\pi$ transfer and the body true anomaly at one of the encounters, it is possible to use Kepler's equation and a true anomaly relation (Eqns. (4.3) and (4.4), respectively) to determine the TOF required for the flyby body to traverse the π radians of true anomaly, which is denoted t_{HR}. This time (t_{HR}) is the same for all transfers for a given departure body location because it is a function only of the flyby body. For a circular body orbit, this time will be half of the orbital period. The total odd $n\pi$ sequence t_S is given by Eqn. (4.5).

$$\sqrt{\frac{\mu_{pr}}{a_B^3}}t_{HR} = E_O - E_I - e_B(\sin E_O - \sin E_I) \tag{4.3}$$

$$\tan\frac{\nu}{2} = \sqrt{\frac{1+e_B}{1-e_B}}\tan\frac{E}{2} \tag{4.4}$$

$$t_S = t_{HR} + 2\pi M_S\sqrt{\frac{a_B^3}{\mu_{pr}}} \tag{4.5}$$

The spacecraft v_∞ at the second encounter is determined by first creating a list of eligible odd $n\pi$ transfers based on the given M_{MAX} and each possible N. The eligible values of N can be explicitly determined from the Lambert algorithm used below, where the upper bound of N is dependent on $M(N_{upper\,bound}(M))$. To reduce the number of cases to evaluate, a user-specified N_{MAX} is used, but a minimum periapse distance with respect to the primary could also be used to determine the upper bound of N. As an example, for $M_{MAX} = 2$ the eligible transfers would be:

$$\text{Example: } \{On\pi\}_{eligible} = \begin{Bmatrix} 0{:}0 & 0{:}1 & \cdots & 0{:}N_{upper\,bound}(0) \\ 1{:}0 & 1{:}1 & \cdots & 1{:}N_{upper\,bound}(1) \\ 2{:}0 & 2{:}1 & \cdots & 2{:}N_{upper\,bound}(2) \end{Bmatrix}$$

Each eligible transfer is used as the input to a Lambert solver with the previously determined t_S and the initial and final body positions, where the Lambert solver determines semimajor axis of the transfer but the transfer plane is still undefined. When $N = 0$ there will be either zero or one solutions and when $N > 0$ there will be either zero or two solutions (short period and long period). The Lambert solutions found populate a new list of Lambert odd $n\pi$ transfers where the sign of N is used to specify which of the two possible Lambert solutions is used.

Define notation:
$N < 0 \rightarrow$ long period transfer with $|N|$ spacecraft revolutions
$N > 0 \rightarrow$ short period transfer with $|N|$ spacecraft revolutions
$N = 0 \rightarrow$ only transfer with zero complete spacecraft revolutions

With the semimajor axis known, equations in [3] are used to map the transfers to pairs of points (above and below plane) on the v_∞ sphere. Not every Lambert solution will map to the v_∞ sphere for a given v_∞, but every Lambert solution that does map to the v_∞ sphere will map to two points, or transfers. All transfers that map to the v_∞ sphere form the list of feasible odd $n\pi$ transfers: $\{On\pi\}$.

Each feasible odd $n\pi$ transfer is now completely defined on the sphere, and so the departure and arrival velocities for each transfer can be found using equations 17 and 22 in [160]. This velocity and the previously determined flyby body velocity at this point together can be used to determine $v_{\infty O}$, completing the information needed to continue the trajectory search after the gap. Each different possible value of M: $\pm\, N$ for the odd $n\pi$ transfer of the odd $n\pi$

sequence corresponds to a particular v_∞ at the end of the sequence, and so care must be taken to keep track of the differences (i.e. different potential branches) when interfacing with the broader trajectory search. The number of branches (p) is one of the outputs of the gap filling algorithm in Table 4.2.

Because the TOF of the $n\pi$ sequences is well defined and the v_∞ after the sequence can be determined, $n\pi$ sequences can be left as gaps and the search can be allowed to proceed into the future without knowing all the details of the $n\pi$ sequence. Then a later check on the feasibility of the $n\pi$ sequence determines feasibility of the entire trajectory. This approach means that the MGA GS needs only to correctly handle this gap and then the $n\pi$ sequences can be readily included in existing software.

4.2.3 Populating the v_∞ Sphere

Once the gap calculator is completed and the trajectory segment following the gap is solved (v_∞ matched), then the pathfinding of the $n\pi$ sequence can proceed. The pathfinding problem is begun by first computing the possible $n\pi$ transfers and mapping them onto the v_∞ sphere as described in [3]. This process populates the sets $\{En\pi\}$ and $\{On\pi\}$ defined above and used in Table 4.1. The resulting mapping of intersections is the elements on the v_∞ sphere, and they may be either even $n\pi$ circles or odd $n\pi$ points as shown in Figure 4.2 and Figure 4.3. The given boundary value $\mathbf{v}_\infty$ vectors are also treated as elements on the v_∞ sphere, but the magnitude of $v_{\infty O}$ is not independent, as discussed above.

Specifically, the sequence time t_S, the semimajor axes of all the possible odd $n\pi$ transfers, and the required $v_{\infty O}$ magnitude(s) are found in the gap calculator. This information and the known $\mathbf{v}_\infty^I$ are used to determine the latitude of the even $n\pi$ circles and the latitude and longitude of the odd $n\pi$ above-plane and below-plane points on the v_∞ sphere [3]. See [3, 121] for more details on the parameters and degrees of freedom of $n\pi$ transfers and the $\mathbf{v}_\infty$ vectors of an $n\pi$ transfer. With the possible $n\pi$ transfers mapped to the v_∞ sphere(s), candidate sequences that navigate the v_∞ sphere can be determined.

4.2.4 Pathfinding: Enumerating Potential Sequences

Once the v_∞ sphere is populated with all $n\pi$ transfers up to some given M_{MAX} (generally M_S), all of the possible combinations of sequences that add up to M_S are determined. The number of candidate sequences can be combinatorially large for large M_S and large v_∞, since a larger v_∞ sphere will in general intersect more resonant spheres and half-rev circles, though a larger v_∞ normally provides less turn angle per gravity-assist flyby. As a result of the many potential combinations, it is useful to filter the available set of transfers $\{n\pi\}$ to exclude ones which are not reachable under any circumstance with the given δ_{MAX}, $\mathbf{v}_\infty^I$, and $\mathbf{v}_\infty^O$.

In order to aid in describing and distinguishing sequences, define B as the total number of half-revolutions that the body passes through to reach M_S so that the sequence angle is $B\pi$.

$$\begin{gathered}\text{If } \{\text{odd } n\pi \text{ sequence}\} \\ \text{Then } \{B = 2M_S + 1\} \\ \text{Else } \{B = 2M_S\}\end{gathered}$$

4.2.4.1 Enumerating Even $n\pi$ Sequences

First generate a list of permutations of all the available transfers so that each permutation has a total body-revolution sequence angle of $B\pi$ (i.e. the flyby body completes a total of B half revolutions). Note that the order of the transfers matters and that the same transfer angle can be repeated in a sequence. The third column of Table 4.3 demonstrates this process for $B \leq 6$ using the v_∞ sphere map in Figure 4.3b. For example, setting $B = 6$ ($M_S = 3$), the possible combinations of even $n\pi$ transfers by transfer angle are [6π], [$4\pi\,2\pi$], [$2\pi\,4\pi$], and [$2\pi\,2\pi\,2\pi$].

When there is more than one even $n\pi$ circle on the v_∞ sphere for a given M (each having a different N), then each of these permutations is further expanded to include all possible permutations of the different candidate even $n\pi$ circles. Again, order is important and repeating the same transfer is allowed. This enumeration is demonstrated in the fourth column of Table 4.3. For example, choosing the transfer angle set [$4\pi\,2\pi$] for $B = 6$, there are two different 4π transfers (2:1 or 2:3) and two different 2π transfers (1:1 or 1:2) from which to choose. The 14^{th} sequence in Table 4.3 shows one of the resulting sequences, which consists of a 2:3 resonance followed by a 1:1 resonance. Note that each of the resonant circles is labeled in Figure 4.3b.

4.2.4.2 Enumerating Odd $n\pi$ Sequences

In order to simplify the combinatorial nature of the problem and to reduce difficulties associated with changing v_∞, it is assumed that the spacecraft performs exactly one odd $n\pi$ transfer in an odd $n\pi$ sequence (a sequence where B is odd). Therefore, the single odd $n\pi$ transfer divides the sequence into two even $n\pi$ subsequences, where subsequences encounter the flyby body on oppo-

Table 4.3: Enumeration of even $n\pi$ sequences for $B \leq 6$ for the v_∞ sphere in Figure 4.2 and labeled in Figure 4.3b

Sequence number	**B**	**Transfer angle permutation**	**Candidate sequences**
1	2	$[2\pi]$	$[\ \mathbf{v}_\infty^I\ 1{:}1\ \mathbf{v}_\infty^O\ ]$
2	2	$[2\pi]$	$[\ \mathbf{v}_\infty^I\ 1{:}2\ \mathbf{v}_\infty^O\ ]$
3	4	$[4\pi]$	$[\ \mathbf{v}_\infty^I\ 2{:}1\ \mathbf{v}_\infty^O\ ]$
4	4	$[4\pi]$	$[\ \mathbf{v}_\infty^I\ 2{:}3\ \mathbf{v}_\infty^O\ ]$
5	4	$[2\pi\ 2\pi]$	$[\ \mathbf{v}_\infty^I\ 1{:}1\ 1{:}1\ \mathbf{v}_\infty^O\ ]$
6	4	$[2\pi\ 2\pi]$	$[\ \mathbf{v}_\infty^I\ 1{:}1\ 1{:}2\ \mathbf{v}_\infty^O\ ]$
7	4	$[2\pi\ 2\pi]$	$[\ \mathbf{v}_\infty^I\ 1{:}2\ 1{:}1\ \mathbf{v}_\infty^O\ ]$
8	6	$[6\pi]$	$[\ \mathbf{v}_\infty^I\ 3{:}1\ \mathbf{v}_\infty^O\ ]$
9	6	$[6\pi]$	$[\ \mathbf{v}_\infty^I\ 3{:}2\ \mathbf{v}_\infty^O\ ]$
10	6	$[6\pi]$	$[\ \mathbf{v}_\infty^I\ 3{:}4\ \mathbf{v}_\infty^O\ ]$
11	6	$[6\pi]$	$[\ \mathbf{v}_\infty^I\ 3{:}5\ \mathbf{v}_\infty^O\ ]$
12	6	$[4\pi\ 2\pi]$	$[\ \mathbf{v}_\infty^I\ 2{:}1\ 1{:}1\ \mathbf{v}_\infty^O\ ]$
13	6	$[4\pi\ 2\pi]$	$[\ \mathbf{v}_\infty^I\ 2{:}1\ 1{:}2\ \mathbf{v}_\infty^O\ ]$
14	6	$[4\pi\ 2\pi]$	$[\ \mathbf{v}_\infty^I\ 2{:}3\ 1{:}1\ \mathbf{v}_\infty^O\ ]$
15	6	$[4\pi\ 2\pi]$	$[\ \mathbf{v}_\infty^I\ 2{:}3\ 1{:}2\ \mathbf{v}_\infty^O\ ]$
16	6	$[2\pi\ 4\pi]$	$[\ \mathbf{v}_\infty^I\ 1{:}1\ 2{:}1\ \mathbf{v}_\infty^O\ ]$
17	6	$[2\pi\ 4\pi]$	$[\ \mathbf{v}_\infty^I\ 1{:}2\ 2{:}1\ \mathbf{v}_\infty^O\ ]$
18	6	$[2\pi\ 4\pi]$	$[\ \mathbf{v}_\infty^I\ 1{:}1\ 2{:}3\ \mathbf{v}_\infty^O\ ]$
19	6	$[2\pi\ 4\pi]$	$[\ \mathbf{v}_\infty^I\ 1{:}2\ 2{:}3\ \mathbf{v}_\infty^O\ ]$
20	6	$[2\pi\ 2\pi\ 2\pi]$	$[\ \mathbf{v}_\infty^I\ 1{:}1\ 1{:}1\ 1{:}1\ \mathbf{v}_\infty^O\ ]$
21	6	$[2\pi\ 2\pi\ 2\pi]$	$[\ \mathbf{v}_\infty^I\ 1{:}2\ 1{:}1\ 1{:}1\ \mathbf{v}_\infty^O\ ]$
22	6	$[2\pi\ 2\pi\ 2\pi]$	$[\ \mathbf{v}_\infty^I\ 1{:}1\ 1{:}2\ 1{:}1\ \mathbf{v}_\infty^O\ ]$
23	6	$[2\pi\ 2\pi\ 2\pi]$	$[\ \mathbf{v}_\infty^I\ 1{:}1\ 1{:}1\ 1{:}2\ \mathbf{v}_\infty^O\ ]$
24	6	$[2\pi\ 2\pi\ 2\pi]$	$[\ \mathbf{v}_\infty^I\ 1{:}2\ 1{:}2\ 1{:}1\ \mathbf{v}_\infty^O\ ]$
25	6	$[2\pi\ 2\pi\ 2\pi]$	$[\ \mathbf{v}_\infty^I\ 1{:}2\ 1{:}1\ 1{:}2\ \mathbf{v}_\infty^O\ ]$
26	6	$[2\pi\ 2\pi\ 2\pi]$	$[\ \mathbf{v}_\infty^I\ 1{:}1\ 1{:}2\ 1{:}2\ \mathbf{v}_\infty^O\ ]$
27	6	$[2\pi\ 2\pi\ 2\pi]$	$[\ \mathbf{v}_\infty^I\ 1{:}2\ 1{:}2\ 1{:}2\ \mathbf{v}_\infty^O\ ]$

site sides of the primary (illustrated in the lower part of Figure 1.2). Either or both of these even $n\pi$ subsequences may be collapsed into a single flyby if the odd $n\pi$ transfer is the first, last, or only transfer in the sequence. The process for enumerating the odd $n\pi$ sequences follows. This process described below is illustrated in Tables 4.4-4.5 for the v_∞ sphere given in Figure 4.3b, with sequences 17 and 18 corresponding to the specific example mentioned below.

1. List the possible odd $n\pi$ transfers on the inbound v_∞ sphere with $M_i \leq M_S$, taking into account that each one has an above-plane and a below-plane solution and that there can be two solutions for a given M_i:N_i when $N_i > 0$. For example, using the v_∞ sphere in Figure 4.3b with $M_S = 2$ $(B = 5)$, the possible odd $n\pi$ transfers are $\{0{:}0a, 0{:}0b, 1{:}{-1}a, 1{:}{-1}b, 2{:}2a, 2{:}2b\}$.

2. Split the given M_S into subsequences with each odd $n\pi$ transfer placed in every possible combination. For example, using $B = 5$ and Figure 4.3b with the below plane, odd $n\pi$ transfer angle of 3π (resonance 1:-1), the possible ways to split the 5π transfer angle with the $3\pi b$ transfer are $[3\pi b\,\{2\pi\}]$ or $[\{2\pi\}\,3\pi b]$ where $\{2\pi\}$ is an even $n\pi$ subsequence with total transfer angle of 2π

3. Enumerate any even $n\pi$ subsequences that may result from this placement, accounting from the resulting v_∞ change. Continuing the example given above, the transfer angle permutation $[3\pi b\,\{2\pi\}]$ can yield the sequences $[\mathbf{v}_\infty^I\ 1{:}{-1}b\ 1{:}1\ \mathbf{v}_\infty^O\ ]$ or $[\mathbf{v}_\infty^I\ 1{:}{-1}b\ 1{:}2\ \mathbf{v}_\infty^O\ ]$.

In the sequences given in Tables 4.4-4.5, the odd $n\pi$ transfers always have an "a" or "b" to specify whether they are above or below plane, and the even $n\pi$ transfers are described by only the resonance numbers with no

Table 4.4: Enumeration of odd $n\pi$ sequences: one odd $n\pi$ transfer, $B \leq 5$, v_∞ sphere in Figure 4.3b. Enumeration of sequences continued in Table 4.5

Seq.	**B**	**Odd** $n\pi$	**Odd** $n\pi$ **xfer**	t_S	$v_{\infty O}$,	**Candidate**
num.		**transfer**	**placement**	**TU**	**LU/TU**	**sequences**
1	1	[1πa]	[1πa]	2.795184	0.552632	[$\mathbf{v}_\infty^I$ 0:0a $\mathbf{v}_\infty^O$]
2	1	[1πb]	[1πb]	2.795184	0.552632	[$\mathbf{v}_\infty^I$ 0:0b $\mathbf{v}_\infty^O$]
3	3	[3πa]	[3πa]	9.078369	0.549842	[$\mathbf{v}_\infty^I$ 1:-1a $\mathbf{v}_\infty^O$]
4	3	[3πb]	[3πb]	9.078369	0.549842	[$\mathbf{v}_\infty^I$ 1:-1b $\mathbf{v}_\infty^O$]
5	3	[1πa]	[1πa 2π*]	9.078369	0.552632	[$\mathbf{v}_\infty^I$ 0:0a 1:1* $\mathbf{v}_\infty^O$]
6	3	[1πa]	[1πa 2π*]	9.078369	0.552632	[$\mathbf{v}_\infty^I$ 0:0a 1:2* $\mathbf{v}_\infty^O$]
7	3	[1πb]	[1πb 2π*]	9.078369	0.552632	[$\mathbf{v}_\infty^I$ 0:0b 1:1* $\mathbf{v}_\infty^O$]
8	3	[1πb]	[1πb 2π*]	9.078369	0.552632	[$\mathbf{v}_\infty^I$ 0:0b 1:2* $\mathbf{v}_\infty^O$]
9	3	[1πa]	[2π 1πa]	9.078369	0.552632	[$\mathbf{v}_\infty^I$ 1:1 0:0a $\mathbf{v}_\infty^O$]
10	3	[1πa]	[2π 1πa]	9.078369	0.552632	[$\mathbf{v}_\infty^I$ 1:2 0:0a $\mathbf{v}_\infty^O$]
11	3	[1πb]	[2π 1πb]	9.078369	0.552632	[$\mathbf{v}_\infty^I$ 1:1 0:0b $\mathbf{v}_\infty^O$]
12	3	[1πb]	[2π 1πb]	9.078369	0.552632	[$\mathbf{v}_\infty^I$ 1:2 0:0b $\mathbf{v}_\infty^O$]
13	5	[5πa]	[5πa]	15.361554	0.552631	[$\mathbf{v}_\infty^I$ 2:2a $\mathbf{v}_\infty^O$]
14	5	[5πb]	[5πb]	15.361554	0.552631	[$\mathbf{v}_\infty^I$ 2:2b $\mathbf{v}_\infty^O$]
15	5	[3πa]	[3πa 2π**]	15.361554	0.549842	[$\mathbf{v}_\infty^I$ 1:-1a 1:1** $\mathbf{v}_\infty^O$]
16	5	[3πa]	[3πa 2π**]	15.361554	0.549842	[$\mathbf{v}_\infty^I$ 1:-1a 1:2** $\mathbf{v}_\infty^O$]
17	5	[3πb]	[3πb 2π**]	15.361554	0.549842	[$\mathbf{v}_\infty^I$ 1:-1b 1:1** $\mathbf{v}_\infty^O$]
18	5	[3πb]	[3πb 2π**]	15.361554	0.549842	[$\mathbf{v}_\infty^I$ 1:-1b 1:2** $\mathbf{v}_\infty^O$]
19	5	[3πa]	[2π 3πa]	15.361554	0.549842	[$\mathbf{v}_\infty^I$ 1:1 1:-1a $\mathbf{v}_\infty^O$]
20	5	[3πa]	[2π 3πa]	15.361554	0.549842	[$\mathbf{v}_\infty^I$ 1:2 1:-1a $\mathbf{v}_\infty^O$]
21	5	[3πb]	[2π 3πb]	15.361554	0.549842	[$\mathbf{v}_\infty^I$ 1:1 1:-1b $\mathbf{v}_\infty^O$]
22	5	[3πb]	[2π 3πb]	15.361554	0.549842	[$\mathbf{v}_\infty^I$ 1:2 1:-1b $\mathbf{v}_\infty^O$]
23	5	[1πa]	[1πa 4π*]	15.361554	0.552632	[$\mathbf{v}_\infty^I$ 0:0a 2:1* $\mathbf{v}_\infty^O$]
24	5	[1πa]	[1πa 4π*]	15.361554	0.552632	[$\mathbf{v}_\infty^I$ 0:0a 2:3* $\mathbf{v}_\infty^O$]
25	5	[1πa]	[1πa 4π*]	15.361554	0.552632	[$\mathbf{v}_\infty^I$ 0:0a 1:1* 1:1* $\mathbf{v}_\infty^O$]
26	5	[1πa]	[1πa 4π*]	15.361554	0.552632	[$\mathbf{v}_\infty^I$ 0:0a 1:2* 1:1* $\mathbf{v}_\infty^O$]
27	5	[1πa]	[1πa 4π*]	15.361554	0.552632	[$\mathbf{v}_\infty^I$ 0:0a 1:1* 1:2* $\mathbf{v}_\infty^O$]
28	5	[1πa]	[1πa 4π*]	15.361554	0.552632	[$\mathbf{v}_\infty^I$ 0:0a 1:2* 1:2* $\mathbf{v}_\infty^O$]
29	5	[1πb]	[1πb 4π*]	15.361554	0.552632	[$\mathbf{v}_\infty^I$ 0:0b 2:1* $\mathbf{v}_\infty^O$]
30	5	[1πb]	[1πb 4π*]	15.361554	0.552632	[$\mathbf{v}_\infty^I$ 0:0b 2:3* $\mathbf{v}_\infty^O$]
31	5	[1πb]	[1πb 4π*]	15.361554	0.552632	[$\mathbf{v}_\infty^I$ 0:0b 1:1* 1:1* $\mathbf{v}_\infty^O$]
32	5	[1πb]	[1πb 4π*]	15.361554	0.552632	[$\mathbf{v}_\infty^I$ 0:0b 1:2* 1:1* $\mathbf{v}_\infty^O$]
33	5	[1πb]	[1πb 4π*]	15.361554	0.552632	[$\mathbf{v}_\infty^I$ 0:0b 1:1* 1:2* $\mathbf{v}_\infty^O$]
34	5	[1πb]	[1πb 4π*]	15.361554	0.552632	[$\mathbf{v}_\infty^I$ 0:0b 1:2* 1:2* $\mathbf{v}_\infty^O$]

* Evaluated on a different v_∞ sphere with $v_\infty = 0.552632$

** Evaluated on a different v_∞ sphere with $v_\infty = 0.549842$

Table 4.5: Continuation of Table 4.4 giving enumeration of odd $n\pi$ sequences: one odd $n\pi$ transfer, $B \leq 5$, v_∞ sphere in Figure 4.3b

Seq.	**B**	**Odd $n\pi$**	**Odd $n\pi$ xfer**	t_S	$v_{\infty O}$**,**	**Candidate**
num.		**transfer**	**placement**	**TU**	**LU/TU**	**sequences**
35	5	[1πa]	[2π 1πa 2π*]	15.361554	0.552632	[$\mathbf{v}_\infty^I$ 1:1 0:0a 1:1* $\mathbf{v}_\infty^O$]
36	5	[1πa]	[2π 1πa 2π*]	15.361554	0.552632	[$\mathbf{v}_\infty^I$ 1:2 0:0a 1:1* $\mathbf{v}_\infty^O$]
37	5	[1πa]	[2π 1πa 2π*]	15.361554	0.552632	[$\mathbf{v}_\infty^I$ 1:1 0:0a 1:2* $\mathbf{v}_\infty^O$]
38	5	[1πa]	[2π 1πa 2π*]	15.361554	0.552632	[$\mathbf{v}_\infty^I$ 1:2 0:0a 1:2* $\mathbf{v}_\infty^O$]
39	5	[1πb]	[2π 1πb 2π*]	15.361554	0.552632	[$\mathbf{v}_\infty^I$ 1:1 0:0b 1:1* $\mathbf{v}_\infty^O$]
40	5	[1πb]	[2π 1πb 2π*]	15.361554	0.552632	[$\mathbf{v}_\infty^I$ 1:2 0:0b 1:1* $\mathbf{v}_\infty^O$]
41	5	[1πb]	[2π 1πb 2π*]	15.361554	0.552632	[$\mathbf{v}_\infty^I$ 1:1 0:0b 1:2* $\mathbf{v}_\infty^O$]
42	5	[1πb]	[2π 1πb 2π*]	15.361554	0.552632	[$\mathbf{v}_\infty^I$ 1:2 0:0b 1:2* $\mathbf{v}_\infty^O$]
43	5	[1πa]	[4π 1πa]	15.361554	0.552632	[$\mathbf{v}_\infty^I$ 2:1 0:0a $\mathbf{v}_\infty^O$]
44	5	[1πa]	[4π 1πa]	15.361554	0.552632	[$\mathbf{v}_\infty^I$ 2:3 0:0a $\mathbf{v}_\infty^O$]
45	5	[1πa]	[4π 1πa]	15.361554	0.552632	[$\mathbf{v}_\infty^I$ 1:1 1:1 0:0a $\mathbf{v}_\infty^O$]
46	5	[1πa]	[4π 1πa]	15.361554	0.552632	[$\mathbf{v}_\infty^I$ 1:2 1:1 0:0a $\mathbf{v}_\infty^O$]
47	5	[1πa]	[4π 1πa]	15.361554	0.552632	[$\mathbf{v}_\infty^I$ 1:1 1:2 0:0a $\mathbf{v}_\infty^O$]
48	5	[1πa]	[4π 1πa]	15.361554	0.552632	[$\mathbf{v}_\infty^I$ 1:2 1:2 0:0a $\mathbf{v}_\infty^O$]
49	5	[1πb]	[4π 1πb]	15.361554	0.552632	[$\mathbf{v}_\infty^I$ 2:1 0:0b $\mathbf{v}_\infty^O$]
50	5	[1πb]	[4π 1πb]	15.361554	0.552632	[$\mathbf{v}_\infty^I$ 2:3 0:0b $\mathbf{v}_\infty^O$]
51	5	[1πb]	[4π 1πb]	15.361554	0.552632	[$\mathbf{v}_\infty^I$ 1:1 1:1 0:0b $\mathbf{v}_\infty^O$]
52	5	[1πb]	[4π 1πb]	15.361554	0.552632	[$\mathbf{v}_\infty^I$ 1:2 1:1 0:0b $\mathbf{v}_\infty^O$]
53	5	[1πb]	[4π 1πb]	15.361554	0.552632	[$\mathbf{v}_\infty^I$ 1:1 1:2 0:0b $\mathbf{v}_\infty^O$]
54	5	[1πb]	[4π 1πb]	15.361554	0.552632	[$\mathbf{v}_\infty^I$ 1:2 1:2 0:0b $\mathbf{v}_\infty^O$]

* Evaluated on a different v_∞ sphere with $v_\infty = 0.552632$

letters. All of the sequences have the same t_{HR} because this time is determined by propagating the body orbit and all sequences use the same flyby body initial and final conditions.

Overall, this example in Tables 4.4-4.5 produces 54 candidate sequences to evaluate, but the number of possible sequences to evaluate can be reduced by applying filters that consider reachability or some heuristic methods to choose between the above and below plane solutions without evaluating both.

Since the flyby body orbit is non-circular, any even $n\pi$ subsequences following an odd $n\pi$ transfer need to be mapped on a different v_∞ sphere based on the new $v_{\infty O}$ as indicated in the table footnotes. Since the $[1\pi]$ and the $[3\pi]$ transfers each change the v_∞ by different amounts, each one requires a different v_∞ sphere for even $n\pi$ subsequences following them. The $[5\pi]$ transfers also produce a small change in $v_{\infty O}$ but there are no even $n\pi$ subsequences following the $[5\pi]$ transfers, so they do not require an additional v_∞ sphere. For the example given in Tables 4.4-4.5, the change to different v_∞ spheres is assumed to not affect available even $n\pi$ transfers to choose from. For large changes in v_∞ or when the v_∞ sphere is densely populated, the two v_∞ spheres used in an odd $n\pi$ sequence may not have the same transfers available for enumerating candidate sequences.

Therefore, when an odd $n\pi$ transfer changes the spacecraft v_∞, it changes the sphere on which to navigate successive transfers in the sequence. In addition to the change from the magnitude of v_∞, the change in body flight path angle shifts the location of the odd $n\pi$ circles. As a result, a total of three v_∞ spheres would be required to enumerate candidate sequences for the same conditions as in Tables 4.4-4.5, although no more than two v_∞ spheres would be needed for evaluating any one sequence.

By splitting the odd $n\pi$ transfer sequence into two even $n\pi$ sequences patched with a single odd $n\pi$ transfer, the boundary value problem is well defined allowing for rapid inclusion in a broad search. The limitation of considering just one odd $n\pi$ transfer is not very restrictive as each leg in the overall sequence can also include up to one odd $n\pi$ transfer. The restriction essentially limits back to back odd $n\pi$ transfers to the same body. Although it would be a rare find, back to back odd $n\pi$ transfers could provide a change in v_∞ at the original location if the back to back transfers utilized different M or N.

4.2.5 Pathsolving: Choosing the Degrees of Freedom

The process of mapping the v_∞ sphere and enumerating the candidate sequences satisfies all of the constraints given in Table 4.1 except the turn angle constraints. Although enumerating the sequences chooses the number of elements and the elements for any one candidate sequence, the DOFs of the even $n\pi$ transfers are not yet determined in the candidate sequence enumeration. What remains is a pathsolving problem for each candidate sequence to determine its feasibility and/or optimality. There may exist many (infinitely many) solved paths (choices for DOFs) for a given sequence.

The DOF for even $n\pi$ transfers is longitude on the v_∞ sphere (λ_i), which is continuous on $[0, 2\pi)$ where zero and 2π are equivalent. There are many ways to choose the DOFs, based on science objectives or other mission constraints. One approach is a minimax optimization on δ_i that would minimize the maximum turn angle required [126]. For the purposes of an initial design space search, an exact optimum may not be necessary, and instead the goal is initially to choose the DOFs such that a sequence is feasible with respect to the flyby constraints in Table 4.1: $\delta_i \leq \delta_{MAX}$ and $\delta_i \geq \delta_{MIN}$ for all $i = 1 \ldots j$

Determining feasibility is equivalent to choosing DOFs such that the sequence is feasible, and considering infeasible any sequence for which the selection process fails to give a feasible turn angle. There may be many choices of DOF that make a sequence feasible, but in this study the goal is just to determine feasibility and not initially to pick optimum values for the DOFs. Because all transfers are ballistic and further optimization would take time, the algorithm only determines one set of feasible DOFs for a given sequence and moves on to the next sequence when one feasible set of DOFs is found. Further computational speedup is possible if only one sequence is required for a given set of boundary conditions. For post processing different feasible sequences for science or other purposes, the algorithm can stop after finding the q^{th} feasible sequence instead, where q is a given integer less than or equal to the number of feasible sequences. The resulting set of DOFs does not necessarily minimize any performance index like a traditional optimization problem, but an optimization could be applied instead.

In order to choose DOFs and determine sequence feasibility, the value for each successive λ_i is chosen to reach the next element and also minimize the remaining distance on the v_∞ sphere to the next half-rev point (if it exists) or the final condition ($\mathbf{v}_\infty^O$). If this terminal point cannot be reached, then a step of maximum size is taken in its direction. This algorithm may lead to one or more of the flybys towards the end of the sequence violating the $\delta_i < \delta_{MIN}$ when two successive elements are on the same resonant circle. Violating this constraint is an undesirable condition because very small turn angle flybys are not possible in an integrated trajectory with a massive body (without being too far from the body, in essence violating the patched conic position assumption). This problem is rectified by postprocessing complete sequences solutions that are feasible with respect to δ_{MAX}. By resolving the problem in reverse (starting at $\mathbf{v}_\infty^O$ and working backwards) and requiring δ_{MIN} steps, some or all of the earlier steps will be relaxed from their $\delta_i = \delta_{MAX}$ boundaries: therein making the entire solution sequence feasible. However, the basic premise of first taking the maximum step possible leads to rapid filtering of infeasible solutions, and accordingly leads to implementation speedups.

Implementation speedups are also possible by filtering out sequences that will always be infeasible: A sequence will always be infeasible if the minimum angle between any of its successive elements on the v_∞ sphere (e.g. given $\mathbf{v}_\infty$ vectors, even $n\pi$ circles, and odd $n\pi$ points) is greater than δ_{MAX}. This minimum flyby altitude feasibility requirement allows for simple checks that reduce the number of calculations in practical implementation, especially when δ_{MAX} is small (due to large minimum flyby altitudes, small gravitational parameter, or high v_∞). For example, if at least one of the $\mathbf{v}_\infty$ vectors has no other elements on the v_∞ sphere within a cone of half-angle δ_{MAX}, then all of the $n\pi$ transfers are unreachable, and that entire sequence can be discarded.

4.2.6 Algorithm summary

The implementation process described in the preceding sections is summarized as a pseudocode algorithm in Tables 4.6-4.7. Many of the special considerations for odd $n\pi$ sequences are not required if the flyby body orbit is circular, but the algorithm still handles this circular subset of the elliptic orbit case.

Table 4.6: Algorithm outline for the $n\pi$ sequence pathfinding problem on the v_∞ sphere. Algorithm continued in Table 4.7

	INPUTS		OUTPUTS
Problem:	$\mathbf{v}_\infty^I$, $\mathbf{v}_\infty^O$ *, M_S, $N_{S,MAX}$, ν_{BI}	Set of xfers:	$\epsilon_i \in \{n\pi\}\ \forall\ i = 1, 2, \cdots, j-1$
System:	μ_{Pr}, a_B, e_B, i_B, Ω_B, ω_B	DOFs:	$\lambda_i\ \forall\ \epsilon_i \in \{En\pi\}$
Constraint:	M_{MAX}, N_{MAX}, δ_{MAX}, δ_{MIN}	# flybys:	j

Set Up Boundary Value Problem

```
-1. GAP CALCULATOR using algorithm in Table 4.2: find number of branches and $t_S$,
    $v_{\infty O}$ for each branch
0. FOR each branch
   0.1. SOLVE completely the trajectory segment following the $n\pi$ sequence
      0.1.1. Satisfy $v_\infty$ matching constraint by root solving to meet the $v_{\infty O}$ for the branch
   0.2. IF no feasible solutions for branch THEN remove branch from consideration
```

IF even $n\pi$ sequence (only one branch): **Even $n\pi$ Sequence Calculator**

```
1. POPULATE the set of even $n\pi$ elements $\{En\pi\}$ using the given body orbit and $v_{\infty I}$ as
   described in [3]
   1.1. Use the transfer constraints to limit the elements mapped to the $v_\infty$ sphere:
      1.1.1. $M_{MAX} \geq M_i$, $N_{MAX} \geq N_i$
2. FILTER ELEMENTS, removing any which cannot be reached by a step of $\delta_{MAX}$ from
   any other elements
   2.1. FOR each element $\epsilon_i$
      2.1.1. IF min(angle between $\epsilon_i$ and $\epsilon_j\ \forall\ i \neq j) > \delta_{MAX}$ THEN remove $\epsilon_i$ from $\{En\pi\}$
3. ENUMERATE candidate sequences by all permutations of remaining elements
   3.1. Use the sequence constraints to bound the set of candidate sequences:
      3.1.1. $M_S = \sum_{i=1}^{j-1} M_i$, $N_{S,MAX} \geq \sum_{i=1}^{j-1} N_i$
4. FILTER SEQUENCES, removing any which cannot be feasible due to the choice/order
   of elements
   4.1. FOR each candidate sequence
      4.1.1. FOR each element in the sequence except $\mathbf{v}_\infty^O$
         4.1.1.1. IF min(angle between $\epsilon_i$ and $\epsilon_{i+1}) > \delta_{MAX}$ THEN discard candidate sequence and move to next candidate sequence
5. CHOOSE DEGREES OF FREEDOM ($\lambda$ for each even $n\pi$ return) in order to make
   feasible sequences or reject sequences as infeasible
   5.1. FOR each candidate sequence
      5.1.1. FOR each element
         5.1.1.1. $\lambda_i^*$ is nearest point to $\mathbf{v}_\infty^O$ on $\epsilon_i$
         5.1.1.2. IF required $\delta_i$ to reach $\lambda_i^* \leq \delta_{MAX}$ THEN $\lambda_i = \lambda_i^*$
         5.1.1.3. ELSE $\lambda_i$ is a $\delta_{MAX}$ step from $\epsilon_{i-1}$ towards $\lambda_i^*$
      5.1.2. POSTPROCESS starting from the last element
         5.1.2.1. WHILE $\delta_i < \delta_{MIN}$ DO $\{\delta_i = \delta_{MIN}\}$
         5.1.2.2. Recalculate last $\delta_i \geq \delta_{MIN}$ to match the chronologically first of the postprocessed $\epsilon_i$
```

* $|\mathbf{v_{\infty O}}| = v_{\infty O}$ is not an independent input, but depends on the transfers in the sequence, as explained in the text

Table 4.7: Continuation of algorithm outline for the $n\pi$ sequence pathfinding problem on the v_∞ sphere. Algorithm begins in Table 4.6

```
...
ELSE IF odd $n\pi$ sequence: **Odd $n\pi$ Sequence Calculator**
**1.** POPULATE the sets $\{En\pi\}$ and $\{On\pi\}$ using the given body orbit and $v_{\infty I}$
  1.1. Use $\{On\pi\}$ stored from the Gap Calculator
  1.2. $\{En\pi\}$ calculated using equations in [3, 121]
  1.3. Use transfer constraints to limit the elements mapped to the $v_\infty$ sphere (as with even
       $n\pi$ sequence)
**2.** FILTER ELEMENTS, removing any which cannot be reached by a step of $\delta_{MAX}$ from
   any other elements
  2.1. Same filter process as for even $n\pi$ transfers
**3.** FOR each branch with solutions
  3.1. POPULATE the $v_\infty$ sphere for $v_{\infty O}$ for the branch with the set of even $n\pi$ elements
       $\{En\pi\}$ using the given body orbit and $v_{\infty O}$ for the particular branch, as described
       in [3, 121]
  3.2. ENUMERATE THE ODD $n\pi$ OPTIONS using the odd $n\pi$ transfer that corresponds
       to the branch
    3.2.1. Enumerate options such that each option has a distinct even $n\pi$ subsequence
           following the odd $n\pi$ transfer, including a single gravity-assist flyby as one subsequence
  3.3. FOR each odd option
  3.4. SOLVE PRECEDING EVEN $n\pi$ SUBSEQUENCE, if it exists, using the $v_\infty$ sphere
       for $v_{\infty I}$
    3.4.1. Follow even $n\pi$ sequence algorithm starting at step 2, with odd $n\pi$ point at
           terminal condition
  3.5. SOLVE FOLLOWING EVEN $n\pi$ SUBSEQUENCE if it exists, using the $v_\infty$ sphere
       for $v_{\infty O}$ which corresponds to the odd $n\pi$ transfer in the sequence (the branch)
    3.5.1. Follow even $n\pi$ sequence algorithm starting at step 2
```

* $|\mathbf{v_{\infty O}}| = v_{\infty O}$ is not an independent input, but depends on the transfers in the sequence, as explained in the text

4.3 Some Example Trajectory Searches

Starting from the same flyby body condition as that used to generate the v_∞ sphere shown in Figure 4.3b, several example cases are presented. Note that the problem is posed in normalized quantities such that $\mu_{Pr} = 1$. The search constraints for the four examples are given in Table 4.8. A summary of the number of solutions and run times are given in Table 4.9. The odd $n\pi$ sequences are further broken down by which odd $n\pi$ transfer they use. Run times are calculated on a desktop machine with an Intel Xeon W3550 processor, using one processor core at 3.07 GHz for the program performing the search. The routine itself is written in Fortran and compiled with Intel Fortran Composer XE 2011 using the O2 optimization flag. In Examples A, C, E, G, and I, the combination of gravitational parameter and periapse radius allows for large turn angles, so the turn angle constraint is not binding. Examples B, D, F, H, and J have more realistic turn angle constraints.

Table 4.8: Constraints for the sequence search examples

Ex.	ν_{BI}	M_S	$N_{S,MAX}$	$r_{P,MIN}$	$\mathbf{v}_\infty^I$			$\mathbf{v}_\infty^O$		
					$v_{\infty I}$	Lon	Lat	$v^*_{\infty O}$	Lon	Lat
A	$\pi/6$	6	10	1.0e-8	0.18	$-\pi/10$	$-\pi/4$	0.18	$3\pi/2$	$\pi/3$
B	$\pi/6$	6	10	1.0e-7	0.18	$-\pi/10$	$-\pi/4$	0.18	$3\pi/2$	$\pi/3$
C	$\pi/6$	6	10	1.0e-8	0.25	$-\pi/10$	$-\pi/4$	0.25	$3\pi/2$	$\pi/3$
D	$\pi/6$	6	10	1.0e-7	0.25	$-\pi/10$	$-\pi/4$	0.25	$3\pi/2$	$\pi/3$
E	$2\pi/3$	3	6	1.0e-8	0.18	$\pi/4$	$\pi/6$	0.25	$-3\pi/4$	$\pi/3$
F	$2\pi/3$	3	6	1.0e-7	0.18	$\pi/4$	$\pi/6$	0.25	$-3\pi/4$	$\pi/3$
G	$2\pi/3$	3	6	1.0e-8	0.25	$\pi/4$	$\pi/6$	0.25	$-3\pi/4$	$\pi/3$
H	$2\pi/3$	3	6	1.0e-7	0.25	$\pi/4$	$\pi/6$	0.25	$-3\pi/4$	$\pi/3$
I	$\pi/3$	10	10	1.0e-8	0.25	$\pi/8$	$\pi/10$	0.25	$-3\pi/4$	$-\pi/3$
J	$\pi/3$	10	10	1.0e-7	0.25	$\pi/8$	$\pi/10$	0.25	$-3\pi/4$	$-\pi/3$

*The magnitude of $v_{\infty O}$ can be changed by odd $n\pi$ transfers

Because the evaluation of all sequences can be computationally expensive, the algorithm can be run to just find a single solution for any set of

Table 4.9: The number of $n\pi$ sequences with solutions in the example search, with odd $n\pi$ sequences distinguished by which particular odd $n\pi$ transfer is used

Ex.	Total	Even $n\pi$	Odd $n\pi$ solns.		Even $n\pi$ run time, s		Odd $n\pi$ run time, s	
	solns.	solns.	0:0 (1π)	Other	1^{st} soln.	All solns.	1^{st} soln.	All solns.
A	331	103	222	6	3.17e-5	2.63e-3	3.28e-4	3.28e-2
B	190	51	139	0	3.43e-5	2.63e-3	2.34e-4	3.12e-2
C	1350	166	1120	64	3.59e-5	8.31e-3	4.84e-4	4.79e-1
D	376	28	348	0	3.75e-5	5.77e-3	4.68e-4	2.71e-1
E	47	7	40	0	1.09e-5	4.37e-5	5.46e-5	1.61e-3
F	23	4	19	0	1.09e-5	3.58e-5	5.62e-5	1.23e-3
G	56	8	48	0	1.09e-5	5.77e-5	4.68e-5	1.95e-3
H	5	1	4	0	1.09e-5	3.57e-5	6.24e-5	1.22e-3
I	9508	3774	5734	6	7.33e-5	4.87e0	7.05e-4	1.39e1
J	5851	1835	4016	6	7.64e-5	2.95e0	7.18e-4	1.44e1

constraints. The first sequence found may not be the preferred solution for the given constraints (for some science or flyby geometry reasons) but just finding the first solution allows the existence of a sequence from an orbit mechanics only perspective to be quickly verified. The different run times for just finding the first solution as compared to finding all solutions are also shown in Table 4.9. When finding just one solution, the run time of the algorithm is further improved by ordering the most likely sequences first. Sequences that have a $B = 1$ transfer or mostly $B = 2$ transfers are good candidates for evaluating first because these transfers have short flight times, allowing more flybys, and so these trajectories are not as bound by turn angle constraints.

Figure 4.5 through Figure 4.8 provide visuals of some of the sequences calculated in the four examples defined in Table 4.8. Each of these figures shows four views of the trajectory and four views of the v_∞ sphere, a view along each axis and a three-dimensional view. In all plots, the spacecraft initial condition is a diamond and the spacecraft final condition is an asterisk

(*). In the trajectory plots, the body orbit is a dotted line and the spacecraft orbit is a solid line. On the v_∞ spheres, the even $n\pi$ elements are circles around the surface of the sphere and odd $n\pi$ elements are x's and o's. The spacecraft $\mathbf{v}_\infty$ states are given by black shapes connected by a dotted black line. The initial and final conditions are a diamond and an asterisk and intermediate flyby $\mathbf{v}_\infty$ vectors are squares.

Figure 4.5 shows a sequence that uses only 1:1 resonances, with successive flybys moving along the 1:1 circle until reaching the closest point on the circle to $\mathbf{v}_\infty^O$. As described previously, the flybys along this resonant circle have been distributed so that none of them has a turn angle of zero. The resulting trajectory is a series of orbits of constant energy but different inclinations.

Performing the same search, but allowing for the inclusion of one odd $n\pi$ transfer and updating the sequence TOF with Eqn. (4.5), yields many more solutions, as indicated in Table 4.9. Figure 4.6 shows an example sequence from this search. There are actually two v_∞ spheres: one before the odd $n\pi$ transfer and one after it. The sequence uses the 0:0 resonance and the resulting change in v_∞ is very noticeable. Each of the two v_∞ spheres in each figure is aligned with its own local planet velocity vector, so the two spheres have a relative rotation about the Y-axis.

Figure 4.7 shows an odd $n\pi$ sequence from Example B, which enforces a smaller turn angle (larger periapse radius) than Example A. This particular sequence also demonstrates the use of an even $n\pi$ transfer on the second (smaller) v_∞ sphere after the odd $n\pi$ transfer. A sequence from Example C is displayed in Figure 4.8, showing a relatively simple even $n\pi$ sequence. These four different sequences demonstrate pictorially that the $n\pi$ sequence algorithm can find many different types of trajectories, essentially providing

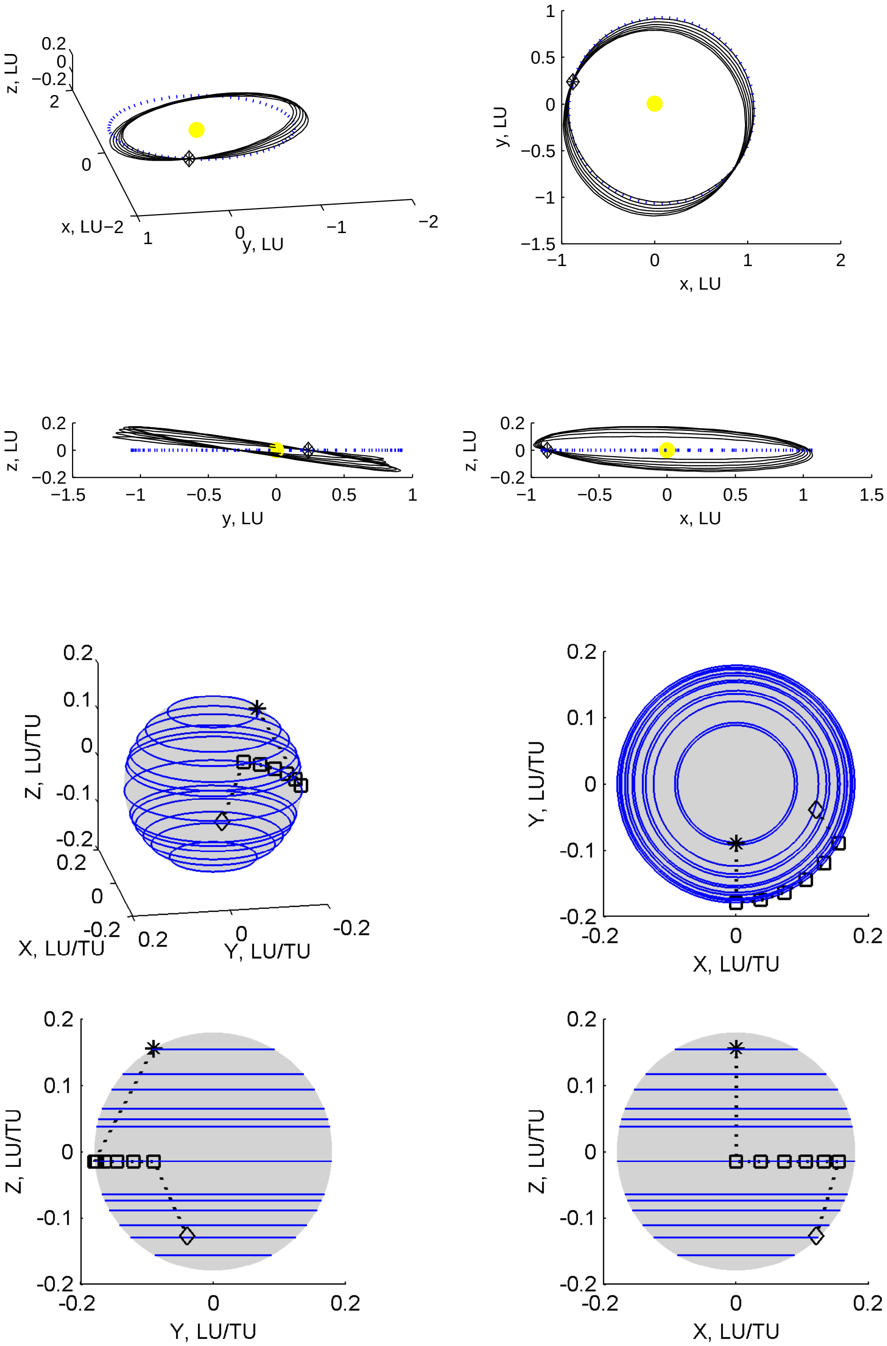


Figure 4.5: Four views of trajectory and of v_∞ sphere for an even $n\pi$ sequence from Example A that uses a sequence of many 1:1 transfers (refer to the text for the definitions of symbols)

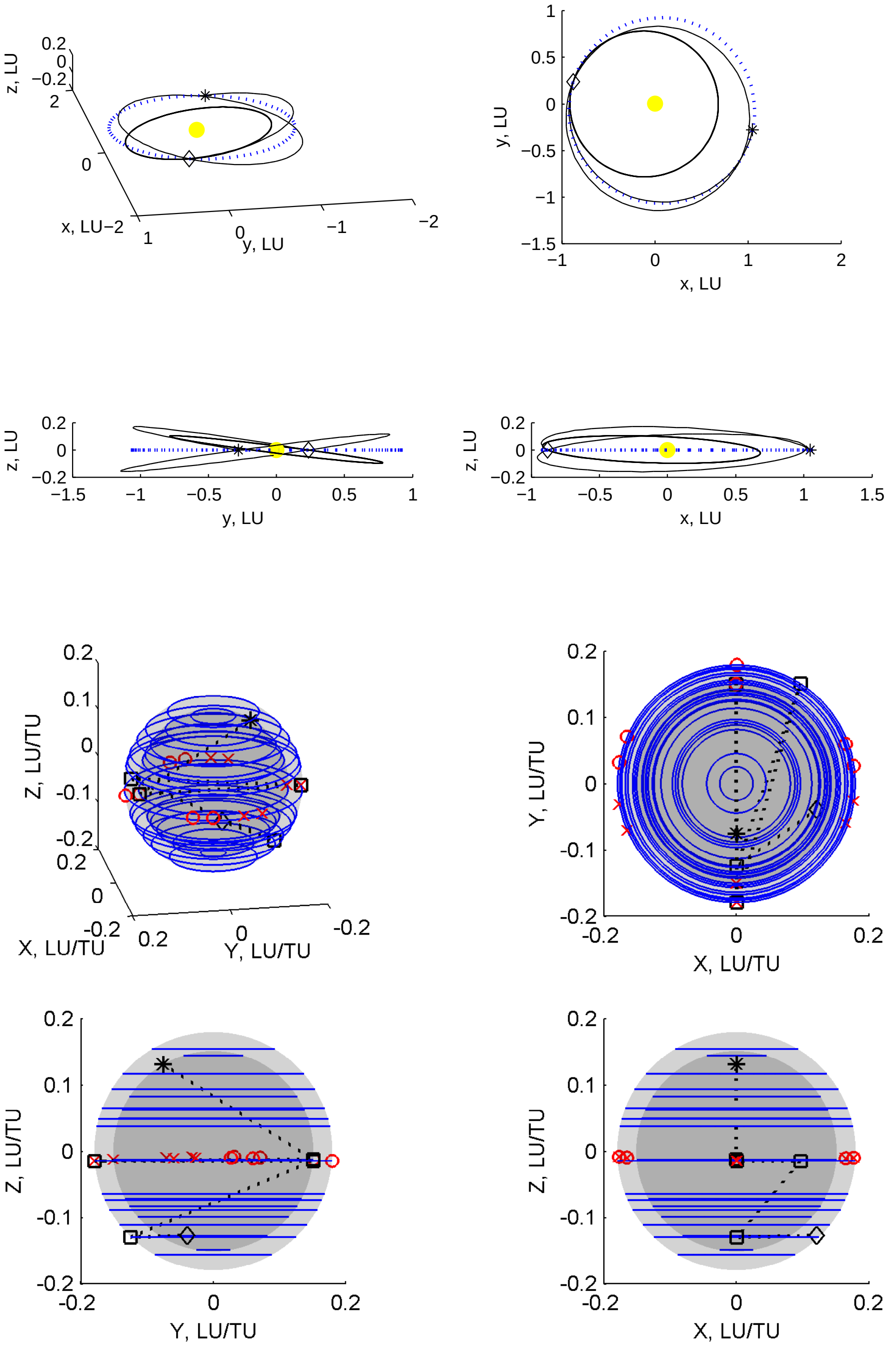


Figure 4.6: Four views of trajectory and of v_∞ sphere for an odd $n\pi$ sequence from Example A that that demonstrates the change in v_∞ caused by the odd $n\pi$ transfer (refer to the text for the definitions of symbols)

many options for loiter orbits which can vastly increase the design space for the ballistic tour problem.

4.4 Mapping bodies with flybys using $n\pi$ sequences: algorithm for the GTOC6 problem

The pathsolving method presented above aims to quickly evaluate sequence feasibility with the box constraints of flyby altitude, but other methods for pathsolving may be more suitable for a particular problem. As an example, the GTOC6 problem [65] is considered, where the goal is to map the four Galilean moons using flybys. In this problem, mapping is done by placing periapse over elements on a grid of each moon, with each grid element mapped contributing a specified amount to the score. The grid, shown in Figure 4.9, is patterned after a soccer ball. Since each moon is tidally locked with Jupiter, the surface grid maps directly to the v_∞ sphere aligned with the body velocity with no time-dependent rotation. Because of time constraints on implementation, only even $n\pi$ sequences are considered but odd $n\pi$ sequences could potentially contribute in an important way.

Once the v_∞ sphere is mapped with the possible $n\pi$ sequences, pathfinding and pathsolving is guided by performance indices instead of by simply attaining feasibility. The $n\pi$ sequences can be long – 31 flybys for the final Io sequence used, for example – so an enumeration of paths becomes combinatorially large. Also, reaching $\mathbf{v}_\infty^O$ is generally possible and not the only goal for such a long sequence. Instead, pathfinding and pathsolving are accomplished simultaneously and sequentially by the following process:

1. Start at $\mathbf{v}_\infty^I$

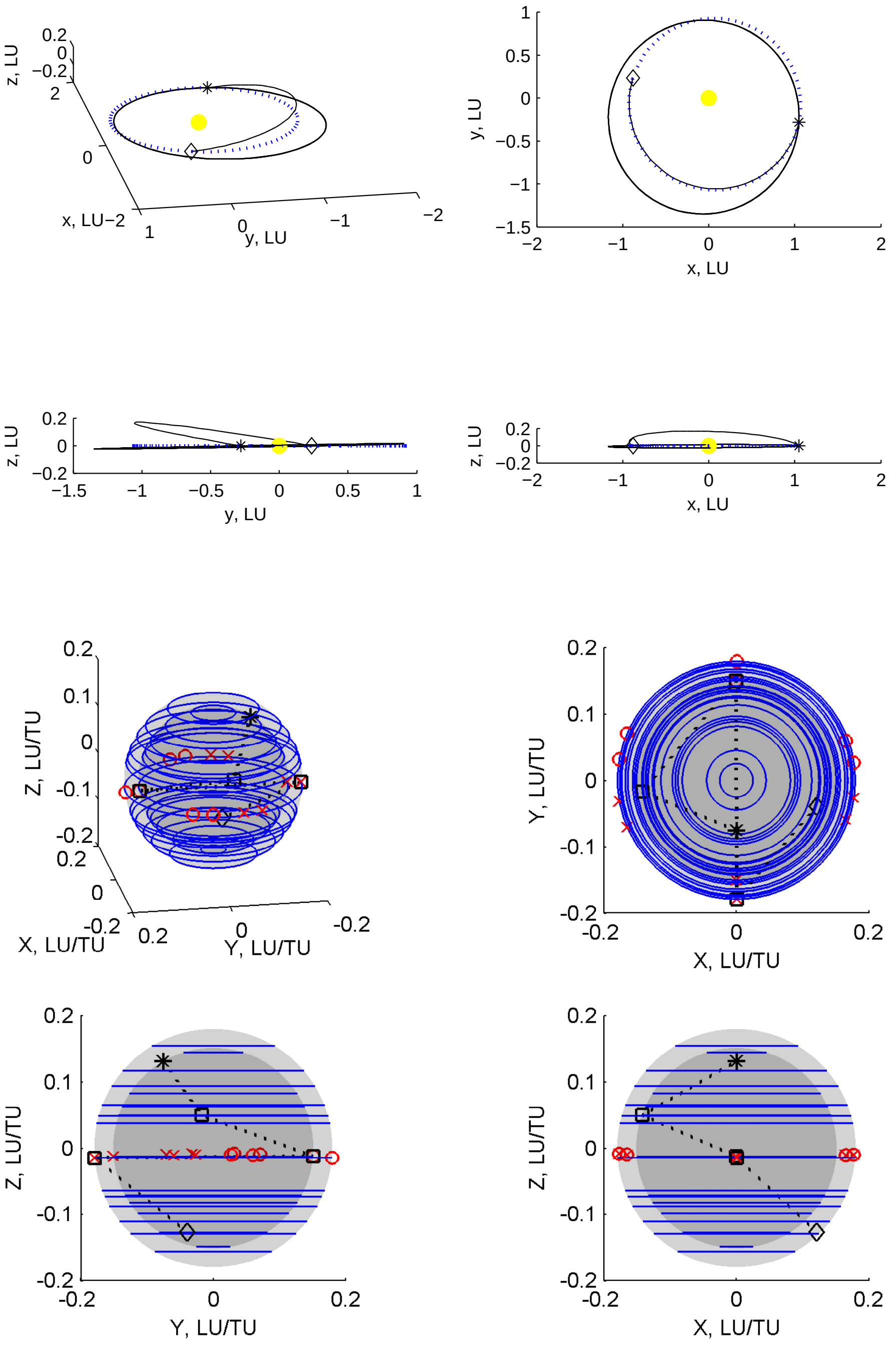


Figure 4.7: Four views of trajectory and of v_∞ sphere for an odd $n\pi$ sequence from Example B, like Example A but with a tighter turn angle constraint (refer to the text for the definitions of symbols)

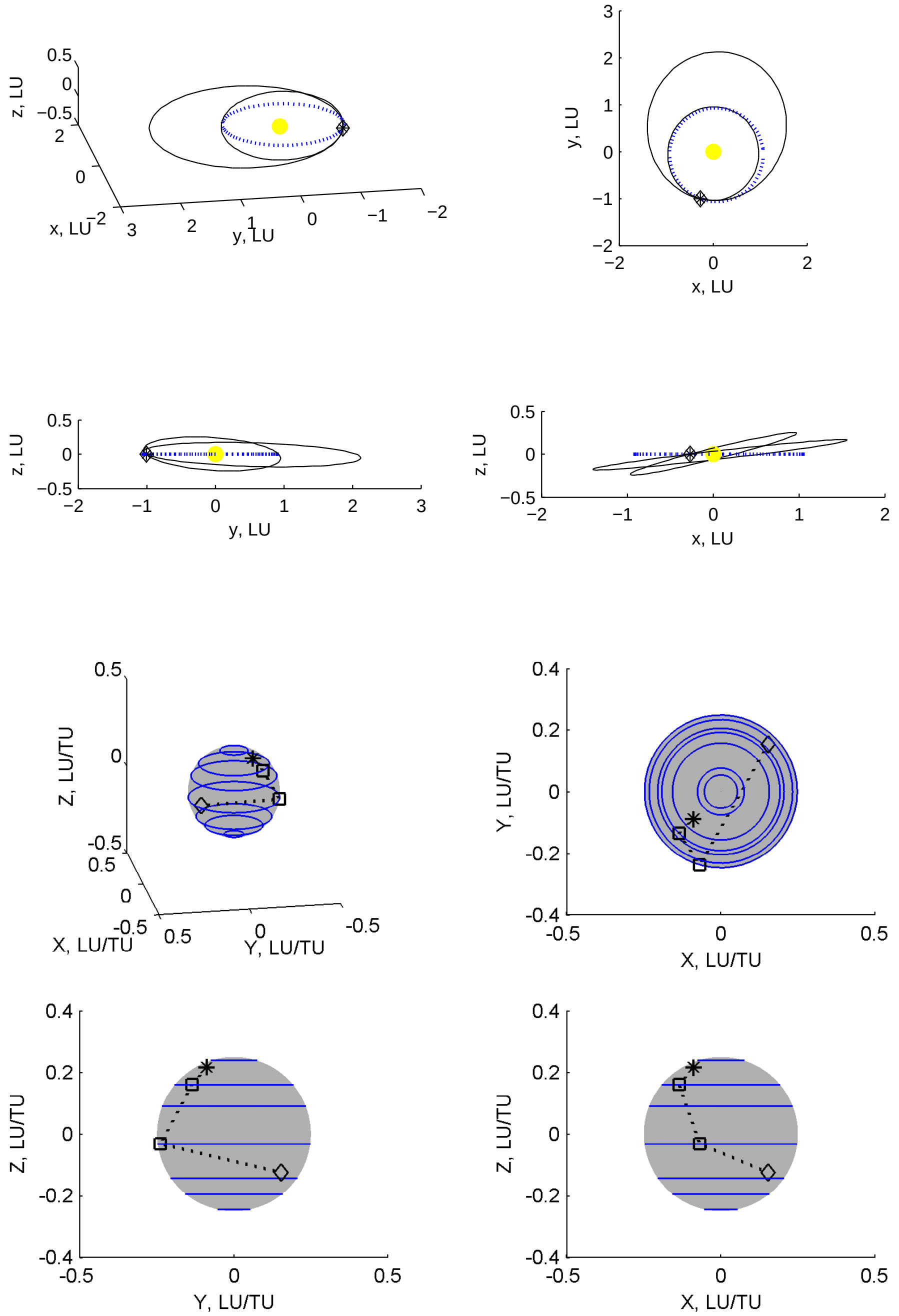


Figure 4.8: Four views of trajectory and of v_∞ sphere for an even $n\pi$ sequence from Example C (refer to the text for the definitions of symbols)

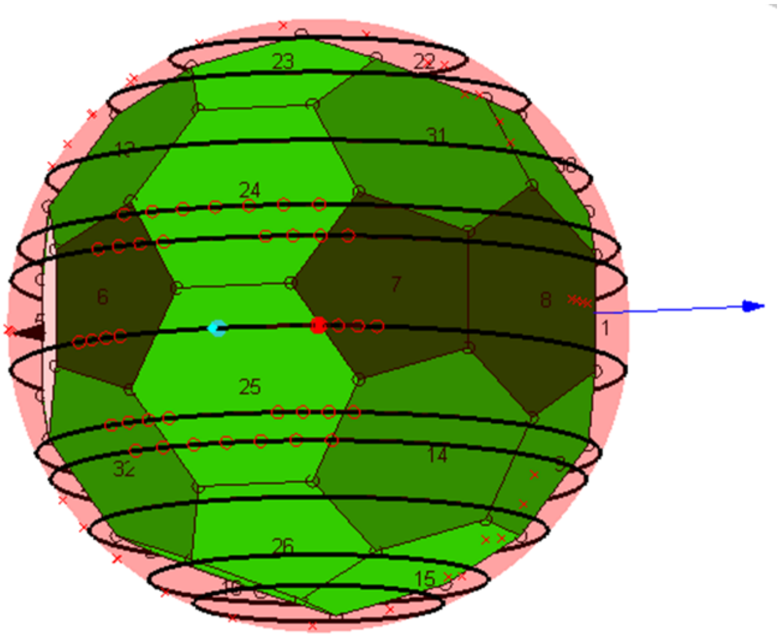


Figure 4.9: Flyby mapping sphere and v_∞ sphere used for the GTOC6 problem. The turquoise arrowhead is the inbound v_∞ vector and the dark red circles are discretized candidate points for the post-flyby $\mathbf{v}_\infty$. The x's show the different periapse locations possible for these candidate $\mathbf{v}_\infty$. The different colors of the grid patches correspond to point values and help the pathfinding algorithm pick which of candidate post-flyby points to rotate the $\mathbf{v}_\infty$ to (the filled red circle)

2. Map a discretized set of options on resonant circles between δ_{MIN} and δ_{MAX}. These are the large circles around the current $\mathbf{v}_\infty$ in Figure 4.9.

3. Choose one option based on a weighted sum of performance index gain (mapping new face), TOF, and radiation penalty as well as a random number makes the search stochastic. Note that which faces have already been mapped needs to be tracked for each sequence.

4. Repeat steps 2-3 from the current $\mathbf{v}_\infty$ until no options in step 2 provide gains in performance index because all the faces that can be mapped next have already been mapped.

5. Repeat steps 1-4 a specified number of times, storing sequences that can reach $\mathbf{v}_\infty^O$ with one flyby at the end and are nondominated in performance index, TOF, and radiation penalty. The random number used in the decision in step 3 makes this equivalent to a stochastic multistart.

6. All the stored sequences become distinct branches in the MGA search and are continued onwards or pruned by the MGA GS (*Explore* in this case).

Application of this pathfinding/pathsolving method was effective for generating many promising sequences. The geometry of periapse placement led to many tours that hopped between different resonances. Because the $n\pi$ sequence terminates when no further increase in score is possible, most sequences do not map the entire moon. As a result, the final MGA tour path used two $n\pi$ sequences at both Ganymede and Europa. Appendix C provides details of the University of Texas solution to the GTOC6 problem.

Chapter 5

Heliotropic orbits as low altitude science orbits at asteroids

The goal of this chapter is to locate and characterize heliotropic orbits as potential low-altitude science orbits for asteroid exploration[1]. Heliotropic orbits and their characteristics are introduced in Chapter 1. Heliotropic orbits are located in three dynamic models of increasing complexity. Then a preliminary analysis of the robustness of these orbits to gravity model uncertainty is presented.

The first model for locating heliotropic orbits in the presence of significant SRP and J_2 perturbations applies a singly-averaged potential in the Lagrange Planetary Equations (LPE). This process yields families of Sun-frozen orbits that maintain a frozen average eccentricity vector with respect to the Sun line. Equatorial heliotropic orbits are one of the families of Sun-frozen

[1] This chapter contains information also published in:
- Demyan Lantukh, Ryan P. Russell, and Stephen B. Broschart. "Heliotropic Orbits at Oblate Asteroids: Balancing Solar Radiation Pressure and J2 Perturbations," Celestial Mechanics and Dynamical Astronomy, Volume 121, Issue 2, pp 171-190, 2015. DOI: 10.1007/s10569-014-9596-x
- Demyan Lantukh, Ryan P. Russell, and Stephen B. Broschart. "Heliotropic Orbits at Asteroids: Zonal Gravity Perturbations and Application at Bennu," 25^{th} AAS/AIAA Space Flight Mechanics Meeting, 11-15 January 2015, Williamsburg, VA

Ryan P. Russell contributed critical insight into averaging and developed high-degree gravity modeling extension. Stephen B. Broschart conducted original numerical studies that found heliotropic orbits at Bennu model and also provided higher-fidelity simulations.

orbits calculated with this process.

Nonequatorial heliotropic orbits are found from a constrained doubly-averaged potential derived to enforce the heliotropic constraint. This doubly-averaged potential is used to develop an analytical formulation for the average orbital elements of inclined heliotropic orbits, enabling a fast, global approach to map zero-obliquity[2] inclined heliotropic orbits. The limits of these inclined heliotropic orbits in the mean elements are investigated and some example orbits are presented.

The third model extends the zonal gravity perturbation beyond J_2 to include higher degree even zonal terms, applied with the constrained doubly-averaged SRP disturbing potential. A model of asteroid Bennu is used to investigate the existence of heliotropic orbits in the presence of some parameter uncertainty.

5.1 Models

The central body is an asteroid around which the significant forces on a spacecraft are the gravity from the small body and SRP. The small body is considered to be an axisymmetric spheroid with its spin axis as its axis of symmetry. This body's gravity is modeled by a point-mass and the zonal gravity terms (J). Solar gravity is neglected because the orbits of interest – where J has an effect comparable to SRP – have sufficiently low altitudes; in terms of relative magnitude, the solar gravity acceleration is often greater than the small body gravity acceleration, but since its effect is almost identical on

[2] For axisymmetric bodies like those assumed in the current study, zero obliquity is equivalent to 180° obliquity.

both the small body and the spacecraft, it has minimal effect on the relative motion of the two bodies when they are very near each other. However, the motion of the small body about the Sun is included because the Sun line direction is important for the SRP. The body obliquity is zero (or equivalently 180°): the equatorial plane is the body orbit plane, and, since the body is axisymmetric, the body spin direction does not affect the spacecraft orbit. Solar gravity and nonzero obliquity are reintroduced as perturbations in the end of this chapter, where a full 12×12 spherical harmonics gravity field is also used to perturb the heliotropic orbit solutions.

Orbit conditions are developed in terms of a constant SRP acceleration (γ) which is directed away from the Sun. This is equivalent numerically to a spherical spacecraft assumption but the actual spacecraft shape is not needed as long as the SRP acceleration can be assumed constant[143, 132, 161, pp. 55–57]. The actual SRP experienced by a spacecraft contributes uncertainty as well as constraints and controls, but those are neglected in the derivation. The shadowing of the spacecraft from the Sun is also not modeled, but the symmetric orientation of heliotropic orbits with respect to the Sun line means that the perturbation from shadowing has no secular effect on orbital element rates [162]. As a result, the effect of shadowing shifts the location of the heliotropic orbit conditions but otherwise does not affect average orbit behavior.

When the ratio of spacecraft semi-major axis to body semi-major axis is small, the Sun-spacecraft distance (d) is approximately the body semi-major axis. Although the averaging process described below does not require a circular body orbit, for the current study a circular orbit with semi-major axis d is assumed for the central body. With these assumptions, the Sun line moves counterclockwise in the plane of the small body orbit with constant rate $\dot{f}$

Table 5.1: List of parameters for the representative SRP+J_2 case in this section

Parameter	Symbol	Value	Units
Body semi-major axis	d	1.684×10^8	km
Body impact (Brouillon sphere) radius	R_0	0.2887	km
Body gravitational parameter	GM_{body}	4.057×10^{-9}	km^3/s^2
Body oblateness: 2^{nd} order zonal gravity	J_2	$0.1\sqrt{5}$	
Body oblateness (normalized)		0.1	
Sun gravitational parameter	GM_{Sun}	1.327×10^{11}	km^3/s^2
SRP acceleration	γ	1.2762×10^{-10}	km/s^2

given by Equation (5.1). Definitions of the other parameters in Equation (5.1) are provided in Table 5.1, which also gives the values of parameters used for the example case presented below.

$$\dot{f} = \sqrt{\frac{GM_{Sun}}{d^3}} \tag{5.1}$$

Spacecraft orbits are defined by orbital elements in a non-rotating reference frame centered at the small body as shown in Figure 5.1. Orbits are visualized in a body-centered Sun-synodic reference frame where the Sun remains in the negative x-direction even though the propagation itself is in the non-rotating reference frame (heliotropic orbits rotate with the Sun line, as shown in Figure 1.3). The orbital elements used in this investigation are the classic set $\{a, e, i, \omega, \Omega, \nu\}$, with the angles defined in Figure 5.1 (except ν, which is the angle from $\hat{\mathbf{e}}$ to spacecraft position). The angle λ from the anti-Sun line ($\hat{\mathbf{d}}$) to the line of nodes ($\hat{\mathbf{n}}$) is also shown in Figure 5.1. The LPE for applying a disturbing potential R to this element set are well-known and can be found, among other texts, in [163, pp. 288–289]. These equations of motion and other equations throughout use the intermediates: $n^* = \sqrt{GM_{body}/a^3}$; $p = a(1-e^2)$; $M^* = n^*t$, where M^* is the mean anomaly.

Unless otherwise specified, the parameter values used in this chapter

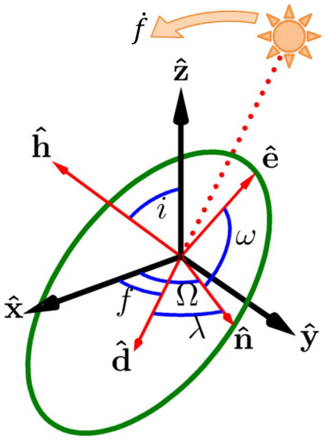


Figure 5.1: Reference frame definition and orbit geometry for asteroid proximity operations discussion

are given in Table 5.1, where the body impact radius is the minimum allowed spacecraft periapse, which is assumed equal to the Brouillon sphere radius. These parameters approximate a spacecraft around a body similar to Bennu but with higher J_2. This simplification is used to better characterize the heliotropic orbit space, and a more accurate Bennu model is then used to find the heliotropic orbit space at Bennu.

5.2 Single averaging and resulting Sun-frozen orbits

Sun-frozen orbit solutions in the presence of J_2 and SRP – of which equatorial heliotropic orbits are a subset – are investigated by first applying a singly-averaged disturbing potential to the LPE. The disturbing potential applied to the LPE is the sum of the averaged potential from J_2 and the averaged potential from the SRP, with the singly-averaged potentials given in Equations (5.2) and (5.3) respectively [133, 161, 142, pp. 293–294]. These disturbing potentials are averaged over a single spacecraft orbit and are valid assuming the change in the remaining orbital elements over a single orbit is sufficiently small.

$$\bar{R}_{J_2} = \frac{\mu R_0^2 J_2}{2a^3(1-e^2)^{3/2}}\left(1 - \frac{3}{2}\sin^2 i\right) \tag{5.2}$$

$$\bar{R}_{SRP} = -\frac{3ae\gamma}{2}\hat{\mathbf{d}}\cdot\hat{\mathbf{e}} = -\frac{3ae\gamma}{2}\left[\cos\omega\cos\left(\Omega - f\right) - \sin\omega\sin\left(\Omega - f\right)\cos i\right] \tag{5.3}$$

These disturbing potentials are applied to the LPE to generate equations of motion equivalent to Equations (11) to (14) of [142] with zero obliquity. [142] further convert these dynamics to nonsingular, nondimensional elements but the current investigation keeps the classical elements for their simple physical interpretation.

5.2.1 Sun-synodic equations of motion

The resulting equations of motion can be expressed more concisely and intuitively by applying the auxiliary element $\lambda = \Omega - f$ (shown in Figure 5.1). These equations of motion of the orbital elements and dynamics of λ are given in Equation (5.4).

$$\begin{aligned}
\dot{a} &= 0\\
\dot{e} &= -\frac{3\gamma}{2n^*a}\sqrt{1-e^2}\left[\sin\omega\cos\lambda + \cos\omega\sin\lambda\cos i\right]\\
\dot{i} &= -\frac{3\gamma}{2n^*a}\frac{e}{\sqrt{1-e^2}}\cos\omega\sin\lambda\sin i\\
\dot{\omega} &= -\frac{3\gamma}{2n^*a}\frac{1}{e\sqrt{1-e^2}}\left[(1-e^2)\cos\omega\cos\lambda - \sin\omega\sin\lambda\cos i\right] + \frac{3n^*R_0^2J_2}{2p^2}\left[2 - \frac{5}{2}\sin^2 i\right]\\
\dot{\Omega} &= -\frac{3\gamma}{2n^*a}\frac{e}{\sqrt{1-e^2}}\sin\omega\sin\lambda - \frac{3n^*R_0^2J_2}{2p^2}\cos i\\
\dot{\lambda} &= \dot{\Omega} - \dot{f}
\end{aligned} \tag{5.4}$$

5.2.2 Sun-frozen orbit conditions

The Sun-frozen condition applied in the singly-averaged dynamics and translated to the orbital element rates results in Equation (5.5).

$$\dot{\lambda} = \dot{e} = \dot{i} = \dot{\omega} = 0 \tag{5.5}$$

For equatorial orbits, the Sun-frozen orbit conditions are modified to account for the fact that Ω and ω cannot be defined individually. Following the example of Vallado, a retrograde factor is defined: $\delta_{dir} = 1$ for prograde orbits and $\delta_{dir} = -1$ for retrograde orbits [6, p. 116]. Using the retrograde factor, the undefined elements are replaced by Equation (5.6).

$$\Pi = \omega + \delta_{dir}\lambda \tag{5.6}$$

With this substitution, the frozen orbit conditions for equatorial orbits are given by Equation (5.7).

$$\dot{\Pi} = \dot{e} = \dot{i} = 0 \tag{5.7}$$

Although requiring $\dot{a} = 0$ is not necessary for an orbit to be frozen with respect to the eccentricity vector, the LPE dictate that all orbits under SRP and J_2 perturbations without shadowing will indeed have a constant mean semi-major axis.

5.2.2.1 Equatorial orbits

For equatorial orbits, $\sin i = 0$ and $\cos i = \delta_{dir}$. With these simplifications, $\dot{i} = 0$ and the equatorial dynamics simplify to Equations (5.8) and (5.9).

$$\dot{e} = -\frac{3\gamma}{2n^* a}\sqrt{1-e^2}\sin\Pi \tag{5.8}$$

$$\dot{\Pi} = \frac{3n^* R_0^2 J_2}{2p^2} - \frac{3\gamma}{2n^* a}\frac{\sqrt{1-e^2}}{e}\cos\Pi - \delta_{dir}\dot{f} \tag{5.9}$$

Using Equation (5.8), the corresponding Sun-frozen condition in Equation (5.7) is satisfied by either $\Pi = 0$ or $\Pi = \pi$. These two possible values of Π correspond to the "heliotropic" and "antiheliotropic" orbits, respectively, as described in past studies [139, 138].

Looking at Equation (5.9), $\dot{\Pi}$ can be considered a sum of a J_2 term, an SRP term, and $-\delta_{dir}\dot{f}$; The J_2 term has the sign of J_2 ($J_2 > 0$ for most bodies) and the SRP term has the sign opposite of $\cos\Pi$. As a result, if $J_2 > 0$ and $\Pi = \pi$, then the Sun-frozen solution can only exist when the body orbit direction is the same as the spacecraft orbit direction: for an oblate body in prograde motion, the $\Pi = \pi$ solution must be prograde and for an oblate body in retrograde motion, the $\Pi = \pi$ solution must be retrograde.

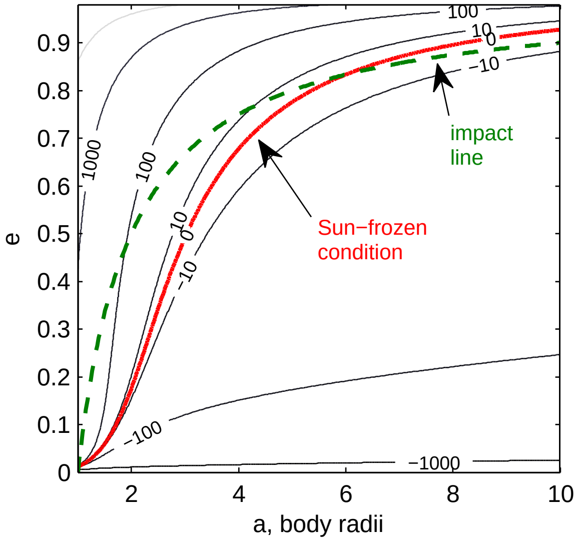


Figure 5.2: Sample contours of the $\dot{\Pi}$ rate equation (deg/day) solved for equatorial prograde orbits with $\Pi = 0$, using parameters from Table 5.1

The equatorial frozen orbit families are enumerated for each of the four possible cases $\{\delta_{dir} = \pm 1, \cos\Pi = \pm 1\}$ by choosing one of the two unknowns

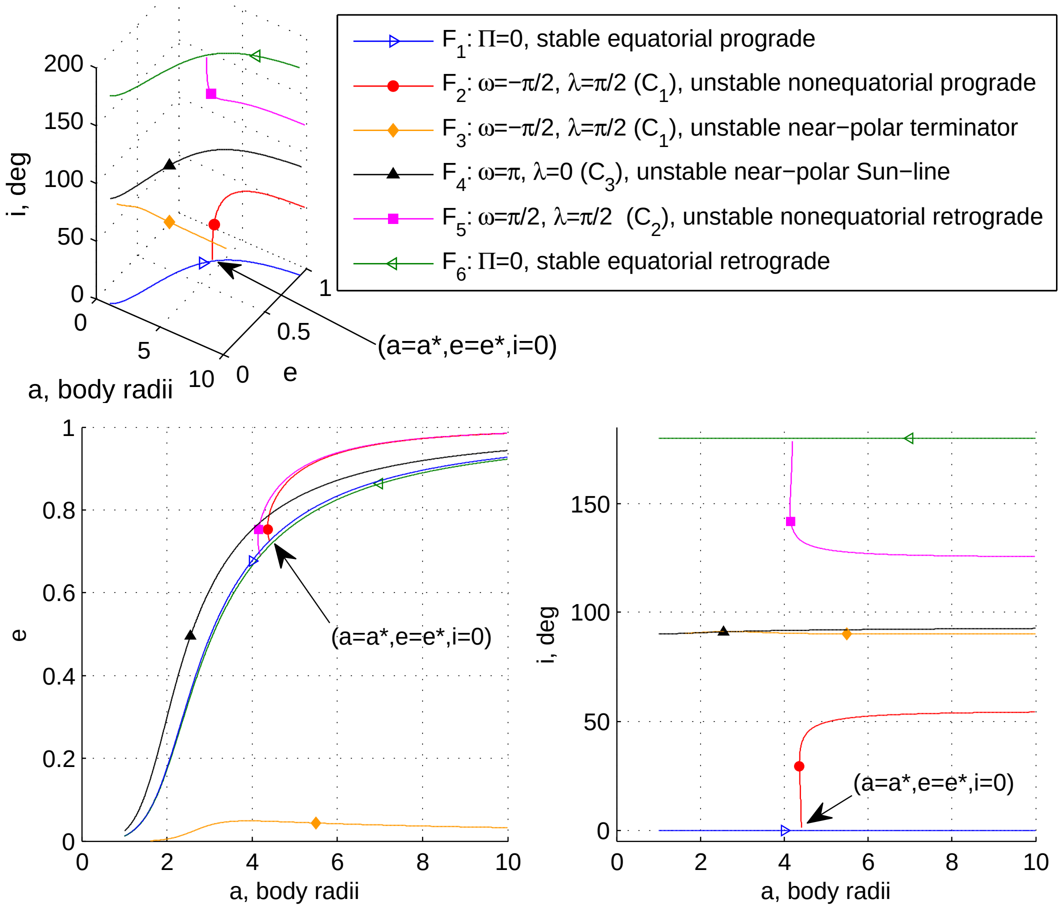


Figure 5.3: Sun-frozen orbit families from the singly-averaged analysis. Nonequatorial orbit case numbers (C_i) correspond to those given in Table 5.2.

$(a$ or $e)$ as the independent variable and solving for the other with $\dot{\Pi} = 0$ within a specified range of the independent variable. Using e as the independent variable is convenient because it is bounded. Evaluating the frozen orbit families for the representative case in the current study results in one prograde family of equatorial orbits and one retrograde family of equatorial orbits for $\Pi = 0$. Figure 5.2 gives the contour plot of $\dot{\Pi}$ for the prograde, $\Pi = 0$ case. No solutions with $\Pi = \pi$ exist for this case, though such solutions can exist

Table 5.2: Possible Sun-frozen orbit cases for nonequatorial orbits

Case	$\cos\omega\cos\lambda$	$\sin\omega\sin\lambda$	Possible Pairs of $\{\omega,\lambda\}$
C_1	0	-1	$\{\ -\pi/2, \pi/2\}, \{\ \pi/2, -\pi/2\}$
C_2	0	1	$\{\ \pi/2, \pi/2\}, \{\ -\pi/2, -\pi/2\}$
C_3	-1	0	$\{\ 0, \pi\}, \{\ \pi, 0\}$
C_4	1	0	$\{\ 0, 0\}, \{\ \pi, \pi\}$

when the perturbations are smaller relative to the body gravitational parameter [138, 141]. The resulting Sun-frozen families for the presented case are given in Figure 5.3.

5.2.2.2 Nonequatorial orbits

For nonequatorial orbits, finding frozen orbit families also begins with $\dot{e} = 0$ and $\dot{i} = 0$; these two conditions together lead to Equation (5.10), the primary frozen orbit requirement for nonequatorial orbits.

$$(\cos\omega = 0 \ \ AND \ \ \cos\lambda = 0) \quad OR \quad (\sin\omega = 0 \ \ AND \ \ \sin\lambda = 0) \qquad (5.10)$$

The trigonometric functions of ω and λ always exist in a product of two such terms in the LPE. The frozen orbit condition then leads to $\sin\omega\cos\lambda = 0$ and $\sin\lambda\cos\omega = 0$. Further, the other products of trigonometric functions of ω and λ can be restricted to a small set: $\cos\omega\cos\lambda = \{-1, 0, 1\}$ and $\sin\omega\sin\lambda = \{-1, 0, 1\}$. The resulting feasible combinations allow four possible cases of frozen orbit conditions, enumerated in Table 5.2.

Of the four cases in Table 5.2, C_1 is considered first: In this case the line of nodes begins normal to the Sun line and periapsis begins on the night side of the body. Applying the Sun-frozen orbit conditions maintains this geometry relative to the Sun: Using the rates from Equation (5.4) in Equation (5.5), and applying all the simplifications that C_1 allows, leads to the Sun-frozen orbit conditions given in Equation (5.11).

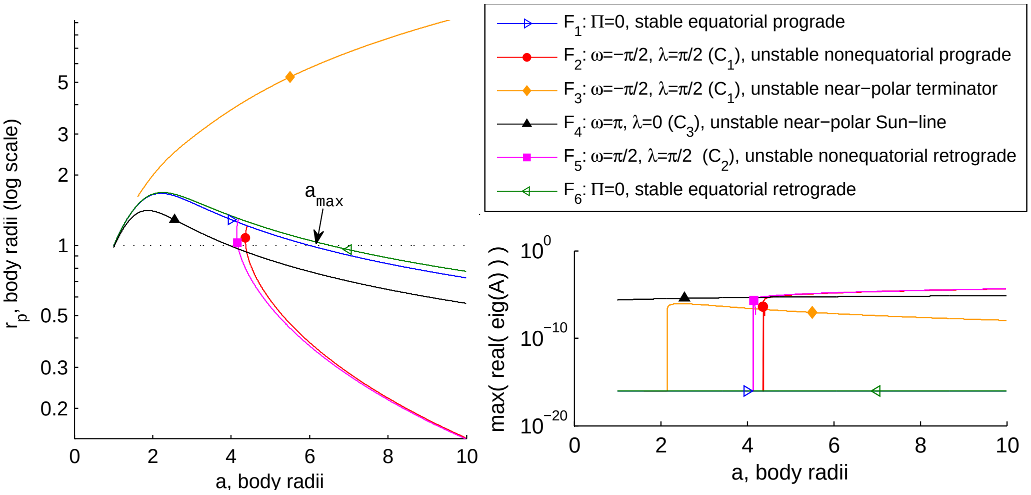
$$\begin{aligned} \dot{\omega}_{C_1} &= 0 = -\frac{3\gamma}{2n^* a} \frac{1}{e\sqrt{1-e^2}} \cos i + \frac{3n^* R_0^2 J_2}{4p^2} \left[4 - 5\sin^2 i\right] \\ \dot{\lambda}_{C_1} &= 0 = \frac{3\gamma}{2n^* a} \frac{e}{\sqrt{1-e^2}} - \frac{3n^* R_0^2 J_2}{2p^2} \cos i - \dot{f} \end{aligned} \tag{5.11}$$

Equation (5.11) provides two equations with three unknowns, $\{a, e, i\}$, for a given body. Solving $\dot{\lambda}_{C_1} = 0$ for $\cos i$ and using $\sin^2 i = 1 - \cos^2 i$ reduces the Sun-frozen orbit conditions to one equation in two unknowns: $\{a, e\}$. Choosing one of these as an independent variable, the other can be solved as the dependent variable to generate families of solutions, taking into account that the dependent variable may be undefined or may not be single-valued. (For example, F_2 and F_3 in Figure 5.3 are two distinct solutions of Equation (5.11).)

The solution process is repeated for the other three cases, resulting in a pair of non-equatorial frozen orbit equations that can be reduced to one equation in two parameters for each case. Each set of equations is mapped globally to bound solution regions, then solved locally using a numerical root-finding technique to find the frozen orbit families. Note that this process yielded no solutions for C_4 (with the given parameters). The resulting Sun-frozen families are shown in Figure 5.3.

5.2.3 Stability of Sun-frozen orbit families

To aid in determining practically useful orbits, the Sun-frozen families can be categorized by evaluating their stability. The LPE can be written in vector form as $\dot{\mathbf{X}} = f(\mathbf{X})$ where $\mathbf{X}$ is the vector of the orbital elements. Then, if the state for a frozen orbit is $\mathbf{X}^*$, the frozen orbit conditions are $\dot{\mathbf{X}}^* =$



(a) Periapse distance of the Sun-(b) Stability criteria, with values below the frozen orbit families. The horizontalnoise threshold of 1×10^{-16f} set to the dashed line is the body impact radiusthreshold

Figure 5.4: Characteristics of the Sun-frozen families from the singly-averaged equations of motion

$f(\mathbf{X}^*) = 0$. Considering a small perturbation around $\mathbf{X}^*$ where $\mathbf{X} = \mathbf{X}^* + \delta\mathbf{X}$, the dynamics of $\delta\mathbf{X}$ to first order are:

$$\delta\dot{\mathbf{X}} = A\delta\mathbf{X} \tag{5.12}$$

$$A = \left.\frac{\partial f(\mathbf{X})}{\partial \mathbf{X}}\right|_{\mathbf{X}^*} \tag{5.13}$$

The linear stability of a particular frozen orbit $\mathbf{X}^*$ (i.e. an equilibrium solution to the nonlinear equations of motion) is determined using the eigenvalues of A. If any eigenvalue has a positive real part then the orbit is unstable. If all eigenvalues have nonpositive real parts then the orbit is linearly stable. For the singly-averaged equations of motion given in Equation (5.4) applied at the Sun-frozen conditions, the partial derivatives in $\mathbf{X}$ are simplified because $\sin\omega\cos\lambda = 0$ and $\cos\omega\sin\lambda = 0$. For the equatorial case, derivative calculation is simplified by the condition $\sin\Pi = 0$.

Evaluating the derivatives shows that one of the eigenvalues of A is always zero. For unstable orbits, the size of the real component of the eigenvalues gives some indication of the characteristic time for an orbit to leave the neighborhood of its initial condition. Therefore, the stability metric for orbits in the current study is the supremum of the real parts of the eigenvalues of A. Figure 5.4b shows this supremum for each of the families of frozen orbits. The result is that most of the nonequatorial orbits are unstable and all the equatorial orbits are linearly stable. The low-eccentricity terminator family becomes stable for very small J_2, which is expected since terminator orbits are linearly stable with only SRP. The point where the unstable family F_2 approaches the equatorial F_1 is designated $\{a = a^*, e = e^*, i = 0\}$ for use as a reference point in studying nonequatorial orbits from a doubly-averaged potential.

5.2.4 Discussion on the Sun-frozen orbit families

The prograde equatorial orbit family F_1 in Figure 5.3 corresponds to the planar heliotropic types of orbits presented and analyzed in literature. The current analysis verifies that the retrograde equatorial family F_6 exists and can be determined by the process described above. The families F_1 and F_6 do not coincide in a and e, as shown in Figure 5.3.

The two near-polar families correspond to J_2-perturbed frozen orbit families that exist with SRP alone. F_2 and F_5 span a wide range of inclinations and, at first glance, appear to make good candidate science orbits. However, for the specific case of parameters considered, these two families have large impacting regions and the regions that do not impact are unstable.

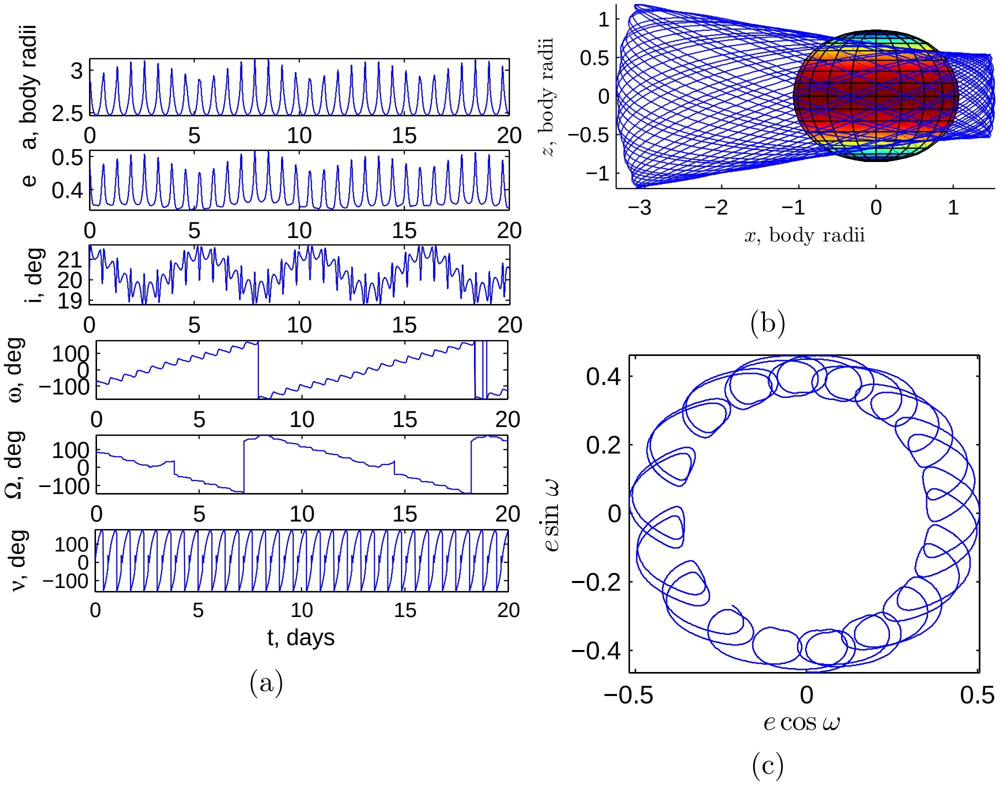
Figure 5.4a shows the average expected periapse radius along the different Sun-frozen families in Figure 5.3. Comparing these average periapse distances to the impact radius bounds the usable regions of the Sun-frozen families. The semi-major axis for F_1 that marks the border between impacting and nonimpacting orbits is defined as a_{max}, as shown in Figure 5.4a. Further analysis in the following sections shows that inclined Sun-frozen orbits do not exceed a_{max}.

5.3 Constrained double averaging for inclined heliotropic orbits

The stable heliotropic orbits from past analytical studies (verified with the current singly-averaged Sun-frozen orbit analysis) are equatorial [143, 141]. Past work [138] has looked at numerically following families of orbits to find inclined heliotropic orbits, but a more systematic and global means to find inclined heliotropic orbits is desirable. In pursuit of this goal, consider Figure 5.5 which shows an orbit perturbed by twenty degrees in inclination from the equatorial family of heliotropic orbits. Although simply changing the initial inclination from the equatorial family can produce long-lifetime orbits like the one presented in Figure 5.5, this method is based on trial and error and provides little understanding about the design space. However, observing the resulting inclined heliotropic orbit provides at least one path forward: the secular rates of both Ω and ω appear to be constant (and so the rate of λ is also constant). In addition, these angles are approximately related to each other by Equation (5.14), which maintains the heliotropic geometry of the orbit (see Figure 1.3).

$$\Pi = 0 \tag{5.14}$$

Observing these characteristics of inclined heliotropic orbits motivates the determination of the secular rates of λ and ω. Since the rates in both λ and ω have small oscillations about a mean, a second average can isolate the secular components (assuming, rather liberally, that the typical averaging assumptions still apply: i.e. all elements except the argument of the average are approximately constant over the averaging period). Since the angular condition in Equation (5.14) keeps the eccentricity vector in the anti-Sun hemisphere, no additional constraints are needed to find families of heliotropic inclined orbits.



(a)

(b)

(c)

Figure 5.5: Example A: Heliotropic orbit found by perturbing the equatorial family of solutions. Initial conditions for the orbit are given in Table 5.3

5.3.1 Constrained doubly-averaged LPE

Double-averaging the potential is achieved by averaging the constrained singly-averaged potential over one period of λ as detailed below. For the example in Figure 5.5, the second averaging period is approximately 11 days. Note that $\dot{\Omega}$ is thirty times greater than $\dot{f}$ and so contributes primarily to $\dot{\lambda}$. This averaging assumes that the change in a, e, and i over one period of λ is small, an assumption that may not hold with the large perturbations under consideration. Even with this limitation, the results in the following section demonstrate cases of practical use for the doubly-averaged results, validated by simulations in the unaveraged dynamics. Figure 5.6 shows how the averaged and unaveraged dynamics compare in the evolution of the eccentricity vector from the same initial conditions. Furthermore, it is noted that doubly-averaged potentials have been successfully applied to the other dynamical systems, such as the restricted three body problem [164] and Hill's model [135]. In the problem investigated here, the constrained doubly-averaged potentials are used to provide the mean values for $\dot{\lambda}$ and $\dot{\omega}$ as well as estimates of initial conditions for inclined heliotropic orbits in the full dynamics.

The process for calculating heliotropic orbits is summarized in Algorithm 5.1 and detailed below. The process is first conducted with only J_2 of the zonal gravity terms and then expanded to include higher degree terms. Starting with only J_2 allows for a simple solution that lends itself to analysis; the higher degree zonal fields are needed for accurate modeling of highly nonspherical bodies.

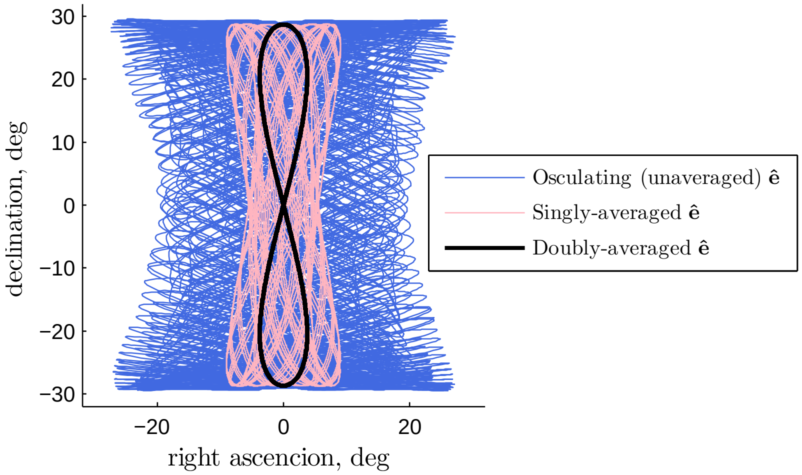


Figure 5.6: Eccentricity vector evolution of a heliotropic orbit propagated for one body orbit around the Sun. Spherical coordinates for the eccentricity vector show that it remains near the anti-Sun line (0,0). Initial conditions are Example B in Table 5.3.

Algorithm 5.1 Generating a heliotropic orbit using constrained double averaging

```
1: Determine $\bar{\bar{R}}^*_{SRP}$, the constrained doubly-averaged disturbing potential from SRP. For the assumptions in this chapter $\bar{\bar{R}}^*_{SRP}$ is given in Equation (5.16)
2: Determine averaged secular disturbing potential from zonal gravity harmonics ($\bar{\bar{R}}_{J,sec}$) using Algorithm 5.2
3: Assemble the doubly-averaged disturbing potential $\bar{\bar{R}}^* = \bar{\bar{R}}^*_{SRP} + \bar{\bar{R}}_{J,sec}$
4: Use $\bar{\bar{R}}^*$ in the Lagrange Planetary Equations to set up Eq. (5.17)
5: Pick values for independent variables (nominally $a$ and $e$)
6: solve Eq. (5.17) numerically for remaining unknown (nominally $i$)    ▷ A solution may not exist for every pair of independent variable values
```

5.3.2 The constrained doubly-averaged SRP disturbing potential

Performing a simple average of $\bar{R}_{SRP}$ from Equation (5.3) over either λ or ω would give zero as a result. However, by applying a constraint based on the heliotropic condition given by Equation (5.14), the resulting doubly-averaged potential is nonzero: Using this heliotropic constraint leads to the substitution $\lambda = -\omega$ (for prograde orbits) in $\bar{R}_{SRP}$. The resulting constrained disturbing potential for SRP is shown in Equation (5.15), where the * superscript indicates the constraint is active.

$$\bar{R}^*_{SRP} = -\frac{3ae\gamma}{2}\left[\cos^2\lambda + \sin^2\lambda\cos i\right] \tag{5.15}$$

The one-period averages over λ of $\sin^2\lambda$ and $\cos^2\lambda$ in Eq. (5.15) are both 1/2, leading to Equation (5.16).

$$\bar{\bar{R}}^* = -\frac{3ae\gamma}{4}\left[1 + \cos i\right] \tag{5.16}$$

The heliotropic geometry is maintained by applying the rate constraint given in Equation (5.17). Note that Equation (5.17) is the time derivative of Equation (5.14) and is equivalent to Equation (5.7) for prograde orbits.

$$\dot{\Pi} = \dot{\omega} + \delta_{dir}(\dot{\Omega} - \dot{f}) = 0 \tag{5.17}$$

The angular heliotropic constraint, given in Eq. (5.14), only affects the SRP portion of the disturbing potential because the zonal gravity portion is already independent of ω and Ω after it is averaged to isolate the secular effects. As a result, the LPE can be split into a sum of SRP terms and gravity terms; expanding Eq. (5.17) leads to:

$$\left(\dot{\omega}_{SRP} + \delta_{dir}\dot{\Omega}_{SRP}\right) + \left(\dot{\omega}_{J,sec} + \delta_{dir}\dot{\Omega}_{J,sec}\right) - \delta_{dir}\dot{f} = 0 \tag{5.18}$$

5.3.3 Heliotropic orbits with SRP and J_2

For the simplified case where the small body is modeled by only a point mass and J_2 zonal gravity term, the LPE with the constrained doubly-averaged disturbing potential are given in Equation (5.19)

$$\begin{aligned}
\dot{a}^* &= \dot{e}^* = \dot{i}^* = 0 \\
\dot{\omega}^* &= -\frac{3\gamma}{4n^* a}\frac{1}{e\sqrt{1-e^2}}\left[1-e^2+\cos i\right] + \frac{3n^* R_0^2 J_2}{2p^2}\left[-\frac{1}{2}+\frac{5}{2}\cos^2 i\right] \\
\dot{\lambda}^* &= \frac{3\gamma}{4n^* a}\frac{e}{\sqrt{1-e^2}} - \frac{3n^* R_0^2 J_2}{2p^2}\cos i - \dot{f}
\end{aligned} \tag{5.19}$$

A surface of orbit conditions for heliotropic orbits can be developed from the constrained doubly-averaged LPE. In the doubly-averaged problem there remain three unknowns and one constraint; solving Equations (5.19) and (5.17) for prograde orbits leads to a quadratic equation in $\cos i$ given by Equation (5.20).

$$\begin{aligned}
-\frac{4}{3}\dot{f} &= \left[K_{J_2} + (1-2e^2)K_{SRP}\right] + \left[2K_{J_2} + K_{SRP}\right]\cos i - 5K_{J_2}\cos^2 i \\
K_{J_2} &= \frac{n^* R_0^2 J_2}{p^2} \\
K_{SRP} &= \frac{\gamma}{n^* a e\sqrt{1-e^2}}
\end{aligned} \tag{5.20}$$

One of the solutions to this quadratic, Equation (5.21), is the defining equation for this constrained doubly-averaged system. Equation (5.21) gives the solution for inclination at a given pair $\{a, e\}$ where such inclination exists. The other solution to the quadratic equation violates the prograde assumption made in the derivation. An inclination exists that enables a doubly-averaged heliotropic orbit when $B_4 \geq 0$ and $-1 \leq \cos i \leq 1$.

$$\begin{aligned}
\cos i &= \frac{1}{5} + \frac{B_2 + \sqrt{B_4}}{30\,B_1\,B_3} \\
B_1 &= n^{*2} R_0^2 J_2 \\
B_2 &= 3\,\gamma\,p^2 \\
B_3 &= ae\sqrt{1-e^2} \\
B_4 &= 120\,\left(3/5 - e^2\right) B_1 B_2 B_3 + 216\,B_1^2 B_3^2 + 240\,np^2 B_1 B_3^2 \dot{f} + B_2^2
\end{aligned} \tag{5.21}$$

Table 5.3: Initial conditions for the integrated orbit examples. All examples begin at $f = 0$. Table notes provide further information for periodic orbits.

Ex.	a (body radii)	e	i (deg)	ω (deg)	λ (deg)	ν (deg)
A	2.8987341	0.46481127	20	-90	90	180
B	2.7454004	0.52314381	28.715568	-90	90	180
B_P[a]	3.1417561	0.51510321	30.626542	-68.4474	77.5531	68.4474
C	4.2500558	0.73173913	20.429662	-90	90	180
D	4.9121042	0.77478261	11.734502	-90	90	0
E[b]	1.9048110	0.25724790	18.635145	-146.268	-157.517	146.268

[a] Repeat period: 12.995380 days, 18 spacecraft revolutions. Largest eigenvalue magnitude: 1.0000027. Initial conditions are given at the x-y plane crossing.

[b] Repeat period: 4.1014777 days, 11 spacecraft revolutions. Largest eigenvalue magnitude: 1.0255871. Initial conditions are given at the x-y plane crossing.

Equation (5.21) leads to a surface of mean orbital element conditions that produce heliotropic orbits. The resulting surface of orbit conditions, shown in Figure 5.7, is bounded on its low-eccentricity end by the equatorial family of frozen heliotropic orbits. On the high-eccentricity end of the surface, the orbits impact the body. Figure 5.7 is similar to the surface of inclined heliotropic orbits for an Earth-orbiting spacecraft, found numerically by [144], which is also bounded by the equatorial solutions and extends into the region of impacting orbits.

5.4 Orbit design considerations for heliotropic orbits at small bodies

Because the doubly-averaged solutions do not account for the possibility of impact or escape, defining the useful portion of the surface of heliotropic conditions is desirable. The heliotropic surface and its bounds are shown in Figure 5.7. In addition, the doubly-averaged solutions assume that the orbital elements do not change significantly over the course of one orbit or one period of λ circulation, so there may be cases where the surface is not a good predictor of the full dynamics. Examples B and C from Table 5.3, shown in Figures 5.8 and 5.9, respectively, illustrate this possibility: Both examples use initial conditions from the heliotropic surface, but B stays well bounded for 60 days in a heliotropic orbit while C departs from the heliotropic condition and impacts the body. Figure 5.6 shows the evolution of the eccentricity vector of Example B with respect to the anti-Sun line ($\hat{\mathbf{d}}$). The orbit maintains its heliotropic character, with periapse confined to an angular region around $\hat{\mathbf{d}}$.

The low-eccentricity boundary of the heliotropic surface is the Sun-frozen orbit equatorial case (F_1) from the singly-averaged analysis. Orbits along this boundary are stable in the singly-averaged elements. However, there exists a bifurcating unstable family of frozen orbits (F_2), and orbits near this unstable family are likely unstable. This unstable family provides a useful boundary, as discussed below. For a given value of a there is a range of possible e for heliotropic orbits, as defined by the heliotropic surface. Each of these possible $\{a, e\}$ pairs also corresponds to a particular inclination.

The high-eccentricity end of the heliotropic surface consists of body-impacting orbits. This impact boundary can be approximated as the point where the periapse distance is the Brouillon sphere radius: $a(1-e) = R_0$. (In practice a higher minimum radius limit may be desirable.) Substituting this maximum eccentricity into Equation (5.21) allows the calculation of the maximum inclination for a given value of a, a function defined as i_{impact} since this high eccentricity boundary is also the high-inclination boundary for heliotropic orbits. The boundary i_{impact} begins along the low-e boundary at $\{a = R_0, e = 0, i = 0\}$, then rises steeply in inclination to a maximum value

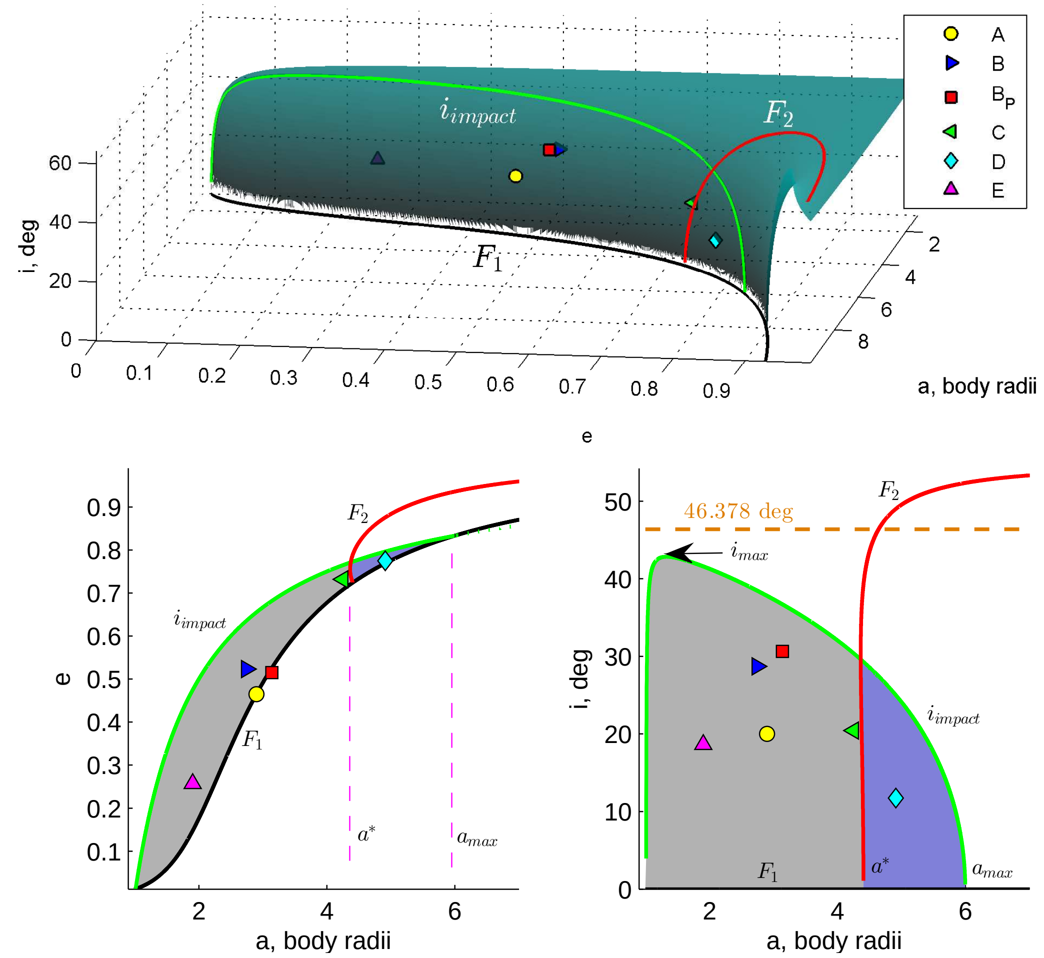


Figure 5.7: The heliotropic orbit surface in three dimensions and from two projections. The shaded region of interest is bounded by F_1 and i_{impact}, the impact boundary. This region is divided by F_2. Specific cases are defined in Table 5.3 (Case E is behind the semi-transparent surface in the three-dimensional view).

(i_{max}), and then returns to the equatorial family (the low-e boundary) again at $\{a = a_{max}, e = 1 - R_0/a_{max}, i = 0\}$. For a particular case, the value of i_{max} can be found numerically by solving $\mathrm{d}i_{impact}/\mathrm{d}a$ for $a_{i_{max}}$ and then calculating $i_{impact}(a_{i_{max}})$. Bounding cases for maximum possible inclination for all

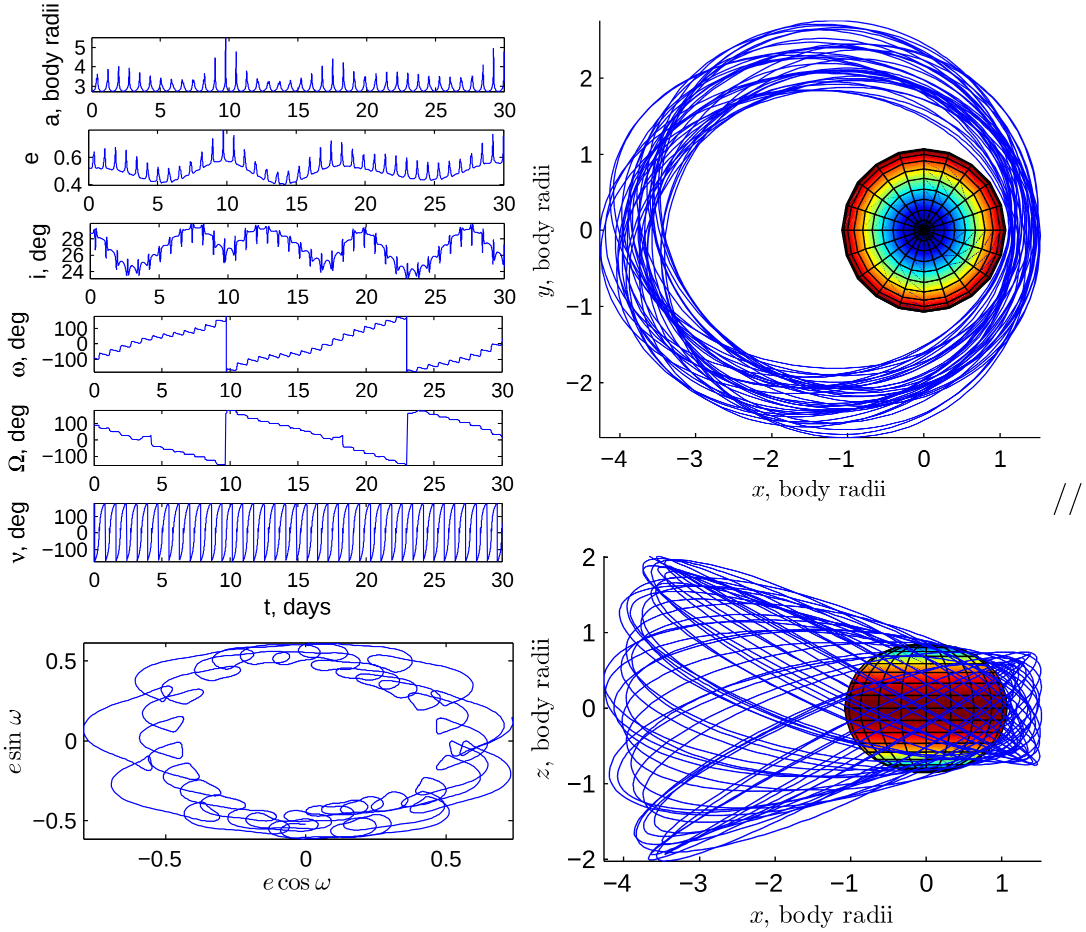


Figure 5.8: Example B: Long-lifetime heliotropic orbit found using the surface of doubly-averaged heliotropic conditions. Initial conditions for the orbit are given in Table 5.3

heliotropic orbits are discussed below.

5.4.1 Limiting cases for heliotropic orbits

Although a_{max} provides an upper bound on a based on impact, the unstable Sun-frozen orbit family (F_2, found in the singly-averaged analysis) may

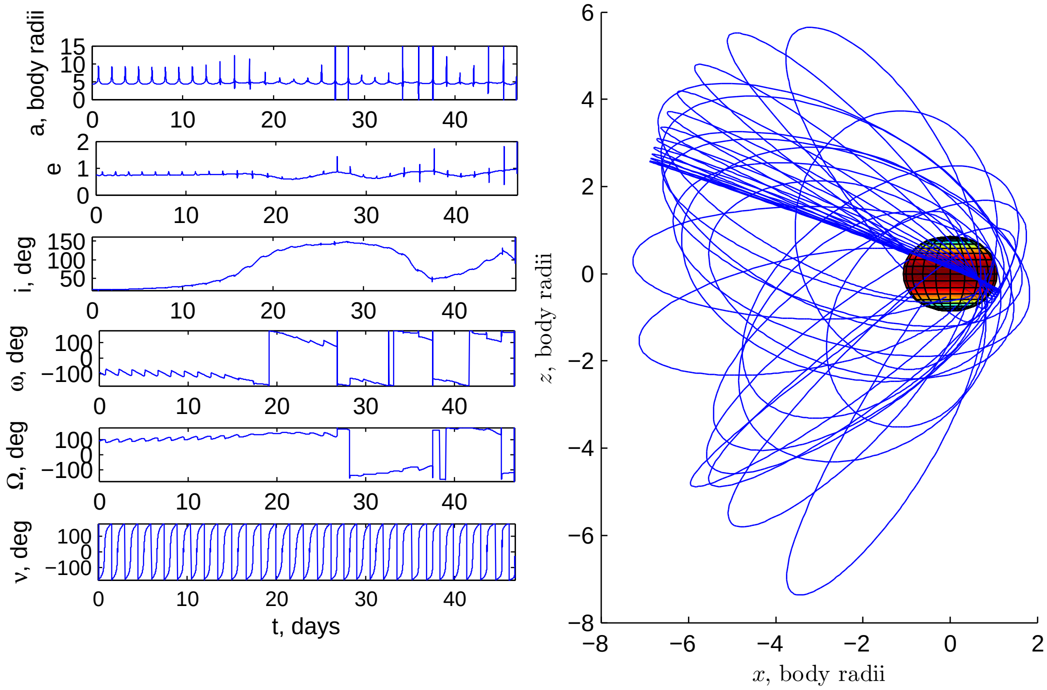


Figure 5.9: Example C: Unstable orbit from the surface of doubly-averaged heliotropic conditions near the unstable singly-averaged Sun-frozen family F_2. Initial conditions for the orbit are given in Table 5.3

provide a better upper bound on a for long-lifetime heliotropic orbits (when F_2 is within the bounds $[R_0, a_{max}]$). Although F_2 only crosses the heliotropic surface at a^*, its proximity suggests that the true dynamics of orbits in the vicinity are unstable; example C in Figure 5.9 illustrates such a case. Conveniently, the steepness of (F_2) at low inclinations allows the use of its end at $i = 0$ (a^*) as a single upper bound value for a. For values of a approaching and exceeding a^*, the singly-averaged equations are no longer stable for inclined orbits when integrated. Even so, long-lifetime orbits may still exist in the full dynamics with $a^* \leq a \leq a_{max}$: Example D, shown in Figure 5.10, is a long-lifetime orbit where $a > a^*$. This orbit is notable for its large periodic variation in e. One practical suggestion is to use initial conditions near periapsis for $a \geq a^*$. This empirical rule was found to aid in generating non-impacting osculating elements from the mean elements. For $a < a^*$, the initial true anomaly does still substantially affect the orbit, but there are often long-lifetime orbits for any choice of initial true anomaly (contrary to the case of $a \geq a^*$).

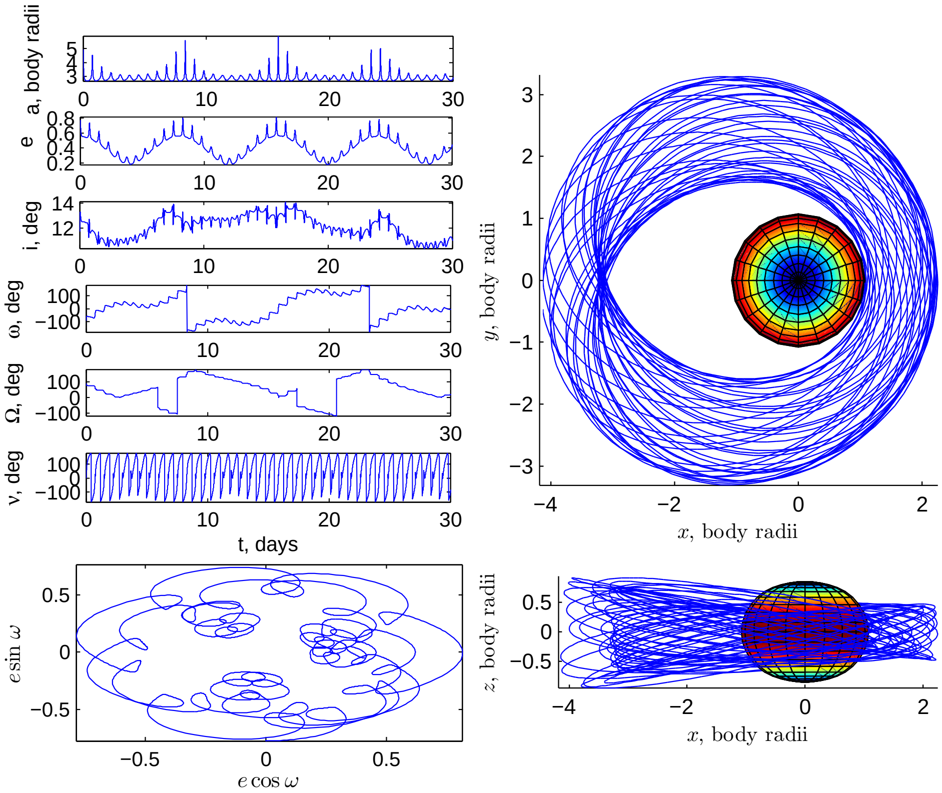


Figure 5.10: Example D: Long-lifetime heliotropic orbit with $a > a^*$ found using the heliotropic surface. Initial conditions for the orbit are given in Table 5.3

The limiting cases for heliotropic orbits can be estimated by manipulating Equation (5.21). Here the oblateness effect is contained entirely in B_1 and the effect of solar radiation pressure entirely in B_2. Also, B_1, B_2, B_3, and B_4 are all real and nonnegative for captured orbits about an oblate spheroid (for $J_2 > 0$). Noting that increasing the strength of SRP (increasing B_2) decreases i if all else is held constant, the maximum possible inclination is expected to come about by reducing B_2. Setting $B_2 = 0$ and ignoring the effect of the small body motion ($\dot{f} = 0$), Equation (5.21) simplifies to Equation (5.22), yielding an inclination of $\approx 46.4°$. Under these conditions, the J_2 terms in the numerator and denominator of Equation (5.21) cancel each other out (again assuming $J_2 > 0$). Note that B_2 also approaches zero when e approaches one, even with the effect of SRP, so the result still holds for high eccentricity orbits. Although higher inclination is possible on the heliotropic surface, the highest-inclination areas (visible in the three-dimensional view in Figure 5.7) occur well into the zone of impacting orbits at relatively high values of a (using the parameter values in Table 5.1).

$$\cos i_{B_2=0,\dot{f}=0} = \frac{1+\sqrt{6}}{5} \tag{5.22}$$

This maximum inclination limit is approached as J_2 increases, with everything else held constant. Moving in the opposite direction, trying to eliminate the effect of J_2, leads to $B_1 = 0$ and causes a singularity in Equation (5.21). Considering instead a small, positive value for B_1 allows the approximation $B_4 \to B_2$, which leads to Equation (5.23). In this equation, maximizing $\cos i$ allows the minimum possible B_1, so assuming $\cos i = 1$, then the approximate minimum value of B_1 is given by Equation (5.24). Two intuitive results are verified by Equation (5.24): first, as the effect of SRP (B_2) increases, a larger oblateness is required to maintain the heliotropic orbit and, second, neglecting the effect of SRP entirely does not drive the requirement on J_2 (B_1) to zero.

$$\cos i_{B_1 \to \epsilon} = \frac{3 + B_2}{15\, B_1\, B_3} \tag{5.23}$$

$$B_{1,min} \approx \frac{3 + B_2}{15\, B_3} \tag{5.24}$$

Equation (5.24) being nonzero results from the need to precess the orbital plane to match the Sun line motion. Some perturbation (i.e. oblateness) is needed to drive that precession.

In summary, practical prograde heliotropic orbits around an oblate ($J_2 > 0$) body are expected to fall within the bounds:

$$\begin{aligned} R_0 \leq & a \leq \min(a^*, a_{max}) \\ 0 \leq & e \leq e_{max} \\ 0 \leq & i \leq i_{max} \lesssim 46.4^\circ \end{aligned}$$

The quantities a_{max}, e_{max}, and a^* come from the singly-averaged Sun-frozen orbit solutions. The value of i_{max} can be found for a particular case with a numerical root-finding procedure as outlined above. These bounds for the space of heliotropic orbits will change for different obliquities, GM, γ, and J_2, but the orbits, once found, are expected to be robust to small changes in these parameters. Future studies will focus on the robustness of the heliotropic orbits to parameter uncertainty.

5.4.2 Periodic orbits

Periodic heliotropic orbits can be found using initial guesses from the averaged dynamics and a suitable root finding technique applied in the full dynamics [134]. Periodic heliotropic orbits are periodic in all the orbital elements except Ω, which experiences a shift equivalent to the change in f to keep the orbit apoapse towards the Sun. (The orbits are exactly periodic if λ is used as an element instead of Ω.) Equation (5.19) is useful for determining the nearest periodic orbit by resonance (number of spacecraft revolutions per circulation in ω). Using this resonance, a periodic orbit is found by numerically correcting the initial conditions determined by the heliotropic surface. Details on the employed numerical methods and the general utility of using resonant periodic orbits for science orbit design can be found in [165] and [134, 166]. Performing the described process on example B yields the periodic orbit example B_P shown in Figure 5.11. This periodic orbit repeats (relative to the Sun line) every 18 spacecraft revolutions, and the eigenvalues of its monodromy matrix reveal linear stability, at least to the sixth digit. (The largest eigenvalue magnitude is given in Table 5.3.) Future work includes following and mapping these and other similar periodic orbit families.

It is also possible to find orbits which are not heliotropic but are still stable. Instead of Equation (5.14), another constraint could be applied to achieve a different constrained orbit geometry. One such possibility for future work is to allow for different resonances between full periods of Ω and ω, leading to the constraint in Equation (5.25) to produce prograde orbits with an $M{:}N$ resonance in ω and Ω, respectively. Unless $M = N = 1$ the resulting orbits will not necessarily be heliotropic, but these orbits could still be stable and may still be useful as low-altitude science orbits.

$$M\omega + N\Omega = f \tag{5.25}$$

An example of such an orbit is given in Figure 5.12 (notice the near 2:1 resonance in ω and Ω, respectively). The orbit shown in Figure 5.12 is marginally

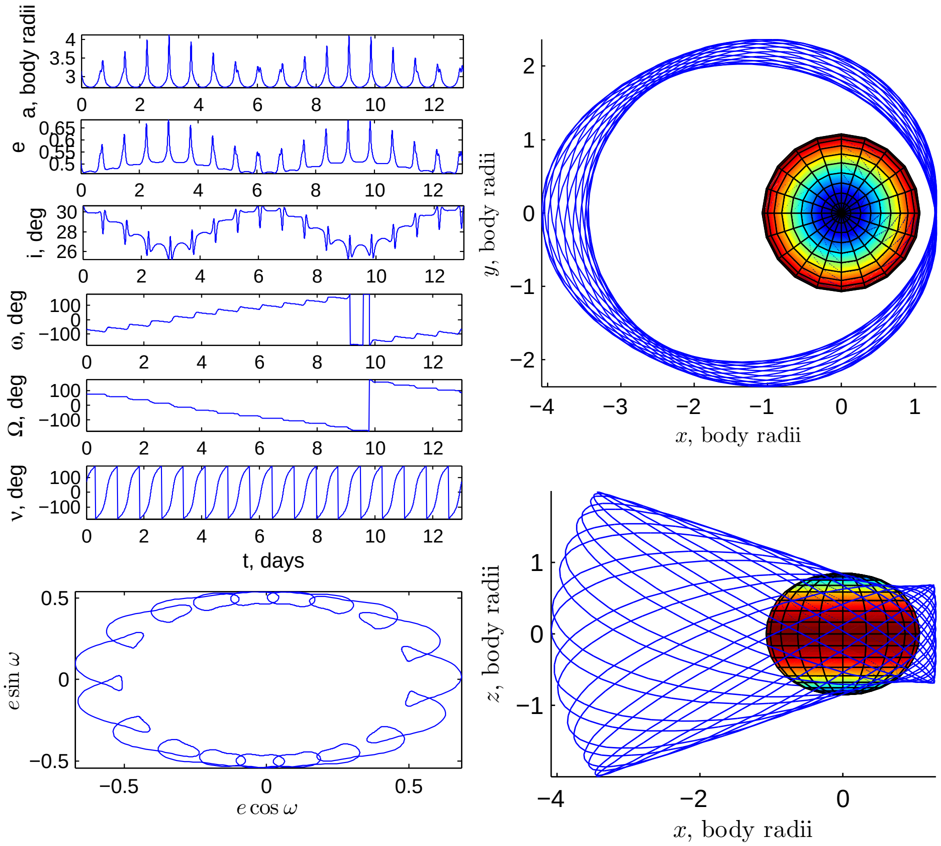


Figure 5.11: Example B_P: Long-lifetime periodic heliotropic orbit corrected from example B. Initial conditions for the orbit are given in Table 5.3

linearly unstable (according to the eigenvalues of its monodromy matrix) but appears to be nonlinearly stable (when propagated for 11,000 orbits). These kinds of orbits can be investigated by using Equation (5.25) to derive a modified doubly-averaged disturbing potential, resulting in a surface of mean element conditions similar to the one developed for inclined heliotropic orbits.

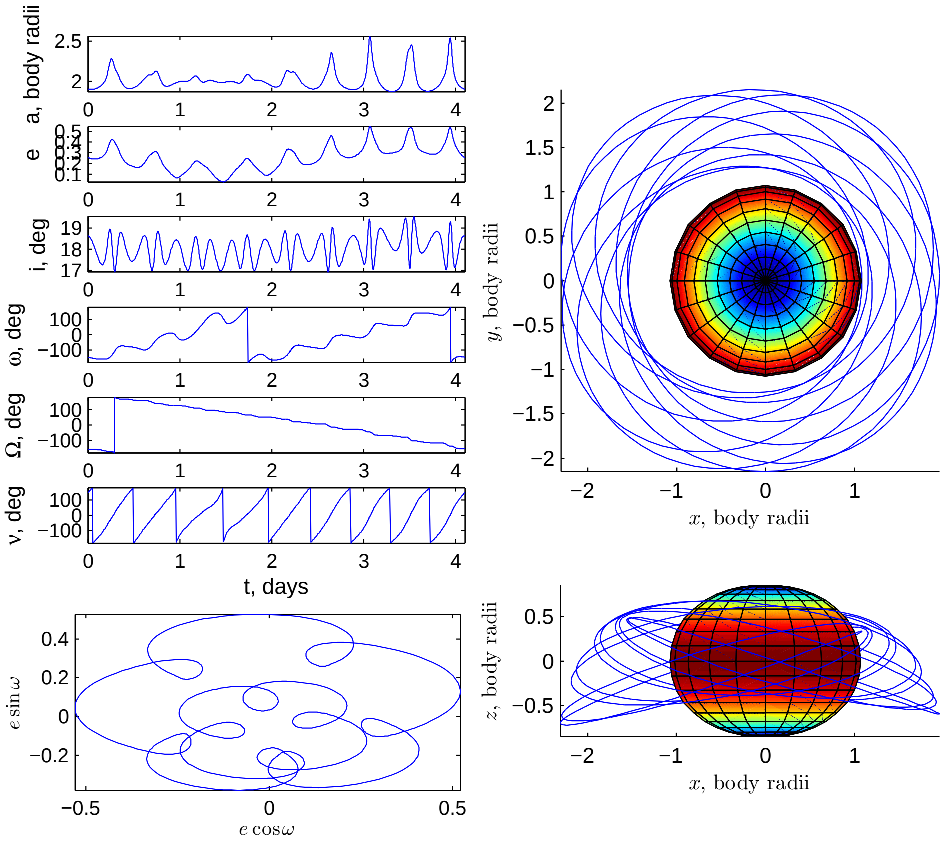


Figure 5.12: Example E: Long-lifetime periodic orbit with similar characteristics to a heliotropic orbit except with a 2:1 resonance in $\dot{\omega}$ and $\dot{\Omega}$. Initial conditions for the orbit are given in Table 5.3

5.5 Implementing High Degree Zonal Gravity

Extending the process of finding 3D heliotropic orbits to higher degree zonal gravity terms leverages the independence of the SRP and zonal gravity terms in the disturbing potential. The constrained doubly-averaged SRP disturbing potential introduced is unaffected by the inclusion of higher degree gravity terms, whose disturbing potential can be determined with Algorithm 5.2. The method for determining the secular effect of J to an arbitrary

Algorithm 5.2 Calculating the secular disturbing potential of the zonal gravity harmonics

```
$\bar{\bar{R}}_{J,sec} = 0$
for $l = 2$ to degree by 2 do
    $R_{J_l}$ : Calculate the potential term from the Legendre polynomial
        $u = \sin(i)\sin(\omega + \nu)$
        $P_l(u) = \dfrac{\frac{\partial^l}{\partial u^l}\left(\left(u^2 - 1\right)^l\right)}{2^l l!}$
                    ▷ There are several alternative methods for determining $P_l(u)$
    $\bar{R}_{J_l}$ : Integrate $R_{J_l}$ over one orbital period, changing the variable of integration to $\nu$
        $\bar{R}_{J_l} = \dfrac{1}{2\pi}\int_0^{2\pi} R_{J_l} dM^* = J_l \left(\dfrac{{R_0}^l}{a^{l+1}}\right) \dfrac{-\mu}{2\pi\left(1 - e^2\right)^{l-1/2}} \int_0^{2\pi} \left(1 + e\cos(\nu)\right)^{l-1} P_l(u)\, d\nu$
    $\bar{\bar{R}}_{J_l,sec}$ : Integrate $\bar{R}_{J_l}$ over period of $\omega$
        $\bar{\bar{R}}_{J_l,sec} = \dfrac{1}{2\pi}\int_0^{2\pi} \bar{R}_{J_l} d\omega$
    $\bar{\bar{R}}_{J,sec} \leftarrow \bar{\bar{R}}_{J,sec} + \bar{\bar{R}}_{J_l,sec}$
end for
return $\bar{\bar{R}}_{J,sec}$
```

degree is summarized in Procedure 5.2, where R_0 is the spherical harmonics normalization radius and ν is the spacecraft true anomaly. Procedure 5.2 is carried out analytically using a symbolic manipulator program (Maple, in the current implementation). Only the even degrees of J are used in the heliotropic orbit calculation because the secular effect of the odd degrees is averaged out to zero by integration on ω [149]. Combining the averaging over ω used to determine $\bar{\bar{R}}_{J,sec}$ with the constrained double averaging on $\bar{\bar{R}}_{SRP}$

assumes their averaging periods are identical. In practice, this requirement leads to the assumption that $\dot{f}$ is small: that is, $\dot{\omega} \approx \dot{\Omega}$.

For example, the four rates required for Eq. (5.18), after following the process in Procedure 5.1 and using Procedure 5.2 to sixth degree, are shown in Eq. (5.26). Here μ is the small body gravitational parameter and γ is the solar radiation pressure acceleration (assumed constant).

$$
\begin{aligned}
\dot{\omega}_{SRP} =& -\frac{3}{4}\left(\frac{e\gamma\cos i}{n^* a\sqrt{1-e^2}} + \frac{\gamma\,(1+\cos i)\sqrt{1-e^2}}{n^* a e}\right)\\
\dot{\Omega}_{SRP} =& \frac{3}{4}\,\frac{e\gamma}{n^* a\sqrt{1-e^2}}\\
\dot{\omega}_{J,sec} =& \frac{15}{4}\,\frac{R_0^2\mu}{\sqrt{\mu/a^3}}\,\frac{1}{\left(e^2-1\right)^6 a^9}\Biggl(\frac{15015}{512}\,{R_0}^4 J_6\left(e^4+\frac{44}{13}\,e^2+\frac{72}{65}\right)\cos^6 i\\
&-\frac{189}{32}\,\Biggl(a^2e^6J_4+\left(-\frac{26}{27}\,J_4a^2+\frac{275}{48}\,{R_0}^2J_6\right)e^4+\left(\frac{725}{36}\,{R_0}^2J_6-\frac{29}{27}\,J_4a^2\right)e^2\\
&+\frac{28}{27}\,J_4a^2+\frac{125}{18}\,{R_0}^2J_6\Biggr){R_0}^2\cos^4 i\\
&+\Biggl(a^4e^8J_2+\left(-4\,J_2a^4+\frac{63}{16}\,{R_0}^2J_4a^2\right)e^6+\left(6\,J_2a^4-\frac{27}{8}\,{R_0}^2J_4a^2+\frac{4725}{512}\,{R_0}^4J_6\right)e^4\\
&+\left(-4\,J_2a^4+\frac{4375}{128}\,{R_0}^4J_6-\frac{81}{16}\,{R_0}^2J_4a^2\right)e^2+9/2\,{R_0}^2J_4a^2+J_2a^4+\frac{805}{64}\,{R_0}^4J_6\Biggr)\cos^2 i\\
&-1/5\,a^4e^8J_2+\left(-\frac{9}{32}\,{R_0}^2J_4a^2+4/5\,J_2a^4\right)e^6+\left(-6/5\,J_2a^4+3/16\,{R_0}^2J_4a^2-\frac{175}{512}\,{R_0}^4J_6\right)e^4\\
&+\left(\frac{4}{5}\,J_2a^4+\frac{15}{32}\,{R_0}^2J_4a^2-\frac{175}{128}\,{R_0}^4J_6\right)e^2-\frac{35}{64}\,{R_0}^4J_6-\frac{1}{5}\,J_2a^4-\frac{3}{8}\,{R_0}^2J_4a^2\Biggr)\\
\dot{\Omega}_{J,sec} =& -\frac{3}{2}\,\frac{R_0^2\mu}{\sqrt{\mu/a^3}}\,\frac{\cos i}{\left(e^2-1\right)^6 a^9}\Biggl(\frac{17325}{512}\,\left(e^4+8/3\,e^2+\frac{8}{15}\right){R_0}^4J_6\cos^4 i\\
&-\frac{105}{16}\,{R_0}^2\Biggl(\frac{75}{16}\,\left(e^4+8/3\,e^2+\frac{8}{15}\right)J_6{R_0}^2+J_4\left(e^2+2/3\right)a^2\left(e+1\right)^2\left(e-1\right)^2\Biggr)\cos^2 i\\
&+\frac{2625}{512}\,\left(e^4+\frac{8}{3}\,e^2+\frac{8}{15}\right){R_0}^4J_6+\frac{45}{16}\,J_4\left(e^2+\frac{2}{3}\right)a^2\left(e^2-1\right)^2{R_0}^2+J_2a^4\left(e^2-1\right)^4\Biggr)
\end{aligned}
\tag{5.26}
$$

5.6 Heliotropic orbit solutions at Bennu

For application at Bennu, the first six even zonal harmonic coefficients $(J_2, J_4, J_6, \cdots, J_{12})$ are determined by applying the method of [167] to the constant-density polyhedral shape model calculated by [155] with the gravitational parameter estimated by [156]. These coefficients are used with an assumed 180° obliquity and a Brillouin sphere radius of 0.2887 km. The magnitudes of these harmonic coefficients and a power fit to them are shown in Figure 5.13. The harmonic coefficients are used to determine the nominal heliotropic surface and the power fit is used for uncertainty modeling, as described below. Nominal parameter values are $\mu = 5.2 \times 10^{-9}\ km^3/s^2$ [156] and $\gamma = 1.2762 \times 10^{-10}\ km/s^2$. The resulting heliotropic orbit solutions at Bennu accounting for the nominal even zonal coefficients J_2-J_{12} are shown in Figure 5.14.

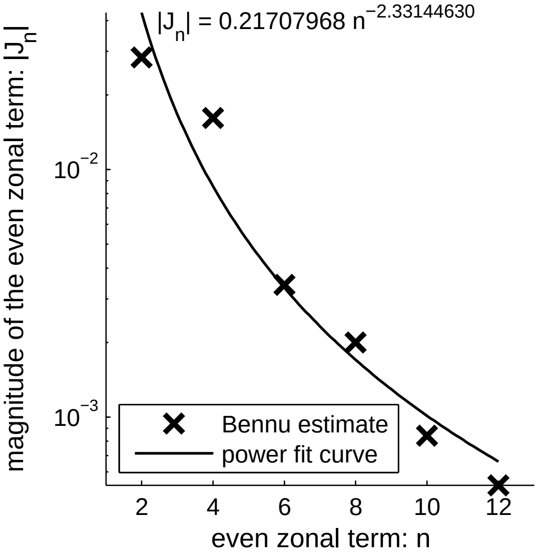


Figure 5.13: Bennu normalized even zonal gravity terms and power fit

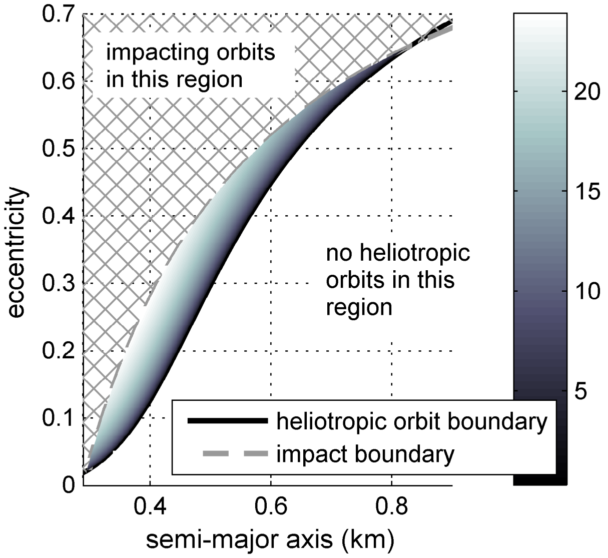


Figure 5.14: Heliotropic orbit solutions at Bennu using even J_2-J_{12}. Surface shading indicates inclination (deg)

Figure 5.14 can be used to quickly determine the bounds of possible heliotropic orbits at Bennu, given the assumptions of the approach and the estimated gravity field from Figure 5.13. For example, the maximum non-impacting mean inclination on the surface is about 21°, with impact defined as mean periapse falling below the Brillouin sphere radius. Similarly, the maximum possible mean semimajor axis is about 2.4 body radii in an equatorial orbit.

In the constrained, doubly-averaged dynamics, heliotropic orbits only exist for eccentricity greater than the heliotropic orbit boundary, which is the equatorial family of heliotropic orbits. The average impact boundary provides a simple upper bound on the possible eccentricities of heliotropic orbits for a given value of semimajor axis. Shading in Figure 5.14 shows the inclination of the valid heliotropic surface between these two boundaries. Note that a spacecraft may go below the impact boundary (Brillouin sphere) even though the average orbital elements stay above the boundary, so a more conservative boundary would likely be used for mission planning.

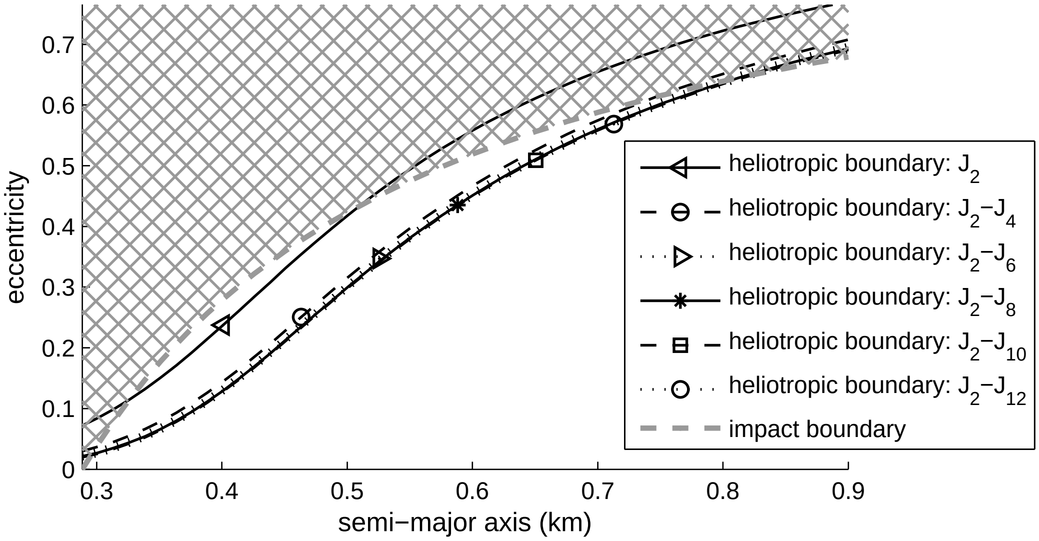


Figure 5.15: The nominal heliotropic boundary shifts as additional degrees of J are included, showing the relative importance of each value at Bennu.

Figure 5.15 illustrates the effect each successive even term in J has on the heliotropic boundary at Bennu. The addition of J_4 makes a significant difference and terms above J_6 appear to have a minor impact only on the nominal boundary.

The surface in Figure 5.14, which shows the heliotropic orbits in the constrained doubly-averaged system, is used to generate initial guesses for heliotropic orbits in the unaveraged, higher fidelity dynamics. As an example, higher fidelity orbits are propagated with the 12×12 spherical harmonics gravity field and the actual 172° obliquity determined by [155], and include SRP and solar gravity perturbations. Figure 5.16 shows one such orbit with initial conditions $\{a, e, i, \Omega, \omega, \nu\} = \{0.5\,\mathrm{km}, 0.38, 21.472^\circ, -90^\circ, 90^\circ, 180^\circ\}$ propagated with an older estimate of Bennu $\mu = 4.057 \times 10^{-9}\ km^3/s^2$, propagated for nearly two full cycles of Ω.

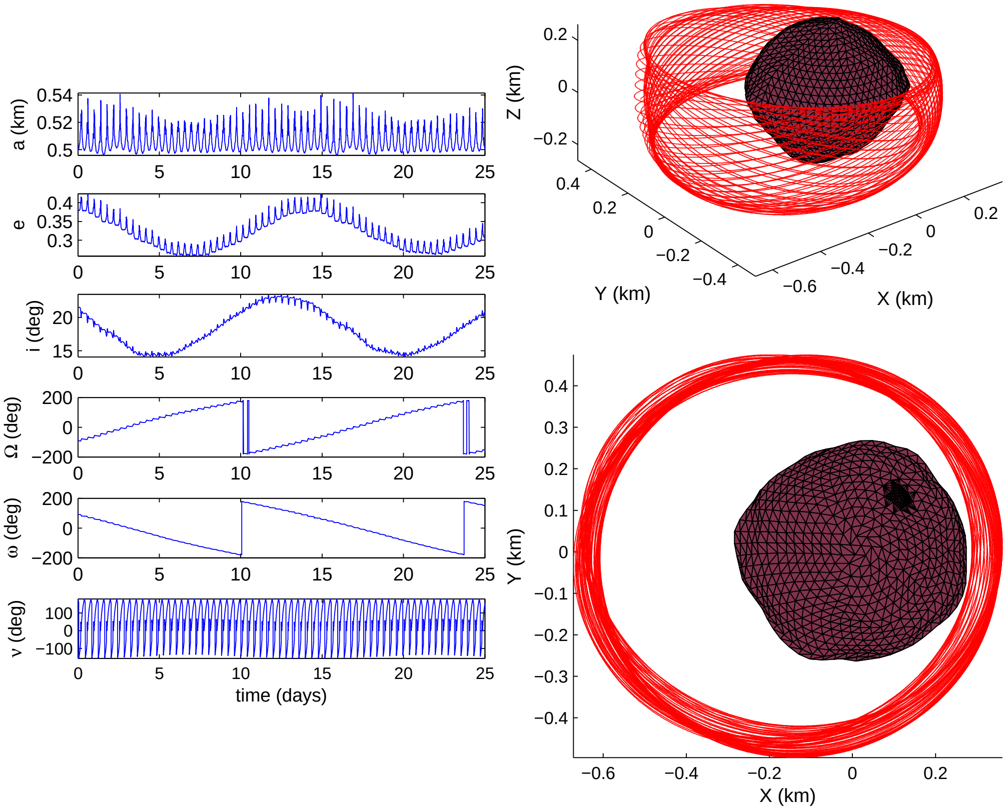


Figure 5.16: Example heliotropic orbit at Bennu integrated in the unaveraged high-fidelity dynamics: including a 12 × 12 spherical harmonics gravity field with a rotating body, 172° obliquity, SRP, and solar gravity – shown in a Sun-synodic frame with the Sun in the negative x-direction

5.7 Sensitivity to gravity field uncertainty

Heliotropic orbits are expected to exist whenever there is a region between the heliotropic orbit boundary and the impact boundary (e.g. the shaded region in Figure 5.14). In order to investigate the effects of gravity field uncertainty on the existence of heliotropic orbits, a Monte Carlo analysis (MC) is performed. The variables being changed as input for each run of the MC are the seven gravity parameters (μ and even $J_2 - J_{12}$) and the SRP acceleration γ, represented as normally-distributed random variables; the output of each MC run is the location of the equatorial heliotropic boundary. An example of the boundary for the nominal case is shown in Figure 5.14. Three different standard deviation cases for J are considered, with the means of the MC given in Figure 5.17.

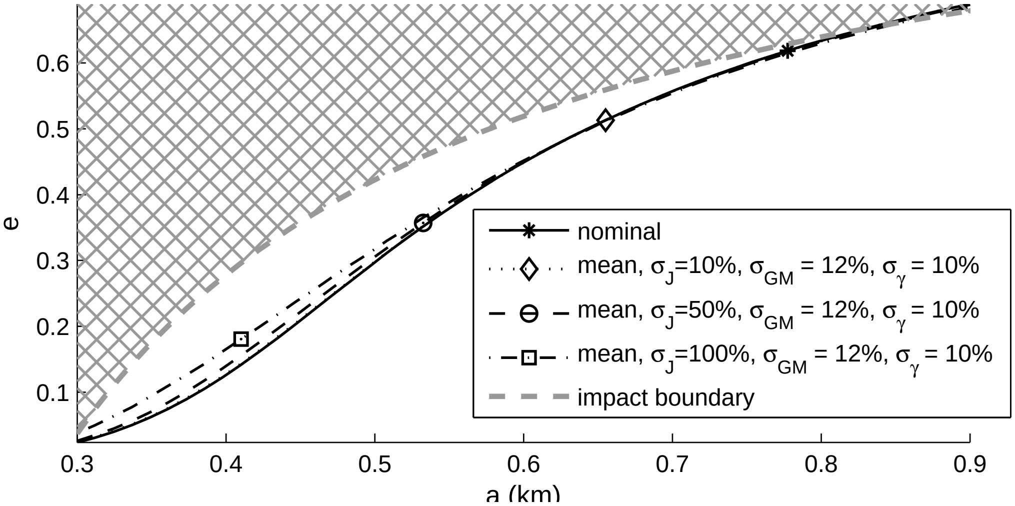


Figure 5.17: Means of three statistical heliotropic boundary cases compared to the nominal case. The mean for $\sigma_J = 10\%$ nearly overlaps the nominal case.

The input random variables (gravity parameters) are normally distributed about their nominal values with standard deviation specified as a percentage of the power curve approximation in Figure 5.13. For μ, the nominal value is 5.2×10^{-9} km^3/s^2 with $\sigma_\mu = 12\%$ of the nominal value [156]. For γ, the nominal value is 1.2762×10^{-10} km/s^2 with $\sigma_\gamma = 10\%$ of the nominal value. For the even terms of the J vector, the mean is the nominal value shown by the "x" marks in Figure 5.13 and the standard deviation is a percentage of the fit value for the same degree. For the standard deviation of J, three different percents are investigated: 10%, 50%, and 100%; corresponding to Figs. 5.18, 5.19, and 5.20, respectively.

Figures. 5.18–5.20 show histograms of the heliotropic boundary location at select values of a. Reference lines in these figures show where the impact, mean, and nominal conditions intersect the histograms. When multiple reference lines intersect one histogram bar in Figs. 5.18–5.20, 'impact' takes priority over 'nominal,' which takes priority over 'mean' in the histogram bar shading.

As an explanation of the process used in the uncertainty analysis, consider the generation of the central histogram in Figure 5.19. First, seven sets of 100,000 normally-distributed random samples are generated to correspond to μ and the even J_2 through J_{12} with the standard deviations for this case ($\sigma = 50\%$ on J). The result is 100,000 independent random parameter sets for Bennu-like gravity fields. Next, a fixed value of a is chosen; in this case $a = 0.6$ km. Then, Procedure 5.1 is applied to each parameter set, with the independent variables for the procedure $a = 0.6$ km and $i = 0$ (the heliotropic boundary is at zero inclination), resulting in a heliotropic value for e. Not every parameter set will necessarily result in a solution (e) because heliotropic orbits may not be possible for every set of gravity parameters (and SRP, which is not varied in the current study). In this example, two of the sets at $a = 0.6$

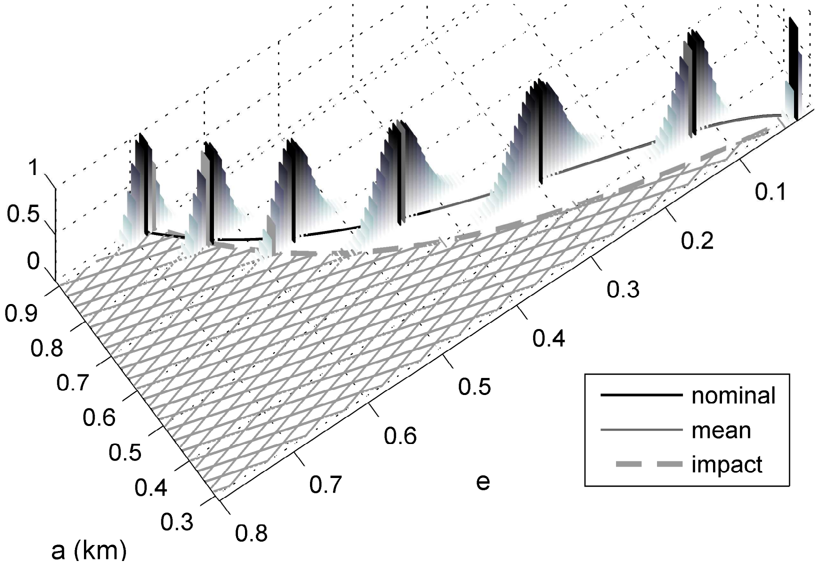


Figure 5.18: Histograms of the heliotropic boundary with $\sigma_\mu = 12\%, \sigma_\gamma = 10\%, \sigma_J = 10\%$. The mean and nominal values nearly overlap. All 100,000 cases evaluated had non-impacting heliotropic orbits at some value of a.

km did not return a solution. The remaining 99,998 values of e form a distribution, whose histogram is generated by creating 150 equally-distributed bins between zero and 0.8 (the cases considered do not exceed $e = 0.8$). The resulting histogram is normalized by the maximum bin count. The same 100,000 random sets of gravity parameters are used to find the heliotropic boundary e for different values of a so that each chosen value of a has its own histogram of the heliotropic boundary. The process described in the example is repeated for each of the three error cases. Although Figs. 5.18–5.20 each show only seven distributions, 50 different values of a were considered for each standard deviation case, with all the results shown in Figure 5.17 and included in drawing conclusions.

As described in the preceding paragraph, it is possible for a heliotropic

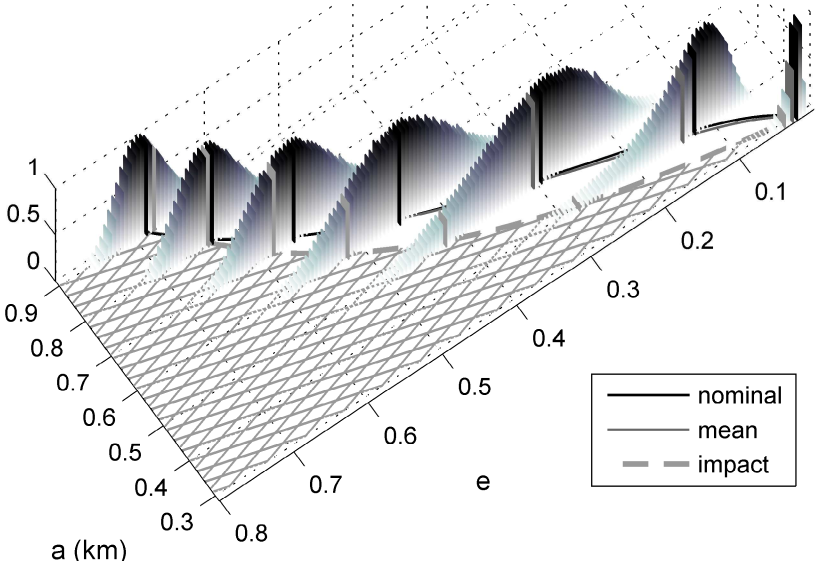


Figure 5.19: Histograms of the heliotropic boundary with $\sigma_\mu = 12\%, \sigma_\gamma = 10\%, \sigma_J = 50\%$. Of the 100,000 cases considered, 98.85% allow for non-impacting heliotropic orbits at some value of a.

orbit boundary not to exist for a given set of gravity parameters. When this occurs, the case is excluded from the histogram, but this exclusion has no significant impact on the displayed results: for $\sigma_J = 10\%$, no cases are excluded for any value of a; for $\sigma_J = 50\%$, $0.0\% - 0.047\%$ of the tested cases are excluded, depending on the value of a; for $\sigma_J = 100\%$, $0.25\% - 2.08\%$ of the tested cases are excluded, depending on the value of a (higher percentage of no-solution values tends to happen at higher values of a).

Figure 5.17 shows that the heliotropic orbits at Bennu are robust to the variations considered, with the mean of the equatorial boundary remaining near the nominal case. The distributions in e are non-Gaussian, and increasingly so for errors with higher standard deviations, as shown in Figs. 5.18–5.20. Other than the distribution shape itself, the difference between the mean and

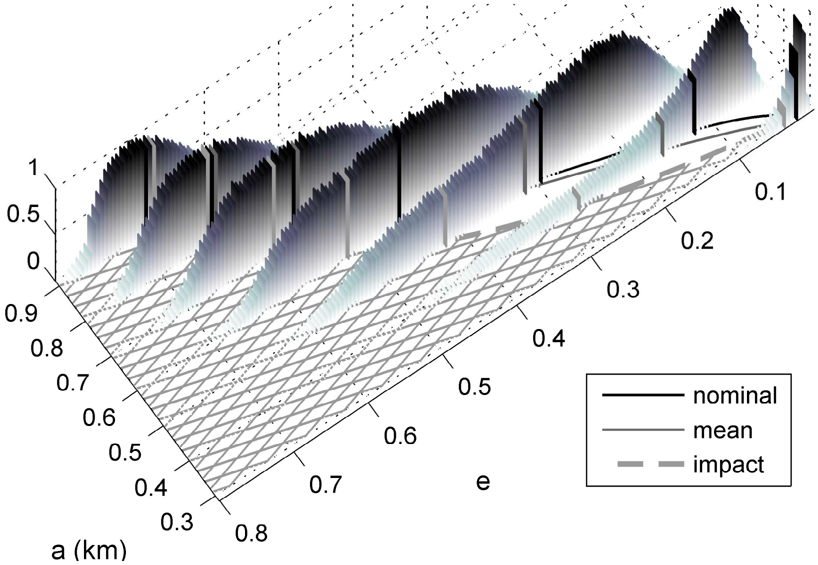


Figure 5.20: Histograms of the heliotropic boundary with $\sigma_\mu = 12\%, \sigma_\gamma = 10\%, \sigma_J = 100\%$. Of the 100,000 cases considered, 88.05% allow for non-impacting heliotropic orbits at some value of a.

the mode (peak) of the distribution – especially visible in Figure 5.20 – provides a good indicator of the non-Gaussian nature of the distribution. Even with this highly non-Gaussian behavior, the mode of the distribution stays below the impact line for the same values of a where the mean is below the impact line. Of all 300,000 hypothetical Bennu gravity model cases considered, about 96% could support heliotropic orbits for some value of a, and of those cases, 100% of the 100,000 $\sigma_J = 10\%$ cases support heliotropic orbits for some value of a.

These uncertainty statistics in the simplified model point to one of the major limitations of practically implementing heliotropic orbits: these orbits are sensitive to body parameters which may not be well defined before the spacecraft arrives at the body. The presence of radar measurements for Bennu provides high confidence in the data available, but most primitive bodies of interest would not have such data. As a result, heliotropic orbits at asteroids are most applicable where the body gravity is sufficiently well defined before the mission or as an extended mission at an applicable body. Alternatively, a mission may be able to fly without nominally using a heliotropic orbit, but with a high "upside potential" in science benefits if a heliotropic orbit can be implemented after characterization of the body from a more distant orbit. In particular, missions which fly a flash LIDAR or active spectrometer may see this as an advantageous possibility because the quality of data can be greatly increased by maintaining a long-lifetime low-altitude orbit. Gravity field measurement could also benefit significantly because the long orbit lifetimes provide for a longer maneuver-free measurement period relatively close to the gravitational perturbations.

Chapter 6

Conclusions

This dissertation has focused on advancing the state of the art in techniques and software used for preliminary design of missions to some challenging destinations, particularly the outer planets and small bodies. Section 1.4 outlines the primary contributions in the four major areas addressed; each of these four is briefly reviewed in this chapter with a view towards applicability and future work.

6.1 Innovating the multiple gravity assist global search

The pathsolving problem is solved with a global search algorithm that combines a grid search with multi-level pruning, as described in Chapter 2. The algorithm approaches the pathsolving as a cascade of leg-solving and node-solving subproblems which are computed breadth-first. The breadth-first approach allows straightforward comparison between all trajectories at the same stage and pruning based on that comparison.

One of the primary advantage of the presented method is its flexibility to incorporate different types of trajectory legs and nodes: v_∞ leveraging, shape-based low thrust, and optimized DSM legs have been implemented with the presented algorithm in a pathsolving tool called *Explore*. Each leg needs to be able to solve a BVP similar to the traditional Lambert problem, so many other options for solving a particular leg are possible. For example, different LT trajectory models could be included as long as they can robustly provide a solution to the BVP. Because the optimized DSM takes significantly longer to compute than a specialized maneuver routine like the VILT, another possible future extension would be other classes of maneuvers; for example, adding a broken plane maneuver routine and an apoapsis plane change maneuver routine would quickly provide other known efficient maneuvers. Implementations for calculating these maneuvers could use additional information available from the search like the inbound $\mathbf{v}_\infty$ to the preceding node and the orbit planes of the bodies to be encountered.

In addition to the flexibility of using different kinds of trajectory legs, the legs can be connected by ballistic flybys, body rendezvous with stay times, or $n\pi$ sequence nodes. Other nodes could also readily be included as well – powered flybys or aerogravity assists, for example. The nodes solve a BVP with provided inbound and outbound $\mathbf{v}_\infty$ and are associated with a single ephemeris object. Although an ephemeris object is nominally a celestial body, further flexibility in the search algorithm comes from the potential to use arbitrary points in space or spacecraft as nodes. Using this flexibility to implement multi-spacecraft missions is one area worth further study; a secondary spacecraft on an externally-defined trajectory could be a node that deploys a primary spacecraft to start the search as illustrated in Figure 6.1. Alternatively, the secondary spacecraft-as-node could be targeted by the primary spacecraft during the trajectory search. One way this capability has been successfully used is by connecting an interplanetary trajectory with a planetocentric capture trajectory; the interplanetary trajectory is the first node of the search, with the search starting by a small maneuver deviating from this trajectory within a given range of TCM times set as launch times. This case is also conceptually illustrated by Figure 6.1.

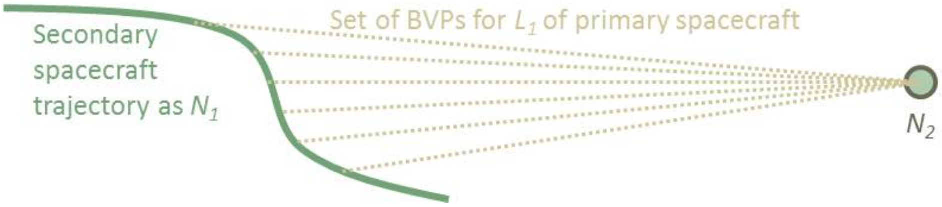

Set of BVPs for L_1 of primary spacecraft
Secondary spacecraft trajectory as N_1
N_2

Figure 6.1: Conceptual illustration of a predefined secondary spacecraft trajectory used as a node within the MGA GS, either to deploy a primary spacecraft or as the transition from interplanetary cruise to the start of a capture sequence search.

The critical role pruning plays in the GS process is also discussed in Chapter 2. Which quantities make the most effective pruning conditions for a particular mission merits further investigation, especially in the context of performance index pruning. Additionally, it has been observed that Pareto sorting with two dimensions which are positively correlated is generally undesirable because each rank often contains only a couple solutions. So further research may involve an automated way to prevent this from occurring during a search or a study of which quantities make the most effective sets for Pareto sorting.

6.2 Including maneuvers and v_∞ leveraging

A boundary value formulation for v_∞ leveraging transfers (VILTs) is described and characterized in Chapter 3; this method enables the fast calculation of transfers including a single impulsive maneuver at a spacecraft apse. These maneuvers, based on VILTs, are a broader class of velocity-aligned apsidal maneuvers (VAM). The resulting transfers naturally incorporate bodies with ephemeris locations and enable leveraging-like transfers between different bodies.

Casting the v_∞ leveraging transfer as a boundary value problem addresses many of the difficulties of including maneuvers in a broad, fast trajectory search. The general VILT reformulation presented in Chapter 3 is based on the solution to a one-dimensional root-solve that requires a similar computational burden and solution process as that of the conventional multi-revolution Lambert problem. The use of the free parameter for tangent and nontangent VILTs requires some understanding of the fundamental mechanics and the types of trajectories expected, but the ability to quickly and effectively include VAMs in the trajectory search is a major advantage, greatly expanding the search space without an exponential increase in run time.

The maneuvers that can be included with the presented VILT routine map to various maneuver types and families, as demonstrated by the different examples. Of particular interest are cases that do not classify as traditional VILTs but can still be found with the same method – for example, orbit insertion maneuvers. In such cases, the limitation to a single maneuver per leg can be detrimental so a possible extension for future work is the inclusion of two apsidal maneuvers between a pair of boundary conditions. Orbit insertion sequences would benefit because the insertion maneuver and the periapse raise maneuver are generally only separated by a coast arc.

Another limitation of the presented VILT method is the assumption that the spacecraft plane is unchanged by the maneuver. In ephemeris-case trajectories, a small plane adjustment can make a big difference in targeting the next encounter, especially for near-π transfers, so a means to incorporate this would be useful. The same TOF based solution structure used for VILTs may be applicable to other classes of maneuvers as well; this is another area worth further investigation.

6.3 Including $n\pi$ sequences

Chapter 4 demonstrates how $n\pi$ transfer sequences can be included in a broad trajectory search without an exponential increase in branches to be considered that would conventionally result from the addition of the many extra DOFs. This inclusion is accomplished by formulating the $n\pi$ sequence design space as a boundary value problem on the v_∞ sphere. The boundary value problem can be defined in the context of a sequential search by treating an $n\pi$ sequence as a gap in time and possibly body states, then evaluating the gap when both boundary conditions have been determined by the primary trajectory search. This algorithm makes the $n\pi$ sequence a "node" in the context of the MGA GS described in Chapter 2.

Two separate algorithms are presented for pathfinding and pathsolving on the v_∞ sphere: the first is a two-level deterministic approach that separates pathfinding and pathsolving, and the second algorithm is a hybrid stochastic sphere-walking approach. The hybrid method is more suitable for very long sequences because there is no need to keep track of a combinatorially large number of possible sequences; although the stochastic nature requires tracking many sequences, this number is set by the user and does not grow with the number of potential transfers in the sequence – as it does with the deterministic approach.

Having both boundary values of the $n\pi$ sequence helps reduce the trajectory search options, but implementing with one boundary condition may be preferable in some cases, such as when the $n\pi$ sequence is the first or last part of the trajectory. Such cases are a good avenue for future research. The sphere-walking algorithm presented is amenable to a single boundary value implementation but the deterministic method presented requires both $\mathbf{v}_\infty$.

Another promising option for extending the presented $n\pi$ sequence method is the inclusion of more than one odd $n\pi$ transfer in a sequence. Although not conceptually different from the method presented, the implementation of such a case is complicated by the additional change in v_∞, as described in Chapter 4. One potential benefit of such a sequence is that a v_∞ leveraging effect could be achieved at a body with an eccentric orbit by using two odd $n\pi$ transfers with different associated resonances. Although such a case would be heavily constrained and may be rare in practice, it merits further investigation.

6.4 Investigating heliotropic orbits

The challenge of finding stable orbits at asteroids is investigated for the case of combined SRP and zonal gravity perturbation in Chapter 5. Sun-frozen and heliotropic orbits are promising categories of orbits in this dynamical regime. The existence and average orbital elements of inclined heliotropic orbits are found using a novel constrained double averaging technique. This technique is also used to determine a maximum upper bound on inclination of about 46° for heliotropic orbits around oblate bodies where J_2 is the dominant zonal harmonic.

The extension of heliotropic orbits into the 3D space has broader implications because these orbits occur naturally for circumplanetary dust and have also been proposed for HAMR spacecraft at Earth. One such extension worth further consideration is the distribution of particles in Saturn's E-ring or “Charming” ringlet, which show heliotropic characteristics. The theory developed could be used to predict the mean elements of ring particles, and such a prediction compared against observations. This type of analysis is complicated by the presence of other perturbations and by the fact that the source mechanism for the ring – cryovolcanic jets on Enceladus for the E ring – may constrain the particle orbital elements more than the heliotropic dynamics.

There may also be other dynamical systems which would benefit from an application of constrained averaging, as done with the heliotropic orbits. In particular, other systems which would yield a trivial solution if averaged in the general case may yield insightful results when averaged with a constraint.

Within the space of heliotropic orbits, there are some periodic orbits. In addition to providing interesting point solutions, these orbits allow a more rigorous stability analysis. Periodic orbits also present another means of finding long-lifetime orbits in the presence of multiple perturbations. As demonstrated by one example within Chapter 5, there may be periodic orbits which are not heliotropic but still maintain long orbit lifetimes; these types of orbits would be interesting to study in more depth, both to find more such orbits in the simplified model and to discover if these orbits persist with more realistic body and perturbation models.

In addition to the these broader applications, the investigation of heliotropic orbits around a model of Bennu shows that such orbits also warrant further study. Including higher degree zonal gravity terms enables a more accurate investigation at small bodies, which tend to have more significant contributions from these terms than planetary bodies do. Additional characterization of perturbations as well as analysis of controllability of spacecraft on heliotropic orbits must precede flying a heliotropic asteroid orbiter, so these are areas of suggested further study. Such additional study will benefit from the groundwork laid in Chapter 5 for determining the likelihood of existence of a heliotropic orbit and for finding initial conditions for higher fidelity modeling.

Appendices

Appendix A

Notation, Acronyms, and Abbreviations

A.1 Sequence abbreviations

Me	Mercury
V	Venus
E	Earth
M	Mars
J	Jupiter
S	Saturn
U	Uranus
N	Neptune
P	Pluto
Ce	Ceres
Ti	Titan
C	Callisto
Eu	Europa
G	Ganymede
I	Io
Te	Tempel-1
Wi	Wild-2
Ve	Vesta
It	Itokawa
Be	Bennu
Ju	1999 JU3
Ch	Churyumov-Gerasimenko
Ha	Halley
Gr	Grigg-Skjellerup
Har	Hartley-2

A.2 Named MGA pathfinding, pathsolving, and pruning tools

EPIC	Stochastic/deterministic hybrid pathsolving
Explore	Deterministic sequential pathsolving, introduced in Chapter 2
GASP	(Gravity Assist Space Pruning) Deterministic pruning for flybys
IGATO	Stochastic GO pathfinding/solving with grid MS
IMAGO	(Interplanetary Mission Analysis Global Optimization) Pathfinding/solving tool based on EPIC
MAnE	(Mission Analysis Environment) Commercial local optimizer with some pathsolving capability
MDTOP	Stochastic pathsolving
PAMSIT	Deterministic sequential pathsolving for AGA trajectories
SOURCE	Deterministic parallel pathsolving
STOUR	(Satellite Tour Design Program) Deterministic sequential pathsolving

A.3 Acronyms

ACO	Ant Colony Optimization
AGA	Aerogravity Assist
BB	Branch and Bound
BVP	Boundary Value Problem
coop	Cooperative Methods (Combined Methods)
DE	Differential Evolution
DOF	Degree of Freedom
GA	Genetic Algorithm
GO	Global Optimization
GS	Global Search
GTOC	Global Trajectory Optimisation Competition
HAMR	High Area-to-Mass Ratio
HDDP	Hybrid Differential Dynamic Programming
IB-VILT	Inter-Body v_∞ Leveraging Transfer
LPE	Lagrange Planetary Equations
LT	Low-thrust
LU	Normalized Length Unit
MBH	Monotonic Basin Hopping
MCS	Multivariate Coordinate Search
MGA	Multiple Gravity Assist
MS	Multiple Start/Restart
NLP	Nonlinear Programming
PSO	Particle Swarm Optimization
SA	Simulated Annealing
SRP	Solar Radiation Pressure
SOI	Sphere of Influence
TU	Normalized Time Unit
VAM	Velocity-Aligned Apsidal Maneuver
VILM	v_∞ Leveraging Maneuver
VILT	v_∞ Leveraging Transfer
YORP	Yarkovsky-O'Keefe-Radzievskii-Paddack effect

A.4 Symbols

Symbols: Part 1

a	Semi-major axis (km or LU, depending on the context)
B	Node or body encounter (ordered) in MGA sequence
b	One particular body (not ordered) in MGA sequence
C	Change: Direction of a VILT with respect to change in e, either increasing (+1) or decreasing (-1)
D	Domain: Specifies whether a VILT is Exterior (+1) or Interior (-1)
E	Eccentric Anomaly (rad unless specified as deg)
e	Eccentricity
F	Number of solutions that a particular set of VILT inputs has, which may be zero, one, or in a rare event more than one
h	Altitude, normally of a flyby (km)
i	Inclination (rad unless specified as deg)
K	Integer number of full spacecraft revolutions in a ballistic coasting arc
k	Pareto rank of an MGA trajectory compared to other trajectories
L	Leg or transfer in MGA sequence
M	Number of approximate body orbit revolutions about the primary
m	spacecraft mass (kg)
M^*	Mean anomaly (rad unless specified as deg)
N	Number of spacecraft orbit revolutions for a transfer (complete for ballistic transfers, approximate for v_∞ leveraging transfers)
n	A positive integer
n^*	Orbital mean motion (rad/s)
O	Offset: Whether the t of a transfer requires an extra $T/2$ (+1) or not (-1) after taking into account K
p	Semi-latus rectum (km)
R	Discretization resolution along a body orbit, normally defined in points per body orbit (ppr)
r	Distance from the primary body (km or LU depending on the context). On a flyby hyperbola, the flyby body is primary.
S	Solution: Which VILT solution to solve for, the lower value of r_C (1) or a higher value (2), if it exists
s	Solution counter or index for a list of solutions
T	Orbital period (s)
TOF	Time of flight of the entire VILT (s)
t	Time of flight of a specific spacecraft arc: to which arc t applies is defined by the subscripts (s)

Symbols: Part 2

V	Velocity magnitude in general (km/s or LU/TU depending on context)
v_∞	Excess hyperbolic speed of a spacecraft relative to a body, the magnitude of the $\mathbf{v}_\infty$ (km/s or LU/TU depending on context)
ΔV	Magnitude of a spacecraft velocity maneuver (km/s)
δ	Turn angle of a flyby
θ	Counterclockwise angle from the first encounter to the second encounter, less than 2π rad (rad)
μ	Gravitational parameter (km^3/s^2 or LU^3/TU^2 depending on context)
ν	True anomaly (rad unless specified as deg)
ξ	Angle between an encounter and the non-leveraging apse (rad)
Ω	Orbit longitude of the ascending node (rad unless specified as deg)
ω	Orbit argument of periapsis (rad unless specified as deg)
Φ	Flight path angle (rad)

A.5 Subscripts and Superscripts

B	Pertaining to the body at an encounter. In Chapter 3, B always refers to the low encounter.
C	Pertaining to the spacecraft, defined at the common apse (the leveraging apse) of a VILT in Chapter 3
E	A placeholder for L or H when an equation is to be applied to both encounters
H	Pertaining to or defined on the high half of a VILT (the transfer with the high encounter)
I	Inbound to a flyby (or encounter or node)
L	Pertaining to or defined on the low half of a VILT (the transfer with the low encounter)
O	Outbound from a flyby (or encounter or node)
P	Pertaining to or defined at orbit periapse
Pr	Pertaining to the primary body (e.g. the Sun for interplanetary transfers)
S	Pertaining to a whole sequence of $n\pi$ transfers (in Chapter 4)

Appendix B

Publications

Refereed Journals

Published

Demyan Lantukh, Ryan P. Russell, and Stephen B. Broschart. "Heliotropic Orbits at Oblate Asteroids: Balancing Solar Radiation Pressure and J2 Perturbations," Celestial Mechanics and Dynamical Astronomy, Volume 121, Issue 2, pp 171-190, 2015. DOI: 10.1007/s10569-014-9596-x — Chapter 5

Demyan Lantukh, Ryan P. Russell, and Stefano Campagnola. "The V-Infinity Leveraging Boundary Value Problem and Application in Spacecraft Trajectory Design," Journal of Spacecraft and Rockets, Volume 52, Issue 3, pp 697–710, 2015. DOI: 10.2514/1.A32918 — Chapter 3

In Review

Demyan Lantukh, Ryan P. Russell, and Stephen B. Broschart. "Heliotropic Orbits at Asteroids with Zonal Gravity Perturbations: Application at Bennu," Journal of Guidance, Control, and Dynamics. Submitted: 23 January 2015 — Chapter 5

Planned

Demyan Lantukh and Ryan P. Russell. "Multi-Objective Search for Multiple Gravity Assist Trajectories," Journal TBD — Chapter 2

Demyan Lantukh and Ryan P. Russell. "Automated Inclusion of n-pi Transfers in Gravity-Assist Flyby Tour Design," Journal TBD — Chapter 4

Conference

Demyan Lantukh and Ryan P. Russell. "Multi-Objective Search for Multiple Gravity Assist Trajectories," AAS/AIAA Astrodynamics Specialist Conference, 9-13 August 2015, Vail, CO — Chapter 2

Demyan Lantukh, Ryan P. Russell, and Stephen B. Broschart. "Heliotropic Orbits at Asteroids: Zonal Gravity Perturbations and Application at Bennu," 25th AAS/AIAA Space Flight Mechanics Meeting, 11-15 January 2015, Williamsburg, VA — Chapter 5

Demyan Lantukh, Ryan P. Russell, and Stephen B. Broschart. "Heliotropic Orbits at Oblate Asteroids: Balancing Solar Pressure and J2 Perturbations," *In* Advances in the Astronautical Sciences, volume 150, 2014 — Chapter 5

Demyan Lantukh and Ryan P. Russell. "Automated Inclusion of n-pi Transfers in Gravity-Assist Flyby Tour Design," *In* Advances in the Astronautical Sciences, volume 143, 2012 — Chapter 4

Demyan Lantukh, Ryan P. Russell, and Stefano Campagnola. "Automated Inclusion of v-infinity Leveraging Maneuvers in Gravity-Assist Flyby Tour Design," AIAA/AAS Astrodynamics Specialist Conference, 13-16 August 2012, Minneapolis, MN — Chapter 3

Appendix C

Solution to GTOC6 by the University of Texas at Austin Team

Department of Aerospace Engineering and Engineering Mechanics,
The University of Texas at Austin

Team Leads Ryan P. Russell • Demyan V. Lantukh
Team Contributors Ricardo León Restrepo • Drew Jones • Nick Bradley • Vivek Vittaldev • Kevin Bokelmann • Sonia Hernandez • Etienne Pellegrini • Nitin Arora • Gregory Johnson • David Ottesen • Marty Brennan • Cesar Ocampo (consultant)

The Global Trajectory Optimisation Competition (GTOC) is a regular international competition to solve challenging spacecraft trajectory problems invented to test both tools and trajectory designers [65]. Begun in 2005 by the European Space Agency Advanced Concepts Team, the competition encourages the development and discussion of global optimization techniques for spacecraft trajectories.

C.1 Problem statement

The 6th GTOC, conducted September-October 2012, aimed to map the four Galilean moons (Io, Europa, Ganymede, and Callisto) using flybys. Each moon was split into 32 faces modeled after a soccer ball (a "football grid" in the terms of the organizers). Each face was assigned a point value which was scored the first time a flyby periapse occurred over that grid face within a specified altitude range.

The spacecraft begins at Jupiter SOI and is captured at Jupiter using a combination gravity assists in a sequence like those found by [52]. Energy reduction and science tours follow capture. The trajectory is constrained to occur within a specified time and mass budget. Final mass is reduced by using fuel in thrusting and by incurring a radiation dose penalty each flyby that depends on the spacecraft orbit. Problem details can be found in [65].

C.2 Submitted solution overview

The University of Texas team took 3^{rd} place of 36 registered teams in GTOC6, the highest-placing team from the United States. The final score of 267/324 came from 97 flybys. A mistake in grid face identification cost 5 points on the score (submitted as 272) but did not affect the final ranking. The submitted solution had both mass and time margin so the solution deadline was the primary limiting factor. Solution properties are given in Table C.1 and the entire trajectory is shown in Figure C.1. The final sequence is:

$$\text{SOI-I-G-VILM-G-VILM-G-VILM-G}^{5}\text{-C}^{11}\text{-G}^{16}\text{-Eu}^{9}\text{-I}^{32}\text{-Eu}^{20}\text{-Eu}$$

Table C.1: Flyby and points breakdowns for the GTOC6 solution submitted

Body	Flybys	Patches (of 32)
Callisto	11	11
Ganymede	24	23
Europa	29	28
Io	33	31

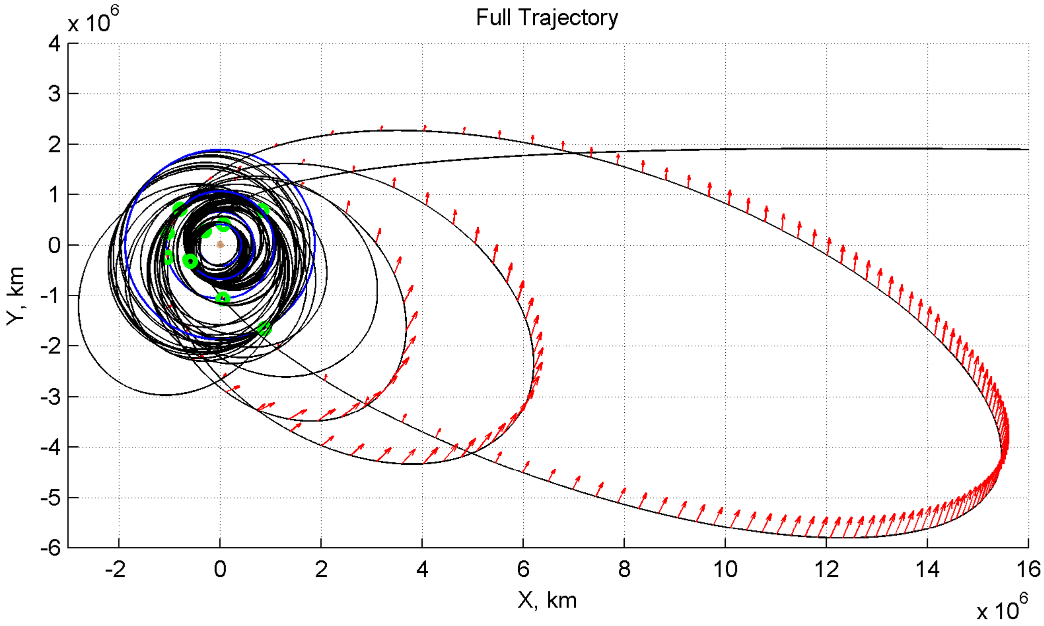


Figure C.1: Galilean moon mapping mission trajectory, the 3^{rd} place solution to the GTOC6 problem.

C.3 Summary of solution methodology

The global trajectory search for Galilean moon mapping trajectories was conducted in *Explore*. As detailed in Chapter 2, two levels of searches were conducted: The first search provided the trade space for flyby-aided capture to a VILT. A few numerical experiments showed that three or four VILTs were sufficient, so the second level search constrained the capture near the best portions of the trade space and then continued the entire tour. The fact that the entire trajectory from before capture through various flybys and maneuvers, from orbit periods on the orders of hundreds of days to orbit periods on the order of a couple days could be found with a global search tool is a testament to the versatility of *Explore*.

Considerable study of the solution space was required in order to properly constrain the global search. While *Explore* was being set up for the problem, several statistical studies were conducted on many randomly-generated cases to tune the $n\pi$ sequence pathfinding/pathsolving method described at the end of Chapter 4 and to determine how much time should be allocated to each of the four moons. Moon visit order was determined by intuition along with trial and error.

Once the problem could be run in *Explore*, the performance indices still needed to be tuned to capture the correct portion of the solution space. As described in Chapter 2, this performance index tuning is not necessarily intuitive but is among different problems since all parameters are physical quantities. If more computational resources could be applied, a more brute-force approach that did less pruning could have been attempted.

The solution from *Explore* contained the entire sequence from SOI to the last Europa encounter but the VILTs were impulsive maneuvers. A local LT optimizer based on differential dynamic programming (HDDP) [168, 169] was applied to convert the VILTs to LT arcs which matched the VILT flyby conditions exactly. This seven-state matching at the start and end of the LT VILTs cost some fuel but meant that the rest of the sequence was unaffected and did not need to be modified as a result of the conversion to LT.

C.4 Details of submitted solution

The solution is composed of seven distinct phases. The last six phases are each composed of an $n\pi$ sequence followed by a nonresonant transfer. Details of the resonant tour implementation are provided in Chapter 4. All phases are computed sequentially and automatically; the user provides performance indices and constraints but does not need to manually patch different phases.

1. Capture and low-thrust v_∞ leveraging
2. First Ganymede resonant tour to Callisto
3. Callisto resonant tour to Ganymede
4. Second Ganymede resonant tour to Europa
5. First Europa resonant tour to Io
6. Io resonant tour to Europa
7. Second Europa resonant tour to Europa

Figure C.2 provides the time histories of spacecraft mass and v_∞ along the whole trajectory, with vertical lines indicating the boundaries between trajectory phases. Figures C.3-C.9 show each phase of the trajectory, making the individual resonant tours visible.

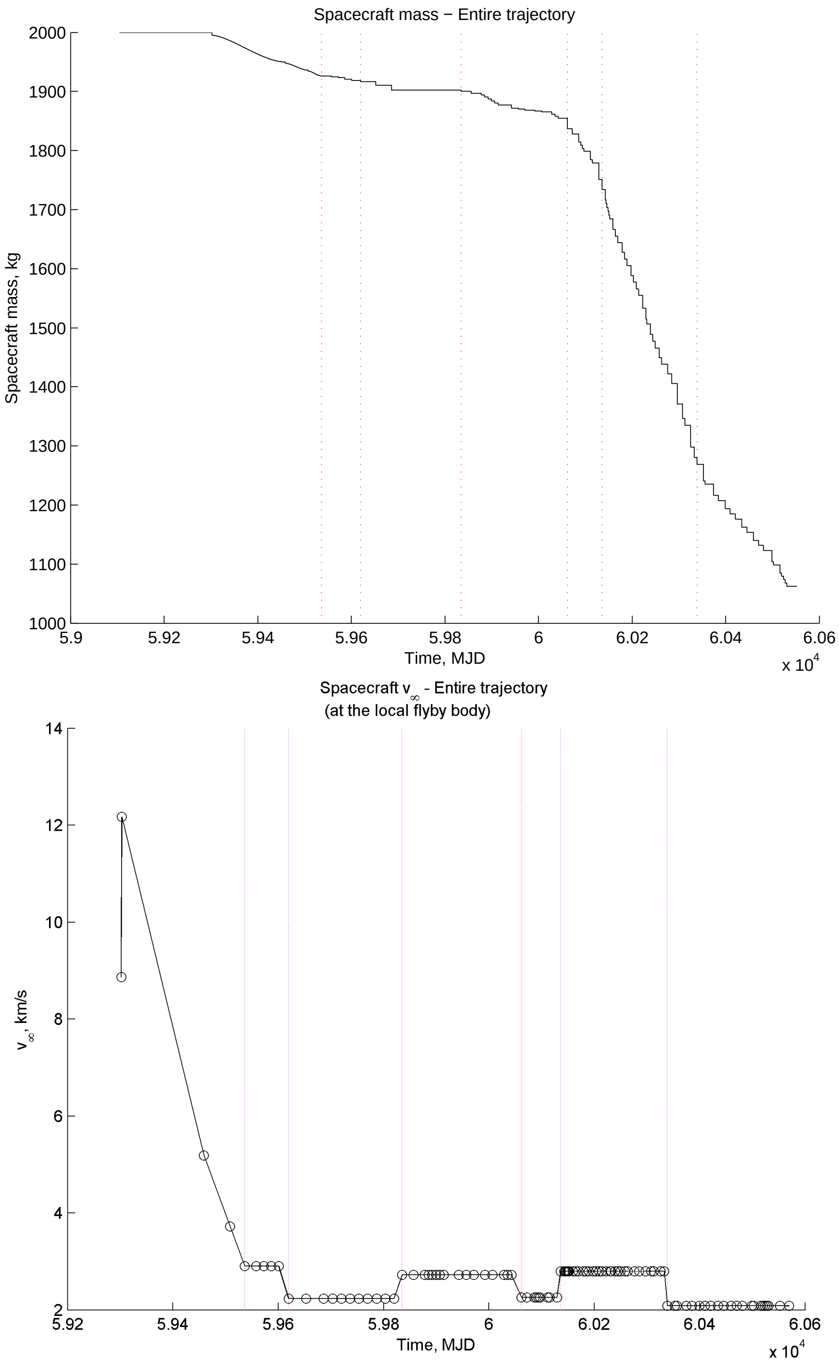


Figure C.2: Galilean moon mapping mission time history of mass and local v_∞ along the entire trajectory. The LT-VILT had a major impact on v_∞ with reasonable cost in propellant mass.

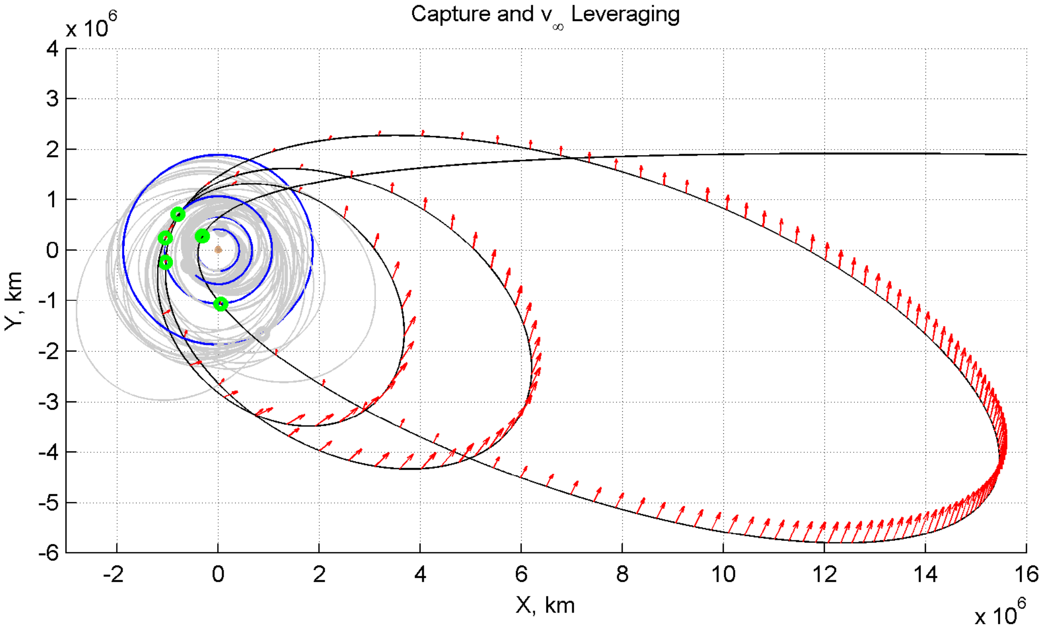


Figure C.3: Galilean moon mapping trajectory for phase 1 of the presented GTOC6 solution

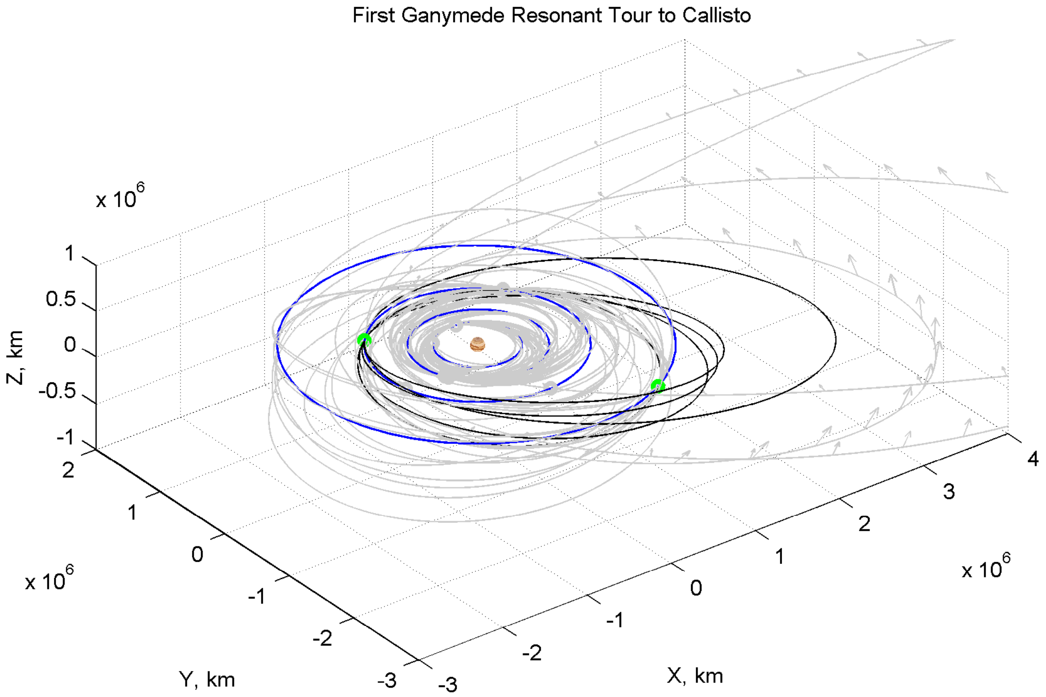


Figure C.4: Galilean moon mapping trajectory for phase 2 of the presented GTOC6 solution

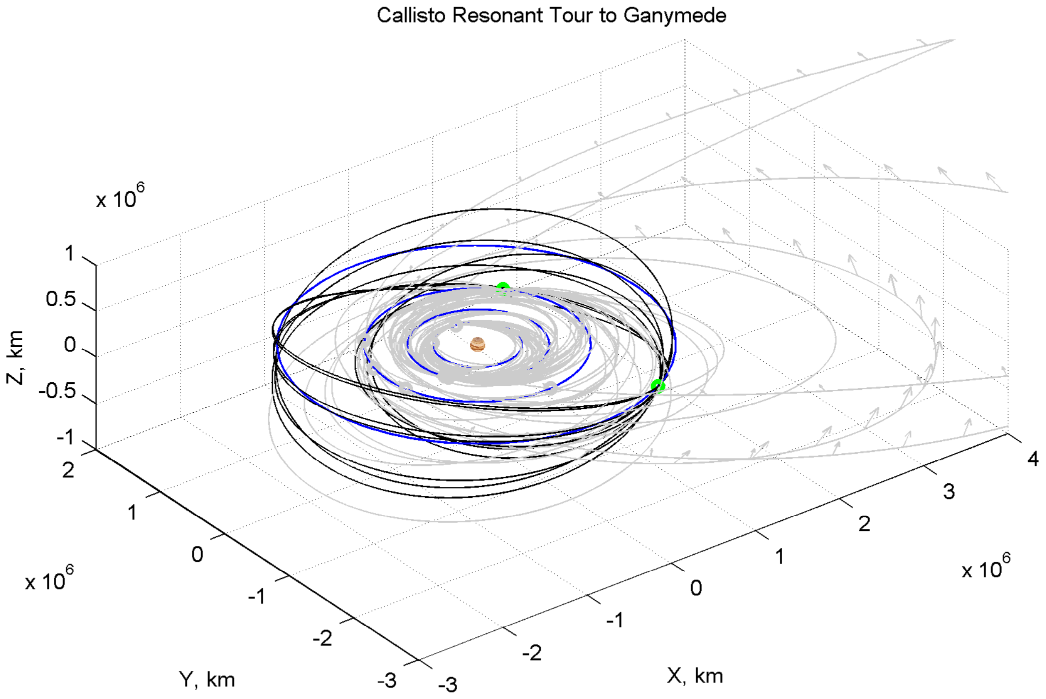


Figure C.5: Galilean moon mapping trajectory for phase 3 of the presented GTOC6 solution

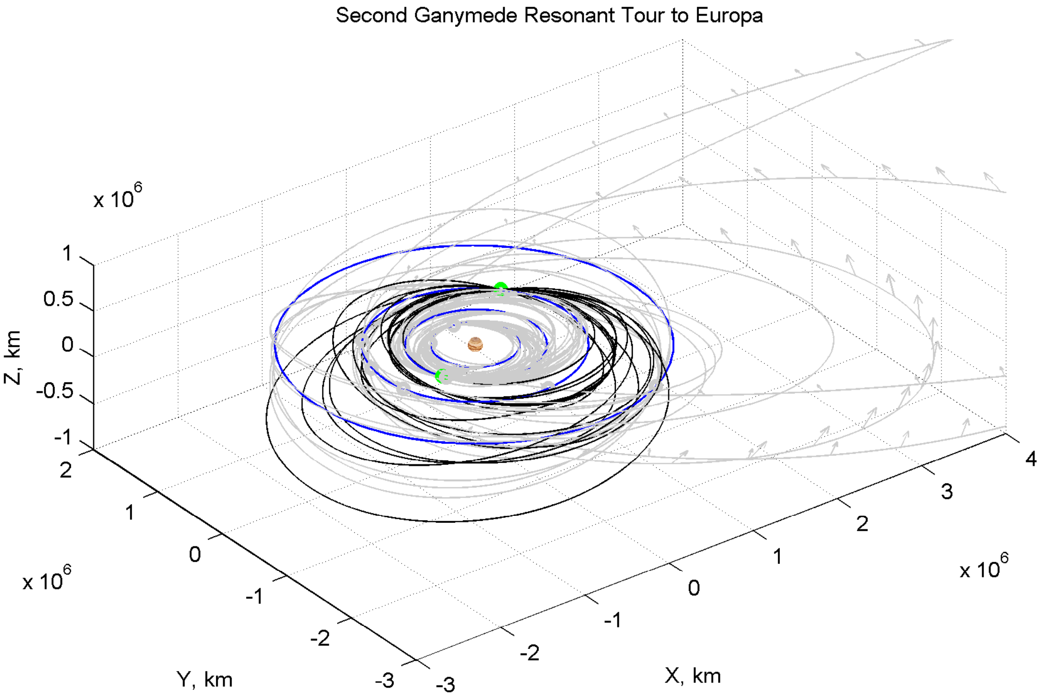


Figure C.6: Galilean moon mapping trajectory for phase 4 of the presented GTOC6 solution

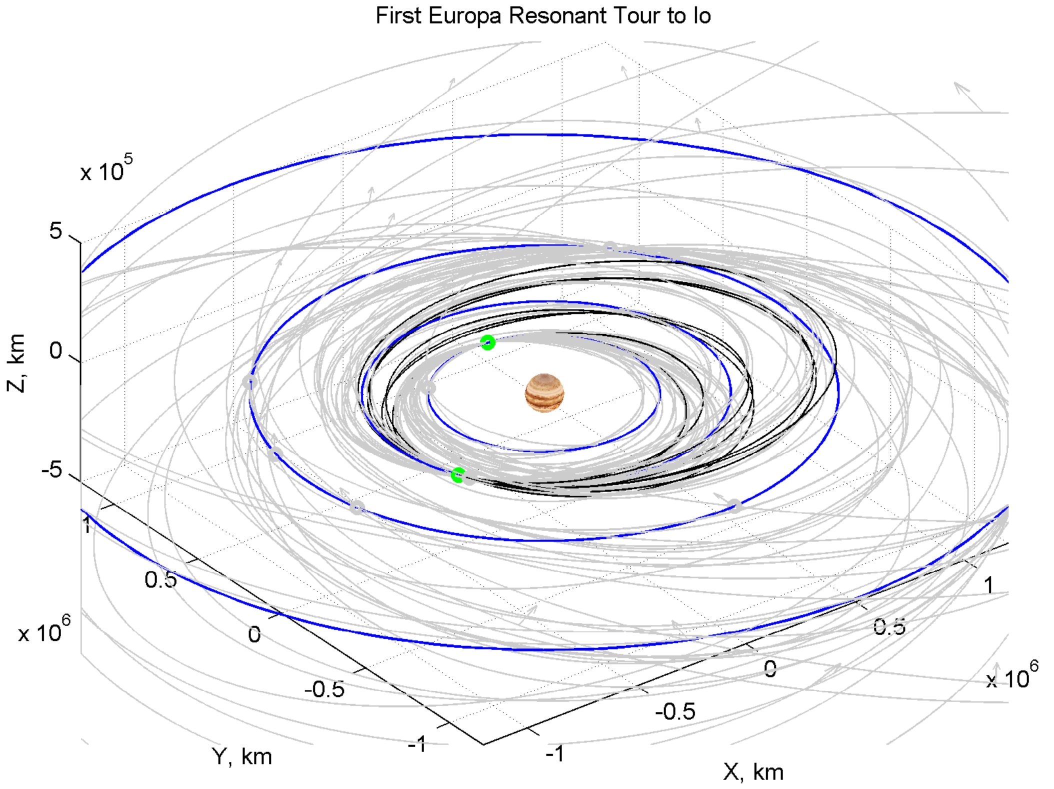


Figure C.7: Galilean moon mapping trajectory for phase 5 of the presented GTOC6 solution

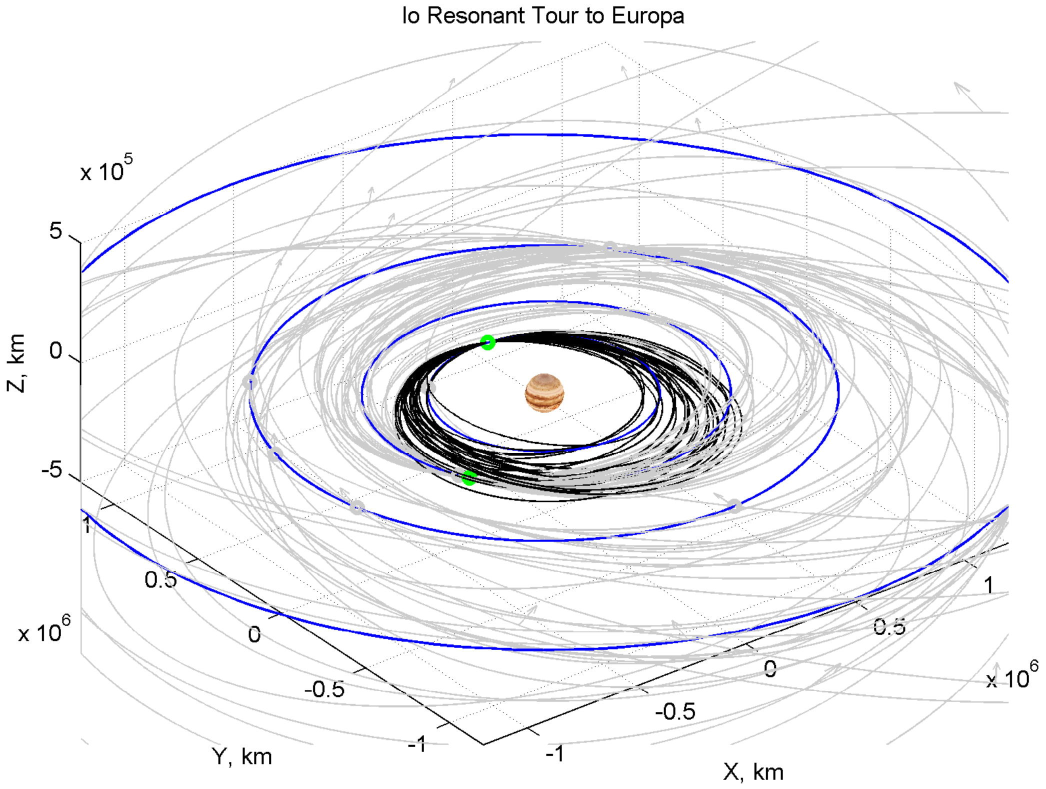


Figure C.8: Galilean moon mapping trajectory for phase 6 of the presented GTOC6 solution

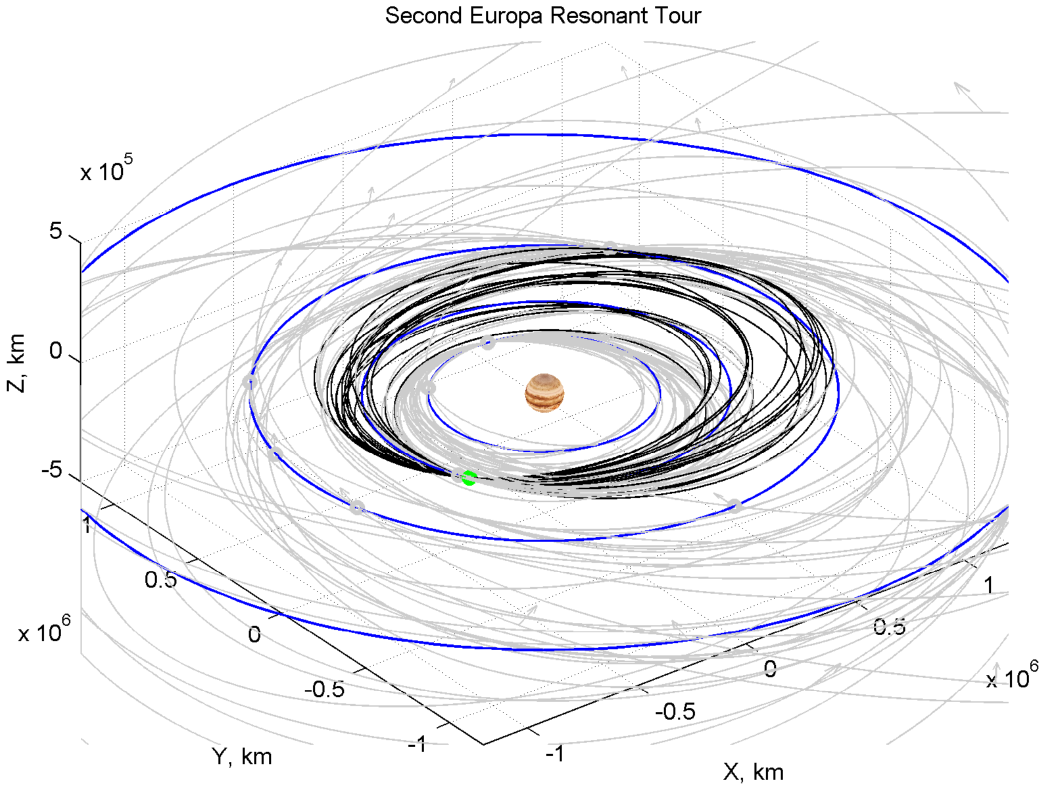


Figure C.9: Galilean moon mapping trajectory for phase 7 of the presented GTOC6 solution

Appendix D

v_∞ Leveraging Transfers with Ephemeris Locations

Several examples are presented to show how the boundary-value VILT method is capable of determining VILTs in the ephemeris case. First, comparisons are made between a circular-coplanar model and an ephemeris model for ΔV-EGA and Earth-Mars transfers. A third case is presented with just ephemeris locations of Ganymede and Europa as an intermoon example.

Ephemeris cases presented here use body states provided by NASA JPL databases de405.bsp and jup230l.bsp[1]. The circular-coplanar bodies are defined by the orbital elements in Table D.1. Table D.2 provides the search constraints used for the presented cases, where ppr indicates discretization sample points per body orbit period. In all the cases presented $\nu_L = 0$, therefore VILTs are tangent. Table D.3 describes the symbols used to differentiate the different types of trajectories in Figs. D.1-D.9. Because VILTs are the focus, no attempt is made to distinguish between different types of ballistic transfers (e.g. number of revolutions, short- or long-period). Note that Fig. D.1 shows solutions with every possible value of K, that is with maneuver placement possible on each spacecraft revolution. Figures. D.2-D.9 restrict the results to $K = 1$ to promote clarity of the figures. Relative geometry of

[1] Available online: ftp://ssd.jpl.nasa.gov/pub/eph/ [Accessed 10 January 2012]

Table D.1: Orbital elements for the circular-orbit cases

	Earth	Mars
Semimajor axis (AU)	1.0000	1.5237
Eccentricity	0	0
Inclination (degrees)	0	0
Argument of periapsis (degrees)	0	0
Longitude of the Ascen. Node (degrees)	0	0
Mean Anomaly (degrees)	0	180
The epoch of the orbits is at launch		

Table D.2: Constraints for the one-leg cases presented as examples of the VILT design space

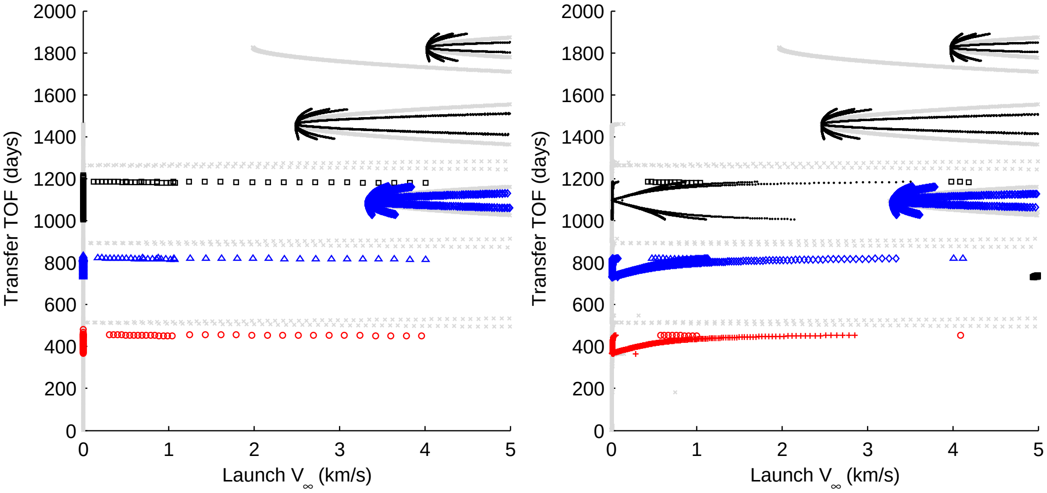
	Earth-Earth	Earth-Mars	Ganymede-Europa
Launch date and time	01/01/2013 0:00	04/18/2013 11:47	01/01/2013 0:00
Max Launch v_∞ (km/s)	5	7	5
Max ΔV (km/s)	1	2	0.3
Max Revolutions	3	3	3
Resolution (ppr)	1000	2000	1000
Max TOF (Earth days)	3600	3600	50

the bodies is important for IB-VILTs so for simplicity and repeatability the trajectory starts at conjunction for the Earth-Mars example and opposition for the Ganymede-Europa example.

Table D.3: Symbol and color scheme for figures in this Appendix

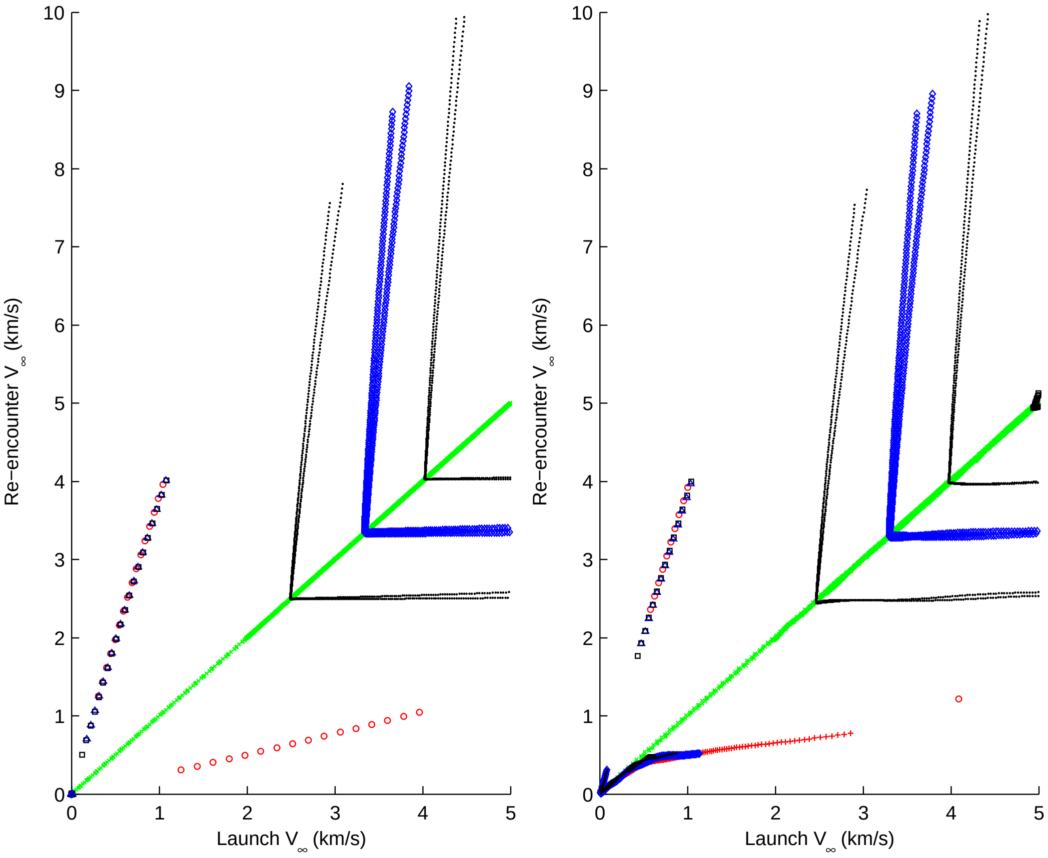
Case	Shape	Color
Ballistic	×	Green
VILT – 1 rev., interior	○	Red
VILT – 1 rev., exterior	+	Red
VILT – 2 revs., interior	△	Blue
VILT – 2 revs., exterior	◇	Blue
VILT – 3 revs., interior	□	Black
VILT – 3 revs., exterior	●	Black

D.1 ΔV-EGA



(a) (b)

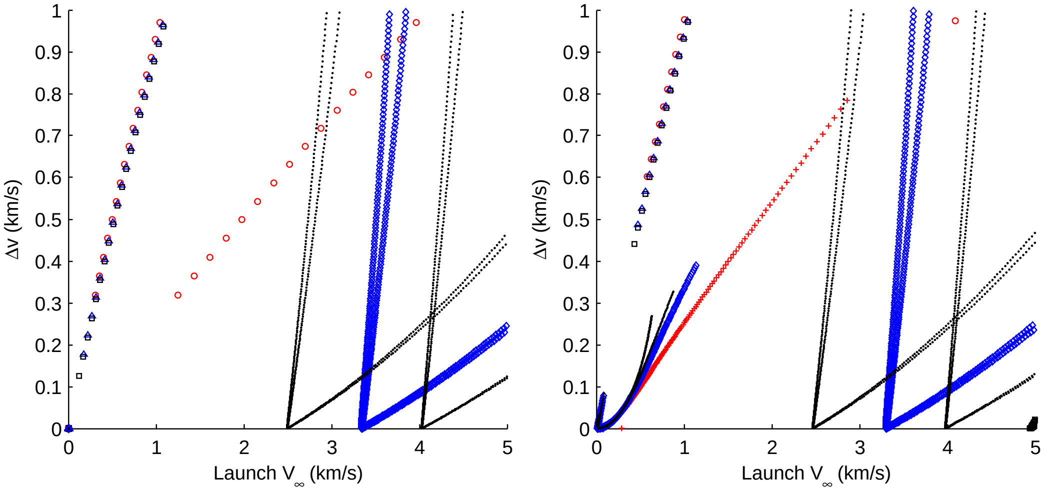
Figure D.1: *TOF* of the ΔV-EGA: circular orbit (a) vs. ephemeris (b) Earth



(a)

(b)

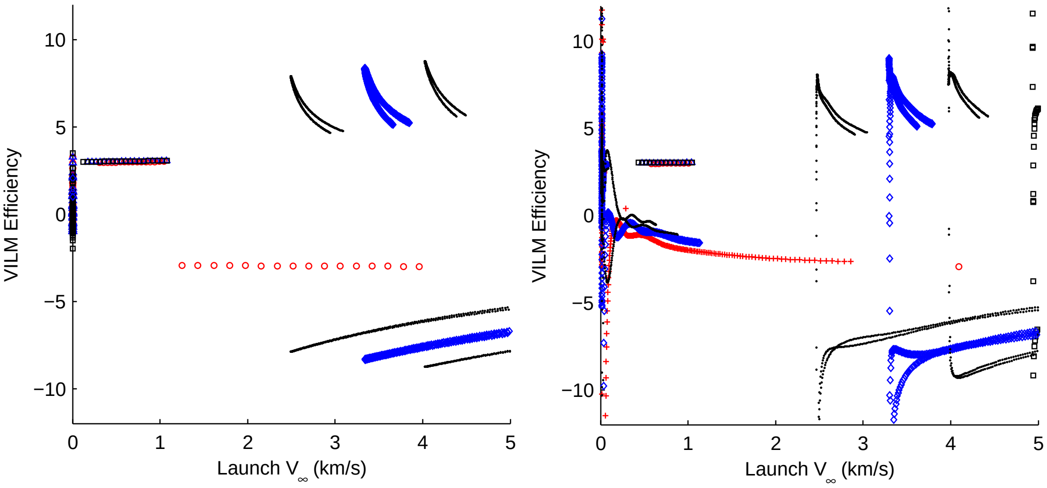
Figure D.2: v_∞ mapping of ΔV-EGA: circular orbit (a) vs. ephemeris (b) Earth



(a)

(b)

Figure D.3: ΔV of the ΔV-EGA for circular orbit (a) and ephemeris (b) cases

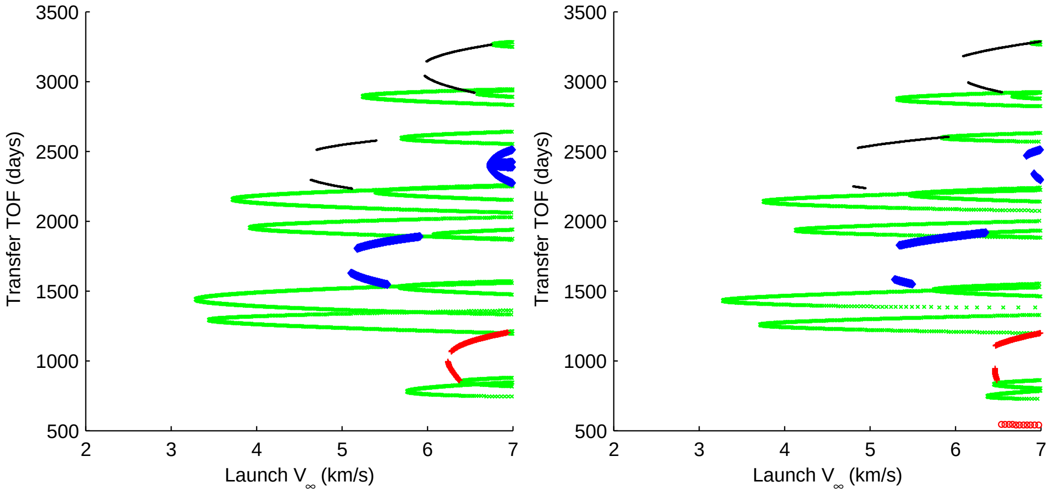


(a)

(b)

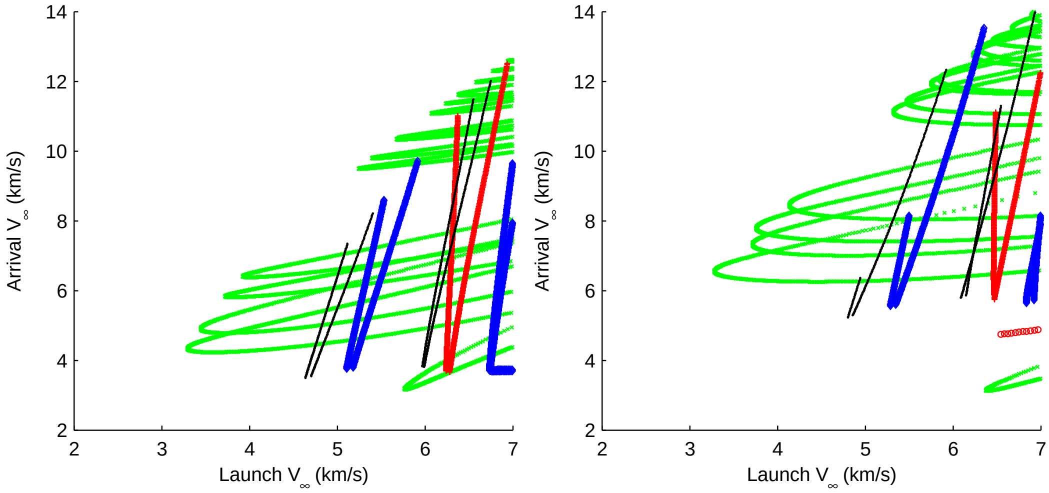
Figure D.4: Efficiency of the ΔV-EGA for circular orbit (a) and ephemeris (b) cases

D.2 Earth-Mars inter-body VILTs



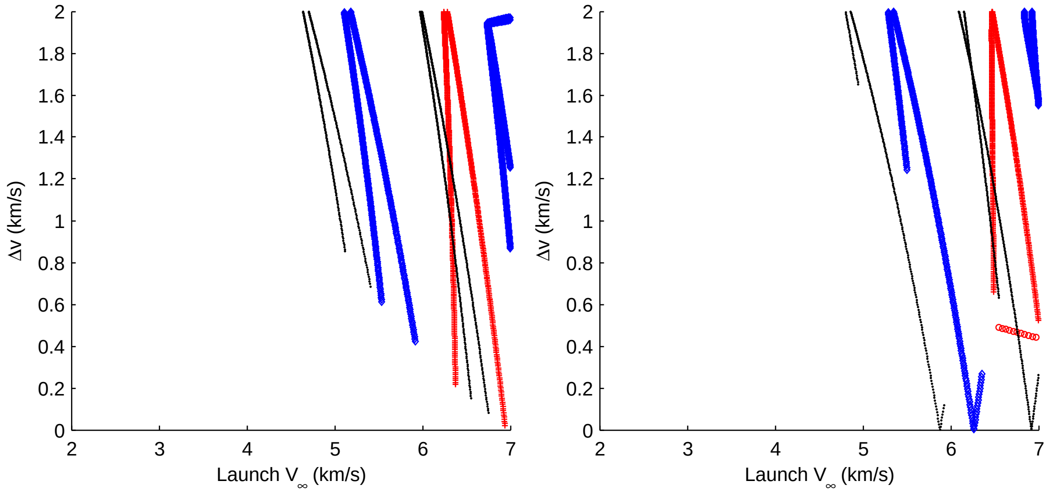
(a) (b)

Figure D.5: *TOF* of ballistic and IB-VILT Earth-Mars transfers for circular orbit (a) and ephemeris (b) cases



(a) (b)

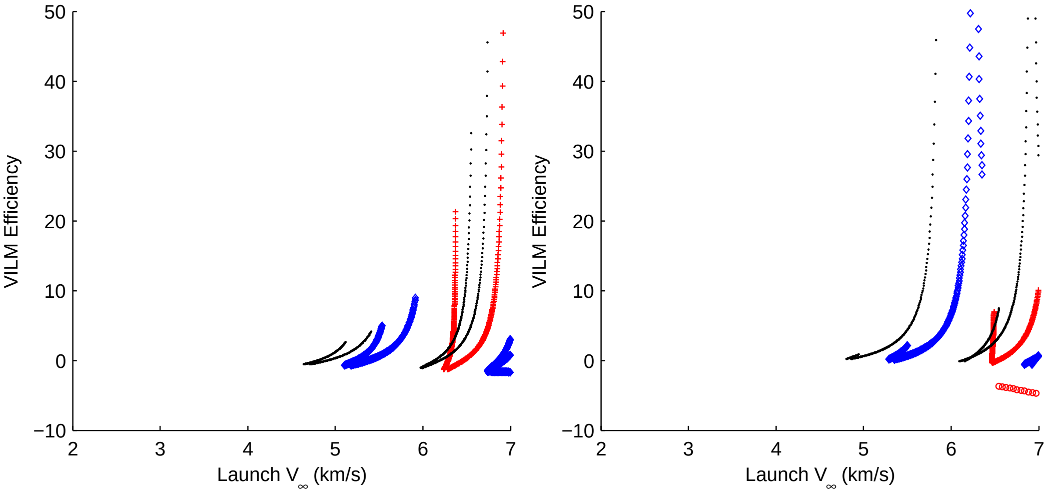
Figure D.6: v_∞ mapping of ballistic and IB-VILT Earth-Mars transfers for circular orbit (a) and ephemeris (b) cases



(a)

(b)

Figure D.7: ΔV of IB-VILT Earth-Mars transfers for circular orbit (a) and ephemeris (b) cases

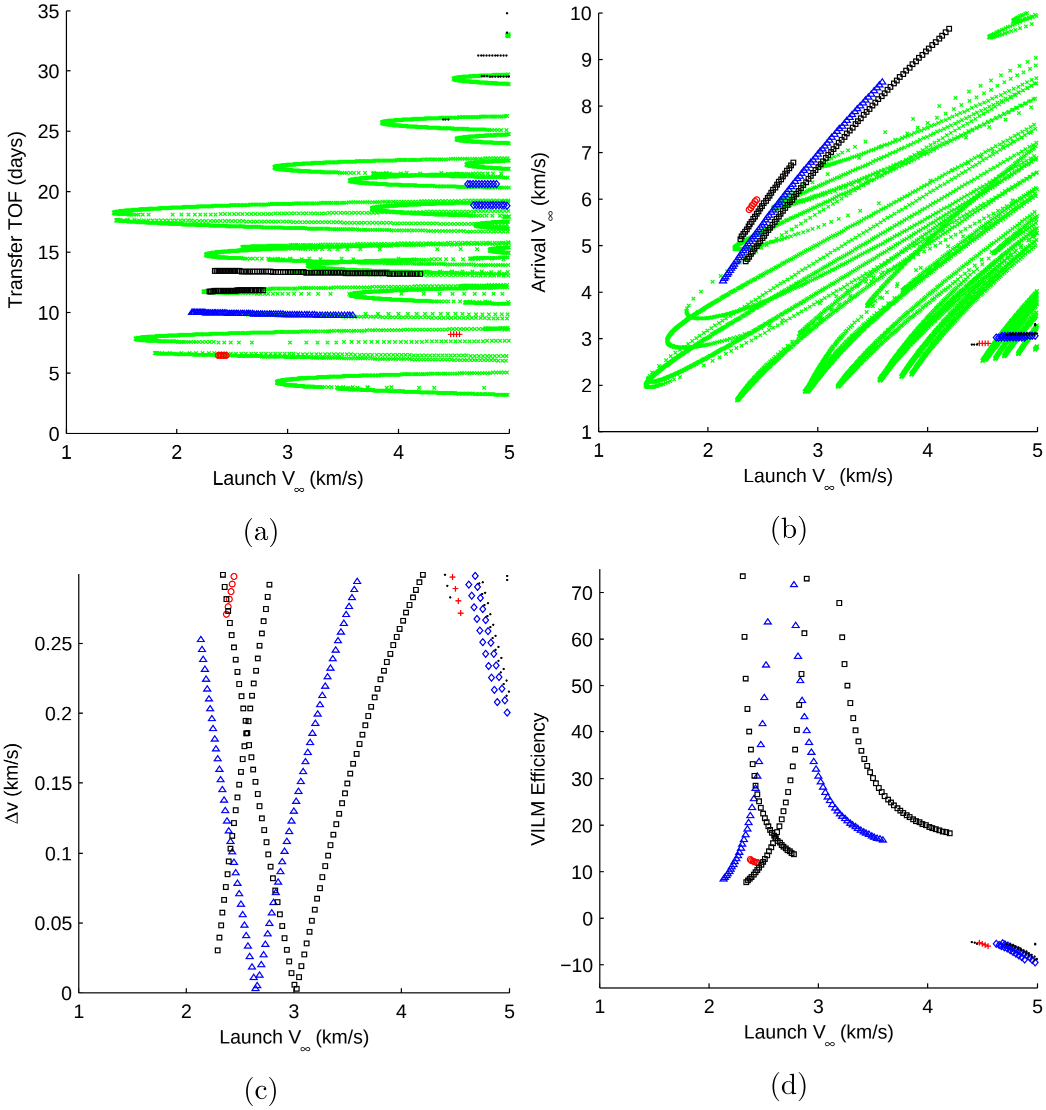


(a)

(b)

Figure D.8: Efficiency of IB-VILT Earth-Mars transfers for circular orbit (a) and ephemeris (b) cases

D.3 Ganymede-Europa inter-body VILTs



(a)

(b)

(c)

(d)

Figure D.9: Ganymede-Europa ballistic transfers and IB-VILTs in the ephemeris case

Bibliography

[1] Minovitch, M. A., "The determination and characteristics of ballistic interplanetary trajectories under the influence of multiple planetary attractions," Tech. Rep. 32-464, Jet Propulsion Laboratory, California Institute of Technology, Pasadena, California, 10 1963.

[2] Broucke, A., "The Celestial Mechanics of Gravity Assist," *Proceedings of the AIAA/American Astronautical Society Astrodynamics Specialist Conference*, 1988, Also AIAA paper 1988-4220.

[3] Russell, R. and Ocampo, C., "Geometric Analysis of Free-Return Trajectories Following a Gravity-Assisted Flyby," *Journal of Spacecraft and Rockets*, Vol. 42, No. 1, 2005, pp. 138–151. doi: 10.2514/1.5571.

[4] Bate, R., Mueller, D., and White, J., *Fundamentals in Astrodynamics*, chap. 1,4, Courier Dover Publications, New York, 1971, pp. 20,24,25,33,185.

[5] Labunsky, A., Papkov, O., and Sukhanov, K., *Multiple Gravity Assist Interplanetary Trajectories*, chap. 1,2.1, Gordon and Breach Science Publishers, Amsterdam, 1998.

[6] Vallado, D., *Fundamentals of Astrodynamics and Applications*, chap. 11, Springer, New York, 2001.

[7] Giberson, W. and Cunningham, N., "Mariner 10 Mission to Venus and Mercury," *Acta Astronautica*, Vol. 2, 1975, pp. 715–743.

[8] Hall, C. F., “Pioneer 10 and Pioneer 11,” *Science*, Vol. 188, No. 4187, 1975, pp. 445–446.

[9] Kohlhase, C. and Penzo, P., “Voyager Mission Description,” *Space Science Reviews*, Vol. 21, 1977, pp. 77–101.

[10] Blamont, J. and Sagdeev, R., “The VEGA Mission,” *Natur wissenschaften*, Vol. 71, 1984, pp. 295–302.

[11] Reinhard, R., “The Giotto Mission to Halley’s Comet,” *Advances in Space Research*, Vol. 2, No. 12, 1983, pp. 97–107.

[12] D’Amario, L. A., Bright, L. E., and Wolf, A. A., “Galileo Trajectory Design,” *Space Science Reviews*, Vol. 60, 1992, pp. 23–78.

[13] Diehl, R., Kaplan, D., and Penzo, P., “Satellite Tour Design for the Galileo Mission,” AIAA Aerospace Sciences Meeting, Reno, Nevada, 1 1983.

[14] Bell, J. and Johanessen, J., “Galileo Europa Mission (GEM) Tour Design,” *Astrodynamics Specialist Conference*, American Astronautical Society, 1997.

[15] Haw, R., Johannesen, J., Halsell, C., Wilson, M., and Pojman, J., “The GEM Extension Satellite Tour: Galileo’s Mission Across the Millennium,” *Advance in the Astronautical Sciences*, Vol. 103, American Astronautical Society, 1999.

[16] Fossi, B. M., Noci, G., and Poletto, G., “The Ulysses Space Mission,” *Il Nuovo Cimento*, Vol. 15, No. 5, 1992, pp. 493–500.

[17] Farquhar, R., Dunham, D., and McAdams, J., "NEAR mission overview and trajectory design," *Journal of the Astronautical Sciences*, Vol. 43, No. 4, 1995, pp. 353–371.

[18] Peralta, F. and Flanagan, S., "Cassini Interplanetary Trajectory Design," *Control Engineering Practice*, Vol. 3, No. 11, 1995, pp. 1603–1610. doi: 10.1016/0967-0661(95)00171-P.

[19] Wolf, A. and Smith, J., "Design of the Cassini Tour Trajectory in the Saturnian System," *Journal of International Federation of Automatic Control*, Vol. 3, No. 11, 1995, pp. 1611–1619. doi: 10.1023/A:1023692724823.

[20] Buffington, B., Strange, N., and Smith, J., "Overview of the Cassini extended mission trajectory," *Astrodynamics Specialist Conference*, Vol. 26, American Institute of Aeronautics and Astronautics, Reston, Virginia, 2008.

[21] Smith, J. and Buffington, B., "Overview of the Cassini Solstice Mission Trajectory," *Advances in the Astronautical Sciences*, Vol. 135, 2009, pp. 829–854.

[22] Veverka, J., Klaasen, K., Duxbury, T., Wolf, A., and Cheuvront, A., "The Stardust NExT Mission to Tempel 1," *Asteroids, Comets, Meteors*, 2008.

[23] Kawaguchi, J., Uesugi, K. T., Fujiwara, A., and Saitoh, H., "The MUSES-C, Mission Description and its Status," *Acta Astronautica*, Vol. 45, 1999, pp. 397–405.

[24] Schoenmaekers, J. and Bauske, R., "Re-design of the Rosetta mission for launch in 2004," *International Symposium on Space Flight Dynamics*, 2004, Report: ESA SP-548.

[25] McAdams, J., Dunham, D., Farquhar, R., Taylor, A., and Williams, B., "Trajectory design and maneuver strategy for the MESSENGER mission to Mercury," *Journal of Spacecraft and Rockets*, Vol. 43, No. 5, 2006, pp. 1054–1064. doi: 10.2514/1.18178.

[26] Chung, M.-K. J., Bhaskaran, S., Chesley, S. R., Halsell, C. A., Helfrich, C. E., Jefferson, D. C., McElrath, T. P., Rush, B. P., Wang, T.-C. M., and wan L. Yen., C., "EPOXI Trajectory and Maneuver Analyses," *Advances in the Astronautical Sciences*, Vol. 142, Univelt, San Diego, 2012.

[27] Guo, Y. and Farquhar, R. W., "New Horizons Mission Design," *Space Science Reviews*, Vol. 140, 2008, pp. 49–74. doi: 10.1007/s11214-007-9242-y.

[28] Rayman, M. D., Fraschetti, T. C., Raymond, C. A., and Russell, C. T., "Dawn: A mission in development for exploration of main belt asteroids Vesta and Ceres," *Acta Astronautica*, Vol. 58, 2006, pp. 605–616. doi: 10.1016/j.actaastro.2006.01.014.

[29] Matousek, S., "The Juno New Frontiers mission," *Acta Astronautica*, Vol. 61, 2007, pp. 932–939. doi: 10.1016/j.actaastro.2006.12.013.

[30] Tsuda, Y., Yoshikawa, M., Abe, M., Minamino, H., and Nakazawa, S., "System design of the Hayabusa 2Asteroid sample return mission to 1999 JU3," *Acta Astronautica*, Vol. 91, 2013, pp. 356–362.

[31] Lauretta, D. S. and Team, T. O.-R., "An overview of the OSIRIS-REx asteroid sample return mission," *Lunar and Planetary Science Conference*, Vol. 43, 2012.

[32] Benkhoff, J., van Casteren, J., Hayakawa, H., Fujimoto, M., Laakso, H., Novara, M., Ferri, P., Middleton, H. R., and Zieth, R., "BepiColomboComprehensive exploration of Mercury: Mission overview and science goals," *Planetary and Space Science*, Vol. 58, 2010, pp. 2–20. doi: 10.1016/j.pss.2009.09.020.

[33] Guo, Y., "Solar Probe Plus: Mission design challenges and trades," *Acta Astronautica*, Vol. 67, 2010, pp. 1063–1072. doi: 10.1016/j.actaastro.2010.06.007.

[34] Grasset, O., Dougherty, M., Coustenis, A., Bunce, E., Erd, C., Titov, D., Blanc, M., Coates, A., Drossart, P., Fletcher, L., Hussmann, H., Jaumann, R., Krupp, N., Lebreton, J.-P., Prieto-Ballesteros, O., Tortora, P., Tosi, F., and Hoolst, T. V., "JUpiter ICy moons Explorer (JUICE): An ESA mission to orbit Ganymede and to characterise the Jupiter system," *Planetary and Space Science*, Vol. 78, 2013, pp. 1–21. doi: 10.1016/j.pss.2012.12.002.

[35] Niehoff, J. C., "Touring the Galilean Satellites," *Journal of Spacecraft and Rockets*, Vol. 4, No. 10, 1971, pp. 1021–1027. doi: 10.2514/3.59764.

[36] Minovitch, M. A., "Gravity Thrust Jupiter Orbiter Trajectories Generated by Encountering the Galilean Satellites," *Journal of Spacecraft and Rockets*, Vol. 9, No. 10, 1972, pp. 751–756. doi: 10.2514/3.30391.

[37] Uphoff, C., Roberts, P., and Friedman, L., “Orbit Design Concepts for Jupiter Orbiter Missions,” *Journal of Spacecraft and Rockets*, Vol. 13, No. 6, 1976, pp. 348–355. doi: 10.2514/3.57096.

[38] Longuski, J. and Wolf, A., “The Galileo orbital tour for the 1986 launch opportunity,” *Astrodynamics Specialist Conference*, 1986.

[39] Petropoulos, A. E., Longuski, J. M., and Bonfiglio, E. P., “Trajectories to Jupiter via Gravity Assists from Venus, Earth, and Mars,” *Journal of Spacecraft and Rockets*, Vol. 37, No. 6, 2000, pp. 776–783. doi: 10.2514/6.1998-4284.

[40] Campagnola, S., Boutonnet, A., Martens, W., and Masters, A., “Mission design for the exploration of Neptune and Triton,” *Workshop on Astrodynamics and Flight Mechanics*, Vol. 23.

[41] Sims, J., Staugler, A., and Longuski, J., “Trajectory Options to Pluto via Gravity Assists from Venus, Mars, and Jupiter,” *Journal of Spacecraft and Rockets*, Vol. 34, No. 3, 1997, pp. 347–353. doi: 10.2514/2.3215.

[42] Vasile, M., Biesbroek, R., Summerer, L., Galvez, A., and Kminek, G., “Options for a Mission to Pluto and Beyond,” *Advances in the Astronautical Sciences*, Vol. 114, Univelt, San Diego, 2003.

[43] Heaton, A. F., Strange, N. J., Longuski, J. M., and Bonfiglio, E. P., “Automated Design of Aerogravity-Assist Trajectories,” *Journal of Spacecraft and Rockets*, Vol. 39, No. 1, 2002, pp. 17–22. doi: 10.2514/2.3801.

[44] Strange, N., Landau, D., Hofer, R., Snyder, J. S., and Randolph, T., “Solar Electric Propulsion Gravity-Assist Tours For Jupiter Missions,”

Proceedings of the AIAA/American Astronautical Society Astrodynamics Specialist Conference, 2012, Also AIAA paper 2012-4518.

[45] Izzo, D., oes, L. F. S., Märtens, M., de Croon, G. C., Heritier, A., and Yam, C. H., "Search for a Grand Tour of the Jupiter Galilean Moons," *Proceedings of the 15th annual conference on Genetic and evolutionary computation*, Vol. 15, ACM, New York, 2013.

[46] Kloster, K. W., Petropoulos, A. E., and Longuski, J. M., "Europa Orbiter tour design with Io gravity assists," *Acta Astronautica*, Vol. 68, 2011, pp. 931–946. doi: 10.1016/j.actaastro.2010.08.041.

[47] Buffington, B., Campagnola, S., and Petropoulos, A., "Europa Multiple-Flyby Trajectory Design," *Proceedings of the AIAA/American Astronautical Society Astrodynamics Specialist Conference*, 2012, Also AIAA paper 2012-5069.

[48] Campagnola, S., Buffington, B. B., and Petropoulos, A. E., "Jovian tour design for orbiter and lander missions to Europa," *Acta Astronautica*, Vol. 100, 2014, pp. 68–81. doi: 10.1016/j.actaastro.2014.02.005.

[49] Izzo, D., Bourdoux, A., Walker, R., and Ongaro, F., "Optimal trajectories for the impulsive deection of near earth objects," *Acta Astronautica*, Vol. 59, 2006, pp. 294–300. doi: 10.1016/j.actaastro.2006.02.002.

[50] Vasile, M., Summerer, L., and Pascale, P. D., "Design of Earth–Mars transfer trajectories using evolutionary-branching technique," *Acta Astronautica*, Vol. 56, 2005, pp. 705–720. doi: 10.1016/j.actaastro.2004.12.002.

[51] Chen, K. J., McConaghy, T. T., Landau, D. F., Longuski, J. M., and Aldrin, B., "Powered EarthMars Cycler with Three-Synodic-Period Repeat Time," *Journal of Spacecraft and Rockets*, Vol. 42, No. 5, 2005, pp. 921–927. doi: 10.2514/1.11610.

[52] Lynam, A., Kloster, K., and Longuski, J., "Multiple-satellite-aided capture trajectories at Jupiter using the Laplace resonance," *Celestial Mechanics and Dynamical Astronomy*, Vol. 109, No. 1, 2011, pp. 59–84. doi: 10.1007/s10569-010-9307-1.

[53] Labroquère, J., Aurélie, Héritier, Riccardi, A., and Izzo, D., "Evolutionary Constrained Optimization for a Jupiter Capture," *Parallel Problem Solving from Nature – PPSN XIII, Lecture Notes in Computer Science*, Vol. 8672, 2014.

[54] Lizia, P. D. and Radice, G., "Advanced Global Optimisation Tools for Mission Analysis and Design," Tech. Rep. Ariadna id: 03/4101, European Space Agency, 2003.

[55] Myatt, D., Becerra, V., Nasuto, S., and Bishop, J., "Advanced Global Optimisation for Mission Analysis and Design," Tech. Rep. Ariadna id: 03/4101, European Space Agency, 2003.

[56] Vasile, M. and Pascale, P. D., "Preliminary Design of Multiple Gravity-Assist Trajectories," *Journal of Spacecraft and Rockets*, Vol. 43, No. 4, 2006, pp. 794–805. doi: 10.2514/1.17413.

[57] Ceriotti, M., *Global optimisation of multiple gravity assist trajectories*, Ph.D. thesis, University of Glasgow, Glasgow, Scotland, United Kingdom, 5 2010.

[58] Carrico, J., Hooper, H., Roszman, L., and Gramling, C., "Rapid Design of Gravity Assist Trajectories," *ESA International Symposium on Spacecraft Flight Dynamics*, Vol. 3, 1991.

[59] Byrnes, L. D. D. and Stanford, R., "A New Method for Optimizing Multiple-Flyby Trajectories," *Journal of Guidance and Control*, Vol. 4, No. 6, 1981, pp. 591–596. doi: 10.2514/3.56115.

[60] Betts, J. T., "Survey of Numerical Methods for Trajectory Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 2, 1998, pp. 193–207. doi: 10.2514/2.4231.

[61] Bryan, J. M., *Global optimization of MGA-DSM problems using the Interplanetary Gravity Assist Trajectory Optimizer (IGATO)*, Master's thesis, California Polytechnic State University, San Luis Obispo, California, 12 2011.

[62] Cassioli, A., Izzo, D., Lorenzo, D. D., Locatelli, M., and Schoen, F., "Global Optimization Approaches for Optimal Trajectory Planning," *Modeling and Optimization in Space Engineering*, edited by G. Fasano and J. D. Pintér, chap. 5, Springer, New York, 2013, pp. 201–213, Series: Springer Optimization and Its Applications 73.

[63] Landau, D., personal communication, 2 2013.

[64] Petropoulos, A., personal communication, 7 2013.

[65] "The Global Trajectory Optimisation Competition Portal," http://sophia.estec.esa.int/gtoc_portal/, Accessed: 8 April 2015.

[66] Strange, N. and Longuski, J., "Graphical Method for Gravity-Assist Trajectory Design," *Journal of Spacecraft and Rockets*, Vol. 39, No. 1, 2002, pp. 9–16. doi: 10.2514/2.3800.

[67] Campagnola, S. and Russell, R. P., "Endgame Problem Part 1: V_∞-Leveraging Technique and the Leveraging Graph," *Journal of Guidance, Control, and Dynamics*, Vol. 33, No. 2, 2010, pp. 463–475. doi: 10.2514/1.44258.

[68] Campagnola, S. and Russell, R. P., "Endgame Problem Part 2: Multibody Technique and the TisserandPoincaré Graph," *Journal of Guidance, Control, and Dynamics*, Vol. 33, No. 2, 2010, pp. 476–486. doi: 10.2514/1.44290.

[69] Woolley, R. C. and Scheeres, D. J., "Applications of V-Innity Leveraging Maneuvers to Endgame Strategies for Planetary Moon Orbiters," *Journal of Guidance, Control, and Dynamics*, Vol. 34, No. 5, 2011, pp. 1298–1310. doi: 10.2514/1.50428.

[70] Izzo, D., Becerra, V. M., Myatt, D. R., Nasuto, S. J., and Bishop, J. M., "Search space pruning and global optimisation of multiple gravity assist spacecraft trajectories," *Journal of Global Optimization*, Vol. 38, 2007, pp. 283–296. doi: 10.1007/s10898-006-9106-0.

[71] Boutonnet, A., Martens, W., and Schoenmaekers, J., "SOURCE: A Matlab-Oriented Tool for Interplanetary Trajectory Global Optimization. Fundamentals," *Advances in the Astronautical Sciences*, Vol. 148, Univelt, San Diego, 2013, pp. 1449–1468.

[72] Olympio, J. T. and Marmorat, J.-P., "Global Trajectory Optimisation: Can We Prune the Solution Space when Considering Deep Space Maneuvers?" Tech. Rep. Ariadna id: 06/4101, European Space Agency, 2006.

[73] Williams, S., *Automated Design of Multiple Encounter Gravity-Assist Trajectories*, Master's thesis, School of Aeronautics and Astronautics, Purdue University, West Lafayette, Indiana, 8 1990.

[74] Longuski, J. and Williams, S., "Automated design of gravity-assist trajectories to Mars and the outer planets," *Celestial Mechanics and Dynamical Astronomy*, Vol. 52, 1991, pp. 207–220. doi: 10.1007/BF00048484.

[75] Bonfiglio, E., *Automated Design of Gravity-Assist and Aerogravity-Assist Trajectories*, Master's thesis, School of Aeronautics and Astronautics, Purdue University, West Lafayette, Indiana, 8 1999.

[76] Bonfiglio, E. P., Longuski, J. M., and Vinh, N. X., "Automated Design of Aerogravity-Assist Trajectories," *Journal of Spacecraft and Rockets*, Vol. 37, No. 6, 2000, pp. 768–775. doi: 10.2514/2.3649.

[77] Henning, G. A., Edelman, P. J., and Longuski, J. M., "Design and Optimization of Interplanetary Aerogravity-Assist Tours," *Journal of Spacecraft and Rockets*, Vol. 51, No. 6, 2014, pp. 1849–1856. doi: 10.2514/1.A32881.

[78] Patel, M., *Automated Design of Delta-V Gravity-Assist Trajectories for Solar System Exploration*, Master's thesis, School of Aeronautics and Astronautics, Purdue University, West Lafayette, Indiana, 8 1993.

[79] McConaghy, T. T., Debban, T. J., Petropoulos, A. E., and Longuski, J. M., "Design and Optimization of Low-Thrust Trajectories with Gravity Assists," *Journal of Spacecraft and Rockets*, Vol. 40, No. 3, 2003, pp. 380–387. doi: 10.2514/2.3973.

[80] McConaghy, T. T., *Design and Optimization of Interplanetary Spacecraft Trajectories*, Ph.D. thesis, Purdue University, West Lafayette, Indiana, 12 2004.

[81] Pessina, S. M., Campagnola, S., and Vasile, M., "Preliminary Analysis of Interplanetary Trajectories with Aerogravity and Gravity Assist Manoeuvres," *International Astronautical Congress*, Vol. 54, 2003.

[82] Qadir, K., *Automated Design of Multiple Encounter Gravity-Assist Trajectories*, Master's thesis, Space Science Department, Luleá University of Technology, Kiruna, Sweden, 6 2007.

[83] Mora, M. B. and Cano, J. L., "1st ACT Global Trajectory Optimisation Competition: Results found at DEIMOS Space," *Acta Astronautica*, Vol. 61, 2007, pp. 794–805. doi: 10.1016/j.actaastro.2007.03.007.

[84] Olympio, J. T., "Design of Optimal Multi-Gravity-Assist Trajectories with Free Number of Impulses," *International Symposium on Space Flight Dynamics*, 2009.

[85] Martens, W., Boutonnet, A., and Schoenmaekers, J., "SOURCE: A Matlab-Oriented Tool for Interplanetary Trajectory Global Optimization. Applications," *Advances in the Astronautical Sciences*, Vol. 148, Univelt, San Diego, 2013, pp. 1449–1468.

[86] Boutonnet, A., Schoenmaekers, J., Martens, W., and Yamaguchi, T., "Mission Analysis Update for the JUpiter ICy moon Explorer (JUICE)," *Advances in the Astronautical Sciences*, Vol. 152, Univelt, San Diego, 2014, pp. 2319–2334.

[87] Bernelli-Zazzera, F., Lavagna, M., Armellin, R., Lizia, P. D., Topputo, F., and Berz, M., "Global Trajectory Optimisation: Can We Prune the Solution Space when Considering Deep Space Maneuvers?" Tech. Rep. Ariadna id: 06/4101, European Space Agency, 2006.

[88] Vasile, M., Ceriotti, M., Radice, G., Becerra, V., Nasuto, S., and Anderson, J., "Global Trajectory Optimisation: Can We Prune the Solution Space when Considering Deep Space Maneuvers?" Tech. Rep. Ariadna id: 06/4101, European Space Agency, 2006.

[89] Becerra, V., Nasuto, S., Anderson, J., Ceriotti, M., and Bombardelli, C., "Search Space Pruning and Global Optimization of Multiple Gravity Assist Trajectories with Deep Space Manoeuvres," *Congress on Evolutionary Computation*, 2007.

[90] Lantukh, D. and Russell, R. P., "Multi-Objective Search for Multiple Gravity Assist Trajectories," Astrodynamics Specialist Conference, Vail, Colorado, 8 2015.

[91] Crain, T., Bishop, R. H., Fowler, W., and Rock, K., "Interplanetary Flyby Mission Optimization Using a Hybrid Global–Local Search Method," *Journal of Spacecraft and Rockets*, Vol. 37, No. 4, 2000, pp. 468–474. doi: 10.2514/2.3607.

[92] Johnson, G., *A Tabu Search Methodology for Spacecraft Tour Trajectory Optimization*, Ph.D. thesis, The University of Texas at Austin, Austin, Texas, 12 2014.

[93] Sentinella, M. R. and Casalino, L., "Cooperative evolutionary algorithm for space trajectory optimization," *Celestial Mechanics and Dynamical Astronomy*, Vol. 105, 2009, pp. 211–227. doi: 10.1007/s10569-009-9223-4.

[94] Englander, J., *Automated Trajectory Planning for Multiple-Flyby Interplanetary Missions*, Ph.D. thesis, University of Illinois at Urbana-Champaign, Urbana, Illinois, 2013.

[95] Hartmann, J., Coverstone-Carroll, V., and Williams, S., "Optimal interplanetary spacecraft trajectories via a Pareto genetic algorithm," *Journal of the Astronautical Sciences*, Vol. 46, No. 3, 1998, pp. 267–282.

[96] Abdelkhalik, O. and Gad, A., "Dynamic-Size Multiple Populations Genetic Algorithm for Multigravity-Assist Trajectories Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 35, No. 2, 2012, pp. 520–529. doi: 10.2514/1.54330.

[97] Izzo, D., Hennes, D., and Riccardi, A., "Constraint Handling and Multi-Objective Methods for the Evolution of Interplanetary Trajectories," *Journal of Guidance, Control, and Dynamics*, Vol. 38, No. 4, 2015, pp. 792–799. doi: 10.2514/1.G000619.

[98] Sentinella, M. R., "Comparison and Integrated Use of Differential Evolution and Genetic Algorithms for Space Trajectory Optimisation," *Congress on Evolutionary Computation*, 2007, pp. 973–978.

[99] Vinkó, T. and Izzo, D., "Global Optimisation Heuristics and Test Problems for Preliminary Spacecraft Trajectory Design," Tech. Rep. ACT-TNT-MAD-GOHTPPSTD, European Space Agency, 2008.

[100] Englander, J. A., Conway, B. A., and Williams, T., "Automated Mission Planning via Evolutionary Algorithms," *Journal of Guidance, Control, and Dynamics*, Vol. 35, No. 6, 2012, pp. 1878–1887. doi: 10.2514/1.54101.

[101] "The Annual "Humies" Awards – 2004-2014," http://www.genetic-programming.org/combined.php, Accessed: 13 April 2015.

[102] Addis, B., Cassioli, A., Locatelli, M., and Schoen, F., "A global optimization method for the design of space trajectories," *Computational Optimization and Applications*, Vol. 48, 2011, pp. 635–652. doi: 10.1007/s10589-009-9261-6.

[103] Vasile, M., Minisci, E., and Locatelli, M., "Analysis of Some Global Optimization Algorithms for Space Trajectory Design," *Journal of Spacecraft and Rockets*, Vol. 47, No. 2, 2010, pp. 334–344. doi: 10.2514/1.45742.

[104] Deb, K., Padhye, N., and Neema, G., "Interplanetary Trajectory Optimization with Swing-Bys Using Evolutionary Multi-objective Optimization," *Advances in Computation and Intelligence, Lecture Notes in Computer Science*, Vol. 4683, 2007.

[105] Conway, B. A., Chilan, C. M., and Wall, B. J., "Evolutionary principles applied to mission planning problems," *Celestial Mechanics and*

Dynamical Astronomy, Vol. 97, 2007, pp. 73–86. doi: 10.1007/s10569-006-9052-7.

[106] Pisarevsky, D. M. and Gurfil, P., "A Memetic Algorithm for Optimizing High-Inclination Multiple Gravity-Assist Orbits," *Congress on Evolutionary Computation*, 2009, pp. 86–93.

[107] Woo, B., Coverstone, V. L., and Cupples, M., "Low-Thrust Trajectory Optimization Procedure for Gravity-Assist, Outer-Planet Missions," *Journal of Spacecraft and Rockets*, Vol. 43, No. 1, 2006, pp. 121–129. doi: 10.2514/1.14665.

[108] Vavrina, M. A., *A Hybrid Genetic Algorithm Approach to Global Low-Thrust Trajectory Optimization*, Ph.D. thesis, Purdue University, West Lafayette, Indiana, 12 2008.

[109] Shen, H., Zhou, J., Peng, Q., Li, H., and Li, J., "Multi-objective interplanetary trajectory optimization combining low-thrust propulsion and gravity-assist maneuvers," *Science China: Technological Sciences*, Vol. 55, No. 3, 2012, pp. 841–847. doi: 10.1007/s11431-011-4705-5.

[110] Wagner, S. A., *Automated trajectory design for impulsive and low thrust interplanetary mission analysis*, Ph.D. thesis, Iowa State University, Ames, Iowa, 2014.

[111] Hollenbeck, G., "New Flight Techniques for Outer Planet Missions," *Supplement to Advances in the Astronautical Sciences*, Vol. 33, Univelt, San Diego, 1975.

[112] Strange, N. and Sims, J., "Methods for the design of v-infinity leveraging maneuvers," *Advances in the Astronautical Sciences*, Vol. 109, Univelt, San Diego, 2001, pp. 1959–1976.

[113] Sweetser, T., "Jacobi's integral and Delta V-earth-gravity-assist (Delta V-EGA) trajectories," *Advances in the Astronautical Sciences*, Vol. 85, Univelt, San Diego, 1993, pp. 417–430.

[114] Sims, J., Longuski, J., and Staugler, A., "V-infinity leveraging for interplanetary missions: Multiple-revolution orbit techniques," *Journal of Guidance, Control, and Dynamics*, Vol. 20, No. 3, 1997, pp. 409–415. doi: 10.2514/2.4064.

[115] Campagnola, S. and Russell, R., "Endgame problem part 1: V-infinity leveraging technique and leveraging graph," *Journal of Guidance, Control, and Dynamics*, Vol. 33, No. 2, 2010, pp. 463–475. doi: 10.2514/1.44258.

[116] Campagnola, S., Strange, N., and Russell, R., "A Fast Tour Design Method Using Non-Tangent V-infinity Leveraging Transfers," *Celestial Mechanics and Dynamical Astronomy*, Vol. 108, No. 2, 2010, pp. 165–186. doi: 10.1007/s10569-010-9295-1.

[117] Brinkerhoff, A. and Russell, R., "Pathfinding and v-infinity leveraging for planetary moon tour missions," *Advances in the Astronautical Sciences*, Vol. 134, Univelt, San Diego, 2009, Also American Astronautical Society paper 09-222.

[118] Strange, N., Campagnola, S., and Russell, R., "Leveraging Flybys of Low Mass Moons to Enable an Enceladus Orbiter," *Advances in the As-*

tronautical Sciences, Vol. 135, Univelt, San Diego, 2009, Also American Astronautical Society paper 09-435.

[119] Kowalkowski, T., Johannesen, J., and Try, L., "Launch period development for the Juno mission to Jupiter," *Proceedings of the AIAA/American Astronautical Society Astrodynamics Specialist Conference*, 2008, Also AIAA paper 2008-7369.

[120] Lantukh, D. V., Russell, R. P., and Campagnola, S., "V-Infinity Leveraging Boundary-Value Problem and Application in Spacecraft Trajectory Design," *Journal of Spacecraft and Rockets*, 2015. doi: 10.2514/1.A32918.

[121] Strange, N., Russell, R., and Buffington, B., "Mapping the V-infinity Globe," *Advances in the Astronautical Sciences*, Vol. 129, Univelt, San Diego, 2007.

[122] Hollister, W., "Periodic Orbits for Interplanetary Flight," *Journal of Spacecraft and Rockets*, Vol. 6, No. 4, 1969, pp. 366–369. doi: 10.2514/3.29664.

[123] Uphoff, C., Roberts, P., and Friedman, L., "Orbit Design Concepts for Jupiter Orbiter Missions," *Journal of Spacecraft and Rockets*, Vol. 13, No. 6, 1976, pp. 348–355. doi: 10.2514/3.57096.

[124] Perozzi, E., Casalino, L., Colasurdo, G., Rossi, A., and Valsecchi, G., "Resonant Fly-by Missions to Near Earth Asteroids," *Modern Celestial Mechanics: From Theory to Applications*, 2002, pp. 49–62. doi: 10.1007/978-94-017-2304-6_4.

[125] Buffington, B. and Strange, N., "Science-Driven Design of Enceladus Flyby Geometry," 57th International Astronautical Congress, Valencia, Spain, 10 2006.

[126] Russell, R. and Ocampo, C., "Global Search for Idealized Free-Return EarthMars Cyclers," *Journal of Guidance, Control, and Dynamics*, Vol. 28, No. 2, 2005, pp. 194–208. doi: 10.2514/1.8696.

[127] Russell, R. and Strange, N., "Cycler Trajectories in Planetary Moon Systems," *Journal of Guidance, Control, and Dynamics*, Vol. 32, No. 1, 2009, pp. 143–157. doi: 10.2514/1.36610.

[128] Lantukh, D. and Russell, R., "Automated Inclusion of n-pi Transfers in Gravity-Assist Flyby Tour Design," *Proceedings of the AIAA/American Astronautical Society Astrodynamics Specialist Conference*, 2012.

[129] "Mission Analysis Environment (MAnE)," http://spaceflightsolutions.com/products/mane.asp, Accessed: 21 August 2012.

[130] Scheeres, D. J., "Orbit Mechanics About Asteroids and Comets," *Journal of Guidance, Control, and Dynamics*, Vol. 35, No. 3, 2012, pp. 987–997. doi: 10.2514/1.57247.

[131] Broschart, S. B., Lantoine, G., and Grebow, D. J., "Quasi-Terminator Orbits Near Primitive Bodies," *Celestial Mechanics and Dynamical Astronomy*, 2014. doi: 10.1007/s10569-014-9574-3.

[132] Mignard, F. and Hénon, M., "About an Unsuspected Integrable Problem," *Celestial Mechanics and DynamicaI Astronomy*, Vol. 33, 1984, pp. 239–250. doi: 10.1007/BF01230506.

[133] Scheeres, D. J., “Satellite Dynamics About Asteroids,” *Advances in the Astronautical Sciences*, Vol. 87, 1994.

[134] Russell, R. P. and Lara, M., “Long-Life Lunar Repeat Ground Track Orbits,” *Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 4, 2004, pp. 982–993. doi: 10.2514/2.5041.

[135] Paskowitz, M. and Scheeres, D., “Design of Science Orbits About Planetary Satellites: Application to Europa,” *Journal of Guidance, Control, and Dynamics*, Vol. 29, No. 5, 2006, pp. 1147–1158. doi: 10.2514/1.19464.

[136] Hart, J., *Plant Tropisms: And Other Growth Movements*, Springer, New York, 1990, p. 36.

[137] Hamilton, D. P., “Motion of dust in a planetary magnetosphere: Orbit-averaged equations for oblateness, electromagnetic, and radiation forces with application to Saturns E ring,” *Icarus*, Vol. 101, 1993, pp. 244–264. doi: 10.1006/icar.1993.1022.

[138] Colombo, C., Lucking, C., and McInnes, C. R., “Orbital dynamics of high area-to-mass ratio spacecraft with J2 and solar radiation pressure for novel Earth observation and communication services,” *Acta Astronautica*, Vol. 81, 2012, pp. 27–32. doi: 10.1016/j.actaastro.2012.07.009.

[139] Hedman, M., Burt, J., Burns, J., and Tiscareno, M., “The shape and dynamics of a heliotropic dusty ringlet in the Cassini Division,” *Icarus*, Vol. 210, No. 1, 2010, pp. 284–297. doi: 10.1016/j.icarus.2010.06.017.

[140] Burt, J., Hedman, M., Burns, J., and Tiscareno, M., "Dynamical analysis of the heliotropic Charming ringlet in Saturns Cassini Division," *Bulletin of the American Astronomical Society*, Vol. 41, 2010, pp. 940.

[141] Hamilton, D. P. and Krivov, A. V., "Circumplanetary Dust Dynamics: Effects of Solar Gravity, Radiation Pressure, Planetary Oblateness, and Electromagnetism," *Icarus*, Vol. 123, No. 2, 1996, pp. 503–523. doi: 10.1006/icar.1996.0175.

[142] Krivov, A., Sokolov, L., and Dikarev, V., "Dynamics of Mars-Orbiting dust: Effects of Light Pressure and Planetary Oblateness," *Celestial Mechanics and DynamicaI Astronomy*, Vol. 63, 1996, pp. 313–339. doi: 10.1007/BF00692293.

[143] Krivov, A. V. and Getino, J., "Orbital evolution of high-altitude balloon satellites," *Astronomy and Astrophysics*, Vol. 318, 1997, pp. 308–314. doi: 10.1.1.388.1205.

[144] Colombo, C. and McInnes, C. R., "Constellations of inclined heliotropic orbits for enhanced earth coverage," *International Astronautical Congress*, Vol. 63, 2012.

[145] Lantukh, D., Russell, R. P., and Broschart, S., "Heliotropic orbits at oblate asteroids: balancing solar radiation pressure and J2 perturbations," *Celestial Mechanics and Dynamical Astronomy*, Vol. 121, 2015, pp. 171–190. doi: 10.1007/s10569-014-9596-x.

[146] Kaula, W. M., *Theory of Satellite Geodesy*, chap. 3, Dover (2000), Mineola, NY, 1966.

[147] Merson, R. H., "The Motion of a Satellite in an Axi-symmetric Gravitational Field," *Geophysical Journal International*, Vol. 4, 1961, pp. 17–52. doi: 10.1007/BF00643817.

[148] King-Hele, D. G., "The Motion of a Satellite in an Axi-symmetric Gravitational Field," *Geophysical Journal International*, Vol. 4, 1961, pp. 3–16. doi: 10.1111/j.1365-246X.1937.tb07101.x.

[149] King-Hele, D. G., Cook, G. E., and Rees, J. M., "Determination of the Even Harmonics in the Earth's Gravitational Potential," *Geophysical Journal International*, Vol. 8, 1963, pp. 119–145. doi: 10.1111/j.1365-246X.1963.tb02902.x.

[150] Smith, D. E., "A Determination of the Even Harmonics in the Earth's Gravitational Potential Function," *Planetary and Space Science*, Vol. 13, 1965, pp. 1151–1159. doi: 10.1016/0032-0633(65)90145-5.

[151] Vokrouhlicky, D., Nesvorny, D., and Bottke, W. F., "The vector alignments of asteroid spins by thermal torques," *Nature*, Vol. 425, 2003, pp. 147–151. doi: 10.1038/nature01948.

[152] Căpek, D. and Vokrouhlický, D., "The YORP effect with finite thermal conductivity," *Icarus*, Vol. 172, 2004, pp. 526–536. doi: 10.1016/j.icarus.2004.07.003.

[153] Hanuš, J., Ďurech, J., Brož, M., Warner, B. D., Pilcher, F., Stephens, R., Oey, J., Bernasconi, L., Casulli, S., Behrend, R., Polishook, D., Henych, T., Lehký, M., Yoshida, F., and Ito, T., "A study of asteroid pole-latitude distribution based on an extended set of shape models derived by

the lightcurve inversion method," *Astronomy and Astrophysics*, Vol. 530, 2011. doi: 10.1016/j.icarus.2011.07.027.

[154] Lhotka, C., Souchay, J., and Shahsavari, A., "Obliquity, precession rate, and nutation coefficients for a set of 100 asteroids," *Astronomy and Astrophysics*, Vol. 556, 2013. doi: 10.1051/0004-6361/201321205.

[155] Nolan, M. C., Magri, C., Howell, E. S., Benner, L. A., Giorgini, J. D., Hergenrother, C. W., Hudson, R. S., Lauretta, D. S., Margot, J. L., Ostro, S. J., and Scheeres, D. J., "Shape model and surface properties of the OSIRIS-REx target Asteroid (101955) Bennu from radar and lightcurve observations," *Icarus*, Vol. 226, 2013, pp. 629–640. doi: 10.1016/j.icarus.2013.05.028.

[156] Chesley, S. R., Farnocchia, D., Nolan, M. C., Vokrouhlicky, D., Chodas, P. W., Milani, A., Spoto, F., Rozitis, B., Benner, L. A. M., Bottke, W. F., Busch, M. W., Emery, J. P., Howell, E. S., Lauretta, D. S., Margot, J.-L., and Taylor, P. A., "Orbit and Bulk Density of the OSIRIS-REx Target Asteroid (101955) Bennu," *Icarus*, Vol. 235, 2014, pp. 5–22.

[157] Vasile, M., Ceriotti, M., and Pascale, P. D., "An incremental approach to the solution of global trajectory optimization problems," *Advances in Global Optimization: Methods and Applications*, 2007.

[158] Ludkovski, M. and Rüschendorf, L., "On comonotonicity of Pareto optimal risk sharing," *Statistics and Probability Letters*, Vol. 78, 2008, pp. 1181–1188. doi: 10.1016/j.spl.2007.11.031.

[159] Lantukh, D., Russell, R., and Campagnola, S., "Automated Inclusion of V-Infinity Leveraging Maneuvers in Gravity-Assist Flyby Tour Design,"

Advances in the Astronautical Sciences, Vol. 143, Univelt, San Diego, 2012, Also American Astronautical Society paper 12-162.

[160] McConaghy, T., Russell, R., and Longuski, "Towards a Standard Nomenclature for Earth-Mars Cycler Trajectories," *Journal of Spacecraft and Rockets*, Vol. 42, No. 4, 2005, pp. 694–698. doi: 10.2514/1.8123.

[161] Scheeres, D. J., *Orbital Motion in Strongly Perturbed Environments*, Springer, New York, 2012, pp. 55–57, 166, 263, 274, 293–294.

[162] Mignard, F., "Effects of Radiation Forces on Dust Particles in Planetary Rings," *Planetary rings*, University of Arizona Press, Tucson, AZ, 1984, pp. 333–366.

[163] Brouwer, D. and Clemence, G. M., *Methods of Celestial Mechanics*, Academic Press, New York, 1961, pp. 288–289.

[164] Broucke, R. A., "Long-Term Third-Body Effects via Double Averaging," *Journal of Guidance, Control, and Dynamics*, Vol. 26, No. 1, 2003, pp. 313–339. doi: 10.2514/1.27104.

[165] Lara, M. and Russell, R. P., "Computation of a Science Orbit about Europa," *Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 1, 2007, pp. 259–263. doi: 10.2514/1.22493.

[166] Russell, R. P. and Lara, M., "On the Design of an Enceladus Science Orbit," *Acta Astronautica*, Vol. 65, No. 1-2, 2009, pp. 27–29. doi: 10.1016/j.actaastro.2009.01.021.

[167] Werner, R. A., "Spherical Harmonic Coefficients for the Potential of a Constant-Density Polyhedron," *Computers & Geosciences*, Vol. 21, No. 10, 1997, pp. 1071–1077. doi: 10.1016/S0098-3004(97)00110-6.

[168] Lantoine, G. and Russell, R. P., "A Hybrid Differential Dynamic Programming Algorithm for Constrained Optimal Control Problems. Part 1: Theory," *Journal of Optimization Theory and Applications*, Vol. 154, 2012, pp. 382–417. doi: 10.1007/s10957-012-0039-0.

[169] Lantoine, G. and Russell, R. P., "A Hybrid Differential Dynamic Programming Algorithm for Constrained Optimal Control Problems. Part 2: Application," *Journal of Optimization Theory and Applications*, Vol. 154, 2012, pp. 418–442. doi: 10.1007/s10957-012-0038-1.

[170] Abouelmagd, E. I., "Periodic orbits under combined effects of oblateness and radiation in the restricted problem of three bodies," *Astrophys Space Sci*, Vol. 341, 2012, pp. 331–341.

[171] Perdiou, A., Perdios, E., and Kalantonis, V., "Periodic orbits of the Hill problem with radiation and oblateness," *Astrophys Space Sci*, Vol. 342, 2012, pp. 19–30.

[172] Singh, J., "Combined effects of oblateness and radiation on the nonlinear stability of L4 in the restricted three-body problem," *Astronomical Journal*, Vol. 137, No. 3286, 2009.

[173] Singh, J. and Umar, A., "Collinear equilibrium points in the elliptic R3BP with oblateness and radiation," *Advances in Space Research*, Vol. 52, No. 8, 2013.

[174] Abouelmagd, E. I. and Sharaf, M. A., “The motion around the libration points in the restricted three-body problem with the effect of radiation and oblateness,” *Astrophys Space Sci*, Vol. 344, 2013, pp. 321–332.

[175] Nolan, M. C., Magri, C., Ostro, S. J., Benner, L. A., Giorgini, J. D., Howell, E. S., and Hudson, R. S., “The Shape and Spin of 101955 (1999 RQ36) from Arecibo and Goldstone Radar Imaging,” *Bulletin of the American Astronomical Society*, Vol. 39, 2007, pp. 433.

[176] Broschart, S. B., Lantoine, G., and Grebow, D. J., “Characteristics of Quasi-Terminator Orbits Near Primitive Bodies,” *Advances in the Astronautical Sciences*, Vol. 148, 2013.

[177] Lantukh, D., Russell, R. P., and Broschart, S. B., “Heliotropic Orbits at Oblate Asteroids: Balancing Solar Pressure and J2 Perturbations,” *Advances in the Astronautical Sciences*, Vol. 150, 2014.

[178] Vasile, M., “A Global Approach to Optimal Space Trajectory Design,” *Advances in the Astronautical Sciences*, Vol. 114, Univelt, San Diego, 2003.

[179] Lizia, P. D., Radice, G., Izzo, D., and Vasile, M., “On the Solution of Interplanetary Trajectory Design Problems by Global Optimisation Methods,” *Proceedings of the International Workshop on Global Optimization*, 2005, pp. 159–164.

[180] Izzo, D., “Advances in Global Optimisation for Space Trajectory Design,” *Proceedings of the International Symposium on Space Technology and Science*, Vol. 25, 2006.

[181] Olds, A. D., *Interplanetary Trajectory Optimization with Differential Evolution*, Master's thesis, University of Missouri, Columbia, Missouri, 12 2005.

[182] Olds, A. D., Kluever, C. A., and Cupples, M. L., "Interplanetary Mission Design Using Differential Evolution," *Journal of Spacecraft and Rockets*, Vol. 44, No. 5, 2007, pp. 1060–1070. doi: 10.2514/1.27242.

[183] Vasile, M., Minisci, E., and Locatelli, M., "On Testing Global Optimization Algorithms for Space Trajectory Design," *Proceedings of the AIAA/American Astronautical Society Astrodynamics Specialist Conference*, 2008, Also AIAA paper 2008-6277.

[184] Vasile, M. and Locatelli, M., "A hybrid multiagent approach for global trajectory optimization," *Journal of Global Optimization*, Vol. 44, 2009, pp. 461–479. doi: 10.1007/s10898-008-9329-3.

[185] Wall, B. J. and Conway, B. A., "Genetic algorithms applied to the solution of hybrid optimal control problems in astrodynamics," *Journal of Global Optimization*, Vol. 44, 2009, pp. 493–508. doi: 10.1007/s10898-008-9352-4.

[186] Ceriotti, M. and Vasile, M., "Automated Multigravity Assist Trajectory Planning with a Modied Ant Colony Algorithm," *Journal of Aerospace Computing, Information, and Communication*, Vol. 7, 2010, pp. 261–293. doi: 10.2514/1.48448.

[187] Gad, A. and Abdelkhalik, O., "Hidden Genes Genetic Algorithm for Multi-Gravity-Assist Trajectories Optimization," *Journal of Spacecraft and Rockets*, Vol. 48, No. 4, 2011, pp. 629–641. doi: 10.2514/1.52642.

[188] Chilan, C. M. and Conway, B. A., “Automated Design of Multiphase Space Missions Using Hybrid Optimal Control,” *Journal of Guidance, Control, and Dynamics*, Vol. 36, No. 5, 2013, pp. 1410–1424. doi: 10.2514/1.58766.

[189] Rogata, P., Sotto, E. D., Graziano, M., and Graziani, F., “Guess Value for Interplanetary Transfer Design Through Genetic Algorithms,” *Advances in the Astronautical Sciences*, Vol. 114, Univelt, San Diego, 2003.

Vita

Demyan Vasilyevich Lantukh was born in Ukraine and emigrated to the United States with his family at the age of three. As a result, he grew up somewhere between an ethnic Slavic community and a suburban Southern community outside Atlanta, Georgia.

After graduating Valedictorian of Milton High School, Demyan applied his passion for learning and his conviction to make the most of every opportunity by earning a B.S. & M.S. in aerospace engineering from Georgia Tech, where he also minored in French. Following through on living to the fullest included not only research in orbital mechanics and interning at the Applied Physics Lab Space Department, but also studying and traveling in Europe, teaching English in Africa, humanitarian work, personal mentorship, and most importantly spiritual development – following Jesus Christ.

Shortly after becoming a NASA Space Technology Research Fellow and beginning his PhD researching global trajectory searches, Demyan transferred to the University of Texas at Austin, continuing to work with Ryan Russell. During this time, Demyan completed a couple internships at the Jet Propulsion Lab, where his research on small body proximity operations began. The move to Texas began a difficult and rewarding period of Demyan's life, with challenges coming from the degree program and from numerous personal and community circumstances. By God's grace Demyan continued to pursue (though sometimes falteringly) excellence in research and a balanced life – a life that demonstrates love and leaves a legacy echoing to eternity.

Permanent address: demyan.lantukh@gmail.com

This dissertation was typeset with LaTeX[†] by the author.

[†] LaTeX is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's TeX Program.