AUTOMATED PLANETARY SATELLITE TOUR PLANNING WITH A NOVEL LOW-THRUST DIRECT TRANSCRIPTION METHOD

BY

DONALD HAMILTON ELLISON

THESIS

Submitted in partial fulfillment of the requirements for the degree of Master of Science in Aerospace Engineering in the Graduate College of the University of Illinois at Urbana-Champaign, 2013

Urbana, Illinois

Adviser:

Professor Bruce A. Conway
ABSTRACT

Upon completion of the heliocentric leg of its mission, it is almost always the case that an interplanetary spacecraft has scientific objectives to accomplish. This is especially true if the spacecraft performs a planetary moon tour. Over the past decade, mission designers have started to consider the use of low-thrust electric propulsion systems on board the spacecraft to enable future tours. The optimal trajectory for a given mission profile using low-thrust may be highly non-intuitive. There are thus many challenging aspects to the design of multiple flyby, low-thrust trajectories. One of the most significant, from the point of view of a numerical optimizer, can be the characteristic time scale of the dynamical system. Trajectories in a setting with a short characteristic time scale (e.g. those occurring in the Jovian system) are more challenging to optimize than those with a longer time scale (e.g. heliocentric trajectories to the outer solar system) because the spacecraft must often perform many revolutions about the central body as well as several flyby maneuvers. In this work, a novel way of parametrizing a low-thrust trajectory is explored and results using this method are presented. In addition to this, methods are outlined to increase the speed of execution and robustness of a numerical optimizer employing the Sims-Flanagan transcription method. To illustrate the difficulty of low-thrust trajectory optimization in a dynamical system characterized by a short time scale, and to provide an example of a tool that would benefit from the previously mentioned improvements, an existing medium-fidelity interplanetary trajectory optimizer, the Evolutionary Mission Trajectory Generator (EMTG), is modified and used to revisit the preliminary design phase of the Jupiter Icy Moons Orbiter (JIMO) reference trajectory. The results of this analysis are presented and compared with the high-fidelity version of the JIMO reference trajectory generated using the software package Mystic.
For Nikita Ann-Nicole and my grandparents Regina, Klaus, Lenore and Tom
ACKNOWLEDGMENTS

No acknowledgment section could ever be sufficient to give the appropriate amount of thanks that I need to, but I will do my best. I first need to thank my adviser, Bruce Conway. Professor Conway, it is an honor to have you as a mentor; thank you for giving me the opportunity to be a part of your research group and for guiding my research. I have not only benefited from your remarkable ability as an instructor but also your wealth of experience in the field.

I would like to thank my classmates and colleagues in the Department of Aerospace Engineering without whom my time so far at UIUC would have been far less enriching, and far less entertaining. Aaron D’Souza, Zaki Sheikh, Richard Strope, Bindu Jagannatha, Galina Shpuntova, Ryne Beeson, Greg Williamson and Vicky Shikova, you not only helped me survive classes but you form a group of friends that I know I will carry with me for the rest of my life. Alex and Laura Ghosh welcomed me to the Champaign-Urbana area and were a great comfort when I came to a town where I knew no one. I felt like I have known them my whole life. Alex, you have been a great collaborator and supported me academically on countless occasions. You are a tremendous source of knowledge and confidence.

A great deal of gratitude goes out to my fellow research group members Pradipto Ghosh and Joanie Stupik. Pradipto you are the embodiment of what it means to be a hard worker, a veritable research machine. Joanie, you have been the “optimal” office mate and truly one of the nicest people that I have ever met. I knew that we would get along from the moment we met through e-mail. You are the type of person whom everyone should strive to be.
It is impossible to appropriately acknowledge Jacob Englander’s impact on, not only this research, but my graduate career in general. Jacob, you not only recruited me to UIUC (with the help of Chris Martin), but you had confidence in my abilities before any one else and basically saw to it that I would be performing astrodynamics research in this department. Not only was your work the seed for this thesis, but there is nothing in the following pages that you and I have not discussed in intimate detail. I have made a friend for life, and feel I have found an enthusiastic collaborator for the rest of my career.

While many people at the University of Illinois have played an important role in my graduate career so far, I would be remiss if I did not mention, and thank, the professors that I had at Eastern Michigan University who prepared me so well for graduate school. Drs. Patrick Koehn, Diane Jacobs, and Ernest Behringer as well as the rest of the faculty in the Department of Physics and Astronomy deserve a lot of credit for getting me to where I am today. Dr. Koehn, in particular, helped me confirm my interest in orbital mechanics and piqued my interest in planetary science when I took his Comparative Planetology class. I would also like to thank Drs. Andrew Ross, Jayakumar Ramanathan and David Folk, among others, in the Department of Mathematics for their rigorous treatment of the tools I use every day.

Just because it seems so long ago does not mean that I can forget my first physics teacher from King City Secondary School, Mr. David Bocknek, who started me down this academic path. Thank you Mr. Bocknek for pushing us hard so early on.

Nikita, you have done more to support me and ensure my success than you could possibly know. Not only did you accept my decision to move away for graduate school, you then followed me to the prairies. You have had, and continue, to sacrifice more than I could ever reasonably expect. I love you.

I am very fortunate to have the family that I do. Michael, you are a continuous source of inspiration. Watching my brother go through law school reminds me what it takes to succeed. My grandparents Regina and Klaus share the dedication of this thesis for a reason. It is shocking how much of
them I see in myself. Mum and Dad, you deserve more credit than anyone else for helping to get me to where I am today. You have never wavered in the support that you give me every single day. If I have one regret over the past eight years it is that I could not have lived closer to you. I love you both very much.
## TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>LIST OF TABLES</th>
<th>ix</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIST OF FIGURES</td>
<td>x</td>
</tr>
<tr>
<td>LIST OF SYMBOLS AND ABBREVIATIONS</td>
<td>xii</td>
</tr>
<tr>
<td>CHAPTER 1 INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Thesis Organization</td>
<td>4</td>
</tr>
<tr>
<td>CHAPTER 2 DYNAMICAL SYSTEM</td>
<td>5</td>
</tr>
<tr>
<td>2.1 The Two-Body Problem</td>
<td>5</td>
</tr>
<tr>
<td>2.1.1 State Propagation with Kepler’s Equation</td>
<td>6</td>
</tr>
<tr>
<td>2.2 Patched Conic Method</td>
<td>8</td>
</tr>
<tr>
<td>2.3 Gravity-Assist Trajectories</td>
<td>9</td>
</tr>
<tr>
<td>CHAPTER 3 DIRECT METHOD OPTIMIZATION OF A LOW-THRUST TRAJECTORY</td>
<td>11</td>
</tr>
<tr>
<td>3.1 Direct Method Optimization of a Low-Thrust Trajectory</td>
<td>11</td>
</tr>
<tr>
<td>3.2 EMTG: Evolutionary Mission Trajectory Generator</td>
<td>12</td>
</tr>
<tr>
<td>3.3 Outer-Loop Transcription</td>
<td>12</td>
</tr>
<tr>
<td>3.3.1 Integer Genetic Algorithm</td>
<td>12</td>
</tr>
<tr>
<td>3.4 Inner-Loop Transcription</td>
<td>14</td>
</tr>
<tr>
<td>3.4.1 Nonlinear Program</td>
<td>14</td>
</tr>
<tr>
<td>3.4.2 The Sims-Flanagan Transcription</td>
<td>15</td>
</tr>
<tr>
<td>3.4.3 Monotonic Basin Hopping</td>
<td>20</td>
</tr>
<tr>
<td>3.4.4 Automated Selection of Decision Vector Bounds</td>
<td>22</td>
</tr>
<tr>
<td>3.5 EMTG_tour: Planetary Satellite Tour Module</td>
<td>22</td>
</tr>
<tr>
<td>3.5.1 Outer-Loop Chromosome Restructuring</td>
<td>23</td>
</tr>
<tr>
<td>3.5.2 Decision Vector Changes</td>
<td>24</td>
</tr>
<tr>
<td>3.5.3 Constraint Equation Rescaling</td>
<td>25</td>
</tr>
<tr>
<td>3.5.4 Ephemeris Model</td>
<td>25</td>
</tr>
</tbody>
</table>
LIST OF TABLES

3.1 Parameters for the Integer GA . . . . . . . . . . . . . . . . . . 13
3.2 Sims-Flanagan transcription decision vector. . . . . . . . . 19

4.1 Performance characteristics for various high-powered HET models. *The NASA-457M thruster has actually achieved a maximum thrust of at least 3.3 N in laboratory testing at elevated power levels of around 100 kW.* 32
4.2 Summary of the JIMO reference trajectory (Table 4 in Whiffen and Lam$^5$). . . . . . . . . . . . . . . . . . . . . . . . . . 33
4.3 Options set by the user prior to inner-loop optimization for the JIMO mission design. . . . . . . . . . . . . . . . . . . . . . . 38
4.4 JIMO preliminary mission design summary. . . . . . . . . . . 39
4.5 Decision variable bounds used for journey 1: Jupiter capture to Callisto rendezvous. . . . . . . . . . . . . . . . . . . . . . . . 40
4.6 Decision variable bounds used for journey 2: Callisto departure to Ganymede rendezvous. . . . . . . . . . . . . . . . . . . . 42
4.7 Decision variable bounds used for journey 1: Jupiter capture to Callisto rendezvous. . . . . . . . . . . . . . . . . . . . . . . . 44

5.1 Example of automatic differentiation using forward accumulation. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 50
5.2 Decision variable bounds used for single-phase Callisto rendezvous starting at 1000 $R_J$. . . . . . . . . . . . . . . . . . . . . . . . . 56
5.3 Options set by the user prior to inner-loop optimization. . . . 56
5.4 Results for the single-phase Callisto rendezvous using mass parametrization . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 57
# LIST OF FIGURES

2.1 Flyby maneuver geometry. ........................................ 9
2.2 Flyby vector geometry. ........................................... 10

3.1 Two-phase low-thrust mission using the Sims-Flanagan Transcription (with non-optimized match point discontinuity). Image courtesy of Jacob Englander. 

3.2 Monotonic Basin Hopping algorithm. ............................... 21
3.3 RA ($\alpha$) and DEC ($\beta$) decision parameters defining the initial spacecraft position vector (in red). The RA is referenced with respect to the ascending node vector $\Upsilon$ and the DEC is referenced with respect to the central body’s mean equatorial plane (in grey). ........................................ 24

4.1 PB1 spacecraft bus model. .......................................... 29
4.2 DSV spacecraft design detailing the proposed multiple stage interplanetary injection and faring-stowed spacecraft configurations. .................................................. 30
4.3 Nuclear fission reactor and Brayton power converter. ............... 31
4.4 Earth injection to Callisto capture. ................................ 33
4.5 Jupiter capture to Callisto capture, energy pump-down phase. 34
4.6 Callisto science orbit to Ganymede capture. ....................... 35
4.7 Ganymede science orbit to Europa capture. ....................... 36
4.8 Spiral down to Europa science orbit in the non-rotating, Europa-centered frame. ........................................ 37
4.9 Jupiter capture phase starting at 226.66 $R_J$. The spacecraft performs three flybys of Callisto before rendezvous occurs. ........................................ 41
4.10 Journey from Callisto escape to Ganymede. The spacecraft performs one flyby of Callisto and one flyby of Ganymede before rendezvous. ........................................ 43
4.11 Journey from Ganymede escape to Europa. The spacecraft performs one flyby of Ganymede and one flyby of Europa before rendezvous. ................................ 45
5.1 Sims-Flanagan Jacobian sparsity pattern for a single-phase planet to planet journey. .......................... 51
5.2 Mass parametrized Sims-Flanagan Jacobian sparsity pattern for a single-phase journey. ......................... 54
5.3 Single-phase Callisto rendezvous using mass parametrization. Analytical expressions for all partial derivatives of the $\Delta V$ magnitude and mass constraints were provided to SNOPT. ...................................................... 58
5.4 Close up of the Callisto rendezvous. ................................. 59
5.5 Two phase Callisto rendezvous using mass parametrization. The problem Jacobian was completely calculated using finite differencing. ...................................................... 61
5.6 Close up of the two phase Callisto rendezvous. .............. 62
5.7 Unit-vector control Sims-Flanagan Jacobian sparsity pattern for a single-phase journey. .......................... 64
LIST OF SYMBOLS AND
ABBREVIATIONS

AU  astronomical unit
C   nonlinear constraint vector
C_l  linear constraint vector
D   thruster duty cycle
DU  normalized length unit
E   eccentric anomaly
G   universal gravitational constant
H   hyperbolic eccentric anomaly
Isp  specific impulse
J   nonsingular skew-symmetric matrix
J   scalar cost function
M   mean anomaly
M_0  mean anomaly at epoch
N   number of time steps in a Sims-Flanagan transcription phase
N_p  number of phases in a Sims-Flanagan transcribed trajectory
P   outer-loop population vector
R_J  Jovian radii
T_{max}  maximum thrust level
T   orbital period
TU  normalized time unit
\( \Upsilon \) first point in Ares/ascending node of central body mean equatorial plane

\( \mathbf{V} \) body/spacecraft velocity vector

\( V \) magnitude of the body/spacecraft velocity vector

\( \alpha \) right ascension of moon tour starting point

\( a \) semimajor axis

\( \beta \) declination of moon tour starting point

\( \delta \) flyby turn angle

\( \Delta \) change in

\( e \) orbital eccentricity

\( f \) true anomaly

\( g_0 \) acceleration due to Earth’s gravity at sea level

\( h \) orbital angular momentum

\( i \) orbital inclination

\( l \) lower bounds vector

\( m \) mass

\( m_{central} \) mass of the central gravitational body

\( m_f \) final mass

\( \mu \) standard gravitational parameter

\( \mathbf{r} \) body/spacecraft position vector

\( r \) magnitude of the body/spacecraft position vector

\( r_{periapse} \) periapse radius

\( r_{planet} \) planetary radius

\( t_0 \) epoch time at the beginning of a phase

\( t_f \) epoch time at the end of a phase

\( t_{flight} \) journey time of flight

\( t_{max} \) upper bound on the time of flight
\( t_{\text{min}} \) lower bound on the time of flight
\( t_{\text{wait}} \) journey wait time
\( u \) upper bounds vector
\( u, v \) journey initial velocity directions
\( \{u_x, u_y, u_z\} \) thrust throttle parameters
\( x \) decision vector
\( x^* \) optimal decision vector
\( x_f \) feasible point for an NLP
\( \{x, y, z\} \) Cartesian components of the position vector
\( \omega \) argument of periapse
\( \Omega \) longitude of the ascending node

**Superscripts**

\( \cdot \) derivative with respect to time
\( \hat{\cdot} \) unit vector
\( \odot \) with respect to the heliocentric frame of reference

**Subscripts**

\( \infty \) hyperbolic excess
\( \odot \) of the sun
\( b \) backwards
\( f \) forwards
\( mp \) match point
\( s/c \) spacecraft
\( x \) x-component
\( y \) y-component
\( z \) z-component
<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AD</td>
<td>Automatic Differentiation/Algorithmic Differentiation</td>
</tr>
<tr>
<td>ADOL-C</td>
<td>Automatic Differentiation by OverLoading in C++</td>
</tr>
<tr>
<td>CT</td>
<td>Coordinate Time</td>
</tr>
<tr>
<td>DE</td>
<td>Differential Evolution</td>
</tr>
<tr>
<td>DOE</td>
<td>Department of Energy</td>
</tr>
<tr>
<td>DRO</td>
<td>Distant Retrograde Orbit</td>
</tr>
<tr>
<td>DSV</td>
<td>Deep Space Vehicle</td>
</tr>
<tr>
<td>EMTG</td>
<td>Evolutionary Mission Trajectory Generator</td>
</tr>
<tr>
<td>EPS</td>
<td>Electric Propulsion Stage</td>
</tr>
<tr>
<td>ESA</td>
<td>European Space Agency</td>
</tr>
<tr>
<td>GA</td>
<td>Genetic Algorithm</td>
</tr>
<tr>
<td>HET</td>
<td>Hall Effect Thruster</td>
</tr>
<tr>
<td>HiPEP</td>
<td>High Power Electric Propulsion</td>
</tr>
<tr>
<td>HOCP</td>
<td>Hybrid Optimal Control Problem</td>
</tr>
<tr>
<td>JIMO</td>
<td>Jupiter Icy Moons Orbiter</td>
</tr>
<tr>
<td>JPL</td>
<td>Jet Propulsion Laboratory</td>
</tr>
<tr>
<td>JUICE</td>
<td>Jupiter Icy Moon Explorer</td>
</tr>
<tr>
<td>MALTO</td>
<td>Mission Analysis Low-Thrust Trajectory Optimization</td>
</tr>
<tr>
<td>MBH</td>
<td>Monotonic Basin Hopping</td>
</tr>
<tr>
<td>MGA</td>
<td>Multiple Gravity Assist</td>
</tr>
<tr>
<td>MGA-DSM</td>
<td>Multiple Gravity Assist with 1 Deep Space Maneuver</td>
</tr>
<tr>
<td>MGA-LT</td>
<td>Multiple Gravity Assist with Low-Thrust</td>
</tr>
<tr>
<td>MJD</td>
<td>Mean Julian Date</td>
</tr>
<tr>
<td>NEXIS</td>
<td>Nuclear Electric Xenon Ion System</td>
</tr>
<tr>
<td>NLP</td>
<td>Nonlinear Program</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>NR</td>
<td>Naval Reactors</td>
</tr>
<tr>
<td>PB1</td>
<td>Prometheus Baseline 1</td>
</tr>
<tr>
<td>PSO</td>
<td>Particle Swarm Optimization</td>
</tr>
<tr>
<td>QP</td>
<td>Quadratic Programming</td>
</tr>
<tr>
<td>SDC</td>
<td>Static/Dynamic Control</td>
</tr>
<tr>
<td>SNOPT</td>
<td>Sparse Nonlinear OPTimizer</td>
</tr>
<tr>
<td>SOI</td>
<td>Sphere of Influence</td>
</tr>
<tr>
<td>SQP</td>
<td>Sequential Quadratic Programming</td>
</tr>
<tr>
<td>STM</td>
<td>State Transition Matrix</td>
</tr>
<tr>
<td>TOF</td>
<td>Time of Flight</td>
</tr>
<tr>
<td>VARITOP</td>
<td>Variational Calculus Trajectory Optimization Program</td>
</tr>
</tbody>
</table>
1.1 Introduction

Historically, robotic exploration of our solar system has been performed by spacecraft using high-thrust chemical propulsion systems. All missions to the planets in our solar system, as of the writing of this document, have made use of chemical engines. As technology advances, the complexity of these missions increases. In order to accommodate this increase in mission complexity, the use of continuous low-thrust propulsion systems is garnering more attention by mission planners. Low-thrust propulsion systems reduce the amount of propellant mass required by a spacecraft which, in turn, allows for the delivery of larger scientific payloads to destinations throughout the solar system. While there are many benefits to using low-thrust, trajectory planning for this class of space missions is a very challenging problem. A great deal of research has been performed over the past few decades towards the development of techniques and tools to aid in the design and optimization of low-thrust trajectories. These techniques fall into two main categories, indirect and direct methods. While indirect methods result in a continuous time history of the spacecraft’s state and control vectors, they are highly sensitive to a required initial guess as well as knowledge of the non-intuitive problem co-states. Direct methods transform the optimization problem into a nonlinear program (NLP) by casting important mission events as explicit NLP decision variables as well as by parametrizing the low-thrust trajectory resulting in a discrete version of the true continuous control history. Direct methods tend to be more robust than indirect methods and allow for the design of much more complex trajectories.
One of the most well-known direct transcription methods is the one introduced by Jon Sims and Steve Flanagan at the Jet Propulsion Laboratory (JPL) in 1999. This method has since been used by many researchers in the preliminary design of low-thrust trajectories and has been incorporated into several software packages including the Mission Analysis Low-Thrust Optimization (MALTO) tool used at JPL. Recently, Englander and Conway at the University of Illinois designed the Evolutionary Mission Trajectory Generator (EMTG) software tool that performs a trajectory analysis which is qualitatively similar to MALTO’s, but incorporates global optimization capability through the use of a Genetic Algorithm (GA) and Monotonic Basin Hopping (MBH). This tool has been shown to be very robust for the optimization of heliocentric trajectories, but lacks the capability to design a potential planetary moon tour upon the completion of the interplanetary phase of the flight.

The first contribution of this thesis is to address the moon tour problem. Specifically, EMTG was given the capability to design a low-thrust moon tour starting at a free point in space, such as a point on the planet’s sphere of influence (SOI) boundary. In order to test its capabilities, the preliminary trajectory design stage of the now-cancelled Jupiter Icy Moons Orbiter (JIMO) mission was revisited. This nuclear electric mission was to be the first of NASA’s now-cancelled Prometheus class of missions which would have helped develop nuclear powered technology for future interplanetary spacecraft. Before JIMO’s cancelation in 2005, Greg Whiffen et al. at JPL designed a reference trajectory using the high-fidelity software program Mystic that would have been used by the JIMO spacecraft. This work details the results of a preliminary design version of this trajectory generated using a modified version of the EMTG called EMTG_tour and discusses the challenges faced while doing so. Most importantly it was discovered that it becomes increasingly difficult for SNOPT to converge on feasible solutions in dynamical systems with fast characteristic time scales. Since the orbital periods of the Jovian moons are several orders of magnitude shorter than the orbital periods of most of the planets, spacecraft trajectories in the Jovian system tend to require a much higher number of revolutions about the central body as well as a greater number of Sims-Flanagan time steps. It is very difficult for EMTG to optimize trajectories of this complex nature, and the JIMO trajectory tested the limits of EMTG’s performance and robustness.
The second main contribution of this paper is a set of general methods that can be used to improve the robustness of a gradient-based optimizer employing the Sims-Flanagan transcription. Since gradient-based solvers like SNOPT use the problem Jacobian in their optimization algorithms, accurate knowledge of the Jacobian sparsity pattern and dense entry values is essential. One can determine the sparsity pattern by randomly sampling the decision space and computing the derivatives at a set of trial points with a finite difference method; however this is not robust — sometimes dense entries in the Jacobian will be missed. By default, SNOPT will compute the entries of the problem Jacobian using a finite differencing routine which is located in the subroutine \textit{snJac}. In this work, the sparsity pattern is constructed at run-time of the optimizer using an analytical method. In addition, analytical expressions for the values of many of the Jacobian entries are derived and provided to SNOPT.

Both of these contributions could potentially advance the EMTG software package which is undergoing continued development at the Goddard Space Flight Center. Future work will involve integrating planetary satellite tour planning capability with the main EMTG program, a process which will involve replacing the Keplerian propagator currently in used in this work with the SPICE ephemeris kernel. Other possible future research efforts will include incorporating additional mission constraints to handle, among other things, spacecraft radiation exposure and planetary preservation considerations. Most importantly, however, will be the incorporation of a method to handle many-revolution trajectories, either through modifications to the Sims-Flanagan transcription or through the use of an existing method from the literature.
1.2 Thesis Organization

Chapters 2 and 3 describe the dynamical system that EMTG and EMTG_tour use to model spacecraft trajectories. Specifically, Chapter 2 describes the two-body problem, state propagation via Kepler’s equation, the patched conics method as well as the equations used to model gravity assist maneuvers. Chapter 3 then describes the general methods available for the optimization of spacecraft trajectories, the details of the core EMTG program as well as the design of the EMTG_tour tool.

A brief history of the robotic exploration of the Jovian system as well as an overview of the now-canceled JIMO mission is given in Chapter 4. The preliminary design of the JIMO mission trajectory is revisited and the results are also presented in Chapter 4 as well.

General methods for improving the robustness of an NLP solver employing the Sims-Flanagan transcription are described in Chapter 5. A novel low-thrust transcription which parametrizes the spacecraft’s mass at the center of each Sims-Flanagan time segment is described and validated. Chapter 5 also includes analyses of the sparsity patterns for the original Sims-Flanagan transcription introduced in 1999, the mass parametrized version introduced here as well as the now standard unit vector control variant of the Sims-Flanagan transcription. It is shown that the mass parametrized and unit vector control variants result in an increase in the sparsity of the Jacobian, which is beneficial to the NLP solver SNOPT. In addition to the sparsity pattern mappings, analytical expressions for the majority of the constraint partial derivatives for the unit vector transcription are derived. The final chapter summarizes the contributions made by this work and indicates possible avenues for future research efforts.
2.1 The Two-Body Problem

The core EMTG program, is a medium-fidelity trajectory planning software package. The two driving factors behind its medium-fidelity classification are:

i) the transcription method used to approximate a low-thrust trajectory

ii) the dynamical system model in which these trajectories are computed

The first item in the above list is discussed in Section 3.4.2. The second item is the famous two-body problem, which was first solved in a geometric fashion by Newton about 1685, given a preliminary analytical treatment by Daniel Bernoulli in 1734 and solved in detail by Leonhard Euler in 1744. The two-body problem, as originally described by Newton states, “Given at any time the positions and velocities of two massive particles moving under their mutual gravitational force, the masses also being known, calculate their position and velocities for any other time.”

For two mass particles \( m_1 \) and \( m_2 \), the motion of \( m_2 \) with respect to \( m_1 \) is described by the following second order ordinary differential equation.

\[
\frac{d^2 r}{dt^2} + \frac{\mu}{r^3} r = 0 \quad (2.1)
\]
where \( \mathbf{r} = \mathbf{r}_2 - \mathbf{r}_1 \) is the vector directed from the center of mass of \( m_1 \) to the center of mass of \( m_2 \), \( \mu = G(m_1 + m_2) \) and \( G = 6.67384 \times 10^{-11} m^3 kg^{-1} s^{-2} \) is the universal gravitational constant. In Cartesian coordinates, Equation (2.1) can be expressed in vector component form using the fact that \( \mathbf{r} = x\hat{i} + y\hat{j} + z\hat{k} \) with magnitude \( r = \sqrt{x^2 + y^2 + z^2} \):

\[
\begin{align*}
\frac{d^2 x}{dt^2} + \mu \frac{x}{r^3} &= 0 \\
\frac{d^2 y}{dt^2} + \mu \frac{y}{r^3} &= 0 \\
\frac{d^2 z}{dt^2} + \mu \frac{z}{r^3} &= 0
\end{align*}
\] (2.2)

The system of differential equations above can, of course, be solved numerically to obtain the position of \( m_2 \) with respect to \( m_1 \) at any point in time, however, an analytical solution exists for the two-body problem which eliminates the inherent error accumulation introduced with numerical integration.

### 2.1.1 State Propagation with Kepler’s Equation

In the two-body dynamical model, the relationship between a satellite’s angular position along its elliptic orbit \( f \) and its radial distance \( r \) from the primary is given by Equation (2.3):

\[
r = \frac{a(1 - e^2)}{1 + e \cos(f)}
\] (2.3)

where \( a \) and \( e \) are the semimajor axis and the eccentricity of the orbit respectively. This does not, however, provide an explicit link between the satellite’s orbital position and time. To establish this link, the concept of the mean anomaly must be introduced; that is the quantity \( M \) which describes the angular position of an imaginary spacecraft traveling along the elliptical path at a constant mean angular rate, instead of the true time-varying rate \( \dot{f} \).

----

8
\[ M = n(t - t_0) + M_0 = \frac{2\pi}{T}(t - t_0) + M_0 = \sqrt{\frac{\mu}{a^3}}(t - t_0) + M_0 \quad (2.4) \]

The quantity \( T \) is the spacecraft’s *orbital period*, \( t_0 \) is an epoch time when a previous value of the mean anomaly, \( M_0 \) was known and \( t \) is the elapsed time since \( t_0 \) (note that \( M = 0 \) at orbital periapse). The key to the link between time and space is Kepler’s Equation (2.5):

\[
M = \begin{cases} 
E - e \sin(E) & \text{if } e < 1 \\
 e \sinh(H) - H & \text{otherwise}
\end{cases} \quad (2.5)
\]

where \( E \) is the *eccentric anomaly* and \( H \) is the *hyperbolic eccentric anomaly*. It is usually more convenient to re-write Equation (2.5) using the Gudermannian transformation:

\[
M = \begin{cases} 
E - e \sin(E) & \text{if } e < 1 \\
 e \tan(H) - \log(\tan(\frac{1}{2}H + \frac{\pi}{4})) & \text{otherwise}
\end{cases} \quad (2.6)
\]

The above are examples of transcendental functions, which must be solved for \( E \) or \( H \) using an iterative method like Newton-Raphson\(^9,10\) or Laguerre-Conway.\(^11\) Once \( E \) has been determined, the *true anomaly* \( f \) may be calculated using Equation (2.7):

\[
\tan\left(\frac{E}{2}\right) = \left(\frac{1 - e}{1 + e}\right)^{1/2} \tan\left(\frac{f}{2}\right) \quad (2.7)
\]

In order to complete the loop, and determine the spacecraft’s position and velocity vectors in Cartesian coordinates, the following two equations may be applied, using the classical orbital elements \( \{a, e, i, \Omega, \omega, f\} \) used to describe the spacecraft’s orbit:
\[ r = r(\cos \Omega \cos \theta - \sin \Omega \sin \theta \cos i) \hat{i} + r(\sin \Omega \cos \theta + \cos \Omega \sin \theta \cos i) \hat{j} + r \sin \theta \sin i \hat{k} \] (2.8)

\[ \mathbf{V} = -\frac{\mu}{h}[\cos \Omega (\sin \theta + e \sin \omega) + \sin \Omega (\cos \theta + e \cos \omega) \cos i] \hat{i} - \frac{\mu}{h}[\sin \Omega (\sin \theta + e \sin \omega) - \cos \Omega (\cos \theta + e \cos \omega) \cos i] \hat{j} + \frac{\mu}{h}(\cos \theta + e \cos \omega) \sin i \hat{k} \] (2.9)

where \( \theta = \omega + f \), \( h = \sqrt{\mu a(1 - e^2)} \) and \( r \) is determined using Equation (2.3).8

2.2 Patched Conic Method

The two-body problem assumes that only one massive body influences the orbital motion of a spacecraft at a time. In reality this, of course, is not true. A spacecraft in orbit around the Earth, for example, experiences non-insignificant gravitational effects from the moon and the sun as well. As long as a spacecraft is within a body’s sphere of influence (SOI), however, its motion can be approximated with the two-body model so long as high-fidelity accuracy is not required. The patched conic method is designed to allow for interplanetary trajectory calculation in the two-body framework and does not treat departure and destination bodies as massless. A spacecraft leaving a body’s SOI does so on a hyperbolic escape trajectory with respect to the departure body, then proceeds on a heliocentric elliptic path during the interplanetary phase of its flight then, finally, approaches its destination on a hyperbolic approach path with respect to the destination body. This method also allows for the incorporation of flyby maneuvers into the trajectory model and is used to model such maneuvers in this work.
2.3 Gravity-Assist Trajectories

The *gravity-assist* (or flyby) maneuver was first performed by the Soviet probe Luna 3 in 1959 in order to photograph the far side of the moon. The use of a gravity-assist trajectory with the express purpose of reaching another planet in our solar system was first made by the Mariner 10 spacecraft as it passed by the planet Venus on its way to the planet Mercury in 1974. This maneuver provides a means of altering a spacecraft’s velocity vector without the use of any on-board propellant. In many cases, the ability to use flyby trajectories may actually determine whether or not a certain mission profile is even feasible, as was the case with the Voyager 1 and 2, Galileo, Ulysses, Cassini and MESSENGER spacecraft as well as, most recently, the Juno mission to Jupiter. The flyby begins with the spacecraft entering a planet’s SOI and ends with its exit of the SOI as shown in Figure 2.1. While within the SOI, the spacecraft makes its closest approach to the planet at a radial distance $r_{periapse}$ corresponding to the periapse of the hyperbolic orbit.

![Figure 2.1: Flyby maneuver geometry.](image)

In Figure 2.1, $\delta$ is the *turn angle* of the hyperbolic orbit and $V_{\infty-in}$ and
**V**∞−out are the incoming and outgoing hyperbolic excess velocities respectively in the frame of the targeted flyby planet. While the magnitude of the spacecraft’s hyperbolic excess velocity does not change as a result of the flyby in the frame of the targeted flyby planet, it is altered when examined in the heliocentric frame of reference. This is most easily observed with the vector diagram in Figure 2.2.

![Figure 2.2: Flyby vector geometry.](image)

In Figure 2.2 **V**⊙s/c−in, **V**⊙s/c−out and **V**⊙planet are the pre-flyby spacecraft velocity, post-flyby spacecraft velocity and planetary velocity vectors all measured in the heliocentric frame. The vectors **V**∞−in and **V**∞−out are the incoming and outgoing spacecraft hyperbolic excess velocity vectors measured in the planet’s frame of reference. The change in the velocity vector of the spacecraft in either frame is ∆**V**flyby. It is important to note that an approximation is introduced with this flyby model. The flyby is treated as an instantaneous event in the heliocentric frame (occurring at a time t_{flyby}), in the sense that the duration of the maneuver is a very small fraction of the planet’s orbital period. This results in an instantaneous velocity change in the heliocentric trajectory resembling a “kink” in the flight path.

The only other unknowns to be calculated are the hyperbolic excess velocity vector which is given by Equation (2.10) and the flyby turn angle which is given by Equation (2.11).

\[ V_\infty = V_{s/c}^{\odot}(t_{flyby}) - V_{planet}^{\odot}(t_{flyby}) \] (2.10)

\[ \delta = 2 \tan^{-1}\left(\frac{\mu_{planet}}{|r_{s/c}(t_{flyby}) - r_{planet}(t_{flyby})| V_\infty^2}\right) \] (2.11)
CHAPTER 3

DIRECT METHOD OPTIMIZATION OF A LOW-THRUST TRAJECTORY

3.1 Direct Method Optimization of a Low-Thrust Trajectory

As explained previously, for most mission architectures, the low-thrust spacecraft trajectory optimization problem is far too complex to take advantage of an indirect method of optimization, that is a method derived from Pontryagin’s Minimum Principle. This is mainly due to the sensitivity of the problem to the required initial guesses of the states and co-states. This disadvantage is made worse by the fact that the problem co-states, and even some of the states themselves, may not be physically intuitive, making the provision of intelligent initial guesses for them extremely challenging, if not impossible. For this reason, a direct method transcription is often utilized that parametrizes the problem control variables and acts to minimize a chosen cost function directly. Direct methods are advantageous primarily because necessary conditions do not have to be derived for each individual problem. These methods also tend to be less sensitive to the initial guess; however, at the same time, the number of problem variables tends to be larger when compared with indirect methods and the discretization of the problem inherently introduces errors. One direct method casts the low-thrust trajectory optimization problem as a hybrid optimal control problem (HOCP) whereby the sequence of flyby targets (categorical variables) is optimized by an “outer-loop” and the real-valued variables defining the trajectory between flybys is optimized by an “inner-loop” which casts the problem as a nonlinear program (NLP).
3.2 EMTG: Evolutionary Mission Trajectory Generator

The Evolutionary Mission Trajectory Generator (EMTG) is a medium-fidelity, direct method, trajectory optimization software package originally developed at the University of Illinois by Jacob Englander.\(^3\),\(^4\) This program defines an automaton capable of optimizing an interplanetary trajectory for a spacecraft using high thrust chemical or low thrust electric engines. EMTG solves HOCP’s conforming to three different standard spacecraft trajectory models: multiple gravity-assist (MGA), multiple gravity-assist with one deep space maneuver per phase (MGA-DSM) and multiple gravity-assist using continuous thrust (MGA-LT). To do so, a HOCP is constructed using the two nested loop structure described previously.

3.3 Outer-Loop Transcription

The outer-loop of EMTG determines the optimal flyby sequence of the spacecraft trajectory. This is done by assigning each possible flyby target an integer and solving the outer-loop as an integer programming problem. Since not all flyby sequences will be of equal length, Englander and Conway developed a method using “null” values where a fixed length decision vector is used to represent all possible flyby sequence lengths.\(^3\),\(^4\) When the outer-loop fitness function parses a decision vector, it omits all null values and constructs the flyby sequence using only values representing allowable flyby targets. This means that the outer-loop fitness function can optimize not only the flyby sequence but its length as well, all without having the sequence length appear as an explicit decision variable.

3.3.1 Integer Genetic Algorithm

The concept of the Genetic Algorithm (GA), first proposed by John Holland,\(^13\) is a population-based search heuristic that mimics the evolution of a species. The integer GA used in EMTG employs four different operators inspired by natural selection: selection, crossover, mutation and elitism. The algorithm starts by initializing a population of vectors \(P\) in an \(n\)-dimensional
vector space and then proceeds as follows for a certain user-specified number of generations. First, the fitness of each member of the population is evaluated. Next, the selection operator is applied to the population to determine the best members of the population which will become the parents of the next generation. The crossover operation then creates child vectors using those vectors selected as parents. These child vectors replace a certain user-specified fraction of the current population known as the crossover ratio. Finally, mutation occurs at random with small probability in the new population. In addition to the three previously mentioned mechanisms, in each generation, a certain number of elite individuals are selected. Those individuals tagged as elite are guaranteed to be members of the new population and are not modified in any way.

The GA employed in EMTG is an in-house developed integer genetic algorithm. The selection operator used is “tournament selection” and the crossover and mutation operators are “binary crossover” and “uniform mutation” respectively. The GA proceeds for a certain number of generations or until the fitness has “stalled” (i.e. not improved for a certain number of generations). The integer GA settings used in this work are summarized in Table 3.1.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Population size</td>
<td>100</td>
</tr>
<tr>
<td>Maximum number of generations</td>
<td>40</td>
</tr>
<tr>
<td>Maximum number of stall generations</td>
<td>5</td>
</tr>
<tr>
<td>Crossover ratio</td>
<td>0.3</td>
</tr>
<tr>
<td>Mutation probability</td>
<td>0.05</td>
</tr>
<tr>
<td>Elite count</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 3.1: Parameters for the Integer GA
3.4 Inner-Loop Transcription

For a given flyby sequence, determined by the outer-loop, EMTG’s inner-loop optimizes the trajectory between control points. The MGA problem is solved using a cooperative evolutionary optimization scheme involving Particle Swarm Optimization (PSO) and Differential Evolution (DE) originally proposed by Tamas Vinkó and Dario Izzo,\textsuperscript{14} whereby one optimization scheme is run for some small number of generations, then its final population is used as the initial population for the other technique. This is repeated until the fitness function does not improve by more than a specified tolerance. The MGA-DSM problem is solved using DE exclusively.

This work focuses on the MGA-LT problem that uses the Sims-Flanagan transcription described in Section 3.4.2 to transform a continuous trajectory into real-valued NLP which is then solved with the NLP software package SNOPT.\textsuperscript{15}

3.4.1 Nonlinear Program

A nonlinear program is an optimization problem whereby the minimization (or maximization) of an objective function $J(x)$ is achieved by varying a set of real variables $x$, subject to a set of equalities $C_l(x)$ and inequalities $C(x)$, commonly referred to as constraints. As mentioned in the previous section, this is usually achieved with a numerical NLP solver such as IPOPT\textsuperscript{16} or SNOPT. The underlying engine in SNOPT is a sparse sequential programming (SQP) algorithm which solves problems of the following form:

\begin{equation}
\text{Minimize } J(x) \\
\text{subject to } 1 \leq \begin{pmatrix} x \\ C(x) \\ C_l(x) \end{pmatrix} \leq u \tag{3.1}
\end{equation}

In Equation (3.1) $J(x)$ is a smooth scalar function, $C(x)$ is a vector of smooth nonlinear constraint functions, $C_l(x)$ is a vector of linear constraint functions.
and \( \{l, u\} \) are vectors of constant lower and upper bounds on \( x, C \) and \( C_i \). A feasible point \( x_f \) for the problem posed in Equation (3.1) is one that satisfies all of the problem constraints.\(^{17}\)

As stated previously, a direct transcription of the low-thrust optimization problem results in a nonlinear program of the form described in Equation (3.1). The specific transcription used in this work is described in the next section.

### 3.4.2 The Sims-Flanagan Transcription

There are several choices when it comes to low-thrust trajectory transcription methods, however, the parametrization introduced by Sims and Flanagan\(^1\) has been shown to be robust and allow for the quick execution needed for preliminary design studies in the context of an MGA-LT design space.\(^{18}\) The Sims-Flanagan transcription discretizes the trajectory into equal-sized time steps. The continuous thrust that may be applied during each step is approximated by applying a bounded impulse at the center of each of these time steps. Since applied thrust is approximated as a discontinuous \( \Delta V \), it is possible to propagate the spacecraft’s position and velocity components using Kepler’s equation either in its elliptic or hyperbolic forms as introduced in Section 2.1.1. The Sims-Flanagan transcription organizes a low-thrust trajectory into \( N_p \) phases. Phases begin and end at control points which can be planets, satellites, small bodies such as asteroids or even free points in space. Each phase is itself divided into two halves. The first half of the trajectory starts at the previous control point and is propagated forward in time. The second half of the trajectory begins at the subsequent control point and is propagated backward in time. The optimizer selects the velocity components of the state vector at the beginning of, and at the end of each phase. The two halves of the trajectory meet at a "match point". The Sims-Flanagan transcription is presented diagramatically in Figure 3.1.
The continuity of the spacecraft’s seven-element state vector is ensured via applied nonlinear constraints. In Equation (3.2), the $mp$ subscript identifies the constraint as one enforcing match point continuity for one of the seven state vector components. The $s/c$ subscript identifies the quantity as belonging to the spacecraft. Finally the $f$ subscript refers to the forward half of the Sims-Flanagan phase starting at the previous flyby target and propagated forward in time to the match point and the $b$ subscript refers to the backward half of the phase starting at the subsequent flyby target or destination and propagated back to the match point.
A nonlinear constraint is also applied to limit the magnitude of the impulse \( \Delta V_i \) that may be applied at each time step such that the velocity change provided by the propulsion system is no greater than what could be achieved if the engine were operated continuously at maximum thrust for the full duration of the time step:

\[
C_{\Delta V} = \Delta V_i - \Delta V_{\text{max},i} \leq 0 \\
= \left[ \Delta V_x^2 + \Delta V_y^2 + \Delta V_z^2 \right]^{1/2} - \frac{D T_{\text{max}} (t_f - t_0)}{m_i N} \leq 0
\] (3.3)

where \( D \) is the thruster duty cycle (modelled continuously), \( T_{\text{max}} \) is the maximum thrust that the engine is capable of producing, \( m_i \) is the mass of the spacecraft at the center of the time step, \( N \) is the number of time steps in the current phase and \( t_0 \) and \( t_f \) are the beginning and ending epochs of the phase respectively which define the phase flight time \( t_{\text{phase}} = t_f - t_0 \). The mass of the spacecraft is not an explicit NLP parameter and is propagated using Tsiolkovsky’s rocket equation:

\[
m_i = m_{i-1} e^{\left( \frac{-\Delta V_{i-1}}{g_0 I_{\text{sp}}} \right)}
\] (3.4)

where \( g_0 = 9.80665 \text{ m/s}^2 \) is the acceleration due to Earth’s gravity at sea level and \( I_{\text{sp}} \) is the specific impulse of the spacecraft’s thruster(s).
As was previously indicated, the optimizer is free to select the spacecraft’s velocity vector components at the two control points defining a particular phase. If the current phase ends with a flyby of a massive body, two non-linear constraints are applied to ensure that the maneuver is feasible. The first constraint forces the magnitudes of the incoming and outgoing velocity asymptotes to be equal:

\[ C_{V_{\infty}} = V_{\infty-out} - V_{\infty-in} = 0 \] (3.5)

The second constraint prevents the altitude of the spacecraft from dropping below the radius of the planet. A “safety factor” is included that actually prevents the spacecraft’s flight path from ever dropping below 2% of the planet’s radius. This buffer can be adjusted if, for instance, it is desirable that the spacecraft always maintain a different minimum stand-off distance from the surface (for planetary preservation considerations, atmospheric avoidance etc.):

\[ C_{flyby} = \frac{r_{periapse}}{r_{planet}} - 1.02 \geq 0 \] (3.6)

\[ = \frac{\mu_{planet}}{V_{\infty}^2 r_{planet}} \left[ \frac{1}{\sin(\delta/2)} - 1 \right] - 1.02 \geq 0 \]

where

\[ \delta = \acos \left( \frac{V_{\infty-in} \cdot V_{\infty-out}}{|V_{\infty-in}|^2 |V_{\infty-out}|^2} \right) \] (3.7)

The last family of constraints are those relating to the time of flight (TOF) \( t_{flight} \). If the current journey flight time \( t_{flight} = \sum_{i=1}^{N} t_{phase_i} \) is unbounded, then the time of flight constraints are undefined. If upper and lower bounds, \( t_{max} \) and \( t_{min} \) respectively, are placed on \( t_{flight} \) then the journey flight time constraint is simply given by Equation (3.8).
\[ C_{TOF} = \begin{cases} 
  t_{\text{min}} - t_{\text{flight}} \leq 0, & \text{if } t_{\text{flight}} < t_{\text{min}}. \\
  t_{\text{flight}} - t_{\text{max}} \leq 0, & \text{if } t_{\text{flight}} > t_{\text{max}}. \\
  0, & \text{otherwise.} 
\] 

(3.8)

Along with the nonlinear constraints described by Equations (3.2) (3.3) (3.5) (3.6) and (3.8), one phase of the Sims-Flanagan transcription is defined by the decision variables described in Table 3.2. Note that the parameters \( t_{\text{wait}}, u, v \) and \( \Delta V_{\text{launch}} \) are only defined for the first phase of a journey. For a rendezvous, the incoming hyperbolic flyby velocity vector is constrained to be zero (i.e. by the definition of a rendezvous, the final velocity of the spacecraft with respect to the destination body is zero).

<table>
<thead>
<tr>
<th>Decision Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( t_{\text{wait}} )</td>
<td>wait time until launch/departure</td>
</tr>
<tr>
<td>( u, v )</td>
<td>angles defining RA and DEC of launch/departure asymptote</td>
</tr>
<tr>
<td>( \Delta V_{\text{launch}} )</td>
<td>magnitude of launch/departure ( \Delta V ) vector</td>
</tr>
<tr>
<td>( \alpha, \beta )</td>
<td>angles defining the start point for the first journey</td>
</tr>
<tr>
<td>( V_{\infty-in} )</td>
<td>incoming hyperbolic flyby velocity vector</td>
</tr>
<tr>
<td>( t_{\text{phase}} )</td>
<td>phase flight time</td>
</tr>
<tr>
<td>( V_{\infty-out} )</td>
<td>outgoing hyperbolic flyby velocity vector</td>
</tr>
<tr>
<td>( m_{fi} )</td>
<td>mass at the end of the phase</td>
</tr>
<tr>
<td>( N \times { \Delta V_x, \Delta V_y, \Delta V_z } )</td>
<td>velocity change vector for each time step</td>
</tr>
</tbody>
</table>

Table 3.2: Sims-Flanagan transcription decision vector.
3.4.3 Monotonic Basin Hopping

One disadvantage of a gradient-based optimization method like SQP, is that it has no global search capabilities, that is to say the solution SNOPT reports as optimal is only guaranteed to be a local minimum and is not necessarily the globally optimal solution. Global search capability is made possible in EMTG by using a global search heuristic called \textit{monotonic basin hopping} (MBH).\textsuperscript{19} This algorithm can be best described as a guided multi-start. The MBH routine selects initial decision variable values for the SNOPT solver to then iterate on, in accordance with Figure 3.2. This method has been successfully used in the optimization of several varieties of spacecraft trajectory problems.\textsuperscript{20–22} Englander incorporated MBH into EMTG in a novel way that makes use of hop sizes selected from a Cauchy distribution instead of the conventionally used uniform distribution.\textsuperscript{3} The MBH algorithm is summarized in Figure 3.2.
Figure 3.2: Monotonic Basin Hopping algorithm.
3.4.4 Automated Selection of Decision Vector Bounds

Equation (3.1) indicates that, by construction, a nonlinear program requires the specification of upper and lower bounds on the decision variables and the constraint equations. The NLP solved by SNOPT in EMTG is defined and solved as part of the program’s inner-loop. The EMTG framework, and most HOCP architectures in general, does not allow for the dynamic specification of the NLP’s upper and lower bounds. In other words, since the outer-loop determines the flyby sequence during program execution, it is impractical for the user to specify intelligent upper and lower bounds a priori and inconvenient to do so dynamically. To circumvent this problem, EMTG uses a basic decision making routine to set these bounds for each outer-loop flyby sequence. More detail on this routine is given in Chapter 4.

3.5 EMTG_tour: Planetary Satellite Tour Module

The purpose of the original EMTG program is the automated design of spacecraft trajectories at a solar system level in the medium-fidelity context of the two-body dynamical model described in Section 2.1. Missions designed within its framework have the spacecraft begin with the same position and velocity as a solar system body, as well as an initial hyperbolic velocity given by a launch vehicle or spacecraft thruster. Similarly, missions end with either a targeted flyby (where the spacecraft and the target’s positions match at the final time and a discontinuous velocity change is applied to determine the spacecraft’s final velocity), an intercept (where only the two position vectors match), a rendezvous (where both the position and velocity vectors match) or a chemical orbital insertion (where the position vectors match and an insertion burn $\Delta V$ is calculated and applied to the spacecraft to determine its final mass after going into orbit about the target body). While the planetary destination itself will certainly be a subject of investigation, recently the scientific community has started to focus its attention on the systems of natural satellites present around nearly all of the planets in our solar system. The presence of satellites in a planetary system not only increases the number of potential scientific points of interest, but also increases the complexity of the dynamical system being analyzed in that, as long as the moons are of sufficient mass, a spacecraft may utilize a sequence of flyby
gravity-assists to aid in its exploration of the system. This, of course, raises the same familiar question of, “what sequence of flybys will minimize the amount of mass needed in order to complete all of the mission objectives?”. The software module EMTG_tour was developed to solve this problem by providing a general medium-fidelity framework for designing low-thrust planetary satellite tour missions.

3.5.1 Outer-Loop Chromosome Restructuring

Fundamentally, the moon tour problem is quite similar to a heliocentric trajectory optimization, and planetary satellite systems can be thought of as scaled-down versions of the solar system itself. One key difference between the core EMTG program and EMTG_tour, is how a mission begins. In EMTG, the spacecraft begins at a solar system body and is given an initial impulse by a launch vehicle or on board engine. In EMTG_tour, the mission begins somewhere on a planet’s sphere of influence, or any other free point in space, with some velocity with respect to the central body. The simulation then proceeds from there. This divides the simulation into two main parts—the interplanetary segment and the moon tour segment, following the patched conic analysis described in Section 2.2.

This change requires that the outer-loop decision vector (i.e. the integer GA chromosome) be restructured. In EMTG, at the start of every generation, the outer-loop generates a vector of flyby sequences. To each sequence it then prepends and appends the mission starting body and destination body respectively. In EMTG_tour, there is no starting body, rather the spacecraft starts at a free point in space, simulating its transition from interplanetary flight, and only the integer corresponding to the destination body is appended to the chromosome. Thus, the first non-null entry in the chromosome will be the integer corresponding to the first flyby target, for a trajectory with multiple phases, or the destination body itself for a single-phase mission.
3.5.2 Decision Vector Changes

Since satellite tour trajectories in EMTG_tour can start at any point in space, selection of this point is usually left up to the optimizer. However, the problem could be over-constrained thereby allowing the user to specify,\textit{ a priori}, the exact point where the trajectory begins. This requires that\two\ additional parameters $\alpha$ and $\beta$ be added to the decision vector. These parameters define two angles, the\textit{ right ascension} (RA) $\alpha$ and the\textit{ declination} (DEC) $\beta$ which uniquely specifies where the trajectory originates in space. The RA is measured clockwise from the system’s x-axis, which is out along the ascending node $\Upsilon$ of the central-body mean equatorial plane, on this reference plane, at the reference epoch$^{23}$ and varies on the interval $[0, 2\pi]$. The reference plane used in this work is the primary body’s mean equatorial plane. The DEC is measured with respect to the primary body’s mean equatorial plane and is specified on the interval $[-\pi/2, \pi/2]$.

![Figure 3.3: RA ($\alpha$) and DEC ($\beta$) decision parameters defining the initial spacecraft position vector (in red). The RA is referenced with respect to the ascending node vector $\Upsilon$ and the DEC is referenced with respect to the central body’s mean equatorial plane (in grey).]
3.5.3 Constraint Equation Rescaling

While the decision vector is altered with the addition of the two new parameters described in the previous section, the constraint vector remains unchanged from its original structure in EMTG. The scaling of select individual constraints, however, must be updated to reflect the fact that the physical system in which the optimization is occurring has changed. In order to ensure good performance on the part of the local optimizer (SNOPT), it is preferable that all constraints have magnitude of order 1 (SNOPT version 7 user’s guide page 73). All of the Sims-Flanagan constraints will naturally be of order 1 except for the match point constraints described by Equation (3.2). These constraints are scaled by normalized length and time units i.e. units such that $\mu_\odot = 1$ and a massive body in a circular orbit with a radius of 1 normalized length unit (DU) has an orbital period of $2\pi$ normalized time units (TU). For the heliocentric missions modeled in EMTG, the distance unit is set equal to the astronomical unit (AU = 149 597 870 700 meters). In EMTG_tour the value of DU must be adjusted depending on the planetary system being analyzed. Once a value for DU is set, the value of TU may be calculated using Equation (3.9),

$$TU = \sqrt{\frac{DU^3}{G \times m_{central}}} \quad (3.9)$$

where $\mu_\odot = G \times m_{central} = 1$.

3.5.4 Ephemeris Model

The core EMTG program uses the SPICE data kernels in order to locate celestial bodies at any point in time. Being still in the design stage, the EMTG_tour module does not have integrated SPICE capabilities. Instead a simple two-body Kepler propagation model is used to specify the location of target bodies as a function of time, as described in Section 2.1.1. The next phase of development will work towards, among other things, the integration of the SPICE kernels into EMTG_tour.
4.1 A Brief Overview of the Exploration of the Jovian System

The \textit{in situ} exploration of Jupiter began in 1973, when Pioneer 10 became the first robotic spacecraft to visit the Jovian system and subsequently was the first to perform a flyby of the planet on its way to the outer solar system. This was followed a year later by Pioneer 11 following a similar mission profile. The Voyager 1 and 2 probes were next in the queue, both visiting the system in 1979 as part of their Grand Tours. These two spacecraft placed an increased emphasis on the exploration of the Jovian satellite system, as well as the planet’s system of rings, and actually discovered several new satellites during their missions. Two of their most significant findings were volcanic activity on the moon Io, and the presence of water ice on Europa. The next visitor to Jupiter was the Ulysses spacecraft. It was designed, primarily, to study the sun, in particular its polar regions; however, in order to gain access to these regions it required a large plane change maneuver, which only a gravity-assist by Jupiter could provide. This gravity-assist occurred in early 1992 and allowed for, among other things, measurements of the planet’s magnetosphere and the Io plasma torus. The spacecraft then revisited the system in 2003-2004 when it was at the aphelion of its orbit (which had a period of approximately six years) allowing it to make further observations of Jupiter at a distance.

The Galileo spacecraft remains, to this day, the only vehicle to actually enter into orbit about Jupiter and study the Jovian system for an extended period of time (1995-2003). During this time it released an atmospheric probe into Jupiter’s upper atmosphere and then made 35 orbits of the planet.
including 25 flybys of the Galilean moons (Io, Europa, Ganymede and Callisto) and one of Almathea which sent it on a deliberately planned crash course with the planet itself. The Galileo spacecraft conducted many scientific investigations and made many important discoveries over the course of its mission. Among others, the Io flux tube and the enormous electrical currents that interact with Jupiter’s atmosphere were studied in great detail, Ganymede was discovered to possess an intrinsic magnetic field independent of Jupiter’s, the structure of Jupiter’s immense magnetosphere was identified and, perhaps most interestingly, additional evidence of the presence of a liquid-saltwater layer ocean under the surface of Europa, Ganymede and Callisto was gathered.

The most recent explorations of Jupiter and its moon system were made by the Cassini spacecraft and the New Horizons probe in 2000 and 2007 respectively. Cassini’s primary scientific contribution was the important observations it made of Jupiter’s atmospheric circulation mechanisms. In addition to this, it sent back more than 26000 images of the planet and its moons. The New Horizons probe made its closest approach to Jupiter in February 2007 during its flyby maneuver which sent it on the final leg of its journey to Pluto. The maneuver had the spacecraft pass within approximately 32 Jovian radii of the planet. At this range, New Horizons used its long range sensors to study Jupiter’s satellites as well as the planet itself. This investigation included making orbital improvement calculations for many of the moons, observation of Io’s volcanoes as well as chemical composition and night side temperature readings of the Galilean satellites. As for Jupiter itself, the probe took data as it flew through the gas giant’s magnetotail, studied the planet’s atmospheric gas dynamics and monitored Jupiter’s night side for lightning and aurora effects.

The next scheduled visit to Jupiter will come in 2016 with the arrival of the Juno spacecraft which is currently in transit to the Jovian system. Juno will enter into a highly elongated, polar orbit around the planet. Its primary mission objectives will be to make detailed maps of Jupiter’s magnetic and gravitational fields, make a more accurate measure of Jupiter’s core mass and determine the abundance of water in Jupiter.
The exploration of the largest planet in our solar system will certainly not end with Juno’s visit. Even now plans are being made by the European Space Agency (ESA) to send a spacecraft to Jupiter for the express purpose of studying the three largest Galilean moons, Europa, Ganymede and Callisto. The Jupiter Icy Moon Explorer (JUICE) spacecraft is currently slated for a 2022 launch, which will have it arrive at Jupiter sometime in 2030.

4.2 The Jupiter Icy Moons Orbiter Mission

The Jupiter Icy Moons Orbiter (JIMO) was a proposed NASA mission that was designed to send a spacecraft to Jupiter and subsequently investigate and enter into orbit around Callisto, Ganymede and finally Europa. The JIMO mission was to be the first of NASA’s Project Prometheus which was established in 2003 to develop nuclear powered systems for long-duration spaceflights. A massive reduction in funding in 2005 effectively ended the project, but not before early planning had occurred for the JIMO mission.

The JIMO spacecraft design was revolutionary for many reasons. The most obvious was its size. The spacecraft’s total mass at the time of its interplanetary orbit injection was projected to be 36000 kg, with a dry mass of 24000 kg. As of the writing of this thesis, the largest unmanned interplanetary spacecraft ever flown is the Cassini spacecraft, currently in orbit around Saturn, which had a wet mass of 5574 kg at launch (including the Huygens Titan probe). In addition to this, JIMO had two major design features that, if the spacecraft had flown, would have allowed for an unprecedented level of exploration of the Jovian satellite system. The first was the spacecraft’s power source, a small fission reactor which would have used a power conversion system based on the Brayton cycle to convert the reactor heat into electricity. The second was the spacecraft’s mode of propulsion, low thrust ion engines for the interplanetary phase of the journey and Hall Effect Thrusters (HET) for proximity operations around the Galilean satellites once JIMO had reached Jupiter space. It was the use of low thrust that would have allowed JIMO to enter into orbit around Callisto, Ganymede and Europa. A chemical propulsion unit would not have allowed for the large ∆V requirement needed to perform the chain of orbital insertions and subsequent
escapes, a sequence which would require a $\Delta V$ of over 10 km/s.$^5$

4.2.1 JIMO Spacecraft

The preliminary planning phase of the JIMO mission detailed a truly unprecedented spacecraft design. The ship was divided into three main segments, the scientific instruments section, the radiator section, and the reactor section. The instrument stage together with the radiator section was referred to as the Prometheus Baseline 1 (PB1) design (Figure 4.1). The PB1 along with the reactor was termed the Deep Space Vehicle (DSV).

Figure 4.1: PB1 spacecraft bus model.$^{26}$

The spacecraft would have been launched by either a Delta IV or Atlas V heavy launch vehicle, and was designed to fit inside a five meter launch fairing by featuring collapsible radiator fins and support boom. The primary reason for such large physical dimensions was to separate the fission reactor from the critical scientific instruments stage. This also allowed for a large radiating surface area, of approximately 422 m$^2$, to dump excess reactor heat.$^{25,26}$ The DSV’s different configurations are detailed in Figure 4.2.

JIMO would have been the first spacecraft to be powered by a nuclear fission reactor. The United States Department of Energy (DOE) tasked the Naval Reactors (NR) office to oversee the design and construction of this reactor. The reactor was to have provided approximately 200 kW of electrical power in order to operate the electric propulsion system and the suite of on board scientific instruments. This power was to have been extracted via a Brayton cycle power converter situated immediately aft of the reactor itself.$^{27,28}$ The reactor and power converter designs are shown in Figure 4.3.
This level of available power would have allowed for very high-quality sensors, including a powerful ice penetrating radar package.

Figure 4.2: DSV spacecraft design detailing the proposed multiple stage interplanetary injection and faring-stowed spacecraft configurations.²⁵
4.2.2 JIMO Propulsion System

The JIMO Electric Propulsion Stage (EPS) was to consist entirely of electric, low-thrust engines. The baseline study called for two thruster pods mounted to the scientific payload bus. Each pod would have contained at least four high efficiency ion engines and three, higher thrust, HETs in addition to smaller HETs for dedicated attitude control. The ion thruster development plan for JIMO was dubbed Herakles and was a combination of the development efforts for the HiPEP and NEXIS thrusters which were already underway. The Herakles thruster was to have a nominal thrust of 0.65 N and operate with an input power of around 28.5 kW with an Isp of 6000-9000 s. There are several possibilities for which Hall thruster model(s)
might have been used in the final EPS design. Most reports indicate that the HETs would have a nominal thrust of around 1.0 N.\textsuperscript{33} Given this information, possible contenders could have included the NASA-400M,\textsuperscript{34} NASA-457M,\textsuperscript{35} Busek BHT-20k\textsuperscript{36} and the Pratt & Whitney T-220HT.\textsuperscript{37} Characteristics for each of these thrusters are summarized in Table 4.1.

<table>
<thead>
<tr>
<th>Hall Effect Thruster</th>
<th>Input Power (kW)</th>
<th>Isp (s)</th>
<th>Thrust (N)</th>
<th>Fuel</th>
</tr>
</thead>
<tbody>
<tr>
<td>NASA-400M</td>
<td>4-47</td>
<td>1322-2995</td>
<td>0.271 - 2.118</td>
<td>Xenon</td>
</tr>
<tr>
<td>NASA-457M</td>
<td>11-73</td>
<td>1911-2929</td>
<td>0.617 - 2.950*</td>
<td>Xenon</td>
</tr>
<tr>
<td>BHT-20k</td>
<td>5-20</td>
<td>up to 2750</td>
<td>up to 1.08</td>
<td>Xenon</td>
</tr>
<tr>
<td>T-220HT</td>
<td>2-22</td>
<td>1300-2600</td>
<td>0.1 - &gt;1.0</td>
<td>Xenon</td>
</tr>
</tbody>
</table>

Table 4.1: Performance characteristics for various high-powered HET models. *The NASA-457M thruster has actually achieved a maximum thrust of at least 3.3 N in laboratory testing at elevated power levels of around 100 kW.\textsuperscript{29}

4.3 The JIMO Reference Trajectory

The JIMO reference trajectory is quite possibly the most complex low-thrust trajectory ever designed for a space mission. The entire reference trajectory was created using the high-fidelity software Mystic,\textsuperscript{38} which relies on the second-order optimization method Static/Dynamic Control (SDC).\textsuperscript{39} Trade studies were also performed for the Earth to Jupiter phase using the low-fidelity software package VARITOP and the medium-fidelity software package MALTO. Trajectories incorporating intermediate flybys were specifically examined with MALTO, however, a direct Earth to Jupiter flight was selected for the final reference trajectory.\textsuperscript{5} The reference trajectory successfully describes a flight plan that delivers 24000 kg of dry mass into a science orbit around Europa with 1095 kg of propellant remaining. Since Mystic uses a high-fidelity multi-body dynamics model, the reference trajectory includes fully integrated spiral-in captures at each of the target Galilean moons and subsequent low-thrust escapes. A summary of the JIMO reference trajectory is given in Table 4.2 and the Mystic output is shown in Figures 4.4, 4.5, 4.6 and 4.7.
<table>
<thead>
<tr>
<th>Leg Description</th>
<th>Flight Time (days)</th>
<th>Fuel Used (kg)</th>
<th>∆V (km/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Earth injection to Jupiter capture</td>
<td>1973.5</td>
<td>5237</td>
<td>10.792</td>
</tr>
<tr>
<td>Jupiter capture to Callisto capture</td>
<td>274.5</td>
<td>464</td>
<td>1.043</td>
</tr>
<tr>
<td>Callisto centered spiral down</td>
<td>217.4</td>
<td>793</td>
<td>1.821</td>
</tr>
<tr>
<td>Callisto science orbit</td>
<td>120.0</td>
<td>0</td>
<td>0.000</td>
</tr>
<tr>
<td>Callisto to Ganymede science orbit</td>
<td>633.8</td>
<td>1782</td>
<td>4.275</td>
</tr>
<tr>
<td>Ganymede science orbit</td>
<td>120.0</td>
<td>0</td>
<td>0.000</td>
</tr>
<tr>
<td>Ganymede to Europa science orbit</td>
<td>262.0</td>
<td>2631</td>
<td>3.396</td>
</tr>
</tbody>
</table>

Table 4.2: Summary of the JIMO reference trajectory (Table 4 in Whiffen and Lam).
Figure 4.5: Jupiter capture to Callisto capture, energy pump-down phase.\textsuperscript{5}
Figure 4.6: Callisto science orbit to Ganymede capture.⁵
Figure 4.7: Ganymede science orbit to Europa capture.\textsuperscript{5}
4.4 Preliminary Recreation of the JIMO Trajectory

The JIMO mission sequence involves four main segments (called journey’s in the EMTG software): an Earth to Jupiter transfer (interplanetary phase), a transfer from interplanetary flight to a science orbit around Callisto, a transfer from the Callisto science orbit to a science orbit around Ganymede and then, finally, a transfer to a science orbit around Europa. In this work, only three of the segments are analyzed. The original EMTG program was designed to handle the interplanetary phase and the Earth to Jupiter transfer for the JIMO mission, in particular, was treated as an example problem in Englander’s Ph.D. dissertation. Candidate sequences were identified using EMTG’s outer-loop, however, it was quickly discovered that the inner-loop optimizer was not always capable of identifying feasible solutions for a given sequence in just a single trial run for this particular problem. That is to say, feasible solutions might exist for a given flyby sequence, however, the
inner-loop will not always discover any with complete reliability with just a single run. For this reason, after the outer-loop was run once for each journey, additional candidate sequences were extensively explored by trial and error based on intuition. For instance, the genetic algorithm identified the sequence Capture-C-C-rendezvous as the optimal solution to achieve a Callisto rendezvous. Using this knowledge, an additional flyby of Callisto was manually added to the inner-loop flyby sequence and several trials were performed, eventually yielding feasible solutions. Once feasible solutions were discovered their corresponding decision vectors could be used as initial guesses for additional runs of the MBH-SNOPT optimization routine, which is otherwise initialized with a completely random decision vector. Proceeding in this manner yielded an increase in the number of feasible solutions discovered. The same homotopy process was repeated for all three journeys. The best solutions found by the inner-loop optimizer are presented in the following sections. The spacecraft parameters as well as other user-specified options are summarized in Table 4.3. The positions of the Galilean satellites with respect to the Jovian body-centered geometric coordinates were seeded with JPL HORIZONS data using an epoch time of 00:00:00.0000 CT 62503 MJD.\textsuperscript{23} After this epoch time, the positions were determined using analytical propagation with Kepler’s equation (Equation (2.6)).

<table>
<thead>
<tr>
<th>User Option</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inner loop run time</td>
<td>8 hours</td>
</tr>
<tr>
<td>Number of Sims-Flanagan time steps ((N))</td>
<td>40</td>
</tr>
<tr>
<td>Earliest mission start date</td>
<td>62503 MJD (Jan. 1st 2030)</td>
</tr>
<tr>
<td>Maximum thrust (T_{\text{max}})</td>
<td>13 N</td>
</tr>
<tr>
<td>Thruster duty cycle (D)</td>
<td>0.98 (modelled continuously)</td>
</tr>
<tr>
<td>Thruster Isp</td>
<td>6000 s</td>
</tr>
<tr>
<td>Total initial spacecraft mass</td>
<td>30800 kg</td>
</tr>
<tr>
<td>Spacecraft dry mass</td>
<td>24000 kg</td>
</tr>
</tbody>
</table>

Table 4.3: Options set by the user prior to inner-loop optimization for the JIMO mission design.
The JIMO trajectory designed using EMTG_tour is summarized in Table 4.4.

<table>
<thead>
<tr>
<th>Mission Event</th>
<th>Date (MJD)</th>
<th>Spacecraft Mass (kg)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jupiter capture</td>
<td>62549.8</td>
<td>30800.0</td>
</tr>
<tr>
<td>Callisto rendezvous</td>
<td>62743.5</td>
<td>30138.1</td>
</tr>
<tr>
<td>Callisto departure</td>
<td>63054.3</td>
<td>29345.0</td>
</tr>
<tr>
<td>Ganymede rendezvous</td>
<td>63244.9</td>
<td>28674.2</td>
</tr>
<tr>
<td>Ganymede departure</td>
<td>63509.6</td>
<td>27082.7</td>
</tr>
<tr>
<td>Europa rendezvous</td>
<td>63807.4</td>
<td>26034.3</td>
</tr>
</tbody>
</table>

Table 4.4: JIMO preliminary mission design summary.

Note that although a minimum of 120 days was allocated for a science orbit around Callisto and Ganymede, the optimizer was allowed to pick a later departure date via the $t_{\text{wait}}$ decision variable.

For all results presented in the following sections, and throughout this work in general, a SNOPT Major Feasibility Tolerance of 1.0e-6 was used. The red lines in the trajectory figures presented in this chapter as well as in chapter 5 represent applied impulse thrust acceleration directions, but are not to scale. They indicate the directions and relative magnitudes of the applied Sims-Flanagan impulses.

4.4.1 Journey 1: Jupiter Capture to Callisto Rendezvous

The first journey begins at a distance of 226.66 Jovian radii ($R_J$) from Jupiter’s barycenter, the same distance indicated in the Mystic reference trajectory where Jupiter capture occurs. The spacecraft begins with a velocity of 3.37 km/s with respect to Jupiter. Three flybys of Callisto are performed before the spacecraft rendezvous with the satellite. The terminal rendezvous condition was chosen because EMTG_tour is not capable of modeling the complex distant retrograde orbit (DRO) capture maneuver used by Mystic to allow the spacecraft to enter into the Callisto science orbit. The propellant consumed to perform this capture maneuver (as reported by Whiffen⁵) is subtracted from the spacecraft’s final mass at the end of the first journey.
The decision variable bounds used by the inner loop optimizer are summarized in Table 4.5.

<table>
<thead>
<tr>
<th>Decision Variable</th>
<th>Bounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_{wait}$</td>
<td>$[0, 365.25]$ days</td>
</tr>
<tr>
<td>$u, v$</td>
<td>$[0, 2\pi]$</td>
</tr>
<tr>
<td>$\Delta V_{launch}$</td>
<td>$[2.5, 6.0]$ km/s</td>
</tr>
<tr>
<td>$\alpha, \beta$</td>
<td>$[0, 2\pi]$ and $[-\pi/2, \pi/2]$</td>
</tr>
<tr>
<td>$V_{\infty-in}$</td>
<td>$[-10, 10]$ km/s (for each component)</td>
</tr>
<tr>
<td>Journey flight time ($t_{phase}$)</td>
<td>$[100, 400]$ days</td>
</tr>
<tr>
<td>$V_{\infty-out}$</td>
<td>$[-10, 10]$ km/s (for each component)</td>
</tr>
<tr>
<td>$m_f$</td>
<td>$[24000, 30800]$ kg</td>
</tr>
<tr>
<td>$N \times {\Delta V_x, \Delta V_y, \Delta V_z}$</td>
<td>$[-3.0, 3.0]$ km/s (for each component)</td>
</tr>
</tbody>
</table>

Table 4.5: Decision variable bounds used for journey 1: Jupiter capture to Callisto rendezvous.
Figure 4.9: Jupiter capture phase starting at $226.66 \ R_J$. The spacecraft performs three flybys of Callisto before rendezvous occurs.
### 4.4.2 Journey 2: Callisto Escape to Ganymede Rendezvous

Allowing for a 120 day science orbit of Callisto, the earliest possible departure date for the second journey was set to 62863 MJD (December 28th 2030). The transfer from Callisto to Ganymede begins at the point of escape from Callisto since, as mentioned in the previous section, EMTG\_tour is not capable of simulating the complex escape maneuver simulated in Mystic. The amount of propellant reportedly used for the Callisto DRO capture, combined with the propellant consumed for the Callisto escape, is subtracted from the final mass of the spacecraft after the Callisto rendezvous and used as the initial mass for the beginning of journey 2. In addition to this, the optimizer was allowed to select from a small initial velocity envelope to simulate conditions immediately after Callisto escape. For this transfer, the sequence used incorporated a flyby of Callisto, followed by one of Ganymede before the rendezvous with Ganymede was achieved (Escape-C-G-rendezvous). The decision variable bounds used by the inner loop optimizer for journey 2 are summarized in Table 4.6.

<table>
<thead>
<tr>
<th>Decision Variable</th>
<th>Bounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_{\text{wait}}$</td>
<td>[0,200.00] days</td>
</tr>
<tr>
<td>$u,v$</td>
<td>$[0,2\pi]$</td>
</tr>
<tr>
<td>$\Delta V_{\text{launch}}$</td>
<td>[0.001,0.1] km/s</td>
</tr>
<tr>
<td>$V_{\infty-\text{in}}$</td>
<td>[-10,10] km/s (for each component)</td>
</tr>
<tr>
<td>Journey flight time ($t_{\text{phase}}$)</td>
<td>[100,200] days</td>
</tr>
<tr>
<td>$V_{\infty-\text{out}}$</td>
<td>[-10,10] km/s (for each component)</td>
</tr>
<tr>
<td>$m_f$</td>
<td>[24000,29345] kg</td>
</tr>
<tr>
<td>$N \times {\Delta V_x, \Delta V_y, \Delta V_z}$</td>
<td>[-3.0,3.0] km/s (for each component)</td>
</tr>
</tbody>
</table>

Table 4.6: Decision variable bounds used for journey 2: Callisto departure to Ganymede rendezvous.
Figure 4.10: Journey from Callisto escape to Ganymede. The spacecraft performs one flyby of Callisto and one flyby of Ganymede before rendezvous.
4.4.3 Journey 3: Ganymede Escape to Europa Rendezvous

Again allowing for a 120 day science orbit of Ganymede, the earliest possible departure date for the third journey was set to 63365 MJD (May 13th 2032). The same mass deduction was used for this journey; that is to say the mass used in the reference trajectory for the Ganymede DRO capture and subsequent escape maneuvers was subtracted from the final mass of the spacecraft calculated by EMTG_tour at the end of journey 2. The sequence used for this transfer was escape-G-E-rendezvous.

\[
\begin{array}{|c|c|}
\hline
\text{Decision Variable} & \text{Bounds} \\
\hline
\var{t_{\text{wait}}} & [0, 200.00] \text{ days} \\
\var{u, v} & [0, 2\pi] \\
\var{\Delta V_{\text{launch}}} & [0.001, 0.1] \text{ km/s} \\
\var{V_{\infty-in}} & [-10, 10] \text{ km/s (for each component)} \\
\text{Journey flight time (t_{phase})} & [100, 300] \text{ days} \\
\var{V_{\infty-out}} & [-10, 10] \text{ km/s (for each component)} \\
\var{m_f} & [24000, 27082.7] \text{ kg} \\
\var{N \times \{\Delta V_x, \Delta V_y, \Delta V_z\}} & [-3.0, 3.0] \text{ km/s (for each component)} \\
\hline
\end{array}
\]

Table 4.7: Decision variable bounds used for journey 1: Jupiter capture to Callisto rendezvous.
Figure 4.11: Journey from Ganymede escape to Europa. The spacecraft performs one flyby of Ganymede and one flyby of Europa before rendezvous.
4.5 Comparison of EMTG_tour and Reference Trajectories

The most obvious difference between the reference trajectory generated using Mystic, and the one generated by EMTG_tour is the accuracy of the trajectory model. The trajectory generated with EMTG_tour is meant to be a preliminary feasibility study of the JIMO mission. It was created by one person over the course of a few days, whereas the high-fidelity version generated using Mystic took many more man hours to design, and far longer to converge to a solution. While the two trajectories are very similar qualitatively, a side-by-side comparison of the two is really not appropriate. As noted in Chapter 3, the Sims-Flanagan transcription does not result in a physically flyable trajectory, however, it does approximate fairly accurately the amount of $\Delta V$ the spacecraft would have to provide in order to fly the real mission. EMTG_tour (as well as the core EMTG program) is not meant to replace a software package like Mystic, the two tools serve different purposes.

Despite the fact that EMTG_tour is meant as a preliminary design tool and Mystic is used to generate fully-integrated flyable trajectories, it is interesting to note that EMTG_tour did come up with somewhat similar solutions as far as the flyby sequences that were used. From the point of Jupiter capture, Mystic used a C-C-C-C-DRO flyby sequence in order to enter into a science orbit around Callisto, followed by a C-C-G-G-DRO and G-E-DRO sequence for the transfer to Ganymede and Europa respectively. The sequence that EMTG found for the Ganymede to Europa transfer, G-E-DRO, was the same as the one used in the reference trajectory designed with Mystic. An attempt was made to find solutions using EMTG that used the same sequences for the first two journeys as well, however, the optimizer was not able to converge for these sequences. The medium-fidelity trajectory does, however, give the mission designer a good starting point in the trajectory design process. For instance, when moving past the preliminary design phase, the trajectory designed by EMTG_tour could be used to help design an initial guess for a higher fidelity program such as Mystic. For example, one might observe that adding an additional Callisto flyby as well as an additional Ganymede flyby will reduce the $\Delta V$ required to get from Callisto to Ganymede.
4.6 Comments on the Convergence Robustness of EMTG_tour

While an experienced mission designer could use EMTG_tour in its current state to help design a moon tour, the trajectory presented in Sections 4.4.1 4.4.2 and 4.4.3 tested the limits of the software and still required some intelligent guidance. The optimizer was not able to converge to the same sequences used by Mystic for several reasons. The primary reason was that the characteristic time scale of the Jovian system is very fast. As the spacecraft flew closer to Jupiter, it became increasingly difficult for EMTG_tour to find feasible solutions. The Ganymede to Europa transfer presented in Section 4.4.3 was one of only five feasible solutions found for one eight hour run of the inner-loop, whereas the transfer shown in Section 4.4.1 was one of over twenty solutions discovered despite the fact that a greater number of flybys was used. This shows that as the time scale gets shorter, and the number of revolutions required increases, EMTG’s ability to discover feasible solutions decreases. By contrast, a heliocentric mission analyzed with the core EMTG program, using a similar number of flybys, will typically result in the discovery of over 100 feasible solutions. Clearly the sensitivity of the problem, in addition to the overall complexity (i.e. NLP dimension) greatly impacts the program’s performance and methods should be explored to help improve the overall robustness of the optimizer. Several methods for doing this are discussed in the next chapter.
CHAPTER 5
EMTG ROBUSTNESS IMPROVEMENTS

5.1 System Jacobian Calculation in SNOPT

All of the low-thrust trajectory optimization performed in this work makes use of the Sims-Flanagan transcription to cast the problem as a nonlinear program. This NLP is then solved using the SNOPT software package. SNOPT, like all NLP solvers, require the user to provide an initial guess for the NLP decision vector. This initial guess is then iteratively improved upon using sequential quadratic programming (SQP) to solve for the optimal decision vector \( x^* \). The SQP method transforms the general nonlinear constraints of the NLP described in Section 3.4.1 into linear constraints by introducing a set of slack variables. Each major iteration of SNOPT generates a decision point \( x_k \) that satisfies the problem’s linear constraints. Then a quadratic programming (QP) subproblem is solved that results in a decision space search direction towards a new iterate \( x_{k+1} \). This process allows for the set of iterates \( \{ x_k \} \) to converge to the point \( x^* \) that satisfies the nonlinear constraints and first order conditions for optimality to within a certain feasibility tolerance.\(^{40} \) The structure of the QP subproblem relies on knowledge of the problem Jacobian, the matrix of the partial derivatives of the objective function and constraint functions with respect to each decision variable. If the problem is simple, the Jacobian can usually be provided in analytical form. It is often the case, however, that the gradients are impossible, or too expensive, to compute in which case SNOPT can estimate their values using finite differencing. It may also be the case that the user may know some, but not all, Jacobian entries. In this situation, the user may specify analytical expressions or values for those entries which are known and SNOPT will estimate the remaining unknown entries with finite differencing.
While SNOPT’s finite differencing routine works very well for many problems, it is always preferable for the user to specify analytical expressions for all gradients. For the specific NLP defined by the Sims-Flanagan transcription, Sims et al. found that analytical derivatives were essential for the rapid and robust convergence of the software package MALTO but provided no detail regarding the nature of these derivatives nor how their computation was carried out. This claim has inspired an effort to explore the Jacobian structure of the Sims-Flanagan transcription in an effort to determine analytical expressions for its entries and, in doing so, hopefully, improve the convergence abilities of the EMTG tool. The first research thrust in this effort, if successful, actually would have allowed for the complete circumvention of a full analytical derivation of the Sims-Flanagan Jacobian.

5.2 Jacobian Determination Using Automatic Differentiation

The first attempt made to fully specify the problem Jacobian in EMTG was done with automatic differentiation (AD) using the open source package ADOL-C. Automatic differentiation, sometimes referred to as algorithmic differentiation, is a class of techniques used to numerically compute the derivatives of a function in a computer program. This can be accomplished in a variety of ways, but all techniques fundamentally rely on the fact that the chain rule can be successively applied to functions in a computer program in order to obtain derivatives of arbitrary order. In addition to this, although the derivatives are calculated numerically and no analytical expressions are produced, the derivatives are computed to machine precision. An example of AD being used to collect the full derivative information for the function $f(x_1, x_2) = x_1 x_2 + \ln(x_1)$ is shown in Table 5.1. Although analytical expressions appear in the table, it should be noted that, in practice, all derivative information is stored numerically without the creation of analytical expressions.
<table>
<thead>
<tr>
<th>Original Line of Code</th>
<th>Accumulated Derivative Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_1 = x_1$</td>
<td>$f_1' = 1$</td>
</tr>
<tr>
<td>$f_2 = x_2$</td>
<td>$f_2' = 1$</td>
</tr>
<tr>
<td>$f_3 = \ln(f_1)$</td>
<td>$f_3' = \frac{1}{f_1} \cdot f_1' = 1/x_1$</td>
</tr>
<tr>
<td>$f_4 = f_1 * f_2$</td>
<td>$f_4' = f_1'f_2 + f_1f_2' = f_2 + f_1$</td>
</tr>
<tr>
<td>$f_5 = f_3 + f_4$</td>
<td>$x_1 + x_2 + 1/x_1$</td>
</tr>
</tbody>
</table>

Table 5.1: Example of automatic differentiation using forward accumulation.

Automatic differentiation is an ideal method for determining the problem Jacobian sparsity pattern and dense entries as dense/sparse entries are identified simply by program execution. In order to test this method, a version of EMTG was built to accommodate the use of ADOL-C in forward tapeless mode. ADOL-C uses overloading of C/C++ operators to implement AD. Full determination of the problem Jacobian using AD was accomplished, however, integration of ADOL-C into the EMTG source code increased the program execution time to the point where it was deemed unusable. Using AD would still be a favorable way to calculate the problem Jacobian, if a package were to be developed which allowed for a faster execution time or perhaps if parallelization were explored. This is because a software developer could easily add additional problem constraints or otherwise make changes to the inner-loop code and, provided the ADOL-C infrastructure is maintained, not have to worry about deriving additional analytical Jacobian entries.

5.3 Sims-Flanagan Transcription Jacobian Structure

The nonlinear program resulting from the Sims-Flanagan transcription is characterized by the decision vector described in Table 3.2. This decision vector combined with the constraints described in Chapter 3 results in the Jacobian sparsity pattern shown in Figure 5.1.
Figure 5.1: Sims-Flanagan Jacobian sparsity pattern for a single-phase planet to planet journey.

The sparsity pattern shown in Figure 5.1 describes a single-phase journey with $N = 40$ time steps, terminating with a rendezvous. This example problem has 50 nonlinear constraints (including the objective function) and 126 parameters. The Jacobian has 2195 dense entries. One important observation of the Sims-Flanagan problem Jacobian is that the partial derivatives of the $\Delta V$ constraint at the $i^{th}$ time step with respect to the control parameters in previous time steps are dense in general. This is because the $\Delta V_{\text{max}}$ quantity in Equation (3.3) for the $i^{th}$ time step is dependent on the amount of mass used in the preceding time steps (controlled by the $\Delta V$ parameters). The partial derivative of a $\Delta V$ constraint $C_{\Delta V_i}$ with respect to a $\Delta V$ parameter from a previous $p^{th}$ time step is determined via the chain rule:

$$\frac{\partial C_{\Delta V_i}}{\partial \Delta V_{jp}} = \frac{\partial C_{\Delta V_i}}{\partial m_i} \cdot \frac{\partial m_i}{\partial \Delta V_{jp}} \cdot \frac{\partial \Delta V_{j_{i-1}}}{\partial m_{i-1}} \cdot \ldots \cdot \frac{\partial m_{p+1}}{\partial \Delta V_{jp}} \quad (5.1)$$

$$j = x, y \text{ or } z \text{ and } 1 \leq p < i \leq N$$

This complex chain of multiplications makes it extremely challenging to obtain analytical expressions for these Jacobian entries. If these entries could be made sparse, through the decoupling of the parameters in adjacent time steps, then the task of deriving analytical partial derivatives would become less daunting. Two strategies which accomplish this are discussed in the following sections.
5.4 Mass-Parametrized Sims-Flanagan: A Novel Low-Thrust Transcription

It is clear that the original Sims-Flanagan transcription problem Jacobian is not very sparse. The more phases that are incorporated into the trajectory, the more sparse it becomes relative to the size of the Jacobian, however, modifications can be made to the transcription, which result in a drastic increase in the sparsity of the Jacobian. One novel way of doing so is to parametrize the mass at the center of each phase segment (i.e. cast the spacecraft’s mass as a decision parameter at the center of every time step). Doing so eliminates the need to propagate the spacecraft’s mass via Equation (3.4), as well as the need for the mass match point constraint in Equation (3.2), as the mass at the match point can be represented by one decision variable which is equal for both directions of propagation. A new complication that arises out of including the mass in the decision vector is that the optimizer has no way of ensuring that the mass at the center of the subsequent time step does not exceed the mass at the center of the current time step (i.e. there is no guarantee that the spacecraft cannot gain mass). To ensure that the mass chosen at a subsequent time step is not in violation of the conservation of mass, a new family of nonlinear constraints is introduced which will hereafter be referred to as the “mass constraints”. This family of constraints is shown in Equation (5.2):

\[ C_{m_i} = m_i - (m_{i-1} - m_{used_{i-1}}) = 0 \quad i = 1, ..., N \] (5.2)

where \( m_{used_i} \) is defined as

\[ m_{used_i} = m_i \left[ 1 - e\left(\frac{-\Delta v_i}{g I_{sp}}\right) \right] = m_i \left[ 1 - e\left(\frac{-\frac{\Delta v_i^2 + v_{y_i}^2 + v_{z_i}^2}{g I_{sp}}}{g I_{sp}}\right) \right] \] (5.3)

In Equation (5.2) \( m_i \) is the mass of the spacecraft at the center of the current time step, \( m_{i-1} \) is the mass of the spacecraft at the center of the previous time step and \( m_{used_{i-1}} \) is the mount of mass that was used as a result of applying control in the previous time step. The initial spacecraft mass \( m_0 \) is not included in the decision vector, as was the case with the original Sims-Flanagan transcription. It is defined by the user prior to running SNOPT. Essentially, this constraint enforces the necessary truth that the amount of
mass remaining after applying thrust is equal to the difference of the amount of mass you had before the burn and the amount of mass you used during the burn. It will be shown in the next two sections that, while this transcription increases the problem dimension, it also drastically increases the sparsity of the Jacobian, which results in a reduction in the number of computations that SNOPT must perform.

5.4.1 Mass-Parametrized Jacobian

Using the spacecraft’s mass as an explicit decision variable means that the current mass of the spacecraft \( m_i \) is no longer determined via Equation (3.4). Since the mass is now a parameter, its partial derivative with respect to other parameters is zero, i.e.

\[
\frac{\partial m_i}{\partial \Delta V_{jp}} = 0
\]  

(5.4)

This removes the sensitivity of the \( \Delta V \) constraints to changes in the applied control during previous time steps by removing the link between them, the propagated spacecraft mass (i.e. the NLP parameters have successfully been decoupled).
5.4.2 Analytical Expressions for the Mass-Parametrized Jacobian

This alternate form of the Sims-Flanagan transcription not only increases the sparsity of the Jacobian but also allows all of the partial derivatives of the $\Delta V$ magnitude and mass constraints to be expressed as simple analytical expressions. The partial derivatives of the $\Delta V$ control constraints described
in Equation (3.3) are non-zero with respect to the control parameters, the spacecraft mass at the center of the current time step and the phase flight time (i.e. each control constraint has five dense Jacobian entries):

\[
\frac{\partial C_{\Delta V_x}}{\partial \Delta V_{x_i}} = \frac{\Delta V_{x_i}}{\sqrt{\Delta V_{x_i}^2 + \Delta V_{y_i}^2 + \Delta V_{z_i}^2}}
\]
\[
\frac{\partial C_{\Delta V_y}}{\partial \Delta V_{y_i}} = \frac{\Delta V_{y_i}}{\sqrt{\Delta V_{x_i}^2 + \Delta V_{y_i}^2 + \Delta V_{z_i}^2}}
\]
\[
\frac{\partial C_{\Delta V_z}}{\partial \Delta V_{z_i}} = \frac{\Delta V_{z_i}}{\sqrt{\Delta V_{x_i}^2 + \Delta V_{y_i}^2 + \Delta V_{z_i}^2}}
\]

(5.5)

\[
\frac{\partial C_{\Delta V_i}}{\partial t_{phase}} = -\frac{D T_{\text{max}}}{m_i N}
\]

(5.6)

\[
\frac{\partial C_{\Delta V_i}}{\partial m_i} = \frac{D T_{\text{max}}}{m_i^2 N}
\]

(5.7)

The partial derivatives of the mass constraints described in Equation (5.2) are non-zero with respect to the mass of the spacecraft in the center of the current time step, the mass at the center of the previous time step, and the three control parameters for the current time step \(\{\Delta V_x, \Delta V_y, \Delta V_z\}\):

\[
\frac{\partial C_{m_i}}{\partial \Delta V_{x_i}} = \frac{m_{i-1} \Delta V_{x_i}}{g \text{ Isp} \Delta V_i} e^{\left(-\frac{\Delta V_i}{g \text{ Isp}}\right)}
\]
\[
\frac{\partial C_{m_i}}{\partial \Delta V_{y_i}} = \frac{m_{i-1} \Delta V_{y_i}}{g \text{ Isp} \Delta V_i} e^{\left(-\frac{\Delta V_i}{g \text{ Isp}}\right)}
\]
\[
\frac{\partial C_{m_i}}{\partial \Delta V_{z_i}} = \frac{m_{i-1} \Delta V_{z_i}}{g \text{ Isp} \Delta V_i} e^{\left(-\frac{\Delta V_i}{g \text{ Isp}}\right)}
\]

(5.8)

\[
\frac{\partial C_{m_i}}{\partial m_{i-1}} = -e^{\left(-\frac{\Delta V_i}{g \text{ Isp}}\right)}
\]

(5.9)

\[
\frac{\partial C_{m_i}}{\partial m_i} = 1
\]

(5.10)

The above equations describe analytical expressions for the Jacobian entries for all of the \(N\) \(\Delta V\) magnitude and mass constraints. Analytical entries for the flyby constraints are derived in Section 5.5.2 and a method for determining the match point constraint derivatives is outlined in Section 6.2.2.
5.4.3 Results using Mass-Parametrization

In order to test whether or not mass parametrization is a viable low-thrust transcription method, two problems were solved. The first is a single-phase Callisto rendezvous starting at 1000 \( R_J \). The decision variable bounds set a priori as well as any additional user-set options are summarized in Tables 5.2 and 5.3. Note that the mass of the spacecraft at the \( i \)th time step is included in the decision vector for the mass parametrized transcription.

<table>
<thead>
<tr>
<th>Decision Variable</th>
<th>Bounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>( t_{\text{wait}} )</td>
<td>([0,365.25]) days</td>
</tr>
<tr>
<td>( u, v )</td>
<td>([0,2\pi])</td>
</tr>
<tr>
<td>( \Delta V_{\text{launch}} )</td>
<td>([3.7] ) km/s</td>
</tr>
<tr>
<td>( \alpha, \beta )</td>
<td>([0,2\pi]) and ([-\pi/2,\pi/2])</td>
</tr>
<tr>
<td>( V_{\infty-\text{in}} )</td>
<td>N/A (no flybys)</td>
</tr>
<tr>
<td>( \text{Journey flight time} ) (( t_{\text{phase}} ))</td>
<td>([100,500]) days</td>
</tr>
<tr>
<td>( V_{\infty-\text{out}} )</td>
<td>N/A (no flybys)</td>
</tr>
<tr>
<td>( m_f )</td>
<td>([24000,30800]) kg</td>
</tr>
<tr>
<td>( N \times {\Delta V_x, \Delta V_y, \Delta V_z} )</td>
<td>([-3.0,3.0]) km/s (for each component)</td>
</tr>
<tr>
<td>( m_i )</td>
<td>([24000,30800]) kg</td>
</tr>
</tbody>
</table>

Table 5.2: Decision variable bounds used for single-phase Callisto rendezvous starting at 1000 \( R_J \).

<table>
<thead>
<tr>
<th>User Option</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inner loop run time</td>
<td>8 hours</td>
</tr>
<tr>
<td>Number of Sims-Flanagan time steps (( N ))</td>
<td>40</td>
</tr>
<tr>
<td>Earliest mission start date</td>
<td>62503 MJD (Jan. 1st 2030)</td>
</tr>
<tr>
<td>Maximum thrust ( T_{\text{max}} )</td>
<td>13 N</td>
</tr>
<tr>
<td>Thruster duty cycle ( D )</td>
<td>0.98 (modelled continuously)</td>
</tr>
<tr>
<td>Thruster Isp</td>
<td>6000 s</td>
</tr>
</tbody>
</table>

Table 5.3: Options set by the user prior to inner-loop optimization.

The optimizer was allowed to run for eight hours. Twelve trials were performed. Six of those trials used SNOPT’s built in \( \text{snJac} \) routine to completely determine the Jacobian’s sparsity pattern and dense entries using finite differencing. The other six used analytical expressions for the \( \Delta V \) magnitude and mass constraints provided to the optimizer as well as the complete sparsity pattern. For the latter cases, SNOPT determined all of the dense Jacobian entries, that were not provided analytically, using finite differencing (i.e. the
match point constraint entries, as well as the flight time constraint entries).
The results of these trials are shown in Table 5.4.

<table>
<thead>
<tr>
<th>$m_f$ kg</th>
<th>No. Feasible Points Found</th>
<th>$m_f$ kg</th>
<th>No. Feasible Points Found</th>
</tr>
</thead>
<tbody>
<tr>
<td>27930.8</td>
<td>34</td>
<td>28028.4</td>
<td>43</td>
</tr>
<tr>
<td>27588.6</td>
<td>31</td>
<td>28342.3</td>
<td>63</td>
</tr>
<tr>
<td>27696.7</td>
<td>35</td>
<td>28235.6</td>
<td>53</td>
</tr>
<tr>
<td>27626.6</td>
<td>36</td>
<td>27817.8</td>
<td>56</td>
</tr>
<tr>
<td>27959.1</td>
<td>31</td>
<td>28077.9</td>
<td>42</td>
</tr>
<tr>
<td>27834.0</td>
<td>34</td>
<td>27727.7</td>
<td>39</td>
</tr>
</tbody>
</table>

Table 5.4: Results for the single-phase Callisto rendezvous using mass parametrization

The best (i.e. fuel-optimal) solution found is shown in Figures 5.3 and 5.4.
Figure 5.3: Single-phase Callisto rendezvous using mass parametrization. Analytical expressions for all partial derivatives of the $\Delta V$ magnitude and mass constraints were provided to SNOPT.
Figure 5.4: Close up of the Callisto rendezvous.
It is clear from Table 5.4 that providing SNOPT with analytical expressions for even some of the dense entries in the problem Jacobian improves its performance by a measurable amount. Not only were the solutions found with analytical derivatives generally better (i.e. the spacecraft’s final mass was higher) but there were also a greater number feasible points discovered by the optimizer.

The second test problem for mass parametrization is a two-phase Callisto rendezvous starting at 1000 $R_J$, employing one intermediate Callisto flyby. The decision vector bounds and user options used were the same as those for the single-phase example (Tables 5.2 and 5.3). This problem was solved with SNOPT which used its subroutine $snJac$ to calculate completely the problem Jacobian. The results are shown in Figures 5.5 and 5.6. No derivative information was provided; solving this problem shows that mass parametrization is not limited to single-phase missions.
Figure 5.5: Two phase Callisto rendezvous using mass parametrization. The problem Jacobian was completely calculated using finite differencing.
Figure 5.6: Close up of the two phase Callisto rendezvous.
5.5 Sims-Flanagan Transcription using Unit-Vector Control

Parametrization of the spacecraft mass at the center of the Sims-Flanagan time steps considerably increases the sparsity of the problem Jacobian, however, another method exists that achieves a similar sparsity increase but does not suffer from an increased problem dimension. This method, which will hereafter be referred to as unit-vector control or “throttle” control is the specific parametrization used in MALTO and is a favorite of other researchers in the field.\(^{42}\)

Instead of parametrizing the \(\Delta V\) that the spacecraft applies at each time step (Equation (3.3)), consider instead, the selection of \(3N\) “throttle” parameters \(\{u_x, u_y, u_z\}\) defining thrust direction vectors whose magnitudes are constrained to be no greater than unity:

\[
C_{\text{throttle}} = \left(u_x^2 + u_y^2 + u_z^2\right)^{1/2} \leq 1 \tag{5.11}
\]

The throttle parameters selected by the optimizer are then scaled by the maximum \(\Delta V\) that the spacecraft may apply during the current time step in order to determine the velocity change in each direction:

\[
\Delta V_x = u_x \Delta V_{\text{max}} \\
\Delta V_y = u_y \Delta V_{\text{max}} \\
\Delta V_z = u_z \Delta V_{\text{max}} \tag{5.12}
\]

where \(\Delta V_{\text{max}}\) is the same quantity defined in Equation (3.3). Equation 5.11 now replaces the nonlinear constraints described by Equation (3.3) and increases the Jacobian sparsity significantly.
5.5.1 Unit-Vector Control Jacobian

The sparsity increase achieved with the unit vector control parametrization is due to the restructuring of the \( \Delta V \) magnitude (throttle) constraints. Specifically, the quantity \( \Delta V_{\text{max}} \) no longer appears in the throttle constraints, resulting in the spacecraft mass not being included in these constraints either. The match point derivatives are unaffected, however, and are still dense.

![Figure 5.7: Unit-vector control Sims-Flanagan Jacobian sparsity pattern for a single-phase journey.](image)

The Jacobian shown in Figure 5.7 is for a case with \( N = 40 \) time steps, starting at a free point in space and terminating with a rendezvous. This example problem has 50 nonlinear constraints (including the objective function) and 128 parameters. The Jacobian has 1012 dense entries.

5.5.2 Analytical Expressions for the Unit-Vector Control Jacobian

As was the case for the mass parametrization method, the dense entries for most of the constraints in the unit-vector control Jacobian have relatively straightforward analytical expressions. The partial derivatives of the throttle constraints with respect to the control parameters in the current time step are as follows:
\[
\frac{\partial C_{\text{throttle}}}{\partial u_x} = \frac{u_x}{\sqrt{u_x^2 + u_y^2 + u_z^2}} \\
\frac{\partial C_{\text{throttle}}}{\partial u_y} = \frac{u_y}{\sqrt{u_x^2 + u_y^2 + u_z^2}} \\
\frac{\partial C_{\text{throttle}}}{\partial u_z} = \frac{u_z}{\sqrt{u_x^2 + u_y^2 + u_z^2}}
\] (5.13)

The derivatives of the flyby constraints in Equations (3.5) and (3.6) were not treated in Section 5.4 but are identical for either transcription. The partial derivatives for Equation (3.5) are

\[
\frac{\partial C_{V_{\infty}}}{\partial V_{\infty-out_x}} = \frac{V_{\infty-out_x}}{\sqrt{V_{\infty-out_x}^2 + V_{\infty-out_y}^2 + V_{\infty-out_z}^2}} \\
\frac{\partial C_{V_{\infty}}}{\partial V_{\infty-out_y}} = \frac{V_{\infty-out_y}}{\sqrt{V_{\infty-out_x}^2 + V_{\infty-out_y}^2 + V_{\infty-out_z}^2}} \\
\frac{\partial C_{V_{\infty}}}{\partial V_{\infty-out_z}} = \frac{V_{\infty-out_z}}{\sqrt{V_{\infty-out_x}^2 + V_{\infty-out_y}^2 + V_{\infty-out_z}^2}} \\
\frac{\partial C_{V_{\infty}}}{\partial V_{\infty-in_x}} = -\frac{V_{\infty-in_x}}{\sqrt{V_{\infty-in_x}^2 + V_{\infty-in_y}^2 + V_{\infty-in_z}^2}} \\
\frac{\partial C_{V_{\infty}}}{\partial V_{\infty-in_y}} = -\frac{V_{\infty-in_y}}{\sqrt{V_{\infty-in_x}^2 + V_{\infty-in_y}^2 + V_{\infty-in_z}^2}} \\
\frac{\partial C_{V_{\infty}}}{\partial V_{\infty-in_z}} = -\frac{V_{\infty-in_z}}{\sqrt{V_{\infty-in_x}^2 + V_{\infty-in_y}^2 + V_{\infty-in_z}^2}}
\] (5.15)

Expressions for the partial derivatives of Equation (3.6) may be found in the Appendix. The partial derivatives for the flight time constraint given in Equation (3.8) are given by Equation (5.16):
\[
\frac{\partial C_{\text{tof}}}{\partial t_{\text{flight}}} = \begin{cases} 
-1, & \text{if } t_{\text{flight}} < t_{\text{min}}, \\
1, & \text{if } t_{\text{flight}} > t_{\text{max}}, \\
0, & \text{otherwise}.
\end{cases}
\] (5.16)
CHAPTER 6

CONCLUSIONS AND FUTURE WORK

6.1 Conclusions and Summary of Contributions

This work details two main contributions to automated low-thrust trajectory planning. The first is the automated design of optimal multiple flyby low-thrust planetary moon tours. This was done by treating the trajectory optimization process as a hybrid optimal control problem which, for this work, utilized an integer genetic algorithm to select the flyby sequence in the outer-loop and monotonic basic hopping to select initial decision variable values for the nonlinear programming solver SNOPT as the inner-loop solver. This complements recent work by Englander\textsuperscript{2–4} who investigated heliocentric trajectories using a similar framework. The preliminary trajectory design phase of the JIMO mission was revisited and it was shown that solutions resembling those produced by the high-fidelity tool Mystic could be obtained. This process also highlighted the fact that trajectory optimization in the context of a dynamical system with a fast characteristic time scale is much more challenging than for a system with a slow time scale. This is mainly due to the relatively higher number of revolutions and flybys required and the overall sensitivity of the system. This fact, as well as claims from other researchers in the field, prompted a comprehensive study of the Sims-Flanagan transcription Jacobian, which is the second main contribution of this work. The first attempt made to fully determine the sparsity pattern and dense entries of the system Jacobian was with algorithmic differentiation. A version of EMTG was successfully built that used algorithmic differentiation, however, its execution time was found to be too slow to be of any practical use. Next, a novel variant of the transcription was also proposed and analyzed, which results in a drastic increase in sparsity of the Jacobian; however, it also increases the problem dimension. The other advantage of
this mass parametrized transcription is that many of the partial derivatives of the NLP constraints are simple analytical expressions. When analytical expressions for a portion of the problem Jacobian were provided to SNOPT it was shown that the optimizer not only discovered a greater number of feasible solutions but that those solutions had, on average, a higher cost function value (final mass). Finally, the sparsity pattern of the unit vector control variant of the Sims-Flanagan transcription was analyzed and analytical expressions for the majority of its Jacobian entries were derived. Although analytical derivatives for this transcription are utilized in the popular design tool MALTO, the details of these derivatives and how they are computed remains absent from the literature.

6.2 Future Work

6.2.1 Complete Analytical Derivation of the Sims-Flanagan Jacobian

One of the primary contributions of this work was an analysis of the problem Jacobian for the Sims-Flanagan transcription as well as introducing some methods for improving the robustness of a solver employing this trajectory model. While expressions for the majority of the Jacobian entries have been provided, some were not treated in this document. For instance, the derivatives for the match point mass constraint were not discussed at all. An immediate research goal will be to build the capability to calculate the complete Jacobian into EMTG. Providing this information to the optimizer would completely eliminate the need to employ finite differencing in the calculation of the system Jacobian. One of the most challenging aspects will be to provide the partial derivatives of the match point constraints in Equation (3.2). Some investigation into how to accomplish this has already been done and a method that could be used to obtain these Jacobian entries is provided in Section 6.2.2.
6.2.2 Sims-Flanagan Match Point Constraint Derivatives

The majority of the dense Jacobian entries for mass parametrized Sims-
Flanagan or Sims-Flanagan using throttle vector control are tractible ana-
lytical expressions. The partial derivatives of the match point constraints
with respect to the control parameters are far more complicated to calcu-
late. It is clear from Figures 5.1, 5.2 and 5.7 that the Jacobian entries for
the match point constraints are, in general, dense with respect to all of the
decision variables (except the masses for the mass parametrized transcription). The majority of these dense entries are the partial derivatives of the
match point constraints with respect to the control parameters. These con-
trol parameters introduce a discontinuity in the spacecraft’s velocity vector
at every impulse. A natural strategy for determining the sensitivity of the
match point constraints to a downstream control parameter is to use succes-
sive multiplications of the state transition matrix (STM) (also known as the
fundamental matrix). The state transition matrix $\Phi(t, t_0)$ provides a means
of determining how a deviation of the state vector from the reference trajec-
tory, at an initial time $t_0$, affects the value of the state vector at some later
time $t$. This is easily accomplished using the following equation:

$$x(t) = \Phi(t, t_0)x(t_0) \quad (6.1)$$

In this way, the STM can be used to propagate a spacecraft’s state vector
or propagate an initially perturbed state vector. Specifically, if at an initial
time $t_0$, contemporaneous perturbations of a spacecraft’s position and ve-
locity vectors $\delta r$ and $\delta v$ are known, then the corresponding deviations from
the reference trajectory at a later time $t$ can be calculated using an STM
multiplication:

$$\begin{bmatrix} \delta r \\ \delta v \end{bmatrix} = \begin{bmatrix} \frac{\partial r}{\partial r_0} & \frac{\partial r}{\partial v_0} \\ \frac{\partial v}{\partial r_0} & \frac{\partial v}{\partial v_0} \end{bmatrix}_{ref} \begin{bmatrix} \delta r_0 \\ \delta v_0 \end{bmatrix} = \Phi(t, t_0) \begin{bmatrix} \delta r_0 \\ \delta v_0 \end{bmatrix} \quad (6.2)$$
The subscript \textit{ref} in Equation (6.2) denotes the fact that the entries in the STM are to be evaluated on the reference trajectory while holding the transfer time constant. A method for computing the entries of the STM for a general conic orbit is provided in detail by Shepperd. The calculation of the Jacobian entries of the $\Delta V$ constraints with respect to the throttle parameters \{\(u_x, u_y, u_z\}\} in a particular time step is a slightly more complicated problem in that not all of the $\Delta V$ perturbations are immediately adjacent to the match point (in fact most are not). The bounded impulses of the Sims-Flanagan transcription are essentially a series of velocity perturbations that must be propagated between control points. This means that the calculation of the match point Jacobian entries can be achieved with successive applications of Equation (6.1) (i.e. the state transition matrix has a transitive property). For example, the partial derivative of the \(x\)-component match point constraint with respect to the \(y\)-component of the \(i^{th}\) $\Delta V$ control point in the forward half of a Sims-Flanagan phase with \(N\) time steps would be calculated as follows:

$$\frac{\partial C_{mpx}}{\partial u_y} = \Phi(t_{N/2}, t_{N/2-1})\Phi(t_{N/2-1}, t_{N/2-2}) \ldots \Phi(t_2, t_1)\Phi(t_1, t_0) \begin{bmatrix} 0 \\ \Delta V_{\max} \end{bmatrix}$$  (6.3)

The derivative calculation method in Equation (6.3) can be applied to phases with any number of time steps due to the transitivity of the STM’s. The only possible issue with this method is that a long chain of STM’s must be multiplied together, which could potentially accumulate numerical error. Specifically, as STM’s are multiplied together, the resultant state transition matrix may cease to be symplectic, and is therefore, by definition, no longer a state transition matrix. A symplectic matrix \(\Phi\) is one such that the following condition holds

$$\Phi^T \cdot J \cdot \Phi = J = \begin{bmatrix} 0 & \mathbb{I} \\ -\mathbb{I} & 0 \end{bmatrix}$$  (6.4)
A useful numerical accuracy test would be to check whether or not the STM multiplication chain in Equation (6.3) satisfies Equation (6.4) to within an acceptable error tolerance. Preliminary experiments indicate that at least for as many as \( N = 100 \) time steps, the numerical error remains close to machine precision and does not seem to appreciably impact the accuracy of the partial derivatives. While match point derivative calculation via STM’s has not yet been fully implemented in EMTG, preliminary testing using SNOPT’s derivative check option indicates that this method is capable of providing accurate values for these Jacobian entries.

6.2.3 Further Improvements to EMTG Convergence Robustness

Obtaining solutions using EMTG in the Jovian system proved to be somewhat challenging for the optimizer. This was especially true for the complex moon transfers presented in Chapter 4. There are several fundamental differences between multiple flyby trajectory optimization in a dynamical environment like the Jovian satellite system compared with the heliocentric system. The primary difference is that the characteristic time scale of the Jovian satellites is very fast compared to the planets. For example Io’s orbital period is 1.77 days, which is several orders of magnitude smaller than the orbital periods of most of the planets. As a result of this, trajectories in a fast system must use a greater number of revolutions about the central body as well as a potentially higher number of flyby maneuvers. This means that, in order to sufficiently approximate a low-thrust trajectory with the Sims-Flanagan transcription a larger number of time steps must be used, meaning that the resulting NLP that SNOPT must solve starts to suffer from the curse of dimensionality. A method needs to be developed to handle these many-revolution trajectories robustly that can then be integrated into EMTG. These types of trajectory problems have been studied previously and it is possible that some of these techniques could be incorporated into a future build of EMTG.\textsuperscript{44–46}
6.2.4 Radiation Considerations in the Jovian System and Other Trajectory Constraints

While solver robustness and general performance are the primary foci of this work and any immediate future research thrusts, there are other ways to make the NLP model used in this work more realistic and, therefore, useful. For example, in its current form, it does not take certain environmental factors into account, for example the hazardous radiation levels present in the Jovian system. While spacecraft hardware is generally radiation hardened, it is not impervious to extreme levels like those found in the vicinity Europa and Io. Ideally, the amount of radiation that a spacecraft receives while flying its mission should be minimized and the identification of feasible solutions that keep radiation exposure under a minimum threshold is important. This could be achieved using a trajectory constraint that integrates the spacecraft’s radiation flux over the course of the mission. Other constraints might include a condition that does not allow a spacecraft to thrust within a certain temporal window before and after periapse passage for planetary preservation purposes as was implemented for the design of the JIMO reference trajectory.

6.2.5 SPICE Integration and Reference Frame Transformations

Perhaps one of the most important future research goals will be to integrate the work presented in this thesis (which is currently referred to as EMTG_tour) into the core heliocentric EMTG program. One challenging aspect of this will be how to properly transition from interplanetary flight to the planet-centered moon tour phase of the mission. The coordinate transformations required will ideally be handled by the SPICE ephemeris tool. Not only will this allow for the optimization of a trajectory from Earth departure through planetary system capture, it will also mark a major increase in EMTG’s overall utility as a mission design tool.
These are the partial derivatives for the flyby altitude constraints in Equation (3.6) with respect to the components of the spacecraft’s hyperbolic approach and departure vector components. Note that $V_{\infty-in}$ and $V_{\infty-out}$ have been shortened to $V_i$ and $V_o$ respectively, for presentation here only.

$$\frac{\partial C_{\text{flyby}}}{\partial V_{ix}} = -\mu \cos \left( \frac{\alpha}{2} \right) \left( \frac{V_{ox} V_{ix}^2 - V_{ix} V_{oy} V_{iy} + V_{ox} V_{iz}^2 - V_{ix} V_{oz} V_{iz}}{r_{\text{planet}}(\alpha - 1)} \right) \left[ 1 - \frac{(V_{ix} V_{ox} + V_{iy} V_{oy} + V_{iz} V_{oz})^2}{\gamma \beta} \right]^{1/2} \gamma^{3/2} \beta^{3/2}$$

$$\frac{\partial C_{\text{flyby}}}{\partial V_{iy}} = -\mu \cos \left( \frac{\alpha}{2} \right) \left( \frac{V_{oy} V_{ix}^2 - V_{ix} V_{oz} V_{iz} + V_{oy} V_{iz}^2 - V_{iy} V_{ox} V_{iz}}{r_{\text{planet}}(\alpha - 1)} \right) \left[ 1 - \frac{(V_{ix} V_{ox} + V_{iy} V_{oy} + V_{iz} V_{oz})^2}{\gamma \beta} \right]^{1/2} \gamma^{3/2} \beta^{3/2}$$

$$\frac{\partial C_{\text{flyby}}}{\partial V_{iz}} = -\mu \cos \left( \frac{\alpha}{2} \right) \left( \frac{V_{oz} V_{ix}^2 - V_{ix} V_{ox} V_{iz} + V_{oz} V_{iz}^2 - V_{iz} V_{oy} V_{iy}}{r_{\text{planet}}(\alpha - 1)} \right) \left[ 1 - \frac{(V_{ix} V_{ox} + V_{iy} V_{oy} + V_{iz} V_{oz})^2}{\gamma \beta} \right]^{1/2} \gamma^{3/2} \beta^{3/2}$$

where

$$\alpha = \frac{V_{ix} V_{ox} + V_{iy} V_{oy} + V_{iz} V_{oz}}{\beta^{1/2} \gamma^{1/2}}$$

$$\beta = V_{ox}^2 + V_{oy}^2$$

$$\gamma = V_{ix}^2 + V_{iy}^2$$
\[
\frac{\partial C_{\text{flyby}}}{\partial V_{ox}} = \frac{2V_{ox}\mu}{r_{\text{planet}}\xi^2} - \frac{2V_{ox}\mu}{r_{\text{planet}} \sin \left(\frac{\cos(\epsilon)}{2}\right) \xi^2} - \frac{\mu \cos \left(\frac{\cos(\epsilon)}{2}\right) \left(V_{ix}V_{oy}^2 - V_{iy}V_{ox}V_{oy} + V_{ix}V_{oz}^2 - V_{iz}V_{ox}V_{oz}\right)}{r_{\text{planet}} \left(\epsilon - 1\right) \left[1 - \frac{\phi^2}{\psi^2}\right]^{1/2} \psi^{1/2} \xi^{5/2}}
\]

\[
\frac{\partial C_{\text{flyby}}}{\partial V_{oy}} = \frac{2V_{oy}\mu}{r_{\text{planet}}\xi^2} - \frac{2V_{oy}\mu}{r_{\text{planet}} \sin \left(\frac{\cos(\epsilon)}{2}\right) \xi^2} - \frac{\mu \cos \left(\frac{\cos(\epsilon)}{2}\right) \left(V_{iy}V_{ox}^2 - V_{ix}V_{ox}V_{oy} + V_{iy}V_{oz}^2 - V_{iz}V_{oy}V_{oz}\right)}{r_{\text{planet}} \left(\epsilon - 1\right) \left[1 - \frac{\phi^2}{\psi^2}\right]^{1/2} \psi^{1/2} \xi^{5/2}}
\]

\[
\frac{\partial C_{\text{flyby}}}{\partial V_{oz}} = \frac{2V_{oz}\mu}{r_{\text{planet}}\xi^2} - \frac{2V_{oz}\mu}{r_{\text{planet}} \sin \left(\frac{\cos(\epsilon)}{2}\right) \xi^2} - \frac{\mu \cos \left(\frac{\cos(\epsilon)}{2}\right) \left(V_{iz}V_{ox}^2 - V_{ix}V_{ox}V_{oz} + V_{iz}V_{oy}^2 - V_{iy}V_{oz}V_{oy}\right)}{r_{\text{planet}} \left(\epsilon - 1\right) \left[1 - \frac{\phi^2}{\psi^2}\right]^{1/2} \psi^{1/2} \xi^{5/2}}
\]

where

\[
\epsilon = \frac{\phi}{\psi^{1/2} \xi^{1/2}}
\]

\[
\xi = V_{ox}^2 + V_{oy}^2 + V_{oz}^2
\]

\[
\phi = V_{ix}V_{ox} + V_{iy}V_{oy} + V_{iz}V_{oz}
\]

\[
\psi = V_{ix}^2 + V_{iy}^2 + V_{iz}^2
\]
REFERENCES


