Design of a Continuous-Thrust Solar Polar Mission

Thesis Report

Willem van der Weg
19\textsuperscript{th} of April, 2010

A rare picture of Venus transiting the Sun, taken by the TRACE spacecraft. The last transit of Venus occurred in 2004. The transfer before that occurred in 1882, and was used to calculate the distance between the Earth and the Sun.
Design of a Continuous-Thrust Solar Polar Mission

Willem van der Weg

19th of April, 2010

Astrodynamics & Satellite Systems
Faculty of Aerospace Engineering
Delft University of Technology
Kluyverweg 1, 2629 HS Delft
The Netherlands
“Everything has a natural explanation. The Moon is not a god, but a great rock, and the Sun a hot rock.”

- Anaxagoras, ca. 475 B.C.
In 2009 I was able to begin work on a thesis topic that aptly suited my interests; designing a mission that is able to insert a spacecraft with low-thrust propulsion into a solar polar orbit at 0.4 AU distance. During work on my thesis I looked at the analytical approach to designing low thrust transfers and how they could be used to bring a spacecraft towards the Sun. The work has already been partially successful; it is with some small measure of pride that I can say that other students have already been able to benefit from my computer code here during my last few months as a student in Delft.

I would like to offer my thanks to the various students before me that have provided the necessary foundation of knowledge in their own thesis reports. Also, thanks to my fellow students on the 9th floor, who made slaving away ones hours behind a computer screen a much more pleasant activity. To those not quite finished yet, hang in there! I would also like to thank my supervisor Ron Noomen, who was extremely helpful and always available to talk, offering advice, encouragement, and criticism as necessary throughout the project. Last but not least, my thanks go out to friends, family, and my girlfriend for their support.

Delft, 13 March 2010, Willem van der Weg
ABSTRACT

The Sun is responsible for all life on our planet. It addition to providing the energy necessary for photosynthesis, it also drives our climate and weather. In spite of its importance there are many questions about the Sun that are left unanswered. Understanding the Sun and its natural processes helps us to understand how the Earth responds to the Sun’s variations and may help improve our ability to predict its behavior. Prediction is especially crucial to spaceflight. Spacecraft are becoming increasingly sensitive to the solar environment and future manned missions to the Moon or Mars will rely upon accurate prediction of the solar activity.

A probe in polar orbit around the Sun would be able to provide us with a more complete picture of the Sun by performing measurements over the Sun’s complete surface area. Moreover, it would be able to provide in-situ measurements. Previous reports have tackled this problem by using solar radiation to propel a spacecraft through space (solar sailing). This is a very innovative approach, but does possess certain drawbacks. An alternative is a spacecraft equipped with electric propulsion. This type of propulsion features high exhaust velocities, giving it the capability to attempt these energy-expensive transfers.

To assess whether this type of propulsion would be suitable, a model is created based on an analytical approach. This approach includes Lambert’s solution of low thrust transfer shapes, multiple gravity assists, and Edelbaum’s theory. As such, this thesis is as much a test of the analytical methods as it is of the propulsion system for this mission.

A good solution is sought using an optimization process featuring random, global, and local search methods. The goal is primarily to minimize propellant mass, but also transfer time, while maintaining a realistic transfer that does not violate imposed constraints such as departure and arrival velocity. Minimizing these two parameters will generally lead to reduced costs. The optimization strategy is to use a random search to perform an initial exploration of the search space, a global heuristic Genetic Algorithm to search for an optimum close to the global optimum, and a local Nelder-Mead search to improve upon this located optimum. As expected, the Genetic Algorithm brought the most significant improvement of objective fitness to the table.

After extensive experimentation it was found that the model is useful for certain transfers, but requires further work to give it the means to tackle more intricate problems. Two answers are offered with regards to the problem of a solar polar probe. One answer involves the launch of a 318 kg (initial mass) spacecraft
that performs a swing-by at Venus and inserts a 40 kg spacecraft into science orbit in 1,367 days. The other answer launches a 233 kg (initial mass) spacecraft and uses a swing-by at Jupiter to insert a 40 kg spacecraft into science orbit in 3,341 days. For reference, the solar sailing design launches a 204 kg spacecraft that is inserted into science orbit after 2,219 days of transfer flight. In all three cases the payload mass is assumed to be 5 kg and the science orbit is a circular orbit at 0.4 AU distance from the Sun, which plane is perpendicular to the ecliptic plane. Without further research, one cannot say with absolute certainty whether either of these designs is superior to the solar sailing design. However, a preliminary indication can be made by using a function $J = m_0 t m_p^{-1}$ that combines initial mass $m_0$, payload mass $m_p$, and transfer time $t$ (the function $J$ is minimized i.e. smaller values of $J$ are superior). The electric propulsion design using a swing-by at Venus has a $J$ value of 86,941, the design with a swing-by at Jupiter has $J = 155,690$, and the solar sailing design has a $J$ value of 90,535. The electric propulsion with swing-by at Venus and the solar sailing design score quite similarly in this indication, while the design with swing-by at Jupiter scores significantly worse due to its much longer transfer time.
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preface</td>
<td>vii</td>
</tr>
<tr>
<td>Abstract</td>
<td>ix</td>
</tr>
<tr>
<td>Nomenclature</td>
<td>xv</td>
</tr>
<tr>
<td>Symbols</td>
<td>xv</td>
</tr>
<tr>
<td>Constants</td>
<td>xvi</td>
</tr>
<tr>
<td>Abbreviations</td>
<td>xvi</td>
</tr>
<tr>
<td>Subscripts</td>
<td>xvi</td>
</tr>
<tr>
<td>Notes</td>
<td>xvii</td>
</tr>
<tr>
<td>Introduction</td>
<td>1</td>
</tr>
<tr>
<td>Mission Overview</td>
<td>3</td>
</tr>
<tr>
<td>2.1 Science Objectives</td>
<td>3</td>
</tr>
<tr>
<td>2.2 Mission Requirements</td>
<td>4</td>
</tr>
<tr>
<td>2.3 Design Considerations</td>
<td>5</td>
</tr>
<tr>
<td>2.4 Low-Thrust Spaceflight</td>
<td>5</td>
</tr>
<tr>
<td>2.5 Spacecraft Specifications and Engine Characteristics</td>
<td>8</td>
</tr>
<tr>
<td>Designing Continuous Thrust Transfers</td>
<td>9</td>
</tr>
<tr>
<td>3.1 The Exponential Sinusoid</td>
<td>9</td>
</tr>
<tr>
<td>3.2 Solving Lambert’s Problem for Exponential Sinusoids</td>
<td>12</td>
</tr>
<tr>
<td>3.3 Terminal Velocity</td>
<td>14</td>
</tr>
<tr>
<td>3.4 Time of Flight</td>
<td>14</td>
</tr>
<tr>
<td>3.5 Propellant and Power Considerations</td>
<td>15</td>
</tr>
<tr>
<td>3.6 Gravity Assists</td>
<td>21</td>
</tr>
<tr>
<td>3.7 Orbit Plane Changing</td>
<td>25</td>
</tr>
<tr>
<td>Numerical Integration</td>
<td>41</td>
</tr>
<tr>
<td>4.1 Method Overview</td>
<td>41</td>
</tr>
</tbody>
</table>
4.2 Numerical Verification ........................................................................................................... 47

5 Optimization ............................................................................................................................ 57
5.1 Fitness Function ...................................................................................................................... 57
5.2 Penalty Functions ................................................................................................................... 58
5.3 Employed Optimization Methods .......................................................................................... 61

6 Implementation ....................................................................................................................... 77
6.1 Optimization Objective ........................................................................................................... 77
6.2 Calculation of a Single Transfer Leg .................................................................................... 78
6.3 Penalties .................................................................................................................................. 81
6.4 Time of Flight Curve Aberrations ....................................................................................... 90
6.5 Lunar Gravity Assist ............................................................................................................. 91
6.6 Reaching The Sun ............................................................................................................... 93
6.7 Analysis and Plotting .......................................................................................................... 95
6.8 Program Structure .............................................................................................................. 104

7 Testing .................................................................................................................................... 109
7.1 Terminal Velocity (Sensitivity Analysis) ............................................................................. 109
7.2 Previous Results (Validation) ............................................................................................ 112
7.3 Earth-Mars Transfer (Validation) ..................................................................................... 115
7.4 Optimization Robustness (Tuning) .................................................................................. 121
7.5 Perturbations .................................................................................................................... 128

8 Candidate Designs ................................................................................................................ 145
8.1 Earth-To-Sun Direct Transfer ............................................................................................. 145
8.2 Earth-Venus-Sun ................................................................................................................. 153
8.3 Earth-Venus-Venus-Sun .................................................................................................... 157
8.4 Earth-Moon-Venus-Sun ...................................................................................................... 161
8.5 Earth-Moon-Earth-Venus-Sun ........................................................................................ 165
8.6 Earth-Jupiter-Sun .............................................................................................................. 169
8.7 Best Solutions ................................................................................................................... 173

9 Conclusions & Recommendations ....................................................................................... 183

Bibliography ............................................................................................................................. 191
Appendix A ............................................................................................................................... 195
Appendix B ............................................................................................................................... 197
Appendix C ............................................................................................................................... 201
Appendix D .......................................................................................................................... 209
Appendix E .......................................................................................................................... 213
NOMENCLATURE

SYMBOLS

Greek
\( \alpha \) Asymptotic Deflection Angle
\( \beta \) Thrust Yaw Angle
\( \gamma \) Flight Path Angle
\( \delta \) Latitude
\( \varepsilon \) Reflectivity Coefficient, Energy
\( \eta \) (Thruster) Efficiency
\( \theta \) True Anomaly
\( \lambda \) Longitude
\( \mu \) Gravitational Constant
\( \rho \) Density
\( \tau \) Local Truncation Error
\( \omega \) Argument of Periapsis
\( \Delta \) Delta (change in velocity, energy, etc.)
\( \Phi \) Phase Angle, Solar Flux
\( \Omega \) Longitude of the Ascending Node (equivalently Right Ascension of the Ascending Node)

Roman
\( A \) Area
\( a \) Semi-major Axis, Acceleration
\( B \) Magnetic Induction
\( C \) Capacitance
\( c \) Velocity of Light
\( \mathcal{E} \) Energy
\( E \) Eccentric Anomaly
\( e \) Eccentricity
\( F \) Force
\( g \) Gravitational Acceleration
\( h \) Specific Angular Momentum
\( I \) Impulse, Irradiance
\( i \) Inclination
\( M \) Mean Anomaly
\( m \) Mass
\( n \) Normalized Position Vector
\( P \) Power
\( p \) Semi-latus Rectum, Momentum
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Value/Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \mathcal{P} )</td>
<td>Pressure</td>
<td></td>
</tr>
<tr>
<td>( q )</td>
<td>Electric Charge</td>
<td></td>
</tr>
<tr>
<td>( r )</td>
<td>Radial Distance, Position Vector</td>
<td></td>
</tr>
<tr>
<td>( T )</td>
<td>Thrust</td>
<td></td>
</tr>
<tr>
<td>( t )</td>
<td>Time</td>
<td></td>
</tr>
<tr>
<td>( U )</td>
<td>Potential</td>
<td></td>
</tr>
<tr>
<td>( u )</td>
<td>Argument of Latitude</td>
<td></td>
</tr>
<tr>
<td>( v )</td>
<td>Velocity</td>
<td></td>
</tr>
</tbody>
</table>

## CONSTANTS

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Value/Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{AU} )</td>
<td>Astronomical Unit</td>
<td>149,598,000 km</td>
</tr>
<tr>
<td>( c )</td>
<td>Velocity of Light</td>
<td>299,792,458 m/s</td>
</tr>
<tr>
<td>( g_0 )</td>
<td>Gravitational Acceleration at Sea Level</td>
<td>9.80665 m/s²</td>
</tr>
<tr>
<td>( G )</td>
<td>Universal Gravitational Constant</td>
<td>( 6.67428 \times 10^{-11} ) m³kg⁻¹s⁻²</td>
</tr>
<tr>
<td>( \mu )</td>
<td>Gravitational Constant</td>
<td>( = G \cdot m )</td>
</tr>
</tbody>
</table>

## ABBREVIATIONS

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CME</td>
<td>Coronal Mass Ejection</td>
</tr>
<tr>
<td>COTS</td>
<td>Commercial Off The Shelf</td>
</tr>
<tr>
<td>GA</td>
<td>Genetic Algorithm</td>
</tr>
<tr>
<td>GCD</td>
<td>Gregorian Calendar Date</td>
</tr>
<tr>
<td>GEO</td>
<td>Geostationary Earth Orbit</td>
</tr>
<tr>
<td>GSFC</td>
<td>Goddard Space Flight Center</td>
</tr>
<tr>
<td>GTO</td>
<td>Geostationary Transfer Orbit</td>
</tr>
<tr>
<td>LEO</td>
<td>Low Earth Orbit</td>
</tr>
<tr>
<td>LTP</td>
<td>Low Thrust Propulsion</td>
</tr>
<tr>
<td>MJD</td>
<td>Modified Julian Date</td>
</tr>
<tr>
<td>MJD2000</td>
<td>Modified Julian Date of epoch January 1, 2000, 12h.</td>
</tr>
<tr>
<td>ODE</td>
<td>Ordinary Differential Equation</td>
</tr>
<tr>
<td>SEP</td>
<td>Solar Electric Propulsion</td>
</tr>
<tr>
<td>SPE</td>
<td>Solar Particle Event</td>
</tr>
<tr>
<td>TOF</td>
<td>Time Of Flight</td>
</tr>
</tbody>
</table>

## SUBSCRIPTS

<table>
<thead>
<tr>
<th>Subscript</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( p )</td>
<td>At Periapsis, Power Supply</td>
</tr>
<tr>
<td>( a )</td>
<td>At Apoapsis</td>
</tr>
<tr>
<td>( \text{esc} )</td>
<td>Escape (Escape Velocity)</td>
</tr>
<tr>
<td>( c )</td>
<td>Circular (Velocity)</td>
</tr>
<tr>
<td>( \infty )</td>
<td>Hyperbolic Excess (Velocity)</td>
</tr>
<tr>
<td>( T )</td>
<td>Due to Thrust (Acceleration), Target, Translation</td>
</tr>
<tr>
<td>( \text{max} )</td>
<td>Maximum</td>
</tr>
<tr>
<td>( \text{min} )</td>
<td>Minimum</td>
</tr>
<tr>
<td>( \odot )</td>
<td>Sun</td>
</tr>
<tr>
<td>( \oplus )</td>
<td>Earth</td>
</tr>
<tr>
<td>( D )</td>
<td>Drag</td>
</tr>
</tbody>
</table>
\textbf{atm} Atmosphere
\textbf{sat} Parameter pertaining to the spacecraft
\textbf{sp} Specific (Impulse)
\textbf{sc} Spacecraft
\textbf{sc_p} Perfect Property (for example pertaining to perfect circular velocity)
\textbf{N} Normalized
\textbf{e} Exhaust

\textbf{Solar, Sun} Properties pertaining to the Sun
\textbf{prop} Propellant
\textbf{av} Available
\textbf{SO} Solar Orbit
\textbf{Used} Spent Propellant
\textbf{Panel} Property of a Solar Panel
\textbf{IR} Infrared
\textbf{D} Drag
\textbf{c} Circular
\textbf{pen} Penalty
\textbf{L} Lorentz

\textbf{Notes}

Vectors are printed in bold.

Sometimes a different quantity is assigned to the same symbol in a section. The text makes it obvious to the reader when this is the case. In addition, some symbols represent two different quantities. In these cases, the context in the text makes it clear which quantity the symbol signifies.
On August 7 1972, between the Apollo 16 and Apollo 17 lunar missions, a massive solar event occurred, sending protons, electrons, and heavier ions hurtling towards the Moon. The effects of this radiation would have been deadly, or at the least life threatening, to the Apollo astronauts if they had been on the surface of the Moon at the time.

Observatories on the surface of the Earth have kept track of solar events such as these ever since the late 1800s. NASA was fully aware of the possibility of such an event during the lunar landings, but in its race against the USSR to the Moon chose to accept the relatively small risk. The organization’s test pilots-turned-astronauts were expendable to a certain extent, and the lunar missions were only ten days in length… short enough for the organization to safely wager against an event occurring during a lunar mission.

For a mission of greater duration however, such as a mission to Mars, a space agency can no longer afford to simply take its chances. Some form of solar event prediction will be a necessity for this particular type of mission, allowing time for spacecraft to shutdown their electronics and astronauts to seek cover. Not only directly useful to the space industry, it would also be extremely useful to be able to predict the solar weather in general. Solar events can not only damage (or even destroy) spacecraft in orbit around the Earth, but also disrupt communications and electronic systems on the surface of the Earth.

One of the steps leading to the ability of predicting the Sun’s behavior is the placement of a spacecraft in a polar orbit around the Sun. To be able to predict the Sun’s behavior, measurements of the polar regions of the Sun are necessary, and to accomplish this, a spacecraft could be placed into a polar orbit of the Sun. This has been done to a limited extent in the past; Ulysses (launched October 1990) took measurements when it passed the Sun’s poles, but only did so three times (in its mission lifetime of almost 20 years) as it passed the Sun in its highly eccentric orbit. This orbit had a perihelion of just over 1 AU, so the spacecraft never got closer to the Sun than the Earth already is. To help create more accurate predictions something more permanent will have to be established, that can remote sense the Sun as well as perform in-situ measurements of the surrounding environment.

To this end a mission is to be designed that brings a single spacecraft into a polar orbit utilizing a continuous low-thrust propulsion system while fulfilling specific requirements. To establish these requirements the Trajectory Optimisation of a Solar Polar Mission [Garot, 2006] report is consulted, which
describes a trajectory design for a spacecraft using solar sailing. The requirements listed in Garot’s report are changed only slightly to reconcile the differences between the two propulsion methods. This mission design is then compared to the mission using a solar sail in order to determine which of the two is preferable.

A general mission overview, discussing objectives, requirements, and design considerations, is provided in Chapter 2. Chapter 3 holds information on all the analytical aspects (such as exponential sinusoids, time of flight, gravity assists, etc.) that were involved in the implementation of a solution. Chapter 4 provides information on where and what numerical methods are used to solve particular equations within the software. Additionally, the numerical methods are tested for their accuracy and speed. Information on the topic of optimization and the implemented global and local search algorithms is located in Chapter 5. Chapter 6 describes the implementation of the theory contained within Chapter 3. Testing (validation, tuning, and sensitivity analysis) is performed in Chapter 7, where results are compared to those in previous reports. In addition, an overview of perturbations (and why they may be neglected for a first order study) is offered to the interested reader at the end of the chapter in section 7.5. A variety of different transfers that were inspected is given in Chapter 8. Finally, conclusions and recommendations are provided in Chapter 9.

There are also a number of appendices, which are also referred to in the text. Appendix A contains a derivation of the set of equations of motion modeling the continuous thrust in tangential and radial direction in the 2-body problem. Appendix B contains a derivation of a rotation matrix for a rotation about an arbitrary axis defined by two points in a three dimensional space. Appendix C contains some elementary derivation for the two bodied problem. Appendix D provides an overview of the created functions in the software. Finally, Appendix E shows a selection of search space figures of the Earth Jupiter Sun and Earth Venus Sun transfers.
As mentioned previously in the introduction, the goal of this research thesis is to explore the possibility of a transfer from Earth to a polar orbit at 0.4 AU distance from the Sun using a continuous low-thrust propulsion system. Before delving further into the theory (in the next chapter) this chapter will cover some necessary basics, such as the science objectives in section 2.1, the mission requirements in section 2.2, and the design considerations in section 2.3. General information on electric propulsion is provided in section 2.4 and finally spacecraft specifications are listed in section 2.5.

2.1 SCIENCE OBJECTIVES

The Sun is responsible for all life on our planet. It addition to providing the energy necessary for photosynthesis, it also drives our climate and weather. The Sun itself accounts for more than 99% of the mass in the solar system [Williams, 2004]. Despite being such an important object in our solar system, many questions still remain about the physical processes going on inside, on the surface, and in the surroundings of the Sun. The Earth is the only place in the solar system, where under the relative safety of its protective magnetic field and atmosphere, life has developed and flourished. How the Earth responds to the Sun’s variations is one of the on-going science goals for ESA, NASA and other organizations. With long duration manned space flight missions on the table (e.g. a permanent lunar base or a mission to Mars) there is more than ever a need for the ability to predict the Sun’s behavior.

Due to developments in the industry (e.g. Commercial Off The Shelf, miniaturization) commercial spacecraft are becoming increasingly sensitive to the space environment [Barnaby, 2005]. Even more important, understanding the solar environment is critical to support further human ventures out into space. Future manned spaceflight missions to objects such as the Moon and Mars will rely on the accurate prediction of solar activity. For these reasons, it has become apparent that the understanding of the solar environment must be given a high priority.

The Sun can produce noticeable disturbances in the space environment. Besides emitting the solar wind (a continuous and non-uniform stream of charged particles, also called plasma) the Sun also occasionally releases billions of tons of matter, named Coronal Mass Ejections (CME) [Hathaway, 2007]. When pointed towards the Earth, such an immense cloud of charged matter can cause large magnetic storms in the magnetosphere and upper atmosphere. These storms can be observed on or near Earth by their influence in many forms, such as the aurora borealis, disruptions to communications, spacecraft, and
Mission Overview

many others. The physics that govern the Sun are not yet fully understood, and so scientists would like [Smith, 2008]

- to understand the solar activity cycle,
- to understand the physics of solar particle events,
- to understand how the different regions and layers of the Sun evolve over space and time,
- and to identify the role of the magnetic field in delivering energy to the solar atmosphere and its many layers.

A solar polar probe would provide us with measurements over the Sun’s complete area and would be able to provide in-situ measurements. The scientific objectives of a spacecraft in a polar orbit around the Sun, which would help in addressing the above issues, are [ESA, 2004]

✓ Image the global extent and dynamic effects of coronal mass ejections.
✓ Discover the sources, longitudinal structure, rotational curvature and time-variability of coronal features.
✓ Link particle and field observations to images of the Sun, corona, and heliosphere at all latitudes.
✓ Determine magnetic structures and convection patterns in the Polar Regions.
✓ Follow the evolution of solar structures over a full solar rotation.

To this end a mission is to be designed that brings a single spacecraft into polar orbit utilizing a continuous low-thrust propulsion system, while fulfilling specific requirements.

2.2 MISSION REQUIREMENTS

To establish these requirements the report Trajectory Optimisation of a Solar Polar Mission [Garot, 2006] is consulted, which describes a trajectory design for a spacecraft using solar sailing. The requirements listed in Garot’s report are rewritten slightly to reconcile the differences between the two propulsion methods, and are

Payload The minimum payload mass is 2 kg.
Initial Orbit Assume the spacecraft is launched into a GTO orbit with 0° inclination with respect to the ecliptic plane†.
Final Orbit The Science Orbit is a circular solar polar orbit with radius 0.4 AU and an inclination of 90°.
Operation The spacecraft should arrive at the Science Orbit before the 1st of July, 2020, two years before the solar cycle maximum in 2022. The spacecraft should be able to operate for a period of 4 years once inserted into Science Orbit.
Trajectory The spacecraft is assumed to be able to survive at 0.4 AU distance from the Sun. Should parts of the trajectory bring it much closer to the Sun the spacecraft’s thermal design must allow for this.
Optimization Optimize the trajectory for low cost.

These requirements will be discussed in more detail in the following section.

† The original requirement stated a 0° inclination with respect to an unknown plane, so the ecliptic plane is assumed.
2.3 DESIGN CONSIDERATIONS

Although Garot’s report states a minimum payload of 2 kg the actual design put forth in Trajectory Optimisation of a Solar Polar Mission [Garot, 2006] is a spacecraft of 204 kg (it is not specifically stated whether the mass of the sail and booms is included or not) carrying a payload mass of 5 kg. As we would like to match this payload mass, the payload requirement is essentially increased from 2 to 5 kg. Therefore, the designs put forward in Chapter 8 in this report carry a payload mass of 5 kg. The total time of flight of the solar sailing spacecraft was 2,219 days. The payload mass (5 kg) and time of flight (2,219 days) design parameters will be the reference for a mission using electric propulsion.

The optimization requirement related to cost is approached by minimizing the parameters of spacecraft mass and time of flight, because both are directly linked to cost. Every extra kilogram of mass added will increase the cost to design, build, and launch the vehicle (this is not always the case but serves as a rule of thumb), while additional time spent during the transfer means additional costs in operating the spacecraft and the supporting ground equipment.

The payload can constitute between 15% to 30% of the dry spacecraft mass [Wertz, et al., 1999]. This gives us a minimum dry spacecraft mass of between 6.7 and 13.3 kg for 2 kg of payload. For 5 kg of payload the spacecraft dry mass is between 16.7 and 33.3 kg. The propellant may form up to be a large portion of the mass of a spacecraft with an electric propulsion system. For example, the 2nd Global Trajectory Optimization Competition (GTOC 2) problem definition specifies a spacecraft of 1500 kg, of which 1000 kg is available propellant [Petropoulos, 2008].

2.4 LOW-THRUST SPACEFLIGHT

Low Thrust Propulsion (LTP) is a group of propulsion methods that provide a low thrust, yet may offer advantages compared to more traditional chemical propulsion systems for certain missions. Some examples of LTP are laser propulsion (where a ground based laser propels the spacecraft), solar sailing (where solar radiation propels the spacecraft) and electric propulsion systems. While other forms of LTP have yet to be actually flown in space, electric propulsion has already been used to great success (e.g. Dawn, Deep Space 1, Hayabusha, and SMART-1).

The idea of electric propulsion first came to light at the start of the 20th century, when visionaries Konstantin Eduardovitch Tsiolkovsky and Robert Hutchings Goddard mentioned the possibility of electric propulsion in their published articles.

Tsiolkovsky, perhaps the man who started rocketry with his 1903 article showing the derivation of (what would later be known as) the Tsiolkovsky rocket equation, wrote in 1911 [Choueiri, 2004]:

“It is possible that in time we may use electricity to produce a large velocity for the particles ejected from a rocket device.”

The article then continues with the subsequent text:

“It is known at the present time that the cathode rays in Crookes’ tube, just like the rays of radium, are accompanied by a flux of electrons whose individual mass is 4,000 times less than the mass of the...
helium atom, while the velocities obtained are 30,000 – 100,000 km/s i.e. 6,000 to 20,000 times greater than that of the ordinary products of combustion flying from our reactive tube.”

This is the earliest published mention of electric propulsion. The quotes show that Tsiolkovsky was aware of the possibilities of electrically accelerating particles to provide thrust. In his text he referred to electrons, and not ions, simply because the latter were not discovered yet.

In the earliest documented source (1906) where the possibility of electric propulsion is considered, Goddard writes in his notes on the problem of producing a “reaction with electrons moving with the velocity of light”. Again, electrons are referred to because other charged particles are unknown as of yet. He also writes several questions regarding electric propulsion, in particular [Choueiri, 2004]:

“At enormous potentials can electrons be liberated at the speed of light, and if the potential is still further increased will the reaction increase (to what extent) or will radioactivity be produced?”

This shows that Goddard was well aware of the possibilities and outstanding questions regarding electric propulsion.

The person who first popularized electric propulsion was Hermann Julius Oberth, who published Wege zur Raumschiffahrt (Ways to Spaceflight) in 1929. This book devoted a complete chapter to spacecraft power and electric propulsion, and was distributed and well known throughout the world [Choueiri, 2004].

In the end, despite the idea’s conception at the start of the century, it would take 54 years to realize the first tests of an electric propulsion system in space. And although the Russians have been applying electric propulsion to their spacecraft for the past decades, it would take the West until the nineties to fully start applying the technology. Nowadays, there are over 200 spacecraft using some form of electric propulsion, in most cases GEO satellites using this type of propulsion for station keeping purposes. Electric propulsion designs have also been successfully applied to interplanetary transfers because of their high specific impulse (or high exhaust velocity), enabling the spacecraft to achieve large changes in momentum over time.

2.4.1 WHY USE ELECTRIC PROPULSION?

Chemical propulsion systems create thrust by thermodynamically expanding heated propellant through a nozzle. The energy to heat this propellant is stored in the chemical bonds of the monopropellant, or fuel and oxidizer, and is released through decomposition of the monopropellant, or released through a chemical reaction of the propellant and oxidizer. The amount of energy in chemical propulsion systems is limited by the available reaction energy in the propellant (they are said to be ‘energy limited’), and exhaust velocities are limited to a few thousand meters per second. Many desirable space missions, and thus their trajectories, would benefit from propulsion systems that would be able to provide a greater velocity increment (ΔV) by an order of magnitude, or more.

Electric propulsion systems provide us this opportunity. They achieve a higher velocity increment (ΔV), with much higher specific impulses (i.e. exhaust velocities), than a traditional chemical system can achieve. Unlike chemical propulsion, where energy is stored inside the propellant, electric propulsion systems acquire their energy from other sources (such as solar or nuclear energy). Their main limitation is
the rate of power that can be provided to the propulsion system, hence they are ‘power limited’ [ESA, 2008]. The rate of power is limited by available mass for the power supply, making spacecraft with electric propulsion a low thrust-to-mass ratio (and consequently low acceleration). Yet although the thrust-to-mass ratio is low, they are able to achieve a far greater total change of momentum (i.e. they have a larger amount of impulse) than spacecraft using chemical propulsion. To summarize, a chemical propulsion system offers a higher thrust-to-mass ratio, but the propellant is expended quickly at a lower specific impulse. An electric propulsion system provides a lower thrust-to-mass ratio, but can operate for long periods of time, thereby building up a larger total impulse. To accentuate the differences between chemical and electric propulsion systems, Table 2.1 showcases some of the advantages of electric propulsion.

<table>
<thead>
<tr>
<th>Propulsion Type</th>
<th>Chemical</th>
<th>Electric</th>
</tr>
</thead>
<tbody>
<tr>
<td>Propellant Type</td>
<td>Soyuz Fregat Main Engine (S5.92M)</td>
<td>SMART-1 Hall Effect Thruster (PPS-1350)</td>
</tr>
<tr>
<td>Thruster</td>
<td>Nitrogen tetraoxide / Unsymmetrical dimethyl hydrazine</td>
<td>Xenon</td>
</tr>
<tr>
<td>Specific Impulse</td>
<td>320 s</td>
<td>1640 s</td>
</tr>
<tr>
<td>Thrust</td>
<td>1.96⋅10^9 N</td>
<td>6.80⋅10^{-2} N</td>
</tr>
<tr>
<td>Thrust Time</td>
<td>877 s</td>
<td>1.80⋅10^{-7} s</td>
</tr>
<tr>
<td>Propellant consumed</td>
<td>5350 kg</td>
<td>80 kg</td>
</tr>
<tr>
<td>Total Impulse</td>
<td>1.72⋅10^7 Ns</td>
<td>1.2⋅10^6 Ns</td>
</tr>
</tbody>
</table>

Table 2.1 Comparison of propulsion chemical and electric propulsion technology [ESA, 2008].

It can be seen that the chemical Soyuz engine produces roughly 14 times the amount of impulse than the Hall Effect Thruster on the SMART-1 does, but uses almost 70 times the amount of propellant mass to do so. The hydrazine thruster produces less than a tenth of the total impulse the Hall Effect Thruster produces, while using 65 % as much propellant mass [ESA, 2008].

Solar sailing provides another low-thrust propulsion option, where the energy for propulsion is provided by the Sun. However, electric propulsion has some technical advantages when compared to solar sailing. The list below contains a number of arguments that can be made in favor of electric propulsion with respect to solar sailing.

**Tried and tested** The technology of electric propulsion has already been used in LEO, GEO, and in interplanetary missions. It has proven itself to be a reliable technology that works well. At the time of writing, many COTS engines are available from various manufacturers. In contrast, a solar sail has yet to be successfully tested in space.

**More reliable** Electric propulsion is mechanically less complex. It has no moving parts and does not have the same steering issues (attitude control) that solar sailing has. And because it has been flown many times before there is more experience with the technology. This makes electric propulsion more reliable than solar sailing.

**No deployment** A solar sailing mission will need to deploy its sail and supporting structure after launch; this is a mechanically complex maneuver. Electric propulsion requires no such deployment.

**No separation** After reaching final orbit the solar sail must be jettisoned in order to prevent
interference with the science payload [ESA, 2004]. The study [ESA, 2004] states that the ‘separation must take place with a minimum risk of collision.’ A solar polar orbiter using electric propulsion has no such issues. It can simply turn off its main engine, and easily restart it again should any station keeping or maneuvering be necessary.

Steering A spacecraft with electric propulsion is less limited with respect to the direction it can thrust than a spacecraft with solar sails.

Of course, based on these arguments alone it is impossible to fully ascertain which option is superior; after all, the fact that the thrust is free for a solar sail spacecraft is a major benefit.

2.5 SPACECRAFT SPECIFICATIONS AND ENGINE CHARACTERISTICS

Some generic parameters of the spacecraft are assumed in this study. Unless specified otherwise the following parameters are passed on to the optimization process

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial Mass</td>
<td>1500 kg</td>
</tr>
<tr>
<td>Specific Impulse</td>
<td>5000 s</td>
</tr>
<tr>
<td>Solar Panels Area</td>
<td>33 m²</td>
</tr>
<tr>
<td>Panel Efficiency</td>
<td>25 %</td>
</tr>
<tr>
<td>Thrust Efficiency</td>
<td>75 %</td>
</tr>
</tbody>
</table>

Some remarks about the program must be made here concerning these parameters. These numbers can be set to any value before performing an optimization. In addition, a transfer found by the optimization process can be rerun afterwards with different initial parameters to inspect the differences. The above parameters were chosen to reflect a current state of the art electric propulsion engine, powered by space quality solar panels with a quite large area (for reference, a typical geostationary communications satellite has a panel area of roughly 80 m²). The mass, a relatively large value, was chosen such that any given transfer will not run into trouble with propellant mass exceeding initial mass. Although the optimization runs have been performed using solar panels, the option to use a constant power source is also available. This will provide a constant maximum thrust (independent of distance to Sun) to the spacecraft.

If the optimization is not constrained by any penalties imposed upon the thrust profile then a single transfer shape can easily be recalculated with any initial mass and specific impulse. This is due to the fact that the approximation is shape based. This shape prescribes an acceleration, which then together with spacecraft parameters gives values such as the final mass, etc. In this way, taking initial mass (one of the initial parameters) as an example, a transfer with a specified initial mass can instantly be reexamined with a different initial mass to see how the final values (such as total propellant cost) differ for the exact same trajectory shape.
CHAPTER 3

DESIGNING CONTINUOUS THRUST TRANSFERS

The design of an efficient continuous propulsion trajectory is typically treated as an optimization problem. Generally, the optimization requires some form of initial estimate, which helps to obtain a broad overview of the problem, which provides a starting point for the trajectory optimization. Such an estimate can be provided by the shape-based approach. The shape based approach assumes a powered spacecraft trajectory of a certain shape, from which a corresponding thrust profile is established. If the shape is chosen correctly, a trajectory of satisfactory performance with a feasible thrust profile hopefully follows. One such shape is the exponential sinusoid, which is discussed in detail in section 3.1. Section 3.2 contains the solution of Lambert’s Problem employing an exponential sinusoid. Section 0 describes the determination of terminal velocities, and the time of flight for this type of transfer is discussed in section 3.4. Propellant and power requirements are inspected in section 3.5. The theory of gravity assists is described in section 3.6. Finally, an analytical solution for low thrust orbit plane changing is given in section 3.7.

Although there are other shapes that can also be used to analytically describe a low-thrust transfer the exponential sinusoid shape was chosen to conduct this research with. A single leg solution using exponential sinusoid had already previously been implemented in GALOMUSIT, facilitating verification. Additionally, the large number of research documents exploring this type of shape illustrates its popularity and also allows for verification of, for example, the solution of Lambert’s problem for an exponential sinusoid.

3.1 THE EXPONENTIAL SINUSOID

The equation for the exponential sinusoid is given as [Petropoulos, et al., 2004]

\[ r = k_0 e^{q \theta} + k_1 \sin (k_2 \theta + \phi) \tag{3.1} \]

where

- \( r \) is the radial distance,
- \( q \) is a constant parameter,
Design

- $k_0$ is the scaling factor,
- $k_1$ is the dynamic range parameter,
- $k_2$ is the winding parameter,
- $\theta$ is the true anomaly,
- $\phi$ is the phase angle.

The figure below provides an overview of the geometry of an exponential sinusoid and shows the parameters used in this section.

![Figure 3.1 Exponential Sinusoid geometry configuration [Petropoulos, et al., 2000].](image)

The constant parameter $q$ is taken as zero, as non-zero values are detrimental to the method’s performance [Petropoulos, et al., 2000] [Petropoulos, et al., 2004] [Izzo, 2006]. This makes the shape a purely exponential sinusoid (similarly, if the dynamic range parameter $k_1$ is taken as zero, the spiral is purely logarithmic). Thus, for our purposes the equation of the exponential sinusoid takes the form of

$$ r = k_0 e^{k_1 \sin (k_2 \theta + \phi)} $$

(3.2)

Specifying an arbitrary thrust profile requires the solution of two coupled differential equations, a difficult task. In order to analytically solve the problem the thrust direction is assumed to always be tangential (along or against) with respect to the velocity vector. This assumption results in relatively good velocity profiles for both flyby missions and rendezvous missions. The two body equations of motion in polar coordinates (equation (A.14), as derived in Appendix A) are

$$ \ddot{r} - r \dot{\theta}^2 + \frac{\mu}{r^2} = a_T \sin \alpha $$

$$ r \ddot{\theta} + 2 \dot{r} \dot{\theta} = a_T \cos \alpha $$

(3.3)

The thrust acceleration can be normalized by the local gravity acceleration to give

$$ a_N \equiv \frac{a_T}{\mu/r^2} $$

(3.4)

Thus the normalized thrust acceleration diminishes with the squared distance (this is roughly the same relation for solar powered electric propulsion, which diminishes with regards to distance from the Sun).
The flight path angle $\gamma$ is given geometrically as (cf. Figure 3.1)

$$\tan \gamma = \frac{\dot{r}}{r \dot{\theta}} = \frac{1}{\frac{dr}{dt} \frac{d\theta}{dt}} = \frac{1}{\frac{dr}{d\theta}}$$  \hspace{1cm} (3.5)

The radial velocity as a function of position on the exponential sinusoid can be determined by differentiating equation (3.2) and then substituting equation (3.2) into the result to obtain

$$\dot{r} = k_0 k_1 k_2 \dot{\theta} \cos(k_2 \theta + \phi) e^{k_1 \sin(k_2 \theta + \phi)} = r \dot{\theta} k_1 k_2 \cos(k_2 \theta + \phi)$$  \hspace{1cm} (3.6)

Substituting this differentiated shape equation into equation (3.5) gives the following expression for the flight path angle $\gamma$ [Petropoulos, et al., 2004]

$$\tan \gamma = k_1 k_2 \cos(k_2 \theta + \phi)$$  \hspace{1cm} (3.7)

Under the assumption of a tangential thrust direction (and assuming that $q = 0$) the shape equation (3.2), equations of motion (3.3), and flight path angle equation (3.7) yield the expression for angular rate [Petropoulos, et al., 2004]

$$\dot{\theta}^2 = \frac{\mu}{r^3 \tan^2 \gamma + k_1 k_2^2 s + 1}$$  \hspace{1cm} (3.8)

and the expression for the normalized thrust acceleration [Petropoulos, et al., 2004]

$$a_N = \frac{(-1)^n \tan \gamma}{2 \cos \gamma} \left( \frac{1}{\tan^2 \gamma + k_1 k_2^2 s + 1} - \frac{k_2^2 (1 - 2k_1 s)}{(\tan^2 \gamma + k_1 k_2^2 s + 1)^2} \right)$$  \hspace{1cm} (3.9)

where

$$s = \sin(k_2 \theta + \phi)$$  \hspace{1cm} (3.10)

For the assumption of tangential thrust, the thrust angle is given by

$$\alpha = \gamma + n\pi$$  \hspace{1cm} (3.11)

where $n$ is an integer chosen such that the normalized thrust acceleration is positive. When $n = 0$, the thrust is aligned along the velocity vector. When $n = 1$, the thrust is aligned against the velocity vector.

Equations (3.6) and (3.8) show the circumferential and radial velocities purely as a function of the position on the exponential sinusoid. Similarly, equation (3.9) provides the normalized thrust acceleration solely as a function of position on the exponential sinusoid. It can be seen from equations (3.8) and (3.9) that both the thrust acceleration $a$ and angular rate $\dot{\theta}$ approach infinity at periapsis ($s = -1$) when $k_1 k_2^2$ approach unity from below (i.e. values smaller than one). When $k_1 k_2^2 > 1$ the angular rate $\dot{\theta}^2$ is less than zero in the region of periapsis, meaning that this particular shape cannot be followed using tangential thrust. Thus, the search space is limited by placing boundary conditions upon these parameters.

Dynamic Feasibility

This observation that only trajectories with a certain value of $k_1 k_2^2$ are valid can be shown mathematically by using the trigonometric relation $\cos^2 \theta + \sin^2 \theta = 1$ and equations (3.7) and (3.10) to write
Design

\[ \tan^2 \gamma \frac{k_1}{k_2} + s^2 = 1 \]  
(3.12)

By multiplying the terms in this equation with the quantity \( k_1 k_2 \) and replacing \( \tan^2 \gamma \) using equation (3.7) we arrive at

\[ k_1 k_2 c^2 + k_1 k_2 s^2 - k_1 k_2 = 0 \]  
(3.13)

where

\[ c = \cos(k_2 \theta + \phi) \]  
(3.14)

Equation (3.13) shows that \( k_2 \) prescribes \( k_1 \), or vice versa. To avoid singularities near periapsis (by considering only positive values of \( k \)) we require that [Petropoulos, et al., 2004]

\[ 1 - k_1 k_2^2 > 0 \]  
(3.15)

This constraint ensures that the denominators in equations (3.7) and (3.8) are always positive, thus ensuring that the trajectory can be followed using tangential thrust.

In effect, placing a limit on the value of \( k_1 k_2 \) translates into placing a cap on the values of thrust acceleration \( a \) and angular rate \( \dot{\theta} \). This gives the designer some way of specifying what the acceleration magnitude should be like, effectively allowing the designer to avoid trajectories that exceed the spacecraft thruster capabilities. The effects of this constraint can be investigated by looking at the relation, when \( k_1 \) is very large, \( k_2 \) must necessarily be very small. This means that there is a large dynamic range (apoapsis is much greater than periapsis) which leads to many revolutions around the central body. On the other hand, when \( k_2 \) is large, \( k_1 \) must be small, meaning that few revolutions are required and that apoapsis is not much greater than periapsis (small dynamic range). There are some further considerations on the subject of the parameters; upper limits of \( k_1 \) and \( k_2 \) (e.g. \( k_1 = 2 \) and \( k_2 = 1 \)) can be imposed in order to prevent the thrust levels from becoming unsustainably high, cf. equation (3.9). In addition, if only one parameter is very large (and thus the other very small) the trajectory shape will generally not be useful [Petropoulos, et al., 2004].

3.2 SOLVING LAMBERT’S PROBLEM FOR EXPONENTIAL SINUSOIDS

Lambert’s Problem can be defined as the search for an unperturbed orbit about a given inverse square law focal point (gravitational body) that connects two given points for a specified flight time \( \Delta t \). Or mathematically equivalent, find \( r(t) \) such that [Kriz, 1976]

\[ \dot{r} + \frac{\mu}{r^3} r = 0, \quad r = |r| \]

\[ r(t_1) = r_1, \quad r(t_2) = r_2, \quad \Delta t = t_2 - t_1 \]

(3.16)

This is the equation of motion for a two-body problem with two assigned boundary values. It is also possible to solve Lambert’s Problem using exponential sinusoids [Izzo, 2006]. By assuming that \( k_2 \) is fixed the problem is defined by only three parameters: \( k_0 \), \( k_1 \), and \( \phi \). The polar coordinate system is fixed so that \( \theta_1 = 0 \), allowing us to rewrite equation (3.7) as [Izzo, 2006]
\( \tan \gamma_1 = k_1 k_2 \cos \phi \) \hfill (3.17)

To enforce the exponential sinusoid to fit the through the points \( r_1 \) and \( r_2 \) the shape equation is rewritten for start and endpoint to be [Izzo, 2006]

\[
\begin{align*}
  r_1 &= k_0 e^{k_1 \sin \phi} \\
  r_2 &= k_0 e^{k_1 \sin (k_2 \theta_2 + \phi)}
\end{align*}
\] \hfill (3.18)

where [Izzo, 2006]

\[
\theta_2 = \Delta \theta + 2 \pi N
\] \hfill (3.19)

where \( N \) is the number of additional complete revolutions. By dividing both terms \( (r_1 \) and \( r_2 \) in equation (3.18) by each other and taking the logarithm, the expression for the sign of \( k_1 \) is derived [Izzo, 2006]

\[
\frac{k_1}{|k_1|} \sqrt{\frac{k_1^2 - \tan^2 \gamma_1}{k_2^2}} = \frac{\ln(r_1/r_2) + (\tan \gamma_1/k_2) \sin(k_2 \theta_2)}{1 - \cos(k_2 \theta_2)}
\] \hfill (3.20)

Essentially, the sign of \( k_1 \) is determined by the numerator of the right hand term as the expression within the radical and expression in the denominator must by definition be positive for a valid transfer. Isolating for \( k_1 \) yields [Izzo, 2006]

\[
k_1^2 = \left( \frac{\ln (r_1/r_2) + \tan \gamma_1 \sin(k_2 \theta_2)}{1 - \cos(k_2 \theta_2)} \right)^2 + \tan^2 \gamma_1/k_2^2
\] \hfill (3.21)

For all \( k_2 \) there exists a class of exponential sinusoids (entirely determined by \( \gamma_1 \)) that fit the given geometry of the problem \( (r_1, r_2, \Delta \theta) \), and given the number of revolutions \( N \). Only some of these exponential sinusoids will be feasible trajectories. The condition shown in equation (3.15) is rewritten using equation (3.21) to arrive at [Izzo, 2006]

\[
\left( \frac{k_2^2 \ln(r_1/r_2) + k_2 \tan \gamma_1 \sin(k_2 \theta_2)}{1 - \cos(k_2 \theta_2)} \right)^2 + k_2^2 \tan^2 \gamma_1 < 1
\] \hfill (3.22)

It can be seen that this is a quadratic equation for \( \tan \gamma_1 \) and so the solution is [Izzo, 2006]

\[
\tan \gamma_{1,2} = k_2 \left( -\frac{r_1}{r_2} \cot \frac{k_2 \theta_2}{2} \pm \frac{\sqrt{2(1 - \cos(k_2 \theta_2)) - \ln^2 r_1/\Delta}}{k_2^4} \right)
\] \hfill (3.23)

If \( \Delta < 0 \), no feasible exponential sinusoid from the class of that particular value of \( k_2 \) exist. If \( \Delta > 0 \), feasible exponential sinusoids exist for values of \( \gamma_1 \) that lie between the two roots given by equation (3.22).
3.3 **Terminal Velocity**

Just as was done for the flight path angle $\gamma_1$ in equation (3.17), the same may be done for the flight path angle at the final point $\gamma_2$, yielding

$$\tan \gamma_2 = k_1 k_2 \cos(k_2 \theta_2 + \phi)$$  \hspace{1cm} (3.24)

Using equations (3.18) and (3.21) this can be rewritten to

$$\tan \gamma_2 = -\tan \gamma_1 - k_2 \ln \frac{r_1}{r_2} \cot \frac{k_2 \theta_2}{2}$$  \hspace{1cm} (3.25)

which can be rewritten, by using equation (3.23), to an even simpler form as [Izzo, 2006]

$$\tan \gamma_2 = \tan \gamma_1 + \tan \gamma_{12} - \tan \gamma_1$$  \hspace{1cm} (3.26)

This gives a simple geometric relation between the two terminal flight path angles. The velocity vector at any given position on the shape is given by

$$\mathbf{v} = \dot{r} \hat{e}_r + r \dot{\theta} \hat{e}_\theta$$  \hspace{1cm} (3.27)

From this it is trivial to determine the velocity magnitude as

$$v = \sqrt{\dot{r}^2 + (r \dot{\theta})^2}$$  \hspace{1cm} (3.28)

Substituting equations (3.2), (3.6), and (3.8) into the above allows us to determine the velocity at any point of the shape. The initial velocity is determined using the initial flight path angle $\gamma_1$. Given that $\gamma_2$ easily flows from $\gamma_1$ by using equation (3.26) the terminal velocity vector can then be calculated simply on the basis of the shape parameters.

3.4 **Time of Flight**

The time of flight can be obtained by substituting equation (3.8) into

$$t = \int \frac{dt}{d\theta} d\theta$$  \hspace{1cm} (3.29)

This leads to the expression of the time of flight as

$$t_2 - t_1 = \int_{\theta_1}^{\theta_2} \sqrt{\frac{r^3 (\tan^2 \gamma + k_1 k_2^2 s + 1)}{\mu}} d\theta$$  \hspace{1cm} (3.30)

where $r$ and $\gamma$ are given by equations (3.2) and (3.7). Unfortunately, this equation cannot be solved analytically (as it is not a closed form integral), so it is integrated by numerical quadrature to find the time of flight for a particular class of exponential sinusoids.

Provided with an arbitrary set of shape parameters we may determine the feasible region of initial flight path angles using equation (3.23). Thus, the time of flight can be computed for any initial flight path.
angle (within the feasible region) for the set of shape parameters - a curve can be plotted by performing the computation repeatedly across the entire feasible region. Although one only encounters one particular shape in the literature, there are three possible 'shapes' in total (pictured below): monotone decreasing, monotone increasing, and 'bathtub'.

![Figure 3.2 Overview of possible time of flight - initial flight path angle curves.](image1)

Some examples of these shapes are provided in section. It is possible to graphically demonstrate what different shapes are possible. When the expression for the time of flight is inspected it can be seen that the time of flight changes as a function of $\theta$ and $\gamma_1$. The influence of $\gamma_1$ from the expression is shown as

$$t_2 - t_1 = f(\tan^2 \gamma_1) \quad (3.31)$$

This influence is shown in the figure, along one domain area of $0 \, \pi$. It can be seen that for any selected time of flight, the corresponding horizontal line will only ever intersect the curve twice in one region between asymptotes. From this we may surmise that there are three possible shapes the time of flight curve can assume, depending on the choice of $k_2$ and $N$. Depending on the region of viable initial flight path angles between calculated minimum and maximum (and in the actual case depending on the other variables as well of course) a certain part of the domain is specified, leading to one of three possible shapes of the time of flight curve.

### 3.5 Propellant and Power Considerations

This section will lay out some of the fundamental equations used to examine quantities such as propellant mass, power, and thrust.

#### 3.5.1 Fuel Consumption

The derivation of some basic (but necessary) spacecraft engine equations starts with Newton’s second law, which states that the net force on an object is proportional to the time rate of change of its linear momentum

$$F = \frac{dp}{dt} = \frac{d(mv)}{dt} \quad (3.32)$$

Consider a rocket experiencing no forces (i.e. a gravity free vacuum) (as shown in Figure 3.4).
When the resultant external force acting on a system is zero, the total linear momentum of the system remains constant. This is called the conservation of momentum. For the rocket there is no change in momentum, as there are no external forces.

\[ 0 = \frac{\Delta p}{\Delta t} = \frac{p_1 - p_0}{\Delta t} \]  

(3.33)

Filling in the information from Figure 3.4 for time steps \( t_0 \) and \( t_1 \) yields

\[ 0 = (m - \Delta m)(v + \Delta v) + \Delta m \cdot u - m \cdot v \]  

(3.34)

If \( \Delta t \) is very small (approaching zero) equation (3.34) can be rewritten to

\[ m \frac{dv}{dt} = \frac{dm}{dt} (v + \Delta v - u) \]  

(3.35)

The summation of velocities on the right side of this equation can be seen as the exhaust velocity of the rocket, leading to

\[ m \frac{dv}{dt} = \frac{dm}{dt} v_e \]  

(3.36)

where

\( m \) is the time varying, but sampled at an instantaneous moment, mass of the rocket.

\( \frac{dv}{dt} \) is the vehicle acceleration.

\( \frac{dm}{dt} \) is the rate of change of the spacecraft mass due to propellant expulsion.

\( v_e \) is the velocity of the exhaust stream.

The product of the rate of mass expulsion and exhaust velocity is the thrust

\[ T = \frac{dm}{dt} v_e \]  

(3.37)

The ratio of the thrust to the rate of expulsion of propellant measured in units of weight is identified as the specific impulse [Wertz, et al., 1999]

\[ I_{sp} = \frac{T}{W} = \frac{v_e}{g_0} \]  

(3.38)

where
Design

\(g_0\) is the gravitational acceleration at sea level.

\(W = \frac{dm}{dt}g_0\) is the rate of propellant expulsion in units of weight per time.

The method that GALOMUSIT currently uses to determine propellant use is the solution of an integral that assumes a constant spacecraft mass during transfer. In reality the spacecraft mass while change as propellant is spent during transfer. Under the assumption of a constant mass of the spacecraft we can derive an integral that can be solved numerically to arrive at the total fuel consumption. The resulting integral is used only to compare the differences with a more realistic method (described hereafter) that accounts for spacecraft mass change during transfer.

**Mass Evolution under the Assumption of Constant Mass**

To arrive at an equation of fuel consumption we can integrate the spacecraft mass rate of change over time

\[
\int_{t_0}^{t_1} \frac{dm}{dt} \, dt
\]  

(3.39)

and subsequently substitute equations (3.36), (3.37), and (3.38) to arrive at

\[
\int_{t_0}^{t_1} \frac{ma}{l_{sp}g_0} \, dt
\]  

(3.40)

where \(a\) represents the spacecraft acceleration due to the engine, cf. equation (3.36). An expression for the normalized tangential thrust acceleration on an exponential sinusoid was given previously in equation (3.9), which is converted to the tangential thrust acceleration by writing

\[a = \frac{\mu}{r^2}a_N\]  

(3.41)

As we are often performing calculations on the basis of position along the shape (instead of time passed) it is useful to rewrite equation (3.40) by introducing the true anomaly

\[
\int_{\theta_0}^{\theta_1} \frac{ma}{l_{sp}g_0} \frac{dt}{d\theta} \, d\theta
\]  

(3.42)

Using the expression for the angular rate, shown in equation (3.8), this can be equivalently written as

\[
\int_{\theta_0}^{\theta_1} \frac{ma}{l_{sp}g_0} \frac{1}{\dot{\theta}} \, d\theta
\]  

(3.43)

Just as is the case for the time of flight integral this expression cannot solved analytically, and so it can be integrated by numerical quadrature, e.g. using Composite Simpson’s Rule.

**Mass Evolution with Varying Mass**

Of course it is more realistic, especially in the case of longer transfers, to consider the mass as a variable parameter during the integration itself. In this section, it will be shown that the equation becomes a first order ordinary differential equation. This cannot be solved by the method of Composite Simpson’s rule; instead we must look to other more general purpose quadrature methods. We may start the derivation by beginning with the scalar version of equation (3.36)

17
Design

\[
m \frac{dv}{dt} = \frac{dm}{dt} \nu_e
\]

(3.44)

Here, only the exhaust velocity \( \nu_e \) is assumed to be constant, cf. equation (3.38). And so we may rewrite, using equations (3.37) and (3.38), to arrive at

\[
\frac{dm}{dt} = \frac{a(t)}{g_0} \frac{1}{I_{sp}} m(t)
\]

(3.45)

It can be seen that this is a homogeneous first order linear ordinary differential equation. We would like to know the mass on the basis of position along the shape (instead of on the basis of time) and so it is useful to substitute equation (3.8), relating time and position, to yield the following form

\[
\frac{dm}{d\theta} = \frac{a(\theta) \ dt}{g_0 I_{sp}} (\theta) m(\theta)
\]

(3.46)

or written more concisely, with initial value, as

\[
m' = f(\theta) m(\theta), \quad m(0) = m_0
\]

(3.47)

This ordinary differential equation would be analytically solvable if not for the fact that \( f(\theta) \) (and for that matter \( f(t) \)) cannot be performed in closed form when solving a homogeneous first-order linear ordinary differential equation using standard procedure. Hence we must solve, by numerical quadrature, the following initial value problem

\[
m' = f(\theta) m(\theta), \quad m(0) = m_0
\]

(3.48)

where

\[
f(\theta) = \frac{a}{g_0 I_{sp}} \frac{dt}{d\theta}
\]

\[
a = \frac{\mu}{r^2} \left( -1 \right)^n \tan \gamma \left( \frac{1}{\tan^2 \gamma + k_1 k_2^2 s + 1} - \frac{k_2^2 (1 - 2 k_1 s)}{(\tan^2 \gamma + k_1 k_2^2 s + 1)^2} \right)
\]

\[
\frac{dt}{d\theta} = \frac{r^3 (\tan^2 \gamma + k_1 k_2^2 s + 1)}{\mu}
\]

\[
r = k_0 e^{q \theta} + k_1 \sin (k_2 \theta + \phi)
\]

\[
\tan \gamma = k_1 k_2 \cos (k_2 \theta + \phi)
\]

All other parameters are constant during solving.

Composite Simpson’s method no longer applies for the solution of ordinary differential equations and thus some other integrator such as Euler or Runge-Kutta must be employed in order to find the solution. It should also be noticed that the relation \( f(\theta) \) should always be negative as in all cases, irrelevant of the thrust direction, fuel is being spent and the spacecraft mass is decreasing.
3.5.2 AVAILABLE THRUST (POWER SUPPLY)

The propulsion system of the spacecraft will require a power source to function. As the engine is power limited it is important to implement a model of the available power to find the actual maximum engine performance at an arbitrary point on the exponential sinusoid shape. For a spacecraft powered by solar panels it is important to know that the total radiation per unit area at 1 AU from the Sun is $I_{\odot} = 1,367\, \text{W/m}^2$ [Wertz, et al., 1999]. Based on this fact we can deduce the solar irradiance at any given distance in AU from the Sun as

$$I_{\text{Solar}} = \frac{I_{\odot}}{r^2} \quad (3.49)$$

The equivalent notation using kilometers is

$$I_{\text{Solar}} = I_{\odot} \frac{149597870.7^2}{r^2} \quad (3.50)$$

The power provided by the solar panels is then

$$P = I_{\text{Solar}} A_{\text{Panel}} \eta_{\text{Panel}} \quad (3.51)$$

where

- $I_{\text{Solar}}$ is the solar irradiance.
- $A_{\text{Panel}}$ is the total area of the solar panels.
- $\eta_{\text{Panel}}$ is the efficiency of the solar panels (between 0 and 1).

The mass of a power supply can be written as [Wertz, et al., 1999]

$$m_p = m_s P \quad (3.52)$$

where

- $m_p$ is the power supply mass.
- $m_s$ is the specific power supply mass (mass per unit power).
- $P$ is the power required for propulsion.

The power necessary to attain a certain thrust is easily derived. Power is energy per unit of time, in this case kinetic energy per time, which easily translates to [Wertz, et al., 1999]

$$P_t = \frac{1}{2} m \frac{dv^2}{dt} = \frac{m v_e}{2} \frac{dv}{dt} = \frac{v_e T}{2} = \frac{g_0 I_{sp} T}{2} \quad (3.53)$$

where

- $v$ is the spacecraft velocity.
- $v_e$ is the exhaust velocity.
- $T$ is the desired thrust setting.
- $P_t$ is the power required for the desired thrust setting.

The thruster efficiency is defined as how much the thruster converts input power into thrust power [Wertz, et al., 1999].
Design

\[ \eta_T = \frac{P_t}{P} = \frac{v_e T}{2P} = \frac{g_0 I_{sp} T}{2P} \quad (3.54) \]

By using equation (3.52) this can be solved for the power supply mass.

\[ m_p = \frac{m_s v_e T}{2 \eta_T} = \frac{m_s \frac{dm}{dt} g_0^2 I_{sp}^2}{2 \eta_T} \quad (3.55) \]

And expressing the required power for propulsion in engine parameters yields

\[ P = \frac{g_0 I_{sp} T}{2 \eta_T} = \frac{g_0^2 I_{sp}^2}{2 \eta_T} \quad (3.56) \]

And so the maximum available thrust (when provided with a specified power \( P \)) is

\[ T = \frac{2P \eta_T}{g_0 I_{sp}} \quad (3.57) \]

Dividing by the spacecraft mass leads to the maximum available thrust acceleration

\[ a = \frac{2P \eta_T}{g_0 I_{sp} m} \quad (3.58) \]

It is useful to know the maximum available thrust acceleration because the trajectory shape will prescribe a particular thrust acceleration profile that may exceed the limits. Comparison between the available and prescribed accelerations can show whether a certain transfer is realistic or not. Comparison may also occur during the optimization process itself by incurring penalties (to various degrees) on transfers violating the available acceleration level.

Using the above equations, a plot of the maximum thrust acceleration as a function of solar distance is made and shown in Figure 3.5 for a hypothetical spacecraft (parameters provided in the figure).

![Figure 3.5 Maximum spacecraft engine acceleration as a function of distance from the Sun.](image)

**Figure 3.5** Maximum spacecraft engine acceleration as a function of distance from the Sun.

- Solar panel area = 30 m²
- Solar panel efficiency = 0.25
- Thruster efficiency = 0.75
- Mass = 250 kg
- Specific impulse = 5000 s
The plot clearly shows the quadratic increase in available energy to the spacecraft as it is placed closer and closer to the Sun. Of course, in reality an engine design will only be able to achieve a certain specified level of thrust regardless of how much extra power is provided to the system.

3.6 GRAVITY ASSISTS

A gravity assist, also referred to as gravitational slingshot, or swing-by is used to alter the velocity vector of a spacecraft as it passes a planet in a hyperbolic trajectory. The law of conservation of energy states that if the spacecraft gains energy, the planet must necessarily lose some. However, the planet is so massive compared to the spacecraft that the effect on the planet’s velocity, in the reference frame of the Sun, is negligible. In the reference frame of the planet itself the magnitude of the velocity vector is unaltered; it is merely the direction that changes [Cornelisse, et al., 1979].

Reducing the total energy requirement of a mission is the main advantage (leading to reductions in fuel mass, spacecraft size, and cost) of a gravity assist. One can distinguish between several types of gravity assist: gravity assists that utilize thrust (powered gravity assist), gravity assists that use the atmosphere, if available, of a planet (aero gravity assist) and the simplest form of gravity assist, the unpowered gravity assist, which will be discussed in further detail in this section.

The spacecraft closes in on the planet with a velocity \( \mathbf{v}_2 \) (cf. Figure 3.6) with respect to the target planet. The spacecraft’s state vector on entering the planet’s sphere of influence is determined by the impact parameter \( B \) (cf. Figure 3.6). This parameter essentially determines the exit velocity vector of the spacecraft after it has left the sphere of influence [Cornelisse, et al., 1979]. For example, it can be chosen such that the orbit plane is altered, as was done for Ulysses when it used Jupiter’s gravity field to change its orbit plane out of the ecliptic to observe the poles of the Sun.

Figure 3.6 The in-plane geometry of a hyperbolic encounter trajectory, after [Cornelisse, et al., 1979].

The velocities (shown above in Figure 3.6) are \( \mathbf{v}_T' \), the projection of the planet’s velocity \( \mathbf{v}_T \) on the plane spanned by \( \mathbf{v}_2 \) and \( B \), and the hyperbolic excess velocities \( \mathbf{v}_\infty_1 \) and \( \mathbf{v}_\infty_2 \) of the spacecraft as it enters and
leaves the planet’s sphere of influence, respectively. Note that the hyperbolic excess velocities are identical in magnitude, i.e. \( |\mathbf{v}_{\infty 1}| = |\mathbf{v}_{\infty 2}| \), in the planetocentric reference frame.

As shown in Appendix C the specific angular momentum \( \mathbf{h} \), equation (C.12), remains constant, and so we may write [Cornelisse, et al., 1979]

\[
B \mathbf{v}_{\infty 1} = r_p \mathbf{v}_p
\]  

(3.59)

The minimum (periapsis) distance \( r_p \) can be determined. By substituting equations (3.59) and (C.37) into equation (C.46) to obtain the following quadratic relation

\[
v_{\infty 1}^2 r_p^2 + \frac{2\mu}{v_{\infty 1}} r_p - B^2 = 0
\]

(3.60)

from which the minimum (periapsis) distance \( r_p \) is easily determined by finding its roots, yielding the positive root

\[
r_p = -\frac{\mu}{v_{\infty 1}^2} + \sqrt{\frac{\mu^2}{v_{\infty 1}^2} + B^2}
\]

(3.61)

Evaluating equation (C.22) for a point on the hyperbola at infinite distance yields

\[
\cos \left( \frac{\alpha}{2} + \frac{\alpha}{2} \right) = \sin \frac{\alpha}{2} = -\frac{1}{e}
\]

(3.62)

From equations (C.21) and (C.22) we know that

\[
e = B / \mu
\]

(3.63)

Solving for \( B \) in equation (C.18) yields

\[
B = \mathbf{v} \times \mathbf{h} - \mu \frac{\mathbf{r}}{r}
\]

(3.64)

which is easily rewritten to

\[
e = \mathbf{v} \times \mathbf{h} - \frac{\mathbf{r}}{\mu} - \frac{\mathbf{r}}{r}
\]

(3.65)

Rewriting in scalar notation results in

\[
e = \frac{r_p v_p^2}{\mu_T} - 1
\]

(3.66)

Substituting equations (C.37) and (C.46) into equation (3.66) yields

\[
e = \frac{r_p v_{\infty 1}^2}{\mu_T} + 1
\]

(3.67)

Replacing the periapsis distance \( r_p \) by equation (3.61) and substituting this result into equation (3.62) gives the asymptotic deflection angle \( \alpha \) as a function of the impact parameter \( B \), hyperbolic excess speed \( v_{\infty 1} \), and gravitational parameter of the target \( \mu_T \) [Cornelisse, et al., 1979]
\[
\sin \frac{\alpha}{2} = \frac{1}{\sqrt{1 + \frac{B^2 v_{\infty}^4}{\mu_T^2}}} \quad (3.68)
\]

The minimum fly-by distance must be at least the radius of the target (so as to not hit the surface), thus equation (3.61) becomes [Cornelisse, et al., 1979]

\[
R_T \geq -\frac{\mu_T}{v_{\infty}^2} + \frac{\mu_T^2}{v_{\infty}^3} + B^2 \quad (3.69)
\]

Replacing the gravitational parameter \(\mu_T\) by substituting the relation for the escape velocity at the surface, a slightly modified equation (C.37), into this equation and isolating for \(B\) results in [Cornelisse, et al., 1979]

\[
B \geq R_T \sqrt{1 + \frac{v_{esc}^2}{v_{\infty}^2}}, \quad v_{esc} = \sqrt{\frac{2\mu_T}{R_T}} \quad (3.70)
\]

When this relation does not hold, i.e. \(B\) is smaller than the right-hand side, the spacecraft will impact upon the surface of the planet. Equation (3.68) shows that the maximum deflection angle for a specified hyperbolic velocity is obtained when the impact parameter \(B\) is minimum. The maximum deflection angle may then be written as

\[
\sin \frac{\alpha_{max}}{2} = \frac{1}{1 + \frac{R_T v_{\infty}^3}{\mu_T}} \quad (3.71)
\]

Looking at the planets reference frame no change in the magnitude of velocity (or change in energy) is noticed. Nonetheless, the heliocentric velocity of the spacecraft is altered once it has performed the swing-by and left the sphere of influence of the planet. This is shown in Figure 3.7 below.

\[
\Delta \mathcal{E} = \frac{1}{2}(v_2^2 - v_1^2) = \frac{1}{2}(v_2 + v_1) \cdot (v_2 - v_1) \quad (3.73)
\]

Substituting the relations listed in equation (3.72) into equation (3.73) yields
\[
\Delta \mathcal{E} = \frac{1}{2}(v_2^2 - v_1^2) = v'_T \cdot (v_{\infty_2} - v_{\infty_1})
\]  
(3.74)

This can be rewritten to scalar notation using the angles (between the hyperbolic velocity vectors and the planet velocity vector) as

\[
\Delta \mathcal{E} = v'_T v_{\infty_2} - v'_T v_{\infty_1} = v'_T v_{\infty_2} \cos(\alpha + \phi) - v'_T v_{\infty_1} \cos \left(\frac{3}{2} \pi - \frac{\alpha}{2} + \beta\right) = 2v'_T v_{\infty_1} \sin \frac{\alpha}{2} \cos \beta
\]  
(3.75)

Substituting equation (3.68) into (3.75) gives [Cornelisse, et al., 1979]

\[
\Delta \mathcal{E} = \frac{2v'_T v_{\infty_1} \cos \beta}{\sqrt{1 + \frac{B^2 v_{\infty_1}^4}{\mu_T}}} 
\]  
(3.76)

It can be seen that this energy change for a given \(v_{\infty_1}\) will be maximum when B is at minimum, and when the angles are \(\beta = 0\) (energy gain) and \(\beta = \pi\) (energy loss). Thus, substituting equation (3.70) gives us the relation for the maximum energy change [Cornelisse, et al., 1979]

\[
\Delta \mathcal{E}_{\text{max}} = \frac{2v'_T v_{\infty_1}}{1 + R_T v_{\infty_1}^2 / \mu_T}
\]  
(3.77)

Having described the theory we can now use this to study the influence of a gravity assist on a transfer. Using equation (3.77), the maximum change in energy is plotted versus the hyperbolic excess velocity for the inner planets and Jupiter in Figure 3.8. To create this plot the mean radius of the planet is selected as the minimum (or periapsis) distance. In addition, the angle \(\beta\) (see Figure 3.6) is chosen to be zero such that the component of the planet's velocity is maximum, i.e. \(v'_T = v_T\). Both these assumptions ensure the maximum energy change is plotted in Figure 3.8.

![Figure 3.8 The maximum heliocentric energy change for a flyby.](image-url)
Figure 3.9 shows the maximum deflection angle plotted versus the hyperbolic excess velocity, using equation (3.71). As was the case for the previous figure, the maximum angle is attained when the flyby is assumed to skim the planet’s surface, i.e. the periapsis distance is equal to the mean radius of the planet.

It can be seen that Jupiter is capable of changing both heliocentric energy as well as deflection angle quite considerably. Of the inner planets, the Earth and Venus are attractive swing-by planets. More experimentation will likely show what the possibilities are; in Chapter 8 several transfers using various planets as gravity assists are investigated.

### 3.7 ORBIT PLANE CHANGING

At some point in the mission design the orbital plane must change to accommodate a polar orbit around the Sun. In this section we will examine how to construct an analytical model for an orbit plane change using continuous thrust. T.N. Edelbaum examined the problem of a continuous power-limited thrust transfer between two inclined orbits in 1961 for a two body system (body and orbiter). A more recent reformulation of the problem, that provides a relatively straightforward method of analysis, was published in a number of papers by J.A. Kechichian. The results of an extensive and complete derivation (which can be found in Reformulation of Edelbaum's Low Thrust Transfer Problem using Optimal Control Theory [Kechichian, 1997]) are provided in this section.

After extensive derivation, an expression is found for the initial thrust yaw angle [Kechichian, 1997]

$$\tan \beta_0 = \frac{\sin \left( \frac{\pi}{2} \Delta i \right)}{\frac{v_0}{v_1} - \cos \left( \frac{\pi}{2} \Delta i \right)}$$

where

- $\Delta i$ is the desired change in inclination.
- $v_0$ is the initial circular velocity.
Design

\( v_1 \) is the final circular velocity.

This expression is based on the assumptions that the transfer occurs between two circular orbits in a two body system. In addition, a constant level of thrust is assumed in the derivation.

The total velocity change is given by [Kechichian, 1997]

\[
\Delta v = v_0 \cos \beta_0 - \frac{v_0 \sin \beta_0}{\tan \left( \frac{\pi}{2} \Delta i + \beta_0 \right)}
\]  \hspace{1cm} (3.79)

Because the engine acceleration is assumed to be constant we may write \( \Delta v = a_T t \), and so the total transfer time is [Kechichian, 1997]

\[
t = \frac{\Delta v}{a_T}
\]  \hspace{1cm} (3.80)

Further derivation reveals the thrust yaw angle as a function of time to be [Kechichian, 1997]

\[
\beta(t) = \tan \left( \frac{v_0 \sin \beta_0}{v_0 \cos \beta_0 - a_T t} \right)
\]  \hspace{1cm} (3.81)

where \( a_T \) is the acceleration the spacecraft experiences due to its engine. The velocity as a function of time is given by [Kechichian, 1997]

\[
v(t) = \sqrt{v_0^2 - 2v_0 a_T t \cos \beta_0 + a_T^2 t^2}
\]  \hspace{1cm} (3.82)

The inclination as a function of time is [Kechichian, 1997]

\[
\Delta i(t) = \frac{2}{\pi} \left( \tan \left( \frac{a_T t - v_0 \cos \beta_0}{v_0 \sin \beta_0} \right) + \frac{\pi}{2} - \beta_0 \right)
\]  \hspace{1cm} (3.83)

The evolution of the semi-major axis is easily obtained from

\[
a(t) = \frac{\mu}{v(t)^2}
\]  \hspace{1cm} (3.84)

The expulsed propellant mass can be determined using Tsiolkovsky’s equation [Wertz, et al., 1999]

\[
\Delta v = v_e \ln \frac{m_0}{m_1}
\]  \hspace{1cm} (3.85)

where

- \( m_0 \) is the vehicle mass at the beginning of the thrust period.
- \( m_1 \) is the vehicle mass at the end of the thrust period.
- \( v_e \) is the velocity of the exhaust stream.

This can be rewritten as

\[
\Delta v = g_0 l_s p \ln \frac{m_0}{m_1}
\]  \hspace{1cm} (3.86)
Finding the used propellant mass is then a simple matter; the end mass is defined as the starting mass minus the propellant mass.

\[ m_1 = m_o - m_p \] (3.87)

Substituting this into equation (3.86) and rewriting yields

\[ m_p = m_0 \left( 1 - e^{\frac{\Delta v}{g_p \mu_p}} \right) \] (3.88)

The above set of equations allows for a simple inspection of the propellant use and duration of an orbit plane change.

### 3.7.1 An Example Circle to Circle 90° Orbit Transfer

A test case is examined, where the spacecraft performs an orbital plane change of 90° (i.e. an inclination change) at varying distances from the Sun. The results of this analysis are given in a number of figures below (the first figure also provides the spacecraft parameters that were used). Figure 3.10 shows the propellant use of a 90° inclination change as a function of distance from the Sun.

![Figure 3.10 Propellant use as a function of distance from the Sun.](image)

It can clearly be seen that it is advantageous to perform the plane change away from the Sun in order to save more propellant. Figure 3.11 shows the evolution of the duration of the transfer.
This shows that it is beneficial for the transfer to take place closer to the Sun in order to minimize the time spent performing the maneuver. Thus, two design parameters are in contradiction and must be traded off against each other; does the designer desire a smaller propellant mass or a lower transfer duration? A design aid is the final mass over transfer duration factor, which is displayed in Figure 3.12.

When this value is maximal it can be said that the optimum point has been found. Obviously, this optimum at 0.13 AU is merely a theoretical optimum based only on mass and duration, and is much too close to the Sun to be a realistic solution.

It is important to note that distances closer to the Sun are improbable not only because the spacecraft is not designed to withstand the temperatures at these distances, but also because the engine simply cannot
convert all the available power into thrust (see Figure 3.5 above, where the thrust acceleration (and thus the thrust) is shown to go up to extremely high values close to the Sun). An electric propulsion module is usually only able to operate to a certain maximum thrust level, even if more power is available. For the specifications listed above, the thrust the spacecraft can achieve at 0.4 AU is 2 N, which is still within the realm of reality for an electric propulsion system. At 0.1 AU distance the maximum theoretical thrust is 30 N, which is impossible for an actual engine design. For this reason, it is interesting to look at the case where the power level is a constant 10 kW (roughly the same power level the solar panel specifications above would provide at 1 AU) providing an engine thrust of 0.31 N. An example of constant power level would be the use of a radioisotope thermoelectric generator as power source. The propellant use remains identical as it is independent of the thrust, but the behavior of the duration changes as shown in Figure 3.13 below.

\[ \text{Figure 3.13 Duration of the plane change transfer as a function of distance from the Sun for a fixed level of engine thrust.} \]

As can be seen the behavior of the transfer duration is now reversed for the case of a constant thrust setting of 0.31 N. This results in the following curve for the mass over duration, shown in Figure 3.14.
This clearly shows that the design becomes more optimal as the distance to the Sun increases (the power level is 10 kW regardless of distance away from the Sun, due to the radioisotope thermoelectric generator).

Let us now suppose that our spacecraft engine has a certain specified maximum thrust (2 N) it can deliver, and that it is powered by the earlier specified solar panels. The resultant thrust profile is shown in Figure 3.15 below.

The plot shows that the power to the propulsion system becomes ‘saturated’ at close to 0.4 AU distance, thus we are likely to see no further improvement in transfer duration going closer to the Sun. This intuition is confirmed when the transfer duration is plotted, shown in Figure 3.16.
Figure 3.16 Duration of the orbit plane change as a function of solar distance for an upper thrust limit of 2 N.

The plot clearly shows that the optimum transfer duration occurs at the distance the engine reaches its maximum achievable thrust level. The final mass over transfer duration factor evolution also reflects this same behavior, and is given in Figure 3.17 below.

Figure 3.17 Final mass over duration as a function of solar distance for an upper thrust limit of 2 N.

The plot shows that the optimum occurs at the moment the engine reaches maximum capacity at around 0.4 AU. If the specified maximum thrust were to be higher than the optimum in the previous two graphs, the optimum would shift to the left, while it would shift to the right if the maximum thrust level were lower.
Example Transfer

This section will show an example circle-to-circle transfer utilizing a single continuous maneuver starting from a circular orbit at a distance of 0.4 AU from the Sun. Figure 3.18 displays the evolution of the maneuver by plotting the yaw angle $\beta$, inclination $i$, velocity $v$, semi-major axis $a$, eccentricity $e$, perihelion $r_p$, aphelion $r_a$. The spacecraft parameters are also provided in the plot.

\[
\begin{align*}
\beta (^\circ) & \quad \text{t (day)} \\
i (^\circ) & \\
v (m/s) & \\
a (km) & \\
e (-) & \\
r_p (km) & \\
r_a (km) & \\
\end{align*}
\]

Input Power = 30 kW  
Thruster efficiency = 0.75  
Initial mass = 250 kg  
Specific Impulse = 5000 s

Figure 3.18 Evolution of selected orbital parameters during a circle to circle 90° inclination change.
Design

This transfer occurs under the assumption of a constant power supply of 30 kW to the spacecraft engine, translating to a thrust of just over 0.9 N. This level of thrust leads to a total transfer time of 280 days. It can be seen that the orbital elements (such as altitude at aphelion) change considerably during the transfer. This could have a significant effect on the amount of available engine power when using a solar powered spacecraft. To inspect this, the change in available thrust using solar power and the constant required thrust level should be computed. From the orbital elements we may calculate the position of the spacecraft during the transfer as

\[ r = \frac{h}{v} \]  

(3.89)

where the specific angular momentum \( h \) is constant and needs only to be calculated once using

\[ h = \sqrt{\mu p} = \sqrt{\mu a(1 - e^2)} \]  

(3.90)

where the semi-latus rectum \( p \) is quite easily calculated from the semi-major axis \( a \) and eccentricity \( e \). The resulting position of the spacecraft as a function of time is shown in Figure 3.19 below.

![Figure 3.19](image_url)  

\( r \) (km) \[ 2 \times 10^8 \]  

\( t \) (day) \[ 0, 50, 100, 150, 200, 250 \]

Figure 3.19 Spacecraft distance from Sun as a function of time in days.

Using this position we may determine the thrust along the transfer as a function of time, the result of which is shown in Figure 3.20.
Where the thrust is roughly 2 N at the initial circular orbit at 0.4 AU, the thrust is only 0.21 N at the maximum distance (~1.2 AU) from the Sun during the transfer, far below the necessary constant output of 0.9 N. Hence, this transfer is impossible to perform without greatly improving efficiency or increasing the area of the solar panels. It would be inefficient to size the spacecraft parameters based on this single continuous maneuver where the distance becomes so great, and thus an alternative must be sought where the distance from the Sun is somehow constrained. This can be approximated by modeling successive circle-to-circle orbit transfers, each with smaller individual changes in inclination. It is important to note that the previous analysis is performed on the basis that the thrust level is constant during the maneuver. During an actual transfer the distance from the Sun varies and so the thrust level may vary during transfer itself, this is accommodated by assuming the thrust to be continuously equal to the thrust at the greatest distance from the Sun.

### 3.7.2 Successive Transfers

By performing the same inclination change using a specified number of successive circle-to-circle transfers we may control some of the parameters during the total inclination change (at the cost of propellant mass). It would be expected that the total number of successive transfers would be of influence to the parameters we are interested in, e.g. a 90° circle-to-circle transfer split into 2 parts of 45° inclination change would differ from a transfer split into 9 parts of 10° inclination change. For each number of ‘partial transfers’ the total transfer duration, final mass, maximum aphelion, and the corresponding maximum thrust at this point may be determined and plotted. It should be noted that each partial transfer performs an equal change in inclination e.g. 2 parts contributing 45° each, 3 parts contributing 30° each, etc. When calculating a number of successive transfers, care must be taken to deduct expended mass to arrive at the correct initial mass for each separate part. The total duration of the 90° circle-to-circle (starting and ending at a distance of 0.4 AU) inclination change is plotted against the number of partial transfers in Figure 3.21 below.
Figure 3.21 Total transfer duration as a function of the number of successive individual circle-to-circle transfers.

It is clear that the transfer duration decreases as the number of parts increases. The reason for this is twofold, the first being that the transfer as a whole becomes more circular as the number of parts increases. Another factor beneficially influencing the transfer time is that the fact that the mass is lower for each part, thus increasing the thrust acceleration after each individual part. Notice that the transfer time for a single part is slightly over 130 days, a value which is reflected in Figure 3.11 when taking the transfer time value at 0.4 AU distance from the Sun. Figure 3.22 shows the evolution of the maximum distance the spacecraft will be from the Sun during the whole transfer.

Figure 3.22 Maximum distance as a function of the number of successive individual circle-to-circle transfers.

This distance should not be too large such that the thrust from a solar powered system would become inconsequential. The corresponding thrust at maximum distance is shown below in Figure 3.23.
As expected, the maximum achievable thrust is lower for a smaller number of partial transfers (i.e. greater transfer distances) because the total transfer is less circular and the distances from the Sun are larger (thus the solar panels provide less energy to the spacecraft engine). So far, all the parameters have improved with an increasing number of parts. Unfortunately, the whole transfer itself is less propellant efficient, as shown in Figure 3.24.

While only slightly more than 40 kg is left from the original 250 kg for a theoretical transfer that isn’t split into parts, the ratio quickly becomes even worse as the transfer is split into parts. The propellant mass quickly starts to encompass more than 90% of the total spacecraft mass, leaving little space for dry mass, let alone payload.
Example Transfer

This section will show an example circle-to-circle transfer utilizing 9 successive maneuvers starting from a circular orbit at a distance of 0.4 AU from the Sun. Figure 3.25 (shown below) displays the evolution of the transfer by plotting the yaw angle $\beta$, inclination $i$, velocity $v$, semi-major axis $a$, eccentricity $e$, perihelion $r_p$, aphelion $r_a$. The spacecraft parameters are also provided in the plot.

Figure 3.25 Evolution of selected orbital parameters for 9 successive circle-to-circle transfers of $10^\circ$ each.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solar panel area</td>
<td>$30 \text{ m}^2$</td>
</tr>
<tr>
<td>Solar panel efficiency</td>
<td>0.25</td>
</tr>
<tr>
<td>Thruster efficiency</td>
<td>0.75</td>
</tr>
<tr>
<td>Initial mass</td>
<td>250 kg</td>
</tr>
<tr>
<td>Specific Impulse</td>
<td>5000 s</td>
</tr>
</tbody>
</table>
Design

The individual circle-to-circle transfers can clearly be seen (and counted) because of the cyclic nature of the plots. The effort to constrain the distance has also been successful, as can be seen in Figure 3.26 (the maximum eccentricity is much smaller compared to the maximum eccentricity found in Figure 3.18), making the transfer feasible when using a solar powered spacecraft.

![Figure 3.26 Evolution of the distance from the Sun as a function of time for 9 successive circle-to-circle transfers of 10° each.](image)

The corresponding maximum achievable thrust is shown below in Figure 3.27.

![Figure 3.27 Evolution of the maximum achievable thrust as a function of time for 9 successive circle-to-circle transfers of 10° each.](image)

This shows that for 9 successive transfers the thrust varies only slightly, between 1.924 and 1.96 N. Dependent on the available thrust, the difference in inclination that needs to be achieved, and the desired time of transfer a judgment can be made regarding into how many separate parts the total inclination change should be divided.

The mission requires a polar orbit and thus some form of plane changing will have to take place in the design. Dependent on the available thrust, the desired time of transfer, and on the difference in inclination that needs to be achieved, a judgment can be made into how many separate parts the total inclination change should be divided. Edelbaum’s approach is quite accurate in determining the change in
vehicle mass. In the publication *Low Thrust Circle-to-Circle Orbit Transfer* [Gaylor, 2002] Edelbaum’s approach is compared to a more involving numerical solution. The study tests Edelbaum’s theory in three cases, and for all three gives the results that fall well within a margin of 5% error (both for transfer time and final mass/propellant mass).
Design
CHAPTER 4

NUMERICAL INTEGRATION

The software encounters two computational cases that require some form of numerical method to solve, the time of flight integral (equation (3.30)) and the propellant use expression in one of its two forms (either an integral under the assumption of constant spacecraft mass, cf. equation (3.43), or an initial value problem with decreasing mass, cf. equation (3.48)). The solution is thus approximated at various values along the transfer (named mesh points).

4.1 METHOD OVERVIEW

To solve the above two cases the software can currently make use of three methods: Euler’s Method, Composite Simpson’s Method, and the 4th order Runge-Kutta Method. Although the choice of method is ultimately up to the user, recommendations are given later on in this chapter. Should none of the included methods satisfy the user he or she can easily implement another method because of the modular nature of the software. What follows now is a general description of the currently employed methods, starting with Euler’s Method.

4.1.1 EULER

The most straightforward numerical method is Euler’s method. Although not often used in practice, it illustrates the principle of numerical integration quite well. Euler’s method is derived using Taylor’s Theorem, but is provided here without further derivation [Burden, et al., 2001].

\[ y(a) = y_0 \]
\[ y_{i+1} = y_i + hf(t_i, y_i) \] (4.1)

This method showcases the fundamental principle of any routine solving the initial value problem. The derivatives \( dy \) and \( dt \) are rewritten into finite steps of \( \Delta y \) and \( \Delta t \), multiplied by the step \( \Delta t \), and added to the current value of \( y \) to obtain the next value of \( y \). In other words, the basic idea is to add small increments to the function, where the increment is the derivative multiplied by the step size. The step size is the distance between each mesh point and can be written as [Burden, et al., 2001]

\[ h = \frac{b - a}{n} \] (4.2)
Numerical Integration

Where \( n \) is the total number of steps, and \( a \) and \( b \) are the lower and upper boundaries, respectively. This equation shows that the step size for Euler’s method is constant. It is also possible to vary the step size during the integration (see section 4.1.4). Generally speaking, decreasing the step size leads to greater accuracy, but longer computation time. Euler’s Method is a single step method, as it only uses knowledge of the single previous step to calculate the value of the current step.

Because numerical methods yield an approximate solution it is useful to say something about the difference between the exact and approximate solution. The local truncation error is [Burden, et al., 2001]

\[
\tau_{i+1}(h) = \frac{y_{i+1} - \tilde{y}_{i+1}}{h} \quad (4.3)
\]

where

- \( y_{i+1} \) is the exact solution at step \( i+1 \).
- \( \tilde{y}_{i+1} \) is the method’s approximation at step \( i+1 \).

This error is local because it measures the accuracy of the method at a specified step, assuming that the method was exact at the previous step. Using equation (4.3) it is possible to say something about the order size of a method’s local error. Using Taylor’s Theorem for some number \( \xi_i \) between \( t_i \) and \( t_{i+1} \) we arrive at

\[
y(t_{i+1}) = y(t_i) + h\dot{y}(t_i) + \frac{h^2}{2} \ddot{y}(\xi_i) \quad (4.4)
\]

This equation incidentally also shows us how Euler’s method is derived from Taylor’s Theorem. It merely uses the first two expansion terms, and disregards the rest. Thus, it is also possible to create higher order Taylor Methods simply by disregarding less expansion terms [Burden, et al., 2001].

By substituting equation (1) into (4.3) we can see that Euler’s method has a truncation error at the \( i \)th step of

\[
\tau_{i+1}(h) = \frac{y_{i+1} - y_i}{h} - f(t_i, y_i) \quad (4.5)
\]

Substituting the Taylor expansion from (4.4) into (4.5) shows

\[
\tau_{i+1}(h) = \frac{y_i + h\dot{y}_i + \frac{h^2}{2} \ddot{y}(\xi_i) - y_i}{h} - \frac{\dot{y}_i}{2} \ddot{y}(\xi_i) \quad (4.6)
\]

So the local truncation error in Euler’s method is of order size \( O(h) \) [Burden, et al., 2001]. Some texts have a different definition of the local truncation error, where they do not divide the error by the step size, as in equation (4.6). Thus, depending on definition, the local truncation error in Euler’s method can also be said to be of order size \( O(h^2) \) [Press, et al., 2007]. Euler’s method is said to be of \( 1^{st} \) order, because of its error order size. A method is of \( p^{th} \) order when the local truncation error is of order size \( O(h^p) \), or in the alternate definition \( O(h^{p+1}) \) [Press, et al., 2007]. A selection criterion for ODE solving methods is finding a method with a high-order local truncation error (i.e. large values of \( p \)), while keeping the number and complexity of the method’s calculations as low as possible [Burden, et al., 2001].
Euler’s Method is generally not considered to be a very attractive option, as it is neither very accurate nor very stable [Press, et al., 2007]. Nevertheless, it performs remarkably well for our purposes (and is easy to implement), as will be demonstrated in section 4.2.

### 4.1.2 Composite Simpson

Composite Simpson’s Rule is employed to find the time of flight. For a function \( f \in C^4[a,b] \), where the number of steps \( n \) is even, where the step size is defined as \( h = (b - a)/n \), and \( x_i = a + ih \), for each \( i = 0, 1, ..., n \) Composite Simpson’s Rule can be written for \( n \) subintervals as [Burden, et al., 2001]

\[
\int_a^b f(x)dx \approx \frac{h}{3} \left[ f(a) + 2 \sum_{i=1}^{n/2-1} f(x_{2i}) + 4 \sum_{i=1}^{n/2} f(x_{2i-1}) + f(b) \right]
\]

with its error term as [Burden, et al., 2001]

\[
E(f) = -\frac{b-a}{180} h^4 f^{(4)}(c)
\]

Previous experimentation in [Corradini, 2008] found a total number of steps of 400 to provide sufficient accuracy when performing the integration. This will be investigated in more detail in section 4.2.

### 4.1.3 Runge-Kutta

Runge-Kutta methods have the high-order local truncation error of Taylor methods, but eliminate the need to compute and evaluate the derivatives of the function \( f(t,y) \) (the function \( f \) itself is evaluated instead). Higher order Runge-Kutta methods are competitive with other methods. The 4th order method is very robust; it will almost always succeed [Press, et al., 2007]. In addition, it is usually the fastest method when function evaluations are computationally not too expensive and accuracy requirements are not too stringent \( (\leq 10^{-10}) \) [Press, et al., 2007].

There are numerous 2nd order Runge-Kutta methods: the Midpoint Method, the Modified Euler Method, and Heun’s Method. The best known used Runge-Kutta method is of 4th order and is given here as [Burden, et al., 2001]

\[
k_1 = hf(t_i, y_i)
\]

\[
k_2 = hf\left(t_i + \frac{h}{2}, y_i + \frac{1}{2} k_1\right)
\]

\[
k_3 = hf\left(t_i + \frac{h}{2}, y_i + \frac{1}{2} k_2\right)
\]

\[
k_4 = hf(t_{i+1}, y_i + k_3)
\]

\[
y_{i+1} = y_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)
\]

An interpretation of equation (4.9) is obtained by noting that the final line is Simpson’s Rule, which is a method for the approximation of a definite integral [Van Kan, 2001]. These expressions were developed around the end of the 19th century by the mathematicians Carle Runge and Wilhelm Kutta.
Numerical Integration

The 4th order Runge-Kutta method performs four derivative evaluations per step, once at the initial point, twice at trial midpoints, and once at the trial endpoint. From these 4 derivatives the final function value is computed (cf. Figure 4.1) [Press, et al., 2007].

![Fourth order Runge-Kutta method initial point, trial midpoints, and trial endpoint](image)

The higher order methods require more function evaluations than the lower order methods; the 2nd order methods require two function evaluations per step while the Runge-Kutta method of 4th order requires four function evaluations. There is a diminishing return in the relationship between the number of evaluations and the order of the local truncation error. For example, four function evaluations yield a best possible local truncation error of \(O(h^4)\), while a method using ten function evaluations gives a best possible local truncation error of \(O(h^7)\). This is generally why methods of order less than five with smaller step size are preferred to higher order methods with larger step size [Burden, et al., 2001].

The Runge-Kutta 4th order method requires four evaluations per step, while Euler’s Method requires a single evaluation per step. Consequently, if it is to be superior, the Runge-Kutta 4th order method should be more accurate when it has a step size four times larger than Euler’s method. The same comparison can be made with the second order Runge-Kutta methods, where the Runge-Kutta 4th order method should be more accurate with a two times larger step (because it requires twice the function evaluations per step). This is often the case, usually making the Runge-Kutta 4th order method a better choice when compared to Euler’s Method and 2nd order Runge-Kutta methods [Burden, et al., 2001].

4.1.4 Adaptive Step Size Control

Great gains in efficiency can be made by changing the step size during the integration. A smaller step size leads to greater accuracy but more function evaluations, leading to a longer computation time. The opposite holds for a larger step size, a smaller computation time but less accuracy. It would then be beneficial to somehow combine both positive properties in a single integration in order to profit from increased accuracy in less time. Varying the step size, by having many small steps in critical areas (such as periods of time in the integration where change of position and velocity occurs at a high pace) and greater steps in less critical areas (periods of time where the state vector rate of change is not so great), allows us to better balance accuracy and computation time.

Implementation of adaptive step size control is achieved by studying the local truncation error at each step, and adjusting step size as necessary. The simplest approach is step doubling, where a single step is taken twice. The step is fully taken once, and then taken a second time as two half steps. The difference
between both numerical estimates for the next point indicates the truncation error. The step size is then adjusted accordingly.

Far more efficient step size algorithms are embedded Runge-Kutta formulas. \( N \)th order Runge-Kutta methods above order size four require more than \( N \) function evaluations. This is why the classic 4th order Runge-Kutta methods is popular, anything of higher order will require relatively more function evaluations (e.g. 5th order requires 6 evaluations and 8th order 11 evaluations). Fehlburg found a 5th order method requiring six function evaluations, where a second combination of these six evaluations yields a fourth order method. The difference between the 4th and 5th order estimates can then be used to estimate the truncation error so that the step size can be adapted. In addition to Fehlburg’s method, there are numerous other modern embedded formulas.

The general form of a 5th order Runge-Kutta method is [Press, et al., 2007]

\[
\begin{align*}
\mathbf{k}_1 & = hf(t_i, y_i) \\
\mathbf{k}_2 & = hf(t_i + c_2 h, y_i + a_{21} \mathbf{k}_1) \\
\mathbf{k}_3 & = hf(t_i + c_3 h, y_i + a_{31} \mathbf{k}_1 + a_{32} \mathbf{k}_2) \\
\mathbf{k}_4 & = hf(t_i + c_4 h, y_i + a_{41} \mathbf{k}_1 + a_{42} \mathbf{k}_2 + a_{43} \mathbf{k}_3) \\
\mathbf{k}_5 & = hf(t_i + c_5 h, y_i + a_{51} \mathbf{k}_1 + a_{52} \mathbf{k}_2 + a_{53} \mathbf{k}_3 + a_{54} \mathbf{k}_4) \\
\mathbf{k}_6 & = hf(t_i + c_6 h, y_i + a_{61} \mathbf{k}_1 + a_{62} \mathbf{k}_2 + a_{63} \mathbf{k}_3 + a_{64} \mathbf{k}_4 + a_{65} \mathbf{k}_5)
\end{align*}
\]

\[
\text{and } y_{i+1} = y_i + b_1 \mathbf{k}_1 + b_2 \mathbf{k}_2 + b_3 \mathbf{k}_3 + b_4 \mathbf{k}_4 + b_5 \mathbf{k}_5 + b_6 \mathbf{k}_6
\]

The embedded 4th order formula is [Press, et al., 2007]

\[
y_{i+1}^* = y_i + b_1^* \mathbf{k}_1 + b_2^* \mathbf{k}_2 + b_3^* \mathbf{k}_3 + b_4^* \mathbf{k}_4 + b_5^* \mathbf{k}_5 + b_6^* \mathbf{k}_6
\]

The various constants are assigned values. Values found by Dormand and Prince are given in Table 4.1 below [Press, et al., 2007].

<table>
<thead>
<tr>
<th>( i )</th>
<th>( j )</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>( b_1 )</th>
<th>( b_1^* )</th>
<th>( c_1 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>35</td>
<td>5179</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>384</td>
<td>57600</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>384</td>
<td>3</td>
<td>2</td>
<td>9</td>
<td>10</td>
<td>16</td>
<td>1000</td>
<td>1113</td>
<td>16695</td>
<td>16</td>
</tr>
<tr>
<td>5</td>
<td>40</td>
<td>44</td>
<td>56</td>
<td>32</td>
<td>125</td>
<td>393</td>
<td>4</td>
<td>125</td>
<td>9</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>45</td>
<td>49</td>
<td>15</td>
<td>9</td>
<td>192</td>
<td>640</td>
<td>5</td>
<td>192</td>
<td>640</td>
<td>5</td>
</tr>
<tr>
<td>7</td>
<td>19372</td>
<td>19372</td>
<td>25360</td>
<td>64448</td>
<td>212</td>
<td>0</td>
<td>187</td>
<td>2187</td>
<td>92097</td>
<td>8</td>
</tr>
<tr>
<td>8</td>
<td>6561</td>
<td>6561</td>
<td>2187</td>
<td>6561</td>
<td>729</td>
<td>212</td>
<td>0</td>
<td>6784</td>
<td>339200</td>
<td>7</td>
</tr>
<tr>
<td>9</td>
<td>9017</td>
<td>9017</td>
<td>5355</td>
<td>46732</td>
<td>5103</td>
<td>0</td>
<td>2187</td>
<td>187</td>
<td>11</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>3168</td>
<td>3168</td>
<td>176</td>
<td>5247</td>
<td>176</td>
<td>176</td>
<td>115</td>
<td>87</td>
<td>2100</td>
<td>1</td>
</tr>
<tr>
<td>11</td>
<td>35</td>
<td>35</td>
<td>1113</td>
<td>500</td>
<td>125</td>
<td>2187</td>
<td>11</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 4.1 Dormand-Prince 5(4) Parameters for Embedded Runge-Kutta Method [Press, et al., 2007].

The table shows values going up to seven, where \( y_{i+1} \) itself provides the 7th stage. As it must be evaluated in order to start the next step, there is no additional computational cost. These values for the constants are preferred over the original Fehlburg values because they give a more efficient method, and better error properties [Press, et al., 2007].
The local truncation error estimate (which in this definition is not divided by the step size) is estimated as [Press, et al., 2007]

$$
\tau_{i+1}(h) = y_{i+1} - y_{i+1}^* = \sum_{i=1}^{6} (b_i - b_i^*) k_i
$$

(4.12)

To keep the error within desired bounds we state

$$
|\tau_{i+1}(h)| = |y_{i+1} - y_{i+1}^*| \leq \text{Scale}
$$

(4.13)

where

$$
\text{Scale} = \text{atol} + |y_i| \cdot \text{rtol}
$$

(4.14)

where

- \text{atol} is the absolute error tolerance vector.
- $|y_i|$ is the absolute of the component in the state vector.
- \text{rtol} is the relative error tolerance vector.

The component values in \text{atol} and \text{rtol} can be chosen separately for each component of $y$, but in practice one value is chosen for all components of the vector. Instead of using each separate element of the state vector, the component of absolute highest value of the state vector can also be used to compute the scale vector. The error tolerance vectors decide how stringent the requirements are. For example, \text{atol} and \text{rtol} could both be set to $10^{-6}$ to attempt a solution good to one part in $10^6$.

There are varying possibilities of how to handle the error. For example, it is possible to take the highest component in the error vector $\tau_{i+1}$ to define the error size. A second possibility is to take the Euclidean norm of all of the individual error vector components [Press, et al., 2007].

$$
\text{Error} = \sqrt{\frac{1}{n} \sum_{k=0}^{n-1} \left( \frac{\tau_k}{\text{Scale}_k} \right)^2}
$$

(4.15)

where

- $\tau_k$ is the $k^{th}$ element of the error vector $\tau_{i+1}$.
- $\text{Scale}_k$ is the $k^{th}$ element of the Scale vector.

The step is accepted when the error value is smaller than or equal to one ($\text{Error} \leq 1$). It can be seen from equations (4.10) and (4.11) that the error scales as $h^5$ to the step size. If a step $h_i$ is taken and an error of $\text{Error}_i$ is produced, then the step that would have given $\text{Error}_{i+1}$ is obtained from [Press, et al., 2007].

$$
h_{i+1} = S h_i \left[ \frac{\text{Error}_{i+1}}{\text{Error}_i} \right]^{\frac{1}{5}}
$$

(4.16)

$\text{Error}_{i+1}$ is the desired error, and of satisfactory size when equal to one. When the desired error is larger than one, equation (4.16) describes by how much the step size must be reduced to retry the failed step. If
on the other hand the desired error is smaller than one, information is obtained on how much the step size could be increased for the next step. An additional safety factor $S$ (smaller than 1) is added to the formula because the error estimates are not exact.

It is often shrewd to not let the step size increase or decrease too greatly per step. This can be implemented by not allowing the step to become bigger or smaller by a certain factor. A global maximum for the step size can also be implemented in order to avoid stepping past parts of the integration where smaller steps are necessary.

Unfortunately, at the time of writing Runge-Kutta method with adaptive step size control has not yet been properly tested in the software. Though the current fixed step methods already provide satisfactory performance, it would be interesting to see if worthwhile gains could be made using adaptive step size control.

4.2 Numerical Verification

We inspect the employed numerical methods by looking at a theoretical set of single leg transfers from Earth to Mars. The geometry is fixed by placing the Earth at 1 AU and Mars at 1.5 AU distance from the Sun. The angle between the two lines spanned by the Sun and the respective planets is fixed at 90°. The aspects of the numerical integration for both mass consumption and time of flight will be inspected in this section.

4.2.1 Spacecraft Mass Evolution

We will limit our discussion to four types of numerical quadrature:

- Euler’s Method (Constant Mass)
- Composite Simpson (Constant Mass)
- Euler’s Method (Variable Mass)
- 4th Order Runge-Kutta Method (Variable Mass)

The first two methods assume a constant mass during the transfer and the last two methods change the instantaneous spacecraft mass after each integration step (cf. Section 3.5.1 on Fuel Consumption, where the integral in equation (3.43) represents the assumption of constant mass and the ODE in equation (3.48) represents a more valid model that accounts for the decrease of mass during the transfer). The first two methods are implemented not only to allow for comparison with GALOMUSIT (where mass is assumed constant during a transfer and the integration is performed using Composite Simpson) when designing single leg transfers, but mostly to demonstrate the significant flaw this assumption introduces to the values of the mass variables. This flaw comes to the fore especially when the transfer is of great length (i.e. long time of flight) so we examine a test set of transfers with a total of five complete revolutions ($N = 5$) and a winding parameter $k_2$ of 0.5. The spacecraft parameters are fairly standard with a total initial mass of 2,000 kg, further details are irrelevant for the purposes of this example. The number of steps used for both integrators is 400 (stepping through the true anomaly $\theta$). Bear in mind that this example by no means represents an optimal set of transfers, it merely serves to illustrate the differences between integrators, and the difference between the assumption of constant spacecraft mass and decreasing spacecraft mass. Figure 4.2 shows the set of transfers (note that transfers that reach distances from the Sun
in excess of 5 AU are not shown) and the propellant use as a function of initial flight path angle for our theoretical geometry.

![Figure 4.2 The set of transfers (left) of the class S_{1/2}[1, 1.5, \pi/2, 5] and propellant use as a function of initial flight path angle (right).](image)

The numbers of the class S_{1/2}[1, 1.5, \pi/2, 5] entail that k_2 is equal to 1/2, the initial and final radii are 1 and 1.5, the transfer angle is \pi/2, and the number of complete revolutions is 5.

It is immediately obvious that there is a large difference between the two assumptions of constant and varying mass, where the varying mass model yields a far better result (less propellant used). This is natural, due to the fact that as the spacecraft mass decreases the engine must thrust less and less to achieve the necessary acceleration level, leading to an ever growing decrease in propellant use during the transfer.

The quadrature accuracy is inspected (for the case described as above) by varying the number of integration steps and examining the results. Euler’s Method is applied to find the total mass use for a number of 10, 50, 100, 250, 500, and 2,000 steps. This is shown below in Figure 4.3.

![Figure 4.3 Propellant use as a function of initial flight path angle using Euler’s Method for a variety of steps (k_2=0.5 and N=5).](image)

As commonsense would dictate, the 10 steps quadrature varies the most from the ‘reference’ 2,000 step integration. However, the differences between 50 and 2,000 steps already become quite small, at roughly less than 10 kg difference across the entire spectrum of the departure flight path angle. The reason the close up section is quite unsmooth along the initial flight path angle angle is due to the use of stepping for
Numerical Integration

the discretization of the initial flight path angle (note that this dictates the total number of transfers that are integrated separately, and not the number of steps for the actual integration of the individual transfers). Figure 4.4 below shows the comparison between Euler’s Method and the 4th order Runge-Kutta Method, both using 400 steps (stepping through the true anomaly \( \theta \)).

The difference between both methods is roughly 1 kg across the entire spectrum of initial flight path angles, and thus we demonstrate that a method as simple (and computationally cheap) as Euler’s Method performs remarkably well for the solution of equation (3.48).

We may obtain two points from the above discussion. Firstly, that it is sufficient to employ relatively simple quadrature methods such as Euler’s Method and the 4th order Runge-Kutta Method to describe changes in spacecraft mass along a given transfer leg. Secondly, the optimization process can be performed with a fairly small number of steps. Generally speaking (for all cases of \( k_2 \) and \( N \)), a change in the number of steps leads to a shift in the mass consumption curve and not a significant change to the actual shape, allowing use to safely reduce the number of steps. In the above example 50 steps would already suffice to ensure that the optimization process is able to find the most optimal transfer using the least propellant. After the optimization has been performed a more powerful integration, using a greater number of steps, can then be used to paint a more accurate picture of the propellant consumption during the transfer and to determine the final spacecraft mass.

Another factor that plays a role in preferring one method over another is the difference in computational cost per method. Using the same example, the integration is timed and stored by the system, and then averaged to acquire an integration time per transfer (the integration time is roughly the same per transfer; this is expected as the number of steps remains constant after all). The resulting computation cost per method is shown in Table 4.2.
Table 4.2 Overview of computational cost per numerical method for the integration of a single transfer†.

<table>
<thead>
<tr>
<th>Method</th>
<th>Computation Cost</th>
<th>Computation Time Factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Euler’s Method</td>
<td>0.00035 s</td>
<td>1</td>
</tr>
<tr>
<td>Composite Simpson (Constant Mass)</td>
<td>0.0004 s</td>
<td>1.14</td>
</tr>
<tr>
<td>$4^{th}$ order Runge-Kutta Method</td>
<td>0.001 s</td>
<td>2.86</td>
</tr>
</tbody>
</table>

To sum up, there is no real reason to use Composite Simpson, as it suffers from the assumption of constant mass during a transfer (it cannot solve the ODE that follows from the assumption of variable mass) and is also (barely) slower than Euler’s Method. The satisfactory accuracy and excellent computation speed make Euler’s Method (at 400 steps) an excellent candidate to solve every given mass ODE during the optimization. Following the optimization, the complete design can then be passed through once more thoroughly with a 2000 step $4^{th}$ order Runge-Kutta solution.

4.2.2 TIME OF FLIGHT

An example set of transfers is generated with zero complete revolutions ($N = 0$) and a winding parameter $k_2$ of 0.5. As mentioned above, the spacecraft parameters remain standard with a total initial mass of 2,000 kg (further details are irrelevant for the purposes of this example). Again, this example is not an optimal set of transfers, but a test to inspect and illustrate the differences between integrators. Figure 4.5 below shows the set of transfers, the time of flight evolution as a function of initial flight path angle (i.e. time of flight per individual transfer in the set) and a graphical representation of the optimality, where lighter shades correspond to a more optimal (in terms of propellant use) transfer.

![Graphical representation](image)

Figure 4.5 The set of transfers (left) of the class $S_{1/2}[1, 1.5, \pi/2, 0]$, time of flight as a function of initial flight path angle (middle), and a graphical representation of transfer optimality (right).

The expression for the time of flight, equation (3.30) in Section 3.4, is integrated using numerical quadrature for a single transfer within the set. Therefore, the time of flight curve in Figure 4.5 above represents a multitude of these integrations (one for each initial flight path angle value), each providing an answer for the time of flight. Using a number of numerical methods, we will explore the aspects of accuracy and computation speed. The numerical methods used to inspect the time of flight behavior are

- Euler’s Method
- Composite Simpson
- $4^{th}$ Order Runge-Kutta Method

† Computation performed on an Intel T9300 at 2.50 GHz.
The time of flight curve (central plot) of Figure 4.5 is plotted again in Figure 4.6, this time using several integrators (at 400 steps of the true anomaly $\theta$) shown in the legend.

Figure 4.6 Time of flight as a function of initial flight path angle, computed using 400 steps.

The difference between the integrators, except for Euler’s Method, is negligible (note that the black and green lines are indistinguishable in the figure as the resolution of the figure is too low to display the difference). The results obtained from Euler’s Method seem to show a roughly constant shift downwards of about 0.1 days (cf. expanded portion of the plot), which is also not an impressive discrepancy in itself. A ‘reference’ integration is performed with 2000 steps (of the true anomaly $\theta$) and the other methods are subtracted to inspect the error more closely, shown in Figure 4.7 below.

Figure 4.7 The difference in time of flight between the integrators and the reference integrator (2000 steps)

It is clear that the error of Euler’s Method is several orders of magnitude greater than that of Runge-Kutta and Composite Simpson, and that Runge-Kutta has the least variance in error (when compared to the 2000 step Composite Simpson integration).

Under the assumption that longer times of flight (i.e. longer integration periods) with the same number of steps will lead to larger errors, the number of total revolutions $N$ is increased to 5. Figure 4.2 (above) in the previous section shows the geometry of the transfer, and Figure 4.8 shows the curve for the time of flight as a function of initial flight path angle (cut off at $3.5 \times 10^4$ days) and the difference between the three integrators at 400 steps (of $\theta$) compared to the reference integrator (Composite Simpson at 2000 steps of $\theta$) by subtraction.
Figure 4.8 Time of flight as a function (top) and the difference in time of flight between the integrators and the reference integrator (bottom) as a function of initial flight path angle.

The magnitude of the error is now larger for all methods, but still relatively small when taking the much longer times of flight into consideration. As in the previous example (with $N = 0$) the 4th order Runge-Kutta method has the smallest error, especially around the outer values of the initial flight path angle.

Further increasing the total number of revolutions to $N = 100$ (with identical geometry) leads to the transfer set and time of flight curve shown below in Figure 4.9. The transfer set is shown twice, once with a graphical representation of 400 integration steps over the true anomaly $\theta$ (left) and with 2000 steps over $\theta$ (right), in order to illustrate that large jumps in position occur for $N = 100$ with only 400 steps.

Figure 4.9 The set of transfers (left) of the class $S_{1/2}[1, 1.5, \pi/2, 100]$ integrated at 400 steps, time of flight as a function of initial flight path angle (middle), and the same set of transfers integrated at 2000 steps (right).

The time of flight curve is monotone decreasing, where the lowest point on the curve (at $\gamma = 0.7871$ rad) is $2.3659 \times 10^4$ days. The error is examined by subtracting the integrated values found by the methods (using 400 steps) from a reference integration, in this case a 4th order Runge-Kutta Method (using 2000 steps).
Figure 4.10 The difference in time of flight between the integrators and the reference integrator as a function of initial flight path angle (left) and an enlarged area of the plot (right).

It is slightly surprising to now see Composite Simpson perform the worst (even when the 400 step methods are compared to a reference Composite Simpson 2000 step integration) of all three methods. The leftmost plot shows a significantly larger error for Composite Simpson, while the rightmost plot shows that the Composite Simpson error already diverges more quickly and strongly at 0 rad and that the error is also larger at most initial flight path angles. From -0.3 rad and outward, all three methods start to err considerably.

In order to properly judge which method is most suitable to solve the time of flight integral, equation (3.30), it is useful to have a look at the computation cost of a single function call (one single time of flight integration for a particular value of the initial flight path angle) for each particular method. The average computation cost per method is shown below in Table 4.3.

<table>
<thead>
<tr>
<th>Method (400 steps)</th>
<th>Computation Cost</th>
<th>Computation Time Factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Euler’s Method</td>
<td>0.00014 s</td>
<td>1</td>
</tr>
<tr>
<td>Composite Simpson</td>
<td>0.00016 s</td>
<td>1.14</td>
</tr>
<tr>
<td>4th order Runge-Kutta Method</td>
<td>0.00048 s</td>
<td>3.43</td>
</tr>
</tbody>
</table>

Table 4.3 Overview of computational cost per numerical method for the time of flight integration of a single transfer†.

Euler’s Method and Composite Simpson’s Rule have a very similar computational cost, only Runge-Kutta is substantially more expensive. However, the 4th order Runge-Kutta method has proven to be more accurate than the others.

Armed with the above knowledge we can select a method to perform the quadrature of the time of flight integral. In general, Euler’s Method is not preferred, as it is less accurate than Composite Simpson but not much faster. If one were to favor accuracy above else, then it would be natural to use 4th order Runge-Kutta. Composite Simpson performs quite well, being only slightly slower than Euler’s Method and quite accurate at finding the time of flight within most of the initial flight path angle spectrum (except at the outer edges, which are non optimal values in any case) at lower numbers of $N$ (complete revolutions). Since most designs involve a smaller number of $N$ anyway, Composite Simpson is a valid choice for solving the time of flight integral. Of course, should the user wish to do so the user can simply rewrite the module that performs the integration (no more than 30 lines of code) to implement a more advanced quadrature, perhaps involving adaptive step sizing. For the reader interested in a little history, see the inset below. Otherwise, skip straight towards the next section.

† Computation performed on an Intel Core i7 920 at 4 GHz.
A note on Simpson’s Rule and its varying names

In the field of numerical analysis Composite Simpson’s Rule is well known as a method for numerical integration. GALOMUSIT has an implementation of the method in its code that is similar to the definition found in most textbooks, but refers to it as the Composite Cavalieri-Simpson Rule.

Composite Simpson’s Rule is given here, as it is generally shown in the literature, as [Burden, et al., 2001]

\[
\int_{a}^{b} f(x) \, dx \approx \frac{h}{3} \left[ f(x_0) + 2 \sum_{i=1}^{n/2-1} f(x_{2i}) + 4 \sum_{i=1}^{n/2} f(x_{2i-1}) + f(x_n) \right]
\]

where

\[ x_i = a + i \cdot h \text{ for } x = 0, 1, ..., n - 1, \quad h = \frac{b - a}{n}, \quad x_0 = a, \text{ and } x_n = b. \]

Composite Cavalieri-Simpson as implemented in GALOMUSIT as

\[
\int_{a}^{b} f(x) \, dx \approx \frac{h}{6} \left[ f(x_0) + 2 \sum_{i=1}^{n-1} f(x_{2i}) + 4 \sum_{i=1}^{n} f(x_{2i-1}) + f(x_n) \right]
\]

where

\[ x_i = a + i \cdot \frac{h}{2} \text{ for } x = 0, 1, ..., 2n - 1, \quad h = \frac{b - a}{n}, \quad x_0 = a, \text{ and } x_n = b. \]

When comparing the two, it can be seen that the method implemented in GALOMUSIT is simply defined a little different, performing twice as many steps at half the original size (when \( n \) is defined the same for both cases). In other words, a 400 step Composite Simpson method essentially becomes an 800 step method in GALOMUSIT. The difference is just a matter of definition.

Interestingly enough, a search for ‘Composite Simpson-Cavalieri’ online yields only Italian sources, in which the method is shown to be exactly identical to Composite Simpson’s Rule (no implicit changing of the step size occurs). And so the two methods are essentially identical in the literature. Further research reveals that Bonaventura Cavalieri (a student of Galileo) used the method in 1635 in his publication \textit{Geometria indivisibilibus continuorum nova quadam ratione promota}, while Thomas Simpson employed the method in 1743 in his publication \textit{Mathematical Dissertation on a Variety of Physical and Analytical Subjects}. A Scottish mathematician, James Gregory, uses the method in \textit{Geometriae pars universalis}, published right at the end of his Italian visit in 1667. This is the first known publication in the English speaking world that includes this particular method of solving definite integrals.

It is apparent that the method was already well known to mathematicians before it was published and described by Simpson in 1743 with two known publications preceding his. The most likely reason that the method is generally referred to as Simpson’s Rule is that he introduced a number of variations and first popularized the method in accessible textbook format, causing others to coin it as ‘Simpson’s Rule’.

So was Cavalieri the first to discover this method? It actually turns out that Johannes Kepler had already invented the method previously (the story goes that he became interested in computing the contents of wine barrels, as he was left unsatisfied with the method the merchant employed from which he had bought the casks for his wedding). He published the method, called ‘Keplersche Fassregel’ (Kepler’s Barrel Rule) in his work \textit{Nova Stereometria doliorum vinariorum} published in Linz in 1615. As such, Simpson’s Rule is known as Keplersche Fassregel to German textbooks. Kepler’s work was well known,
and so it is extremely likely that Cavalieri had simply obtained the method from Kepler's previous work, and published it in Italy. Thus the current situation subsists, where the method is known as Simpson’s Rule in the English speaking world, Cavalieri (or sometimes Cavalieri-Simpson) in Italy, and Keplersche Fassregel in Germany. This situation aptly demonstrates the huge benefits a modern worldwide publication system provides, where any single document can be found online within a matter of minutes.

Finally, it is interesting to note that Simpson’s Method has a basis in Egyptian mathematics, demonstrated in the Golenischev Papyrus - a papyrus roll written in Egypt dated at around 1890 BC. The writings contain 25 problems and their solutions, one of them being the calculation of the volume of a truncated pyramid with a base of any shape. This is shown to be solved in the papyrus by the prismoidal formula, identical to the most basic form of Simpson’s Rule! So perhaps the ancient Egyptians deserve some of the credit for being the true inventors of Simpson’s Rule.
Optimization can be described as the process of selecting the best element out of a set of alternatives. In mathematics this translates to the location of extrema (either maxima or minima) of a function. A function $f$ is minimized with respect to $n$ design parameters $x$

$$\min_{x \in \mathbb{R}^n} f_0(x)$$  \hspace{1cm} (5.1)

subject to $m$ inequality constraints

$$f_i(x) \leq 0, \quad i = 1, 2, \ldots, m$$  \hspace{1cm} (5.2)

and subject to $k$ equality constraints

$$f_{m+i}(x) = 0, \quad i = 1, 2, \ldots, k$$  \hspace{1cm} (5.3)

The set of feasible $x$ is said to be the feasible region.

The optimization problem that is encountered in this report is a fitness function that can result in a very unsmooth and chaotic search space, depending on the number of design parameters. Unfortunately, this fitness function has no analytical derivative; forcing the use of numerically determined gradients for optimizers requiring derivatives. The problem is constrained in three ways: (1) the box constraints that give the design parameters limits (such as date of launch), (2) constraints within the model that can declare an individual invalid when violated, and (3) constraints placed upon the fitness by penalty functions. These penalty functions serve to make an individual become less attractive should this individual prescribe a design that has undesirable characteristics (e.g. excess arrival velocity).

Information regarding the possible fitness functions within the program is given in section 5.1. Section 5.2 provides a discussion of penalty functions. Finally, section 5.3 goes into more detail regarding the optimization methods that can be used in the software.

### 5.1 Fitness Function

As will be discussed in Chapter 6, the program seeks the optimal transfer given departure, arrival, and swing-by points within the box constraints specified by the user. This is achieved by an optimization
Optimization

process whose objective is to minimize a given fitness function. Currently the program supports two fitness functions. The first fitness function $f_1$ is

$$f_1(x) = m_{prop}$$

(5.4)

This is purely a minimization of the propellant mass, where less propellant mass coincides with a more optimal design. The second fitness function $f_2$ also takes the total time of flight, in addition to the mass, into account.

$$f_2(x) = -\frac{m_0 - m_{prop}}{t}$$

(5.5)

This fitness function strives to strike a balance between mass and time by minimizing both. This balance can be influenced to one side or another by multiplying elements in this fitness function with coefficients.

The two major components of the optimization process within the program are the determination of the propellant use for each individual leg of the transfer and the determination of the mass penalties for every swing-by that occurs. The time of flight itself is actually a design parameter (an input) in the optimization process. Further mass penalties can be incurred for mismatching departure and arrival velocities, and penalties can be incurred when the thrust exceeds engine limitations. These penalties can be manipulated fairly easily by the user, and even completely switched off. An example of this is not penalizing any departure velocity mismatch at the Earth as this $\Delta v$ may be provided by a boost or by the spacecraft engine itself, and might not steer the design towards an optimum the user intends. Another example is not enforcing any thrust penalties; a mass-optimal transfer is usually quite smooth and will tend to minimize engine thrust already. Additionally, a more improved design could lie slightly out of the region of the specified spacecraft parameters. By changing the design of the spacecraft itself (by adding an extra square meter of solar panel for instance) the user might arrive at a generally superior design. There is always some amount of trade-off to a design, and so there is never one inherently best solution. The process of reaching a good solution requires the insight of the user; the software is merely a tool to aid the user in their judgment.

5.2 PENALTY FUNCTIONS

At the start and end of a transfer the excess velocities may be examined and the design punished for having them. With the knowledge of the characteristics of two individual legs of the transfer, a number of swing-by parameters may be calculated; these can be translated into a penalty on angle and a penalty on velocity mismatches. A design with a thrust profile that cannot properly be flown may also be penalized accordingly. Finally, there is also a rejection penalty in the case of a design violating a hard constraint. Section 6.3 provides more fully detailed information on how these various penalties are determined in the process of computing a single individual. This section limits itself to the discussion of their manipulation and implementation into the optimization process itself.

With the introduction of penalties the first fitness function $f_1$ becomes

$$f_1(x) = m_{prop} + m_{pen}$$

(5.6)
This is purely a minimization of the propellant mass, where less propellant mass coincides with a more optimal design. The second fitness function $f_2$ also takes the total time of flight, in addition to the mass, into account.

$$f_2(x) = - \frac{m_0 - (m_{\text{prop}} + m_{\text{pen}})}{t}$$

(5.7)

The total penalty mass value is comprised of the sum of all the individual penalties, such as the swing-by velocity and angle penalties $m_{\text{sb,v}}$ and $m_{\text{sb,a}}$, the departure and arrival penalties $m_{\text{dep}}$ and $m_{\text{arr}}$, the thrust penalty $m_{\text{thr}}$, and finally, the rejection penalty $m_{\text{rej}}$.

$$m_{\text{pen}} = m_{\text{sb,v}} + m_{\text{sb,a}} + m_{\text{dep}} + m_{\text{arr}} + m_{\text{thr}} + m_{\text{rej}}$$

(5.8)

These mass values are based on the manipulation of the variables discovered in section 6.3. This means that all penalties in the program are expressed in terms of mass, and so a velocity is turned into a representative mass in some manner.

In principle one may choose any shape of function to manipulate a parameter $x$ such that it is translated into a penalty value, in our case a fictional penalty mass. Figure 5.1 below shows some examples of the form penalty functions can take.

![Figure 5.1 Example penalty functions where $x = 0$ is considered to be optimal.](image)

What penalty function is most suitable for a particular problem is difficult to determine and often requires some amount of trial and error. The choice of penalty function can heavily incline the design towards a particular direction. This is not inherently a bad thing, as long as the user is aware of this fact. For example, one may want to impose an extremely heavy penalty on arrival velocity to achieve a design that performs well in that regard. Penalty functions should be treated with caution; they are a powerful tool when used correctly, but it can be tricky to balance multiple penalties.

**Swing-by Deflection Angle Penalty**

$$f(x) = c$$

(5.9)

A hard constant penalty is imposed upon the design when the deflection angle exceeds its maximum at a swing-by. A rule of thumb would be to set this to roughly twice the initial mass of the spacecraft, this will cause the optimization to do its best to stay away from swing-by angle changes that are not possible while still allowing the process to search effectively. Setting this penalty to infinity is not recommended as it would actually hinder the optimization’s convergence because the optimizer then has no way to compare different individuals that both violate this constraint. This affects the convergence behavior of the optimization.
Optimization

Swing-by Velocity Mismatch Penalty

\[ f(x) = \begin{cases} \alpha|x|, & |x| < c \\ \beta|x|, & |x| \geq c \end{cases} \] (5.10)

It is almost impossible to exactly match the entrance and exit velocities for a swing-by. However, negligible differences can be attained by inflicting quite a severe penalty for smaller mismatches, and a much heavier one if the velocity mismatch is greater. For velocity disparities beneath 1 km/s the difference in velocity is multiplied by 100, and for disparities above 1 km/s the difference is multiplied by a factor of 1,000 (i.e. \( a = 100, b = 1000, c = 1\text{km/s} \)). This penalty shape seems to work well in producing a design with multiple swing-bys that all more or less match the entrance and exit velocities with a small degree of error.

Excess Departure Velocity Penalty

\[ f(x) = c|x| \] (5.11)

The departure velocity from the initial object can be suitably penalized with a linear function. This is often set to a low value because it is usually deemed more important to match the final arrival velocity. To some degree, the departure velocity and arrival velocity are linked to each other and placing greater emphasis on one generally comes at a cost to the other.

Excess Arrival Velocity Penalty

\[ f(x) = c|x| \] (5.12)

This can be set to about a quarter of the initial mass to encourage the optimizer to minimize the excess in arrival velocity quite successfully. As stated above, there is a relation between the initial departure and final arrival excesses, and the user must decide which factor weighs heavier. For a spacecraft with a continuous low thrust propulsion system the final arrival velocity is more important. While the spacecraft must be captured at arrival and can’t perform any near-instantaneous changes in velocity, the discrepancy in departure velocity can be solved by slowly spiraling the spacecraft outwards or by implementing a boost from a chemical rocket stage/module.

Thrust Penalty

\[ f(x) = c|x| \] (5.13)

The parameter for the fitness function here is the area between the required thrust and available thrust lines when the former exceeds the latter (for more detail see section 6.3.4). This area is very small as it is typically calculating with very small acceleration values. For this reason a coefficient of 1,000,000 serves well in pressuring the optimizer to find a transfer that fulfils this condition. Note that implementing this penalty slows down the optimization process considerably. Because of the fact that the optimization process is not hugely lengthy in itself this is not a great problem. Nevertheless, one can often do without this penalty as a fuel-optimal design will often automatically reduce the acceleration requirements as a secondary effect.
Rejection Penalty

\[ f(x) = \infty \]  

As discussed in Chapter 6 this penalty only occurs when something unforeseen happens during the calculation of one individual (as a sort of safety feature to keep the optimization running) and when the set of design parameters simply creates an impossible transfer leg.

The above penalties can be combined in any way, and their individual strength can be quickly modified to arrive at a transfer that has good characteristics in the areas the user wishes. It can often be the case that making significant improvements in one particular area (i.e. design parameter) of the design comes at the cost of other parameters of the design.

5.3 Employed Optimization Methods

This section provides an overview (see list below) plus detailed description of the various optimization methods used in the software.

- **Line Search**: Used to find the initial flight path angle (corresponding to the specified time of flight) during the computation of an individual within the optimization.
- **Gradient Descent**: Used to find the minimum of a bathtub time of flight curve during computation of an individual.
- **Monte Carlo**: Optional initial design optimization. Results can be imported to form the initial population of the Genetic Algorithm.
- **Grid Search**: Nested do loop that could in theory be used as an optional optimizer, actually used to make plots of the search space.
- **Genetic Algorithm**: Main design optimizer that attempts to find the best possible individual within a user specified search space.
- **Nelder-Mead Method**: Local optimizer that relies solely on function evaluations to further improve the design, once the Genetic Algorithm has completed its search.

The main optimizers that the user will immediately notice are the Monte Carlo, Genetic Algorithm, and Nelder-Mead Method optimizers. The grid search is implemented separately as a method for plotting the search space while the gradient descent and line search routines are used during the computation of individuals and thus go unnoticed to the user.

The general optimization process is roughly as follows; Monte Carlo is implemented by computing a number (set by the user) of individuals to obtain a random sample. The best individuals of this sample are stored and imported as the initial population of individuals for the Genetic Algorithm (this number depends on the user set population size of the Genetic Algorithm). Following this, the Nelder-Mead Method takes the end result from the Genetic Algorithm (the fitness value and corresponding values for the design parameters) and initiates a local search for a better solution.

In principle the user could suffice with only using the Genetic Algorithm and omitting Monte Carlo and the Nelder-Mead Method, and for many simpler problems (such as a single-leg transfer) this works quite well. For more complex problems however, especially the Nelder-Mead Method can significantly improve the result at very low computational cost (this is very dependent on the settings of the Genetic
Optimization

The extent of this improvement is often dependent on the settings of the Genetic Algorithm (the user can make it more rigorous and as a result computationally expensive, or vice versa) but it is always worthwhile to run the Nelder-Mead Method as it has the potential to considerably improve the end result and computation time is quick (in the order of minutes at most).

For complex problems Monte Carlo can be used to gain a good initial population for the Genetic Algorithm. For a simple problem, such as a single-leg transfer with only 4 design parameters an initial Monte Carlo optimization can be safely omitted. Nevertheless, experimentation has shown that it is beneficial as an initial optimization and search space reduction for more complex problems.

5.3.1 LINE SEARCH, GRADIENT DESCENT

During the optimization process the software must often find the correct initial flight path angle that corresponds to the input time of flight parameter. This initial flight path angle is found using a simple line search method. See section 6.2 for further details regarding this process. Step 4 (in section 6.2) discusses how to determine which shape the time of flight curve is. Once this shape (monotone increasing, monotone decreasing, or bathtub) has been established the line search is begun. The line search algorithm is dependent on the shape of the curve.

The whole process will now be explained for the case of a monotone increasing curve. First a check is made whether the specified time of flight lies within the extreme values to ensure validity. Then, a start is made at the minimum initial flight path angle. From this minima a step is taken through precisely half of the valid initial flight path angle region, defined as

\[ \gamma_{\text{test}} = \frac{\gamma_{\text{max}} - \gamma_{\text{min}}}{2} + \gamma_{\text{min}} \]  \hspace{1cm} (5.15)

The time of flight corresponding to this test point is evaluated. Should this time of flight be greater, the test point is moved half of the initial step upwards (towards the maximum initial flight path angle), and should it be smaller, the test point is moved half of the initial step downwards (towards the minimum initial flight path angle). The time of flight corresponding to the test point is then evaluated again and it is determined whether the test point should move another half of the previous step down or up the curve. After a number of test points the value of the time of flight should be close to the desired value (the margin can be specified) and the algorithm is stopped. This process is shown graphically in Figure 5.2 below.

![Figure 5.2 Line search for the initial flight path angle corresponding to the specified time of flight.](image-url)
Figure 5.2 shows the initial step from position 1 to 2, which is defined as half the total initial flight path angle region. Because the time of flight at position 2 is smaller than the specified value the next step is further up at 3. The step size is half the previous step size. As the time of flight is now greater the next step from 3 to 4 is to the left again (the step size is half the previous step size). One final step is taken to the right, from 4 to 5, to arrive at a value that lies within the error margin. In pseudo code the general process for a monotone increasing line can be written as

\[
\gamma_1_{\text{length}} = \gamma_1_{\text{max}} - \gamma_1_{\text{min}} \\
\gamma_1_{\text{test}} = \gamma_1_{\text{min}} \\
\text{TOF}_{\text{test}} = \text{TOF}_{\text{left}} \\
\text{while} \ abs(\text{TOF}_{\text{test}} - \text{TOF}_{\text{goal}}) > \text{day}\_\text{margin} \\
\hspace{1em} \gamma_1_{\text{length}} = 0.5 \times \gamma_1_{\text{length}}; \\
\hspace{1em} \text{if} \ \text{TOF}_{\text{test}} > \text{TOF}_{\text{goal}} \\
\hspace{2em} \gamma_1_{\text{test}} = -\gamma_1_{\text{length}} + \gamma_1_{\text{test}} \\
\hspace{2em} \text{TOF}_{\text{test}} = \text{func}\_\text{tof}(\text{various parameters, } \gamma_1_{\text{test}}) \\
\hspace{1em} \text{else} \\
\hspace{2em} \gamma_1_{\text{test}} = \gamma_1_{\text{length}} + \gamma_1_{\text{test}} \\
\hspace{2em} \text{TOF}_{\text{test}} = \text{func}\_\text{tof}(\text{various parameters, } \gamma_1_{\text{test}}) \\
\text{end} \\
\text{end}
\]

This same process needs only to be slightly altered for a monotone decreasing curve.

In the case of a bathtub curve the area to search is split up into two regions, where the center point connecting the two regions is the minimum. The left region is then treated as a monotone decreasing curve while the right region is treated as being a monotone increasing curve. The location of this minimum point itself is necessary in both cases for the line search to commence successfully. The minimum is found by using a simple gradient descent method, which can be defined as

\[
x_{n+1} = x_n - h \nabla F(x_n) \\
(5.16)
\]

For a single parameter (initial flight path angle) we may rewrite this to

\[
\gamma_{1n+1} = \gamma_{1n} - h \frac{\text{tof}_{\gamma_{1n}}(+\delta) - \text{tof}_{\gamma_{1n}}(-\delta)}{2\delta} \\
(5.17)
\]

In this case the gradient is approximated by determining the time of flight twice near the test point, once to the left (by a distance of \(-\delta\)) and once to the right (by a distance of \(-\delta\)), and then dividing this by the covered flight path angle region, \(2\delta\). The pseudo code for this process is

\[
\gamma_1_{\text{length}} = \gamma_1_{\text{max}} - \gamma_1_{\text{min}} \quad \% \ Distance \ between \ \gamma_{1\text{max}} \ and \ \gamma_{1\text{min}} \ . \\
\gamma_1_{\text{old}} = \gamma_1_{\text{min}} \quad \% \ Initializing \ parameters. \\
\gamma_1_{\text{new}} = 0.5 \times (\gamma_1_{\text{max}} - \gamma_1_{\text{min}}) + \gamma_1_{\text{min}} \quad \% \ Defining \ the \ step \ size. \\
h = \gamma_1_{\text{length}}/1E6 \quad \% \ Defining \ the \ precision. \\
\text{precision} = 1E-4 \quad \% \ Finding \ the \ initial \ time \ of \ flight. \\
\text{TOF}_{\text{new}} = \text{func}\_\text{tof}(\text{various parameters, } \gamma_1_{\text{new}}) \quad \% \ Distance \ between \ \gamma_{1\text{max}} \ and \ \gamma_{1\text{min}} \ . \\
\text{while} \ abs(\gamma_1_{\text{new}} - \gamma_1_{\text{old}}) > \text{precision} \quad \% \ \gamma_1 \ refreshed. \\
\hspace{1em} \gamma_1_{\text{old}} = \gamma_1_{\text{new}} \\
\hspace{1em} \text{TOF}_{\text{new}\_\text{left}} = \text{func}\_\text{tof}(\text{various parameters, } \gamma_1_{\text{new}} - 0.5 \times h) \quad \% \ Left \ point \ for \ gradient. \\
\hspace{1em} \text{TOF}_{\text{new}\_\text{right}} = \text{func}\_\text{tof}(\text{various parameters, } \gamma_1_{\text{new}} + 0.5 \times h) \quad \% \ Right \ point \ for \ gradient. \\
\hspace{1em} \text{TOF}_{\text{new}\_\text{prime}} = (\text{TOF}_{\text{new}\_\text{right}} - \text{TOF}_{\text{new}\_\text{left}})/h \quad \% \ Gradient \ TOF/\gamma_1 \ .
\]
\[
gamma_1_{\text{new}} = \gamma_1_{\text{new}} - h \cdot \text{TOF}_{\text{new}}' \\
\text{TOF}_{\text{new}} = \text{func}_{\text{tof}}(\text{various parameters,} \gamma_1_{\text{new}}) \% \text{Corresponding TOF value}
\]

After a number of steps the minimum of the curve is found to a certain degree of precision, this minimum then splits this curve into a monotone decreasing and a monotone increasing curve that are separately handled by the algorithm to find two initial flight path angles that correspond to a single value for the time of flight.

5.3.2 MONTE CARLO

The Monte Carlo method relies on repeated random sampling of the search space to reach an optimum. A number of samples are taken in the search domain using some form of random distribution (for example, a uniform distribution). Searching for the global optimum by only employing this method is ill advised as the chance of a good solution increases with the sample size, making this strategy computationally expensive. Monte Carlo’s implementation in this program is to allow the user to set a fixed amount of function evaluations. The best candidates from this fixed amount are then stored and used as an initial population for further optimization using a Genetic Algorithm (and so the number of best candidates stored is equal to the user set population of the Genetic Algorithm). For example, the user could instruct the program to perform 10,000 function evaluations using random sampling of the search space, the program then stores the 100 best individuals it finds and the Genetic Algorithm is started with a population size of 100. The user can also inspect the results of the Monte Carlo optimization, identify areas of interest, and alter the lower and upper search boundaries in order to reduce the search space.

5.3.3 GRID SEARCH

Grid search refers to the method of proceeding systematically through a grid of set points in the search space. At each point the objective function is evaluated and stored. If it is the best value that has been found so far, the previous point is then discarded. After the entire search space has been stepped through a point that lies close to the optimum has hopefully been found. The resolution and computation time are naturally dependent on the step size for this process. As one can most likely deduce, this form of optimization is not particularly computationally effective. The program contains a modified version of this procedure, where the defining difference is the storage of all points in the grid. Therefore, this process is purely for plotting purposes of the search space. The user can set the search space area, and select the variables on the axes, for plotting (usually around an optimum).

5.3.4 GENETIC ALGORITHM

The Genetic Algorithm belongs to the group of optimizers named evolutionary algorithms. Genetic algorithms are used to perform a wide exploration of the search space (i.e. a global search), finding promising areas is ensured by means of mutation, crossover, and selection operators. The main differences between a typical derivative-based algorithm and Genetic Algorithm are twofold, a Genetic Algorithm creates a population of points per iteration instead of just one point, and a Genetic Algorithm creates the next population by using random number generators instead of a deterministic computation.

The basic premise of a Genetic Algorithm entails the creation of an initial population of individuals. Each individual contains values for the collection of design parameters \( x \). The objective function \( f(x) \) is
evaluated for each individual, and a fitness value (the individual’s optimality) is assigned accordingly. Now an iterative process is begun to improve the fitness of the individuals in the population. First, by some method of selection, individuals are selected for manipulation. Their offspring is bred through genetic operations such as crossover and mutation. The fitness of the new generation of individuals is evaluated, members are selected for operation and so on. Each new generation is manipulated by genetic operators to allow the system to converge.

**Basic Terminology**

Throughout the report one can find mention of terms such as fitness, design parameter, etc. This short section aims to elucidate the reader by providing some relevant basic terminology that is used to describe optimization, and in particular the working of a Genetic Algorithm. Those that are familiar with this terminology can skip straight past this section.

As mentioned previously, the *fitness function* is simply the function the user wishes to optimize. This is often referred to as the objective function for any optimization algorithm, and referred to as the fitness function in conjunction with a Genetic Algorithm.

The *design parameters*, or simply *variables*, are those variables on which the fitness function is dependent. For example, the function $f(x,y)$ depends on two design parameters $x$ and $y$.

The *search space* is defined as the area in which the design parameters are of valid value. Continuing with the above example the search space could be defined by $0<x<1$ and $0<y<1$, allowing both design parameters to be between 0 and 1. The minimum values the design parameters are said to be the *lower boundaries* and the maximum values are said to be the *upper boundaries*.

An *individual* is a collection of values of the design parameters that together form a point (within the search space) that can be applied to the fitness function. For example, an individual may be at $x=0.5$ and $y=0.7$.

A group of individuals constitute a *population*. Each individual in the population carries its own values for the design parameters (although nothing stops the population from containing two identical individuals by chance). At each iteration, the Genetic Algorithm performs computations to replace the current population by a new (and hopefully improved) population. Each successive population is referred to as a new *generation*.

The *fitness value* of an individual is the value of fitness function. Because we are dealing with a minimization problem lower values are better. The *best fitness* value of a population is thus the smallest fitness value of any individual within the population.

In creating a new generation, the Genetic Algorithm selects individuals from the current generation – named *parents* – and uses these to create new individuals – named *children* – for the next generation. Generally speaking, the algorithm will tend towards selecting individuals with better fitness value to be parents.

Finally, *diversity* refers to the average distance between individuals in a population. A population has high diversity if the average distance is large and low diversity when it is small. Diversity is important for the
optimization process because it allows the algorithm to more effectively search the entire search space, and not fall into the trap of a local optimum.

**Outline of the algorithm**

The following summary provides an overview of the Genetic Algorithm.

1. Algorithm begins with an initial population, this can be randomly created, created from a grid of the search space, or obtained from a Monte Carlo selection of best candidates found.
2. The algorithm creates a succession of new populations. At each step individuals are used in the current population to create the next population. To create a new population, the following steps are performed:
   a. Each member is ranked by evaluating the fitness function.
   b. The fitness values are scaled in order to convert them to standard range of values.
   c. Members (named parents) are selected based on their fitness.
   d. Some of the individuals in the current population can be passed on to next population. These individuals are named elite.
   e. Children are produced from the parents in one of two ways; either by making random changes to a single parent – mutation – or by combining vector entries of a pair of parents – crossover.
   f. The current population is replaced with the children to form the next population.
3. The algorithm stops once one of the stopping criteria have been met, these criteria are described below.

**Stopping Conditions**

The Genetic Algorithm uses the following conditions to determine when to stop [The Mathworks, 2010]:

- **Generations** The algorithm stops when it reaches a specified number of generations.
- **Time Limit** The algorithm stops after running for a specified amount of time.
- **Fitness Limit** The algorithm stops when the fitness function value of the best point in the current population is less than or equal to the specified fitness limit.
- **Stall Generations** The algorithm stops when the weighted average change in the fitness function value over the specified number of generations is less than the function tolerance.
- **Stall Time Limit** The algorithm stops is there is no improvement in the objective function during an interval of time specified by the user.
- **Function Tolerance** The algorithm runs until the weighted average change in the fitness function value over the specified number of stall generations is less than the function tolerance.

The process is displayed graphically as a flowchart in Figure 5.3. The breeding process itself is a stochastic process; it is not guaranteed that new individuals will possess a higher fitness. However, the probability of fitter individuals is higher, allowing the system to converge to an optimum (hopefully the global optimum) [Goldberg, 1989].
Now, the process of a Genetic Algorithm is described in more detail.

**Initial Population**

An initial population can be created in a number of ways. The user may select a number of options. The default setting (often used in this report to create results) is the creation of a random initial population with a uniform distribution. There is also a second setting which allows only feasible candidates to be in the initial population (these individuals satisfy all bound and linear constraints). Finally, there is also the possibility of sampling (with a uniform distribution) the search space a number of times (specified by the user) using the Monte Carlo function. The best candidates found from this sampling are then fed into the Genetic Algorithm as the initial population.

The size of the population influences the workings of the algorithm. A small population will cause the algorithm to converge more quickly, but the simulation will have a higher chance of stopping at a local optimum (instead of another better local optimum or global optimum). A large population will come at the price of higher computational cost (as the rate of change is slower), but will give the algorithm a higher chance to find the global optimum. Depending on the problem, and on the wishes of the user, a suitable population size should be chosen. The population remains constant throughout the optimization process.

**Creating the next Generation**

At each step the algorithm uses the current population to create the children that will make up the next generation. The algorithm selects individuals in the current population as parents who contribute their genes to their children. The algorithm will usually select individuals that have higher fitness values as parents. The selection process itself has a number of options, described further below under its own subsection.

---

An algorithm that falls too quickly into the trap of a local optimum is often referred to as a ‘greedy algorithm’.
The Genetic Algorithm creates three types of children for the next generation:

- **Elite children** are the individuals in the current generation with the best fitness values. They automatically survive through to the next generation.
- **Crossover children** are created by combining the vectors of a pair of parents.
- **Mutation children** are created by introducing random changes, or mutations, to a single parent.

The user can specify how many of each type of child is created by setting two parameters; (1) the number of elite children can be set by the user i.e. **elite count**, and (2) the user can also sets a **crossover fraction** value. As an example if the population is set to 100, the elite count to 2, and the crossover fraction to 0.8, then the two best individuals will be carried on over unchanged to the next generation, and there will be 98 positions left available for mutation or crossover. The algorithm rounds 98·0.8 = 78.4 to 78 to arrive at the number of crossover children. The remaining 20 individuals are mutation children.

**Elitism**

Saving the best individuals (or single best individual) and allowing them to carry over unaltered to the next generation is referred to as elitism. Elitism increases the speed of algorithm (the population converges in fewer generations, thus faster) but at the cost of robustness, there is a higher chance that the algorithm will terminate at a local optimum.

**Crossover**

The user can set crossover options to specify how the algorithm combines parents to form a crossover child for the next generation. Crossover is usually the primary means of creating new offspring (with new chromosomes) from the parents. The user can choose from a number of crossover functions, listed below.

1. **Scattered** (the default crossover function) creates a random binary vector and selects the genes from the first parent where the vector is a 1, and the genes from the second parent where the vector is a 0. Subsequently, these genes are combined to form the child. For example, if \( p_1 \) and \( p_2 \) are the parents

\[
p_1 = [10\ 11\ 12\ 13\ 14\ 15\ 16\ 17]
\]

\[
p_2 = [20\ 21\ 22\ 23\ 24\ 25\ 26\ 27]
\]

and the binary vector is [1 0 1 0 1 0 1] the function would return the following child \( c_1 \)

\[
c_1 = [10\ 21\ 12\ 23\ 24\ 15\ 26\ 17]
\]

2. Another option is **Single point** chooses a random integer \( n \) between 1 and the number of design parameters and then selects vector entries numbered less than or equal to \( n \) from the first parent, and selects vector entries greater than \( n \) from the second parent. These entries are then brought together to form a child. For example, if \( p_1 \) and \( p_2 \) are the parents

\[
p_1 = [10\ 11\ 12\ 13\ 14\ 15\ 16\ 17]
\]

...
Optimization

\[ p_2 = [20 \ 21 \ 22 \ 23 \ 24 \ 25 \ 26 \ 27] \]

and the crossover point is at 4 then the following child is created

\[ c_1 = [10 \ 11 \ 12 \ 13 \ 24 \ 25 \ 26 \ 27] \]

3. **Two point** is another option, which selects two random integers \( m \) and \( n \) between 1 and the total number of design parameters. The function selects vector entries numbered less than or equal to \( m \) from the first parent, vector entries numbered from \( m+1 \) to \( n \) from the second parent, and vector entries greater than \( n \) from the first parent again. The genes are then combined to form a single child. As an example, if \( p_1 \) and \( p_2 \) are the parents

\[ p_1 = [10 \ 11 \ 12 \ 13 \ 14 \ 15 \ 16 \ 17] \]
\[ p_2 = [20 \ 21 \ 22 \ 23 \ 24 \ 25 \ 26 \ 27] \]

and \( m=2 \) and \( n=5 \) then the following child is created

\[ c_1 = [10 \ 11 \ 22 \ 23 \ 24 \ 15 \ 16 \ 17] \]

4. **Intermediate** creates children by taking a weighted average of the parents. The weights can be specified with a single parameter \( R \) (for Ratio), which is a vector of the same length as the design parameters (the default is a vector containing just 1’s). The child is created from parents \( p_1 \) and \( p_2 \) using

\[ c_1 = p_1 + r R (p_2 - p_1) \quad (5.18) \]

where \( r \) is a random number (from a uniform distribution ranging from 0 to 1)

5. **Heuristic** produces a child that lies on a straight line containing both parents. The child is a small distance away from the parent with the better fitness value in the direction away from the parent with the worse fitness value. The distance between the child and the better parent is specified with a parameter \( R \) (defaults to 1.2). If \( p_1 \) and \( p_2 \) are the parents, and \( p_1 \) has the better fitness value, the child is created by

\[ c_1 = p_2 + R (p_1 - p_2) \quad (5.19) \]

6. The final option, **Arithmetic** creates children that are the weighted arithmetic mean of two parents.

**Mutation**

In addition to crossover, the process of mutation can be employed to manipulate the population. Mutation maintains the genetic diversity by the random change of an individual. In the binary representation of a chromosome this is simply performed by flipping one or more bits randomly in the chromosome. The purpose of mutation is to avoid quick convergence to local minima by preventing the individuals in the population from becoming too similar to each other, i.e. to make the search more robust. When there is insufficient mutation the algorithm could become too greedy, sacrificing long term gain for short term gain, and as a result finding only a local optimum.

The frequency with which mutation occurs can be determined by a probability coefficient, which is usually taken as very small.
When the algorithm uses a representation other than binary the form of mutation described above is no longer possible. When individuals are represented in floating point a randomly selected element can be mutated by a function. This function typically introduces larger changes at the start of the simulation, and smaller changes at the later stages. As a result, the search is globally oriented at the early generations, and throughout the following generations the search becomes more and more local. The following mutation options are available to the user.

1. **Gaussian** (the default mutation function) adds a random number taken from a Gaussian distribution with a mean of zero to each entry of the parent vector. The standard deviation of this distribution is determined by two parameters, scale and shrink. The scale parameter determines the standard deviation at the first generation, while the shrink parameter controls how the standard deviation shrinks as generations go by. The default value of both these parameters is 1.

2. **Uniform** mutation begins by selecting a fraction of the vector entries of an individual for mutation, where each entry has a probability rate of being mutated (the default value is 0.01). Following this, the algorithm replaces each selected entry by a random number selected uniformly from the range for that entry.

3. **Adaptive Feasible** randomly generates directions that are adaptive with respect to the last successful or unsuccessful generation. The feasible region is bounded by the constraints and inequality constraints. A step length is chosen along each direction so that linear constraints and bounds are satisfied.

**Selection**

Once a generation of individuals has been established, parents must be selected to reproduce. This selection can be performed in varying ways, but always relies on the basis of their fitness. The Genetic Algorithm allows for four different functions to perform selection. In addition, the user also has the option of writing a custom function of their own.

The default selection function, **Stochastic uniform**, lays out a line in which each parent corresponds to a section of the line of length proportional to its scaled value. The algorithm moves along the line in steps of equal size. At each step, the algorithm allocates a parent from the section it lands on. The first step is a uniform random number less than the step size.

A more deterministic selection option is **Remainder**. **Remainder selection** assigns parents deterministically from the integer part of each individual’s scaled value and then uses roulette selection on the remaining fractional part. For example, if the scaled value of an individual is 2.3, that individual is listed twice as a parent because the integer part is 2. After parents have been assigned according to the integer parts of the scaled values, the rest of the parents are chosen stochastically. The probability that a parent is chosen in this step is proportional to the fractional part of its scaled value.

**Uniform selection** chooses parents using the expectations and number of parents. Uniform selection is useful for debugging and testing, but is not a very effective search strategy.

**Roulette selection** chooses parents by simulating a roulette wheel, in which the area of the section of the wheel corresponding to an individual is proportional to the individual’s expectation. The algorithm uses a random number to select one of the sections with a probability equal to its area.
Finally, *Tournament selection* chooses each parent by choosing a user set number (the default is 4) of parents at random and then choosing the best individual out of that set to be a parent.

**Genetic Representation**

In order to exchange information between the individuals to create new offspring some form of notation describing the individual’s information (i.e. the information containing that individual’s values of the design parameters \(x\)) is necessary upon which the genetic operators can work. This representation can take different forms, the most straightforward being a binary representation. The algorithm allows for the use of binary (or bitstring) representation but the default option is that the individuals are represented by a vector containing doubles (number that occupies 64 bits in computer memory).

The disadvantage of representing the design parameters in binary is the large amount of necessary bits to attain high resolutions (many decimals). This coupled with a multidimensional problem with many design parameters will make the size of the representations grow out of hand. This makes the default system of notation far superior to binary, allowing the Genetic Algorithm to solve more quickly and at higher precision.

**Further considerations**

The parameters of a Genetic Algorithm should be tuned per specific problem; this can be a time consuming task. If the algorithm tends to converge on local optima, it means the system should sacrifice short term fitness in exchange for long term fitness (by increasing mutation for example). All of the parameters influence the Genetic Algorithm in some way. Depending on the problem, a balance must be found in the parameters. For example, the population must not be too big because the computational cost would be too high, but the population must not be too small either, as the algorithm would most likely find a local minimum. This is why, when employing a Genetic Algorithm (or evolutionary algorithm in general), it is useful to have a good grasp of what the impact each parameter has upon the overall simulation when it is changed.

### 5.3.5 Nelder-Mead Method

The Nelder-Mead method is commonly used for multidimensional nonlinear unconstrained optimization problems. It makes no use of gradients (numerical or analytical) of the function, thus it belongs to the class of direct search methods. The gradients of the function in the program cannot be computed analytically so gradients must be calculated by taking finite distances of computed function values. Because the objective function is very unsmooth there is a significant risk of finding inaccurate derivatives, which could lead a gradient based method astray. Because this method makes no use of gradients and makes almost no assumptions about the function it optimizes, the method is known to be extremely robust [Press, et al., 2007].

This method is sometimes also referred to as the downhill simplex method (not to be confused with the simplex method, which is used to solve problems in linear programming). This method is able to solve an unconstrained minimization problem in \(n\) dimensions by maintaining \(n + 1\) points that define a simplex. This simplex is updated at each iteration by applying certain transformations to it so that it ‘rolls downhill’ until it finds a minimum [Weisstein, 2008]. A simplex takes the concept of a triangle or tetrahedron and generalizes this idea to an arbitrary dimension (in two-space, a simplex is a triangle, in
three-space it is a tetrahedron, and in four-space it is a pentachoron, etc.). If \( n \) is the length of \( x \), then a simplex in \( n \) dimensional space is characterized by the \( n + 1 \) distinct points that are its vertices.

At each step of the search a new point in or near the current simplex is generated. The function value at the new point is compared with the function’s values at the vertices of the simplex and, usually, one of the vertices is replaced by the new point, creating a new simplex. This step is repeated until the diameter of the simplex is less than the specified tolerance.

The Nelder-Mead Algorithm

We will now describe the actual algorithm in further detail. Four scalar parameters are specified to define the necessary coefficients of reflection \( \rho \), expansion \( \chi \), contraction \( \gamma \), and shrinkage \( \sigma \). The original paper ‘A simplex method for function minimalization’ [Nelder, et al., 1965] states that these parameters should satisfy

\[
\rho > 0, \quad \chi > 1, \quad \chi > \rho, \quad 0 < \gamma < 1, \quad 0 < \sigma < 1
\] (5.20)

The standard values (equal to the values used in this implementation) for these parameters are

\[
\rho = 1, \quad \chi = 2, \quad \gamma = \frac{1}{2}, \quad \sigma = \frac{1}{2}
\] (5.21)

The statement for the algorithm here by and large follows a more recent paper ‘Convergence properties of the Nelder-Mead simplex method in low dimensions’ [Lagarias, et al., 1998]. At the beginning of the \( k \)th iteration, \( k \geq 0 \), a nondegenerate simplex (the simplex is nondegenerate when its volume is greater than zero) \( \Delta_k \) is given, along with its \( n + 1 \) vertices, each of which is a point in \( \mathbb{R}^n \). An iteration \( k \) always begins by ordering and labeling these vertices as \( x_1^{(k)}, \ldots, x_{n+1}^{(k)} \), such that

\[
f_1^{(k)} \leq f_2^{(k)} \leq \cdots \leq f_{n+1}^{(k)}
\] (5.22)

where \( f_i^{(k)} \) denotes \( f_i(x_i^{(k)}) \). The \( k \)th iteration creates a set of \( n + 1 \) vertices that define a new simplex for the next iteration, such that \( \Delta_{k+1} \neq \Delta_k \). \( x_1^{(k)} \) is the best point (corresponding to the minimum value of the function) while \( x_{n+1}^{(k)} \) is the worst point (corresponding to the maximum value of the function). A single iteration can have two results, either a single new vertex (accepted point) is created that replaces the previous worst point \( x_{n+1}^{(k)} \) in the set of vertices for the next iteration, or a set of \( n \) new points, together with \( x_{n+1}^{(k)} \), is created to form a new simplex for the next iteration (this is called a shrink step).

A single iteration consists of a number of steps, starting with the ordering of the vertices (the superscript \( k \) to indicate iteration number if omitted for the sake of clarity).

1. **Order.** Order the \( n+1 \) vertices to satisfy equation (5.22).

2. **Reflect.** Compute the reflection point \( x_r \) using

\[
x_r = \bar{x} + \rho (\bar{x} - x_{n+1})
\] (5.23)

where
\[
\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i
\]

is the centroid of \(n\) best points (all vertices except for \(x_{n+1}\)). Evaluate the function value at \(x_r\) to obtain \(f_r = f(x_r)\). If \(f_1 \leq f_r < f_n\) the point \(x_r\) is accepted and the current iteration is finished.

3. **Expand.** If \(f_r < f_i\) the expansion point

\[
x_e = \bar{x} + \chi(x_r - \bar{x}) = (1 + \rho \chi)\bar{x} - \rho \chi x_{n+1}
\]

is computed. The function value at \(x_e\) is obtained, \(f_e = f(x_e)\). If \(f_1 < f_e < f_n\), \(x_e\) is accepted and the iteration is finished. Otherwise, if \(f_e \geq f_r\) the point \(x_r\) is accepted and the iteration is stopped.

4. **Contract.** Contraction between \(\bar{x}\) and the better of \(x_r\) and \(x_{n+1}\) occurs if \(f_r \geq f_{n+1}\).

4a. **Outside contraction.** If \(f_n \leq f_r < f_{n+1}\) an outside contraction is performed and a point

\[
x_{oc} = \bar{x} + \gamma(x_r - \bar{x}) = (1 + \rho \gamma)\bar{x} - \rho \gamma x_{n+1}
\]

is calculated, and \(f_{oc} = f(x_{oc})\) is evaluated. If \(f_{oc} \geq f_r\) \(x_{oc}\) is accepted and the iteration is stopped. Otherwise, a shrink step is performed (cf. step 5).

4b. **Inside contraction.** If \(f_r \geq f_{n+1}\) an inside contraction is performed and a point

\[
x_{ic} = \bar{x} + \gamma(x_r - x_{n+1}) = (1 - \gamma)\bar{x} + \gamma x_{n+1}
\]

is calculated, and \(f_{ic} = f(x_{ic})\) is evaluated. If \(f_{ic} < f_{n+1}\) \(x_{ic}\) is accepted and the iteration is stopped. Otherwise, a shrink step is performed (cf. step 5).

5. **Shrink.** Evaluate \(f\) at the \(n\) points

\[v_i = x_1 + \sigma(x_i - x_1)\]

for \(i = 2, \ldots, n + 1\). Now, the (unordered) vertices for the next iteration are \(x_1, v_2, \ldots, v_{n+1}\).

**Tie-Breaking.** Should the iteration produce a vertex with a function value such that there are two best points some rules must be stated so that vertices can be ordered correctly during the next iteration.

**Nonshrink Ordering Rule.** After a nonshrink iteration (steps 1-4 provide a new vertex) the worst vertex \(x_{n+1}^{(k)}\) is discarded. The newly created point \(v^{(k)}\) during iteration \(k\) becomes a new vertex and takes position \(j+1\) in the order of vertices of \(\Delta_{k,i}\), where \(j\) is defined using set notation as

\[
j = \max_{0 \leq \ell \leq k} \left\{ \ell \mid f(v^{(k)}) < f(x_\ell^{(k)}) \right\}
\]

The right notation denotes the set of all \(\ell\) that satisfy \(f(v^{(k)}) < f(x_\ell^{(k)})\). Once this set of \(\ell\) is found the maximum value (up to \(n\)) is taken to find \(j\). All other vertices maintain their relative ordering from iteration \(k\).
**Optimization**

**Shrink Ordering Rule.** In the case of a shrink step (step 5 provides a new vertex) the only vertex carried over from $\Delta_k$ to $\Delta_{k+1}$ is $x_1^{(k)}$. In case $x_1^{(k)}$ and one or more of the new points are tied as the best point then a check is made. If

$$
\min \left\{ f \left( v_2^{(k)} \right), \ldots, f \left( v_{n+1}^{(k)} \right) \right\} = f \left( x_1^{(k)} \right)
$$

then $x_1^{(k+1)} = x_1^{(k)}$. All other vertices are ordered as normal.

Figure 5.4 shows examples of simplices in two-space (i.e. a triangle) after finding a reflection point or expansion point.

![Figure 5.4 Simplices after reflection (left) and after expansion (right). Original simplex is indicated in dashed lines.](image)

Figure 5.5 shows examples of an outside contraction, inside contraction, and a shrink.

![Figure 5.5 Simplices after outside contraction (left), inside contraction (center), and a shrink. Original simplex is indicated in dashed lines.](image)

A graphical example of the Nelder-Mead algorithm process, showing steps such as reflection and expansion, is provided in Figure 5.6.

![Figure 5.6 An example of the Nelder-Mead Simplex method [Murray, et al., 2010].](image)
There is one final thing to point out regarding the Nelder-Mead method. It is mentioned in the literature [Press, et al., 2007] that the method rarely makes a step that trips termination criteria prematurely before an optimum has been found, and therefore recommends restarting multiple times by providing the previous optimum as input for the new run. This is confirmed by experimentation with the method. Five restarts seem to suffice for all encountered problems.
CHAPTER 6

IMPLEMENTATION

This chapter contains information regarding how the theory was implemented in the optimization program. Section 6.1 describes the basic optimization goal and two fundamental fitness functions that can be used in the optimization process. Section 6.2 discusses the process of determining the propellant use of a single leg of the transfer. Section 6.3 describes the different penalties, and their implementation, that can be imposed on a design if necessary. Sometimes, the numerical determination of the time of flight can give misleading results, section 6.4 presents this problem and offers a solution. Section 6.5 examines lunar swing-bys, their implementation in the program, and examines whether the model is able to correctly assess their benefit. Of course, the program must also be able to provide an approximation for a transfer towards a specified orbit around the Sun; section 6.6 describes how the optimization process can be guided such the final orbit possesses characteristics such as distance and inclination that are desired by the user. Section 6.7 provides information on how the user inputs and optimization outputs (fitness value and design parameters) are processed such that useful data and plots may be created. Finally, section 6.8 gives an overview of the program structure.

6.1 OPTIMIZATION OBJECTIVE

The program’s goal is to find the optimum transfer given departure, arrival, and swing-by points within the box constraints specified by the user. This is achieved by an optimization process whose objective is to minimize a given fitness function (see Chapter 5 for further details on optimization). Currently the program supports two fitness functions. The first fitness function $f_1$ is

$$f_1(x) = m_{prop}$$

(6.1)

This is purely a minimization of the propellant mass, where less propellant mass coincides with a more optimal design. The second fitness function $f_2$ also takes the total time of flight, in addition to the mass, into account, and is

$$f_2(x) = -\frac{m_0 - m_{prop}}{t}$$

(6.2)

This fitness function strives to strike a balance between mass and time by minimizing both. This balance can be influenced to one side or another by multiplying elements in this fitness function with coefficients. The two major components of the optimization process within the program are the determination of the
propellant use for each individual leg of the transfer and the determination of the mass penalties for every swing-by that occurs. The time of flight itself is actually a design parameter (an input) in the optimization process. Further mass penalties are incurred for mismatching departure and arrival velocities, and penalties are incurred when the thrust exceeds engine limitations. These penalties can be manipulated fairly easily by the user, and even completely switched off. An example of this is not penalizing any departure velocity mismatch at the Earth as this ∆\( v \) may be provided by a boost or by the spacecraft engine itself, and might not steer the design towards an optimum the user intends. Another example is not enforcing any thrust penalties; a mass optimal transfer is usually quite smooth and will tend to minimize engine thrust already. Additionally, a more improved design could lie slightly out of the region of the specified spacecraft parameters. By changing the design of the spacecraft itself (by adding an extra square meter of solar panel for instance) the user might arrive at a generally superior design. There is always some amount of trade-off to a design, and so there is never one inherently best solution. The process of reaching a good solution requires the insight of the user; the software is merely a tool to aid the user in their judgment.

6.2 Calculation of a Single Transfer Leg

This section discusses the calculation of a single individual within the optimization process from starting parameters to the final propellant use. Initially, the calculation is commenced with the provided user input and the set of optimization parameters used to calculate this particular individual within the optimization algorithm.

**Step 1: Using the Ephemeris**

The object numbers and relevant starting and transfer times are supplied in order to gain the object position and velocity vectors by using the ephemeris. The arrival dates are simply determined as

\[
 t_{\text{Arrival}_i} = t_{\text{Departure}} + \sum_{i=1}^{n_Legs} t_{\text{TOF}_i} \quad (6.3)
\]

The final arrival date is thus the sum of the departure date and all time of flight periods, while intermediate arrivals (for a swing-by) are the sum of the departure date and the time of flight periods of the thus far travelled legs of the transfer.

**Step 2: Determining the Transfer Angle**

From the position vectors the angle can be determined by equation

\[
 \Delta \theta = \cos \frac{r_1 \cdot r_2}{|r_1| \cdot |r_2|} \cdot \text{sgn}((r_1 \times r_2) \cdot e_z), \quad e_z = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (6.4)
\]

This provides the angle between two arbitrary points \( r_1 \) and \( r_2 \) in 3 dimensional space but forces this angle to be counter-clockwise around the \( z \)-axis to mimic the direction of movement of the planets around the Sun (when viewed from above i.e. from the positive \( z \) direction). Note that the right part of the equation is the signum function, which returns 1, 0, or -1 of the answer between brackets. The angle is easily modulated to be expressed in the domain of \( 0^\circ \) to \( 360^\circ \). Added to this angle is then the number of additional complete revolutions in order to arrive at the total angle that needs to be traversed.
\[\theta_2 = \Delta \theta + 2\pi N\]  \hspace{1cm} (6.5)

**Step 3: Calculation of minimum and maximum flight path angles**

Equation (3.23) is checked for validity by ensuring the part under the radical (i.e. \(\Delta\)) is positive. Should it be negative, the transfer is invalid and mass consumption is penalized such that it is no longer a candidate (mass consumption is set to infinity). Note that the winding parameter \(k_2\) is a design parameter in the optimization and thus a value has been assumed when calculating this step. Equation (3.23) is shown here once more as

\[
\tan \gamma_{1,2} = \frac{k_2}{2} \left( -\ln \frac{r_1}{r_2} \cot \frac{k_2 \theta_2}{2} \pm \sqrt{\frac{2(1 - \cos(k_2 \theta_2))}{k_2^4} - \ln^2 \frac{r_1}{r_2}} \right) \quad (6.6)
\]

Should the transfer be valid, the maximum and minimum initial flight path angles are calculated. Using this information we can determine the actual initial flight path angle that prescribes the transfer in such a way that the specified time of flight is satisfied (described in the next paragraph).

**Step 4: Finding the initial flight path angle**

As discussed previously, there are three possible shapes the time of flight (as a function of initial flight path angle) curve can take: monotone increasing, monotone decreasing, and a ‘bathtub’ shape. To determine which shape we are dealing with for this particular transfer leg, a sample of three points is taken, one at the minimum flight path angle, one at the maximum flight angle, and one at the mean of these two extremes. It is then simply a matter of comparing the time of flight periods to ascertain the curve’s shape, cf. Figure 6.1.

![Figure 6.1](image)

*Figure 6.1 Determining the time of flight curve shape.*

As can be seen above, this simple logic check quickly determines what shape the time of flight curve will be. An alternative, more robust, method is determining the gradient at each extreme by sampling two points near each extreme (for a total of 4). Dependent on whether the gradients are positive or not, the shape of the curve can be determined. Of course, this method comes at a slightly increased computational cost. After extensive testing it was found that a sample set of three points is sufficient to categorize all time of flight curves. This is partially explained by looking at equation (3.30), which seems to indicate that the shapes of these curves are a function of \(\sqrt{\tan^2 \gamma}\). This is a periodical repetition of asymptotes with ‘bathtub’ curves that reach towards positive infinity and relatively flat areas in between (cf. Figure 3.3).

To find the initial flight path angle that corresponds to the specified time of flight a simple line search strategy is implemented. For more information on this procedure, the reader is referred to section 5.3.1.
Implementation

A sample time of flight value is derived from an arbitrary initial flight path angle $\gamma_1$ by first determining the dynamic range parameter $k_1$ with equations (3.20) and (3.21), repeated here for the sake of convenience as

$$k_1^2 = \left( \frac{\ln \left( \frac{r_1}{r_2} \right) + \tan \gamma_1 \sin(k_2 \theta_2)}{1 - \cos(k_2 \theta_2)} \right)^2 + \frac{\tan^2 \gamma_1}{k_2^2}$$

(6.7)

and the sign is then deduced by

$$\text{sign}\{k_1\} = \text{sign}\{\ln(r_1/r_2) + (\tan \gamma_1/k_2) \sin(k_2 \theta_2)\}$$

(6.8)

The phase angle $\phi$ is then determined by rewriting equation (3.17) giving

$$\phi = \arccos \left( \frac{\tan \gamma_1}{k_1 k_2} \right)$$

(6.9)

and the scaling factor $k_0$ is calculated by rewriting equation (3.18)

$$k_0 = \frac{r_1}{e^{k_1 \sin \phi}}$$

(6.10)

The time of flight is then found by solving

$$t_2 - t_1 = \int_{\theta_1}^{\theta_2} \sqrt{r^3 (\tan^2 \gamma + k_1 k_2^2 s + 1) / \mu} \, d\theta$$

(6.11)

by numerical quadrature. This process is repeated using a line search strategy (cf. section 5.3.1 for details) until the calculated time of flight is close enough (depending on the set tolerance for error) to the specified time of flight. At this point the initial flight path angle corresponding to the specified time of flight is found.

**Step 5: Propellant Use**

Once the correct initial flight path angle has been determined, the corresponding other parameters are determined using the equations listed above. The mass consumption ordinary differential equation (3.48) can now be solved using these previously calculated parameters. In the case of a ‘bathtub’ time of flight curve two initial flight path angles are found, and thus two sets of parameters are found. Both these sets of parameters are run through the mass computation, and whichever parameter set stipulates the greatest propellant use is discarded.

**Step 6: Successive transfer legs**

The first leg will consume a certain amount of fuel. The next leg will thus start with a smaller mass, which is beneficial for the design (lower mass at identical thrust leads to a greater acceleration). Thus, at the end of each successive transfer leg the new spacecraft mass is passed on as input for the initial mass into the computation of the next leg.
Penalties help the optimization process to steer away from undesirable combinations of input parameters, and to penalize the design when it violates constraints. This section will discuss how the penalty values are computed in this particular program. These penalties are imposed when encountering the cases listed below.

**Swing-by**  A penalty is imposed upon the candidate when it violates either velocity mismatch or maximum deflection angle constraint.

**Arrival**  The candidate is penalized when the arrival velocity of the spacecraft does not match the velocity of the final object it arrives at.

**Departure**  The candidate is penalized when the departure velocity of the spacecraft does not match the velocity of the initial object it departs from.

**Thrust**  A penalty is imposed upon the candidate when it requires higher values of thrust than the spacecraft engine can deliver.

**Rejection**  A penalty that is tripped in certain cases (when the term under the radical $\Delta$ in equation (6.6) is negative), causing the candidate to be discarded.

Penalties functions are discussed in greater detail in Chapter 5.

### 6.3.1 Swing-by Penalty

With the knowledge of the characteristics of two individual legs of the transfer, a number of swing-by parameters may be calculated. A graphical representation of the geometry of the first leg is given in Figure 6.2 below.

The velocity at the end of the first leg is determined using equations (3.6), (3.8), and (3.10) to arrive at the velocity at the end of the first leg. The angular rate is

$$\dot{\theta}_{r_{2},1} = \frac{\mu}{\sqrt{r_2^3}} \tan^2 \gamma_{r_{2},1} + k_1 k_2^2 s_{r_{2},1} + 1$$

(6.12)

with
\[ s_{r2,1} = \sin(k_2 \theta_{r2,1} + \phi_1) \] (6.13)

The subscript \( r_2, 1 \) signifies that these parameters pertain to the first leg at the \( r_2 \) position i.e. position at the second body. The radial component of the velocity is

\[ \dot{r}_{r2,1} = r_2 \dot{\theta}_{r2,1} k_1 k_2 \cos(k_2 \theta_{r2,1} + \phi_1) \] (6.14)

As shown previously, the velocity vector at the end of the first shape is given by

\[ \mathbf{v}_{r2,1} = \dot{r}_{r2,1} \mathbf{e}_r + r_2 \dot{\theta}_{r2,1} \mathbf{e}_\theta \] (6.15)

Please note that this is the velocity vector in the transfer plane when the unit vectors \( \mathbf{e}_r \) and \( \mathbf{e}_\theta \) are defined along the transfer. The velocity magnitude is easily found from the velocity components to be

\[ v_{r2,1} = \sqrt{\dot{r}_{r2,1}^2 + (r_2 \dot{\theta}_{r2,1})^2} \] (6.16)

To find the actual 3-dimensional velocity vector we must determine \( \mathbf{e}_r \) and \( \mathbf{e}_\theta \) in the 3-dimensional space. This is done by taking the cross product of the two positional vectors

\[ \mathbf{r}_4 = \mathbf{r}_1 \times \mathbf{r}_2 \] (6.17)

To ensure that the correct counter clockwise rotation is performed we check the \( z \) component of \( \mathbf{r}_4 \). If it is negative let

\[ \mathbf{r}_4 = -\mathbf{r}_4 \] (6.18)

The normalized vector of an arbitrary vector \( \mathbf{r} \) is defined as

\[ \hat{\mathbf{r}} = \frac{\mathbf{r}}{|\mathbf{r}|} \] (6.19)

To compute the direction of the angular component of the velocity we perform another cross product

\[ \mathbf{e}_\theta = \frac{\mathbf{r}_4 \times \mathbf{r}_2}{|\mathbf{r}_4 \times \mathbf{r}_2|} \] (6.20)

The direction of the radial component is simply

\[ \mathbf{e}_r = \hat{\mathbf{r}}_2 \] (6.21)

Now that we have the final heliocentric velocity of the spacecraft at the first leg in the 3 dimensional sun centered ecliptic reference frame we can determine the excess arrival velocity as

\[ \mathbf{v}_{\infty, r2,1} = \mathbf{v}_{r2,1} - \mathbf{v}_{r2} \] (6.22)

Having computed the excess arrival velocity it is now desirable to compute the excess departure velocity by inspecting the second leg of the transfer. See Figure 6.3 for a graphical representation.
The procedure is the same as before, but now we substitute the initial parameters of the second leg into equations (3.6), (3.8), and (3.10). The angular rate is now

\[
\dot{\theta}_{r_2,2} = \frac{1}{\sqrt{r_2^3 \tan^2 \gamma_{r_2,2} + k_1 k_2^2 s_{r_2,2} + 1}}
\]

with

\[
s_{r_2,2} = \sin(k_2 \theta_{r_2,2} + \phi_2)
\]

The subscript \( r_2, 2 \) signifies that these parameters pertain to the second leg at the \( r_2 \) position. The \( k_1 \) and \( k_2 \) parameters are also different, and pertain to the second leg (i.e. not the same as the values in the previous equations describing the first leg). The radial component of the velocity is

\[
\dot{r}_{r_2,2} = r_2 \dot{\theta}_{r_2,2} k_1 k_2 \cos(k_2 \theta_{r_2,2} + \phi_2)
\]

The velocity vector in the transfer plane at the end of the first shape is given by

\[
v_{r_2,2} = \dot{r}_{r_2,2} e_r + r_2 \dot{\theta}_{r_2,2} e_\theta
\]

The velocity magnitude is easily found from the velocity components to be

\[
v_{r_2,2} = \sqrt{\dot{r}_{r_2,2}^2 + (r_2 \dot{\theta}_{r_2,2})^2}
\]

To find the velocity vector we must determine \( e_r \) and \( e_\theta \) in the 3 dimensional space. This is done by taking the cross product of the two (normalized) positional vectors

\[
r_5 = r_3 \times r_2
\]

To ensure that the correct counter clockwise rotation is performed we check the \( z \) component of \( r_5 \). If it is negative let

\[
r_5 = -r_5
\]

To compute the direction of the angular component of the velocity we perform another cross product.
Implementation

\[ e_\theta = \frac{r_5 \times r_2}{|r_5 \times r_2|} \]  \hspace{1cm} (6.30)

The direction of the radial component is simply

\[ e_r = \hat{r}_2 \]  \hspace{1cm} (6.31)

Now that we have the final heliocentric velocity of the spacecraft at the second leg in the 3-dimensional sun centered ecliptic reference frame we can determine the excess departure velocity as

\[ v_{\infty r_2,2} = v_{r_2,2} - v_{r_2} \]  \hspace{1cm} (6.32)

For two legs the velocity change at the second body is

\[ \Delta v = v_{\infty r_2,2} - v_{\infty r_2,1} \]  \hspace{1cm} (6.33)

and the magnitude of the velocity change is simply

\[ \Delta v = |v_{\infty r_2,2} - v_{\infty r_2,1}| \]  \hspace{1cm} (6.34)

However, the velocity difference which is of interest is the difference between the magnitudes of both excess velocities

\[ \Delta v = |v_{\infty r_2,2}| - |v_{\infty r_2,1}| \]  \hspace{1cm} (6.35)

This velocity difference represents the velocity increment a spacecraft would have to carry out in order to perform the swing-by. For a low-thrust spacecraft this velocity difference should be zero for a valid transfer.

To determine the angle change between the two excess velocities the smallest angle (the angle we are looking for) between two vectors is sought.

\[ \alpha = \arccos \frac{v_{\infty r_2,1} \cdot v_{\infty r_2,2}}{|v_{\infty r_2,1}| |v_{\infty r_2,2}|} \]  \hspace{1cm} (6.36)

This must be compared with the maximum change of angle, cf. equation (3.71).

\[ \sin \frac{\alpha_{\text{max}}}{2} = \frac{1}{1 + \frac{R_{r_2} v_{\infty r_2,1}}{\mu_{r_2}}} \]  \hspace{1cm} (6.37)

It can be seen that the maximum angle change is a function of the excess arrival velocity, of the minimum allowable radius of swing-by (distance from the center of the swing-by body), and of the body’s gravitational constant.

There are two constraints that a valid swing-by must satisfy for a low-thrust transfer; a (near) zero change in velocity magnitude during swing-by and a change of angle below the maximum allowable angle change. Because there is the assumption of a low-thrust engine on the spacecraft there is no method of employing a change in velocity magnitude at the swing-by, hence a valid transfer must have a result of equation
Implementation

(6.35) equal to zero in order to simulate this limitation of the spacecraft. Thus this constraint is translated into a penalty mass by the penalty function

\[ m_{sb,v} = f(|\Delta v|) \quad (6.38) \]

This value will never be completely zero when patching 2 different shapes together with a swing-by, but the penalty function reduces this to within acceptable tolerances.

In addition, a change of angle greater than the maximum angle change is physically impossible to perform. Thus, violating this constraint must be penalized using some penalty function

\[ m_{sb,\alpha} = \begin{cases} f(\alpha - \alpha_{max}), & |\alpha| > |\alpha_{max}| \\ 0, & |\alpha| \leq |\alpha_{max}| \end{cases} \quad (6.39) \]

The form these penalty functions will take is discussed in further detail in Chapter 5.

6.3.2 DEPARTURE EXCESS VELOCITY PENALTY

A penalty can be placed upon the design when there is a mismatch between the initial spacecraft velocity as prescribed by the exponential sinusoid and the velocity of the object the spacecraft is leaving. The defining geometry of the problem is shown in Figure 6.10 below.

![Figure 6.4 A representation of two arbitrary bodies in the Sun centered ecliptic reference frame, showing the initial radial and angular velocity directions of the transfer leg.](image)

The velocity at the initial point is determined using equations (3.6), (3.8), and (3.10) to arrive at the initial velocity. The angular rate is

\[ \dot{\theta}_{r_1} = \frac{\mu}{r_1^3 \tan^2 \gamma_{r_1} + k_1 k_2 s_{r_1} + 1} \quad (6.40) \]

with

\[ s_{r_1} = \sin(k_2 \theta_{r_1} + \phi_1) \quad (6.41) \]

The radial component of the velocity is

\[ \dot{r}_{r_1} = r_1 \dot{\theta}_{r_1} k_1 k_2 \cos(k_2 \theta_{r_1} + \phi_1) \quad (6.42) \]

As shown previously, the velocity vector at the end of the first shape is given by
\[ \mathbf{v}_{r_1} = \dot{r}_1 \mathbf{e}_r + r_2 \dot{\theta}_1 \mathbf{e}_\theta \]  
\hspace{1cm} (6.43)

Please note that this is the velocity vector in the transfer plane when the unit vectors \( \mathbf{e}_r \) and \( \mathbf{e}_\theta \) are defined along the transfer. The velocity magnitude is easily found from the velocity components to be

\[ v_{r_1} = \sqrt{\dot{r}_1^2 + (r_2 \dot{\theta}_1)^2} \]  
\hspace{1cm} (6.44)

To find the velocity vector we must determine \( \mathbf{e}_r \) and \( \mathbf{e}_\theta \) in the three dimensional space. The direction of the radial component is

\[ \mathbf{e}_r = \hat{r}_1 \]  
\hspace{1cm} (6.45)

and the direction of the angular component of the velocity is

\[ \mathbf{e}_\theta = \frac{\mathbf{r}_4 \times \mathbf{r}_1}{|\mathbf{r}_4 \times \mathbf{r}_1|} \]  
\hspace{1cm} (6.46)

where \( \mathbf{r}_4 \) is made sure to pointed along the positive \( z \) axis by checking the sign of the \( z \) component and mirroring the vector if this is negative.

Now that the velocity has been determined in the Sun centered reference frame a difference in velocity is computed between the velocity of the object and the velocity of the spacecraft at the starting point of the transfer. This difference in velocity is

\[ \Delta \mathbf{v}_{dep} = \mathbf{v}_{r_1} - \mathbf{v}_1 \]  
\hspace{1cm} (6.47)

For this difference in velocity to have influence upon the fitness of a candidate this difference must be translated into penalty mass. One method of translation is simply adding the absolute vector components together and then multiplying them by a coefficient determined by the user. Another method is multiplying the normal of the difference with a coefficient. One final method is based on Tsiolkovsky’s equation, that provides a relation between mass and change in velocity.

\[ \Delta v = g_0 l_{sp} \ln \frac{m_0}{m_1} \]  
\hspace{1cm} (6.48)

Rewriting and substituting yields a penalty mass

\[ m = m_0 - m_0 \left| \frac{\Delta \mathbf{v}_{dep}}{g_0 l_{sp}} \right| e \]  
\hspace{1cm} (6.49)

This mass penalty may then be further augmented by implementing a coefficient. The general penalty function notation is

\[ m_{dep} = f(\Delta \mathbf{v}_{dep}) \]  
\hspace{1cm} (6.50)
6.3.3 ARRIVAL EXCESS VELOCITY PENALTY

A penalty can be placed upon the design when there is a mismatch between the final spacecraft velocity as prescribed by the exponential sinusoid and the velocity of the object the spacecraft is arriving at. The defining geometry of the problem is shown in Figure 6.10 below.

\[ \dot{\theta}_{r_2} = \frac{\mu}{\sqrt{r_2^3 \tan^2 \gamma_{r_2} + k_1 k_2^2 s_{r_2} + 1}} \]  
(6.51)

with

\[ s_{r_2} = \sin(k_2 \theta_{r_2} + \phi_2) \]  
(6.52)

The subscript \( r_2 \) signifies that these parameters pertain to the second leg at the \( r_2 \) position. The radial component of the velocity is

\[ \dot{r}_{r_2} = r_2 \theta_{r_2} k_1 k_2 \cos(k_2 \theta_{r_2} + \phi_2) \]  
(6.53)

The velocity vector in the transfer plane at the end of the first shape is given by

\[ \mathbf{v}_{r_2} = \dot{r}_{r_2} \mathbf{e}_r + r_2 \dot{\theta}_{r_2} \mathbf{e}_\theta \]  
(6.54)

The velocity magnitude is easily found from the velocity components to be

\[ v_{r_2} = \sqrt{\dot{r}_{r_2}^2 + (r_2 \dot{\theta}_{r_2})^2} \]  
(6.55)

To find the velocity vector we must determine \( \mathbf{e}_r \) and \( \mathbf{e}_\theta \) in the three dimensional space. The direction of the radial component is

\[ \mathbf{e}_r = \hat{r}_2 \]  
(6.56)

and the direction of the angular component of the velocity is
Implementation

\[ e_\theta = \frac{r_4 \times r_2}{|r_4 \times r_2|} \]  \hspace{1cm} (6.57)

Now that the velocity has been determined in the Sun centered reference frame a difference in velocity is computed between the velocity of the object and the velocity of the spacecraft at the starting point of the transfer. This difference in velocity is

\[ \Delta v_{arr} = v_{r_2} - v_2 \]  \hspace{1cm} (6.58)

For this difference in velocity to have influence upon the fitness of a candidate this difference must be translated into penalty mass. One method of translation is simply adding the absolute vector components together and then multiplying them by a coefficient determined by the user. Another method is multiplying the normal of the difference with a coefficient. Or finally, just as for the departure excess velocity, Tsiolkovsky’s equation can be used to provide a relation between mass and velocity change. The general penalty function is

\[ m_{arr} = f(\Delta v_{arr}) = f(v_{sp,2} - v_2) \]  \hspace{1cm} (6.59)

6.3.4 Thrust Penalty

The user can apply a penalty to any individual design when it specifies engine thrusts that exceed the capabilities of the engine. Figure 6.6 below provides an example transfer where the specified thrust surpasses the limitations of the spacecraft engine.

![Figure 6.6 Available thrust (blue) and specified thrust (black), red areas indicate the spacecraft engine exceeding limits.](image)

The calculation of a single transfer leg is performed normally (cf. section 6.2) as usual, but now an additional loop must be run (iterating through the true anomaly \( \theta \)) for each individual in the optimization process to determine by how much a design exceeds the thrust limits. The penalty is based on the areas where the thrust design exceeds specification (e.g. the areas indicated in red in Figure 6.6). At each step (each step corresponds to a position on the transfer) in the loop the available thrust acceleration and specified thrust acceleration must be determined. The following equations were also shown in Chapter 3, but are provided here again for the sake of clarity. Starting with the determination of the specified thrust acceleration at an arbitrary true anomaly \( \theta \), this is computed by
where the radial distance is
\[ r = k_0 e^{k_1 \sin (k_2 \theta + \phi)} \] (6.61)

and the normalized thrust acceleration is
\[ a_N = \frac{(-1)^n \tan \gamma}{2 \cos \gamma} \left( \frac{1}{\tan^2 \gamma + k_1 k_2 s + 1} - \frac{k_2^2 (1 - 2k_1 s)}{(\tan^2 \gamma + k_1 k_2 s + 1)^2} \right) \] (6.62)

To compute this normalized thrust acceleration we require the expression for the flight angle, which is
\[ \tan \gamma = k_1 k_2 \cos (k_2 \theta + \phi) \] (6.63)

and the definition
\[ s = \sin (k_2 \theta + \phi) \] (6.64)

These equations provide us the basis to compute the design specified thrust acceleration at each iteration of the true anomaly. The available thrust acceleration is defined as
\[ a_{T_{av}} = \frac{T_{av}}{m} \] (6.65)

The available thrust level is calculated according to the available power
\[ T_{av} = \frac{2 P \eta_r}{g_0 I_{sp}} \] (6.66)

The available power is defined as
\[ P = I_{Solar} A_{Panel} \eta_{Panel} \] (6.67)

where
\[ I_{Solar} = \frac{I_\odot}{r^2} \] (6.68)

The solar irradiance at 1 AU from the Sun is \( I_\odot = 1367 \text{ W/m}^2 \) [Wertz, et al., 1999]. The mass is updated each iteration by
\[ m_{New} = m_{Old} - m_{used} \] (6.69)

The mass use is approximated each iteration by
\[ m_{used} = \Delta t \frac{dm}{dt} \] (6.70)

The propellant rate at this step is
\[ \frac{dm}{dt} = \left| \frac{ma_T}{g_0 I_{sp}} \right| \] (6.71)
Implementation

The approximation of the period of time it took to complete this step (of size $h$) in the true anomaly is

$$\Delta t = \frac{h}{\dot{\theta}}$$

(6.72)

where the angular rate is

$$\dot{\theta}^2 = \frac{\mu}{r^3} \frac{1}{\tan^2 \gamma + k_1 k_2 s + 1}$$

(6.73)

Once $a_{r_{av}}$ and $a_T$ have been determined for each step on the transfer it is made sure that each value is absolute, i.e. $|a_{r_{av}}|$ and $|a_T|$ (the design may require negative thrust accelerations). All of the individual accelerations at each step are stored in vectors $a_{r_{av}}$ and $a_T$. A comparison is made by subtracting these two vectors from each other.

$$\Delta a = a_{r_{av}} - a_T$$

(6.74)

Whenever an element in $\Delta a$ is positive that particular element is set to zero, as no penalty should be induced when the engine operates within specification. This leaves us with the vector $\Delta a_r$ with only negative values. The absolute sum of the leftover elements within vector $\Delta a$ is taken to arrive at the penalty value. This value corresponds to the total area spanned by the true anomaly and thrust (acceleration) parameters where the design is not capable of being achieved by the spacecraft engine (cf. red areas in Figure 6.6). This penalty value must be manipulated in some way to translate it into a penalty mass, this is performed by a user stated penalty function (e.g. a simple coefficient) of the form

$$m_{thr} = f(\Delta a_r)$$

(6.75)

6.3.5 Rejection Penalty

Sometimes a candidate individual must be completely discarded. This can happen when no minimum and maximum pair of initial flight path angles are found (the set of optimization parameters are invalid by definition, see section 3.2) or when no valid initial flight path angles that correspond to a time of flight can be found. This second rejection occurrence is mostly a safety built into the function (that finds the initial flight path angle based on time of flight) to make it robust. Should a set of optimization parameters result into the individual triggering the rejection penalty the penalty mass is simply set to infinite and the individual discarded from the population. The penalty function depends on the design parameters $x$ and is written as

$$m_{rej} = f(x)$$

(6.76)

6.4 Time of Flight Curve Aberrations

An interesting phenomenon occurs when computing the time of flight values at (and near) the extreme values of the initial flight path angle ($\gamma_1 \approx \gamma_{1_{\text{min}}}$ or $\gamma_1 \approx \gamma_{1_{\text{max}}}$). For some cases, the computation of the time of flight near the minimum and maximum flight path angles gives erroneous values. As a consequence, the shape of the time of flight curve (e.g. a 'bathtub' curve) can no longer accurately be determined. The end result is an incorrect evaluation of the curve, and thus the algorithm is no longer
able to find the correct time of flight. This has nothing to do with numerical quadrature inaccuracies but with the behavior of the terms contained in the equation of the time of flight, equation (3.30). The driving force for the inaccuracy in this equation are the terms contained within the cubed radius (i.e. shape equation for an exponential sinusoid), given here (with $q = 0$) again as

$$r = k_0 e^{k_1 \sin (k_2 \theta + \phi)}$$  \hspace{1cm} (6.77)$$

It is useful to look at the behavior of $k_0$, cf. equation (3.18)

$$k_0 = \frac{r_1}{e^{k_1 \sin \phi}}$$  \hspace{1cm} (6.78)$$

and at the behavior of $k_1$, cf. equation (3.21)

$$k_1^2 = \left( \frac{\ln \left( \frac{r_1}{r_2} \right) + \frac{\tan y_1}{k_2^2 \sin (k_2 \theta_2)} \tan y_1}{1 - \cos (k_2 \theta_2)} \right)^2 + \frac{\tan^2 y_1}{k_2^2}$$  \hspace{1cm} (6.79)$$

For certain configurations of the parameters the dynamic range parameter $k_1$ may become extremely large at the fringes of the valid initial flight path angle space. In fact, $k_1$ may become so large that the exponent of this (see equation (6.77) above) may be saved to memory as infinite in a computer. Additionally, $k_0$ can rapidly approach zero for certain configurations (again when $k_1$ is large, cf. equation (6.78)), leading to a zero and infinity multiplication operation within the time of flight computation of which the result is undefined (and is stored as such to memory). Naturally, an infinite, and especially an undefined, answer will play havoc upon any algorithm tasked with the determination of the shape of the time of flight curve, so this problem requires addressing.

The solution is not particularly difficult to implement. The times of flight at the extremes are calculated, and if either of these are infinite or undefined a step is taken inward for that particular extreme (within the defined area of initial flight path angles) and the time of flight is evaluated again. The process is repeated until a real positive value for the time of flight is found at both extremes. Based on this information, the shape of the time of flight curve may now safely be determined. If this check were not to occur, the time of flight algorithm will often arrive at incorrect assessments or run to a standstill in an infinite loop.

No harm is done by disregarding small regions at the edges of the flight path angle spectrum as these are unattractive (yet theoretically feasible) solutions with extremely large or small times of flight that the computer cannot evaluate with sufficient accuracy.

### 6.5 LUNAR GRAVITY ASSIST

We will explore the implementation of a lunar swing-by in the program by investigating a test transfer from Earth to Venus - with and without lunar swing-by. The model in the program is quite simple; it includes an analytical ephemeris of the Moon in Sun Centered Ecliptic reference frame and simply treats it as another planet (at much lower mass) to perform a swing-by with. This model will most likely have shortcomings due to the fact that the whole transfer is treated as being primarily in the Sun’s sphere of
Implementation

influence while the actual Earth-Moon transfer would be much more complex. Nevertheless, an attempt has been made to implement lunar swing-bys into the program and has met with some success.

The Earth-to-Venus transfer optimization was performed with emphasis placed upon the arrival velocity at Venus. Additionally, velocity mismatches at the Lunar swing-by were punished relatively heavily. Relevant spacecraft parameters for this particular example are a specific impulse of 5000 s, a solar panel area of 33 m², a panel efficiency of 25%, and a thruster efficiency of 75%. It should be noted that the penalty functions were identical for both transfers, to ensure a fair comparison. Figure 6.7 below provides an overview of the resulting transfers.

![Figure 6.7 Earth-Venus transfer (left) and Earth-Moon-Venus transfer (right), both showing transfer (top), thrust profile (center), and spacecraft mass evolution (bottom).](image)

![Figure 6.7 Earth-Venus transfer (left) and Earth-Moon-Venus transfer (right), both showing transfer (top), thrust profile (center), and spacecraft mass evolution (bottom).](image)

The plots show relatively similar transfers (the Moon is denoted by the grey colored circle) with a small decrease in propellant use for the case with the Lunar swing-by. Provided with this information, we can already purport that the lunar swing-by is beneficial. In addition, other parameters of the transfer improve when the lunar swing-by is added, cf. Table 6.1.

<table>
<thead>
<tr>
<th></th>
<th>Earth-Venus</th>
<th>Earth-Moon-Venus</th>
</tr>
</thead>
<tbody>
<tr>
<td>Excess Departure Velocity</td>
<td>2.1311 km/s</td>
<td>1.8084 km/s</td>
</tr>
<tr>
<td>Excess Arrival Velocity</td>
<td>0.9439 km/s</td>
<td>0.1547 km/s</td>
</tr>
<tr>
<td>Swing-by velocity mismatch</td>
<td>N/A</td>
<td>0.207 km/s</td>
</tr>
<tr>
<td>Initial Mass</td>
<td>1500 kg</td>
<td>1500 kg</td>
</tr>
</tbody>
</table>
Table 6.1 List of transfer properties for both Earth-Venus and Earth-Moon-Venus transfers.

<table>
<thead>
<tr>
<th></th>
<th>Initial Mass</th>
<th>kg</th>
<th>Mass</th>
<th>kg</th>
</tr>
</thead>
<tbody>
<tr>
<td>Final Date</td>
<td>1329.06</td>
<td>kg</td>
<td>1348.81</td>
<td>kg</td>
</tr>
<tr>
<td>Initial Date</td>
<td>01-01-2000 06:43:57 GCD</td>
<td>09-12-1999 20:06:10 GCD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time of Flight</td>
<td>572.974 day</td>
<td>676.755 day</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The table shows that the excess arrival velocity substantially improves to only 155 m/s difference between the spacecraft and the velocity of Venus at that moment. There is also a small improvement in the excess departure velocity. However, a swing-by velocity mismatch of roughly 200 m/s is introduced. It has proven difficult to completely work away this swing-by velocity mismatch, another run (with a more severe penalty function) gave a mismatch of only 38.8 m/s, but with an excess departure velocity of 1.21 km/s (an improvement) and an excess arrival velocity of 1.16 km/s (even worse than the transfer without swing-by). It seems difficult to find a design that is able to both provide a small velocity mismatch at the swing-by in addition to a small arrival velocity mismatch at the final planet. Although this method of implementing lunar swing-by is quite simplistic, it seems to be able to generally improve transfers to some degree. The user should examine any transfer involving a lunar swing-by on a case by case basis and interpret the results.

### 6.6 REACHING THE SUN

When the design calls for any type of orbit in space that is not meeting with an object (such as a planet) the optimizer must be given special instructions. In our case we wish to insert into a solar polar orbit at 0.4 AU distance. Three methods of making the optimizer tend towards inclined orbits are described in this section.

The first of these methods uses the comparison between spacecraft arrival velocity and ideal science orbit polar velocity. To do this, the user specifies the point and a separate transfer function is called that is specifically written for this final leg to the Sun (or the only leg if no swing-bys are performed). The optimizer is provided with three extra scalars, varying from -1 to 1. Let us say that these scalars constitute a vector originating at the Sun.

\[
\mathbf{r}_{\text{Sun}} = \begin{bmatrix} x_{\text{Sun}} \\ y_{\text{Sun}} \\ z_{\text{Sun}} \end{bmatrix}, \quad -1 \leq x_{\text{Sun}} \leq 1, -1 \leq y_{\text{Sun}} \leq 1, -1 \leq z_{\text{Sun}} \leq 1
\]  

(6.80)

This vector can describe a point anywhere around the Sun within a square box, and the optimizer will create various vectors \( \mathbf{r}_{\text{Sun}} \) composed of varying values within this box. This vector is then normalized and multiplied by the desired distance (in our case \( d_{SO} = 0.4 \text{ AU} \)) to arrive at

\[
\mathbf{r}_{SO} = \frac{\mathbf{r}_{\text{Sun}}}{|\mathbf{r}_{\text{Sun}}|} \cdot d_{SO}
\]

(6.81)

This new vector now gives an arbitrary point on a sphere of \( d_{SO} \) distance surrounding the Sun. The software then creates a transfer between the previous body (the Earth for instance) and this point in space. The position itself is not enough to satisfy the constraints of a solar polar orbit, the velocity must also be
Implementation

that of an object in circular orbit. To achieve this, the cross product of the \( z \)-axis vector \( e_z \) and \( r_{SO} \) is taken.

\[
q = r_{SO} \times e_z
\]  

(6.82)

Using the newly created vector \( q \) the tangential velocity direction can be calculated as

\[
v = r_{SO} \times q
\]  

(6.83)

This vector \( v \) is guaranteed to always give a velocity that lies in a plane orthogonal to the ecliptic plane (i.e. the vector gives a polar orbit), see Figure 6.8.

\[\text{Figure 6.8 The geometry used to find the position and velocity vectors for insertion into the science orbit.}\]

To arrive at the actual science orbit velocity \( v_{SO} \) this vector must be normalized and multiplied by the circular orbital velocity.

\[
v_{SO} = \frac{v}{|v|} \sqrt{\frac{\mu}{d_{SO}}}
\]  

(6.84)

An individual within the optimization process possesses the vectors \( r_{SO} \) (describing position) and \( v_{SO} \) (describing velocity) or \(-v_{SO}\) (the other direction) and this information now serves as the final point for solving Lambert’s problem. At the end of the transfer the velocity of the spacecraft is compared to the circular science orbit velocity and mismatches are punished using a penalty function (see section 0).

There are also additional possibilities to help the optimizer to search for inclined orbits around the Sun. The second method seeks to maximize the angle between the ecliptic and the orbital plane of the spacecraft on its last leg towards the Sun. The spherical geometry described above (for the first method) is used and then the angle is simply determined by computing the angle between the orthogonal vector of each plane. For the ecliptic this is simply the \( z \) axis vector \( e_z \). The transfer plane is defined by the vector \( r_1 \) pointed towards the previous departure (or swing-by) body and the vector \( r_{SO} \). And so the angle between both planes is

\[
\cos \alpha = \frac{e_z \cdot (r_1 \times r_{SO})}{|e_z||r_1 \times r_{SO}|}
\]  

(6.85)

It is made sure that the vector \( (r_1 \times r_{SO}) \) is in the positive \( z \) direction by mirroring the answer if the \( z \) component is negative. Because we seek this angle \( \alpha \) to be as close as possible to 90° this is modeled as a reward in the fitness function by reducing propellant mass as angle increases in the fitness function.
The third method is adding the propellant mass necessary to change the orbital plane (using Edelbaum’s approach, described in section 3.7) to the fitness function. In this way, the optimizer will recognize the propellant required for plane changing and try to incorporate a plane change earlier on by using swing-bys.

In practice, all methods helped the optimizer to find inclined orbits with a certain degree of success. However, after some experimentation it was found that a combination of both first and third method seems to work most effectively.

6.7 ANALYSIS AND PLOTTING

The optimization process provides a set of fundamental parameters that prescribe the specified transfer. This resulting set of parameters is processed in order to fully assess the transfer. This processing can provide data (such as velocity, position, current mass, etc.) at any point in the transfer, provide 3-dimensional plots of the transfer, and plot the behavior of certain variables as a function of other variables (such as time or angle). This helps the user to determine whether a particular trajectory is actually optimal or has shortcomings in some other fashion (such as an unattractive thrust profile) that the user did not foresee when setting up the optimization.

6.7.1 ANALYSIS

All processing is contained within one function, which only requires the input constants for the optimization program (automatically passed on for processing) in addition to the final fitness value and its corresponding optimization variables.

For each leg of the transfer the initial and final conditions have been determined as shown in the previous sections. Intermediate values are found by stepping through the true anomaly of each leg. The position is determined simply by employing the shape equation (3.1), listed here again as

\[ r = k_0 e^{q_\theta} + k_1 \sin(k_2 \theta + \phi) \]  

(6.86)

All variables in this equation are known, and for the current value of the true anomaly \( \theta \) the current position \( r \) is found. The current (for a value of true anomaly \( \theta \)) flight path angle (from equation (3.7)) is

\[ \tan \gamma = k_1 k_2 \cos(k_2 \theta + \phi) \]  

(6.87)

This is used to determine the thrust acceleration at this position on the transfer, adapted from equation (3.9) as

\[ a_T = \frac{\mu}{r^2} \frac{(-1)^h \tan \gamma}{2 \cos \gamma} \left( \frac{1}{\tan^2 \gamma + k_1 k_2^2 s + 1} - \frac{k_2^2 (1 - 2 k_1 s)}{(\tan^2 \gamma + k_1 k_2^2 s + 1)^2} \right) \]  

(6.88)

where

\[ s = \sin(k_2 \theta + \phi) \]  

(6.89)

The rate of change of the true anomaly (from equation (3.8)) is
The rate of change of the radial position (from equation (3.6)) is

\[ \dot{r} = r \dot{\theta} k_1 k_2 \cos(k_2 \theta + \phi) \]  

(6.91)

The current velocity is (from equation (3.28)) then

\[ v = \sqrt{\dot{r}^2 + (r \dot{\theta})^2} \]  

(6.92)

Up until now the equations in this section have been taken from Chapter 3 without any significant changes. Time, thrust, and mass are now inspected as a function of the step \( h \) in true anomaly that was taken. An approximation of the period of time it took to complete this step in the true anomaly is

\[ \Delta t = \frac{h}{\dot{\theta}} \]  

(6.93)

The current propellant rate (obtained from section 3.5) is

\[ \frac{dm}{dt} = \left| \frac{ma_T}{g_0 l_{sp}} \right| \]  

(6.94)

The rate is taken as absolute as the shape can prescribe negative accelerations. The simplest method of determining the expelled propellant during this step in the true anomaly is

\[ m_{Used} = \Delta t \frac{dm}{dt} \]  

(6.95)

The current thrust is calculated as

\[ T = ma_T \]  

(6.96)

The available thrust level (from equation (3.57)) is calculated according to the available power

\[ T_{av} = \frac{2P \eta_T}{g_0 l_{sp}} \]  

(6.97)

that is available to it. The available power (from equation (3.51)) is

\[ P = I_{Solar} A_{Panel} \eta_{Panel} \]  

(6.98)

where the current solar irradiance (in kilometers) is

\[ I_{Solar} = I_\oplus \frac{149597870.7^2}{r^2} \]  

(6.99)

At the end of each step of the true anomaly the time is updated using

\[ t_{New} = t_{Old} + \Delta t \]  

(6.100)

and the update in mass is given by
Implementation

\[ m_{New} = m_{Old} - m_{Used} \]  

(6.101)

The entire leg of the transfer is stepped through in this manner, storing all relevant intermediate values of time, mass, velocity, position, flight path angle, available and prescribed thrust, available and prescribed acceleration. Each separate leg of the transfer is stepped through in this manner to acquire data on every leg.

With an acquired set of design parameters the change in mass can be calculated once more using different numerical quadrature, this time storing intermediate values to obtain an idea of the spacecraft mass evolution. As demonstrated above the spent propellant mass per step in true anomaly \( \theta \) is

\[ m_{Used} = f(h, \mu, m, g_0, l_{sp}, k_0, k_1, k_2, \theta, \phi) \]  

(6.102)

By default, a 4th order Runge-Kutta method is employed to calculate the mass change for each step (the user can choose to use another integrator, or write their own in the relevant function), written as

\[ m_{Used} = \frac{1}{6} c_1 + \frac{1}{3} c_2 + \frac{1}{3} c_3 + \frac{1}{6} c_4 \]  

(6.103)

with the coefficients being

\[ c_1 = f(h, \mu, m, g_0, l_{sp}, k_0, k_1, k_2, \theta, \phi) \]

\[ c_2 = f(h, \mu, m + 1/2 \Delta \theta, g_0, l_{sp}, k_0, k_1, k_2, \theta + 1/2 c_1, \phi) \]

\[ c_3 = f(h, \mu, m + 1/2 \Delta \theta, g_0, l_{sp}, k_0, k_1, k_2, \theta + 1/2 c_2, \phi) \]

\[ c_4 = f(h, \mu, m + \Delta \theta, g_0, l_{sp}, k_0, k_1, k_2, \theta + 1/2 c_3, \phi) \]  

(6.104)

For each transfer leg the steps in the true anomaly are tracked and the final angle is always superposed upon the next leg of the transfer such that this angle is a continuously growing angle without modulation (e.g. 3 revolutions in total, no matter the number of individual legs, around the Sun would coincide with a true anomaly change of 0° to 1080°).

The analysis also stores all relevant data, such as positions and velocities of each object and the spacecraft at relevant times (arrival, departure, and swing-by points), excess velocities and angles at each leg of the transfer, swing-by velocity mismatches, propellant use, shape parameters, etc. to file.

6.7.2 PLOTTING

Each of the separate planar legs of the transfer must now be presented cohesively in one overview. The simplest method of displaying the total transfer in 3-dimensional space is using a mapping technique for every leg. Another method (more accurate), employs rotation matrices to correctly align the transfer arcs. Both methods will be discussed in this section. Before discussing these methods some preliminary work is done on the existing data, and an orthonormal set of vectors is created.

Initially, the dataset for a single leg is given in a 2-dimensional frame, with polar coordinates \((r, \theta)\) and the true anomaly starts from zero which, if changed to Cartesian coordinates, would translate to the transfer leg starting on the x-axis (see equation (6.106) where \(y=0\) for \(\theta=0\)).
Implementation

Either vector mapping or rotation is necessary to correctly find the position and velocity vectors along the entire transfer in the heliocentric reference frame. This is due to the fact that the transfer shapes are computed using scalar values (radii of the objects and the angles between the objects). As a result the points describing the transfer (in polar or Cartesian coordinate system) cannot be introduced directly into the heliocentric reference frame without manipulation (if they were, all transfer legs would lie in the \(xy\) plane, or ecliptic plane, and all the individual legs of the transfer would all start on the \(x\)-axis where the initial true anomaly \(\theta\) was locally defined as equal to zero).

Before we change from polar to Cartesian coordinates the true anomaly values describing the transfer leg are shifted by

\[
\Delta \theta = \cos\left(\frac{e_x \cdot r_1}{|e_x| \cdot |r_1|}\right) \cdot \text{sgn}\left((e_x \times r_1) \cdot e_z\right)
\]

where \(r_1\) is the positional vector of the initial, or departure, object. This process is illustrated in Figure 6.9.

![Figure 6.9 The change in true anomaly for the entire transfer leg.](image)

Following this, it is changed to Cartesian coordinates (using the new values of \(\theta\)) simply by using

\[
\begin{align*}
x &= r \cdot \cos \theta \\
y &= r \cdot \sin \theta \\
z &= 0
\end{align*}
\]

for each data point in the set. A dataset \(x\) is defined that contains a series of Cartesian coordinates that describe a transfer. The size of this dataset is dependent on the number of points that are used to describe the transfer arc. For example, 100 true anomaly steps along the arc would lead to 101 points in space along the transfer, and in Cartesian coordinates this would lead to a 3×101 matrix (where the \(z\) row is as of yet 0). The dataset \(x\) still lies in a 2-dimensional plane and must be manipulated further such that the transfer actually spans between departure point \(r_1\) and arrival point \(r_2\).

Before moving on to the discussion of vector mapping or rotation matrices, an orthonormal set (set of normalized orthogonal vectors) of vectors is created. The initial point is normalized by

\[
u_1 = \hat{r}_1 = \frac{r_1}{|r_1|}
\]

and the destination point is normalized by

\[
\text{(6.107)}
\]
Implementation

\[ \mathbf{u}_2 = \mathbf{r}_2 = \frac{\mathbf{r}_2}{|\mathbf{r}_2|} \]  

(6.108)

An orthonormal vector to the plane formed by \( \mathbf{v}_1 \) and \( \mathbf{v}_2 \) is

\[ \mathbf{u}_3 = \mathbf{u}_1 \times \mathbf{u}_2 \]  

(6.109)

But \( \mathbf{u}_1, \mathbf{u}_2, \) and \( \mathbf{u}_3 \) do not yet form an orthonormal set because \( \mathbf{u}_1 \) and \( \mathbf{u}_2 \) are generally not orthogonal. Thus we create a fourth vector

\[ \mathbf{u}_4 = \mathbf{u}_1 \times \mathbf{u}_3 \]  

(6.110)

Now we have an orthonormal set consisting of \( \mathbf{u}_1, \mathbf{u}_3, \) and \( \mathbf{u}_4 \).

From here we may continue by using vector mapping (discussed first), or by using rotation matrices (discussed after vector mapping).

**Vector Mapping**

By subtracting the \( x \) and \( y \) components of the position vector of the departure object (the point the transfer leg stems from, e.g. the Earth) from all the points stored in dataset \( \mathbf{x} \) the starting point of a transfer leg is translated so that it starts from the origin (the center of the Sun). This process can be mathematically expressed as

\[ \mathbf{x}_T = \mathbf{x} - \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{r}_1 \]  

(6.111)

where

\( \mathbf{x}_T \) is the new dataset containing points that if brought directly into the heliocentric reference frame, would lie the xy plane (the ecliptic, \( z = 0 \)) starting from the origin.

\( \mathbf{x} \) is the original dataset (that would directly translate onto the ecliptic if brought into the heliocentric reference frame).

\( \mathbf{r}_1 \) is the location of the initial, or departure, object.

The data is now a collection of points describing a transfer that originates at the origin, and if it were directly imported into the heliocentric reference frame the transfer lie in the ecliptic plane and would originate from the Sun. From here, the data will be mapped upon the correct plane (and resized) followed by another translation to arrive at the desired set of points that describe a fully three dimensional transfer between two objects arbitrarily located in space in the heliocentric reference frame.

Using vector mapping we can map a vector from one plane onto another plane and then manipulate the vector such that it maintains its original length. A single point \( i \) (from the dataset \( \mathbf{x}_i \) containing the \( x,y,z \) coordinates of all points along the transfer) can be construed as a vector \( \mathbf{u}_i \) that spans from the origin to point \( i \) (thus the vector is in the plane of the ecliptic). This vector is then mapped onto the plane spanned by \( \mathbf{u}_1 \) and \( \mathbf{u}_4 \) by using

\[ \mathbf{u}_{map} = \frac{\mathbf{u}_i \cdot \mathbf{u}_1}{\mathbf{u}_1 \cdot \mathbf{u}_1} \mathbf{u}_1 + \frac{\mathbf{u}_i \cdot \mathbf{u}_4}{\mathbf{u}_4 \cdot \mathbf{u}_4} \mathbf{u}_4 \]  

(6.112)
Implementation

Now the vector has the correct orientation (in the heliocentric reference frame), but is of wrong length (because it has been mapped). So this vector must be normalized and then multiplied by the original length. The normalized mapped vector is

$$\hat{\mathbf{u}}_{i_{\text{map}}} = \frac{\mathbf{u}_{i_{\text{map}}}}{|\mathbf{u}_{i_{\text{map}}}|}$$

(6.113)

The length of a single point $i$ (consisting of three Cartesian coordinates $x$, $y$, and $z$) from the data set $x_T$ is calculated by

$$l_{i_T} = \sqrt{x_{i_T}^2 + y_{i_T}^2 + z_{i_T}^2} = |\mathbf{u}_{i_T}|$$

(6.114)

Thus, the final vector is

$$\mathbf{u}_{i_{\ell_T}} = l_{i_T}\hat{\mathbf{u}}_{i_{\text{map}}}$$

(6.115)

This vector is then translated back again, from the origin towards the initial position of the object (so that the transfer originates from the departure planet instead of the origin), by

$$\mathbf{u}_i = \mathbf{u}_{i_{\ell_T}} + \mathbf{r}_1$$

(6.116)

Figure 6.10 shows a graphical representation of the imaging process. The vector $\mathbf{u}$ points to the $i^{th}$ point that describes the transfer. Both point $i$ and vector $\mathbf{u}$ lie within the ecliptic (i.e. the $xy$ plane) if directly imported into the heliocentric reference frame without change. This vector is then mapped upon the plane spanned by vectors $\mathbf{u}_i$ and $\mathbf{u}_4$. Because the length of the vector is now incorrect, it is normalized and then multiplied by the length of the original vector $\mathbf{u}$ to arrive at the vector $\mathbf{u}_{i_{\ell_T}}$. This vector is then translated back by adding the position of the initial object. This process is repeated for all points that describe the transfer in order to arrive at a data set describing the shape of the transfer in the Sun centered ecliptic reference frame.

Figure 6.10 Vector mapping from the ecliptic plane to the plane of the actual transfer in Sun centered ecliptic reference frame.

If the transfer consists of multiple legs this procedure is repeated for each leg to obtain a complete set of points describing position in the Sun centered reference frame.
**Rotation Matrix**

Instead of mapping the transfer shape from one plane onto another we may instead make use of rotations. A check is performed on vector \( \mathbf{u} \), whether it is pointed in positive or negative \( z \) axis direction. If it is negative then the vector is pointed in the opposite direction. In pseudo code this process could be written as

\[
\text{if vector3}(3) < 0 \\
\quad \text{vector3}_\text{temp} = -\text{vector3} \quad \% \text{The vector is mirrored.} \\
\text{else} \\
\quad \text{vector3}_\text{temp} = \text{vector3} \quad \% \text{The vector is left alone.}
\]

The angle for this first rotation is then determined by employing

\[
\alpha_1 = \arccos \left( \frac{\mathbf{u}_{3\text{temp}} \cdot \mathbf{e}_z}{\left| \mathbf{u}_{3\text{temp}} \right| \cdot \left| \mathbf{e}_z \right|} \right) \cdot \text{sgn}(\mathbf{u}_3 \cdot \mathbf{e}_z) \quad (6.117)
\]

where \( \mathbf{e}_z \) denotes the unit \( z \) vector \(<0,0,1>\) and \( \text{sgn}() \) represents the signum function, returning either -1,0, or 1. This angle is then modulated such that it varies from 0° to 360°. The rotation requires the discovery of the intersection vector of two planes; the \( xy \) plane (ecliptic) and the plane spanned by the two vectors \( \mathbf{u}_1 \) and \( \mathbf{u}_2 \). The standard equation of a plane in 3-dimensional space is

\[
Ax + By + Cz + D = 0 \quad (6.118)
\]

The normal to this plane is \((A,B,C)\). If given three points \((x_1, y_1, z_1)\), \((x_2, y_2, z_2)\), and \((x_3, y_3, z_3)\) the equation of the plane through these 3 points is given by the determinants

\[
A = \begin{vmatrix} 1 & y_1 & z_1 \\ 1 & y_2 & z_2 \\ 1 & y_3 & z_3 \end{vmatrix}, B = \begin{vmatrix} x_1 & 1 & z_1 \\ x_2 & 1 & z_2 \\ x_3 & 1 & z_3 \end{vmatrix}, C = \begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix}, D = -\begin{vmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \end{vmatrix} \quad (6.119)
\]

As the plane of motion is always around the Sun both planes always contains the origin point \((0,0,0)\) and thus the intersection vector of the two planes is guaranteed to go through this point. Because of this fact we may easily deduce the intersection vector as being the cross product of the normals of both planes. The normal of the ecliptic is simply \( \mathbf{e}_z \) and the normal of the plane spanned by vectors \( \mathbf{u}_1 \) and \( \mathbf{u}_2 \) is \( \mathbf{u}_3 \), cf. equation (6.109). The intersection vector is then simply the cross product of both these normal vectors.

\[
\mathbf{u}_{is} = \mathbf{e}_z \times \mathbf{u}_3 \quad (6.120)
\]

Just as in the case of vector mapping we handle each separate point on the transfer individually. The initial data set \( \mathbf{x} \) is taken (see above) and each separate point in this set is described as a vector \( \mathbf{u} \) from the origin. These vectors \( \mathbf{u} \) are currently still in the \( xy \) plane. The rotation is performed according to the rotation function \( f(x,y,z,a,b,c,u,v,w,\theta) \) shown (and derived) in appendix B. For this rotation the input arguments are \((x,y,z) = \mathbf{u}_i, (a,b,c) = (0,0,0), (u,v,w) = \mathbf{u}_s\) and angle \( \alpha_1 \). And so for each vector \( \mathbf{u} \) within the set \( \mathbf{x} \) follows

\[
\mathbf{u}_{R1_i} = f(\mathbf{u}_i, 0, \mathbf{u}_{is}, \alpha_1) \quad (6.121)
\]

After rotation the data set is defined as \( \mathbf{x}_{R1} \) and we refer to an individual rotated vector as \( \mathbf{v}_{R1_i} \).
Figure 6.11 Rotation of point \((x,y,z)\) around vector \(<u,v,w>\) with origin \((a,b,c)\) by angle \(\theta\).

Figure 6.11 above shows the a general rotation from \((x,y,z)\) to \((x_R,y_R,z_R)\). In this case since \((a,b,c) = (0,0,0)\) the vectors coincide with these coordinates, i.e. \(u = (x,y,z)\), \(u_6 = (x_R,y_R,z_R)\). Following this first rotation a second rotation must be performed in much the same way as the first. The second rotation angle is

\[
\alpha_2 = \cos \left( \frac{u_5 \cdot u_6}{|u_5| \cdot |u_6|} \right) \cdot \text{sgn} \left( (u_6 \times u_5) \cdot u_3 \right)
\]  

(6.122)

where \(u_i\) is simply

\[
u_5 = r_2 - r_1
\]  

(6.123)

and \(u_i\) is the initial rotated vector of set \(x_R\) subtracted from the final rotated vector of set \(x_0\), or

\[
u_6 = u_{R_{i=n}} - u_{R_{i=1}}
\]  

(6.124)

The center of rotation for this rotation is

\[
C_R = \frac{u_{R_{i=1}} - r_1}{u_5 - u_6}
\]  

(6.125)

These two vectors \(u_i\) and \(u_6\) and center of rotation \(C_R\) are also shown graphically in an example in Figure 6.13. The second rotation is also performed according to the rotation function \(f(x,y,z,a,b,c,u,v,w,\theta)\) (shown and derived in appendix B) with input arguments \((x,y,z) = u_6\), \((a,b,c) = (C_R)\), \((u,v,w) = u_3\) and angle \(\alpha_2\).

\[
v_{R_{2i}} = f(u_R, C_R, u_3, \alpha_2)
\]  

(6.126)

After rotation the data set is defined as \(x_{02}\) and we refer to an individual rotated vector as \(u_{02}\). Following both rotations a translation is performed to make sure the new transfer fits the initial and final point in space. This translation is performing by subtracting the position of the initial object \(r\) from the initial vector \(u_{R_{2i=0}}\) of the set \(x_{02}\) (this is the spatial distance between the initial point of the transfer shape and the initial object, if the rotation were performed correctly the transfer shape should also connect with the final object). So the final manipulation to each vector within the set \(x_{02}\) is

\[
u_{R_{2i}} = (u_{R_{2i=1}} - r_1)
\]  

(6.127)

Doing this for each vector gives us the final set of vectors that form the final data set \(x\). This set contains the collection of vectors in Cartesian coordinates in the Sun centered ecliptic reference frame, and so can
be plotted directly in this reference frame. For additional clarity a number of plots showing the rotations for an imaginary transfer are shown in Figure 6.12 below.

Figure 6.12 Two successive rotations and one translation ensure the transfer is correct.

The first plot (top left) shows the transfer which is first obtained from the scalar distances of the objects from the Sun and the angle between the vectors they span with the Sun as origin ($r_1$ and $r_2$). This transfer lies within the $xy$ plane ($z = 0$). The first rotation is performed in the second plot (top right) such that the new transfer (in blue) lies within the plane spanned by vectors $r_1$ and $r_2$ (the plane that both objects and the Sun are in). The second rotation causes the new transfer (bottom left, in green) to be rotated such that it is correctly aligned with respect to the objects. The transfer plane is still the same plane as the plane spanned by vectors $r_1$ and $r_2$. The angle of this second rotation is determined by the angle between vectors $u_5$ and $u_6$ with the rotation center $C_R$ being the point where these two vectors intersect (these vectors are guaranteed to intersect as they are not parallel and lie in the same plane). See Figure 6.13 for a graphical representation of the vectors and rotation center.
Finally, cf. Figure 6.12 above (bottom right plot), a translation is performed such that the rotated transfer (in green) is translated towards the correct position, connecting it with the two objects. This transfer is pictured in red.

6.8 PROGRAM STRUCTURE

This chapter will discuss the software implementation of the analytical approximation of a low thrust transfer between arbitrary planets (and other objects) in the solar system using low thrust propulsion and swing-bys. Section 6.8.1 provides a succinct overview of the process within the program and a description of the inputs the user must provide. Section 0 provides a flow diagram showing the separate blocks of the program process. Function names and descriptions are provided in Appendix D.

6.8.1 PROGRAM OVERVIEW

The basic flow of the program is as follows: the user inputs desired settings and constants. This is passed on to the global optimizer, which begins to search for an optimum transfer based on the user’s input. Once done, this result is passed through to the local optimizer for further optimization. This (hopefully) leaves us with the parameters for an optimum transfer. The parameters that describe the transfer are then passed on to a separate function for plotting and interpreting the data. Finally, all data is stored. A graphic representation of this process is supplied in Figure 6.14.

We will now go into further detail by exploring the user inputs, the creation of a single individual, and the optimization process itself.
6.8.2 User Inputs

The user inputs a number of parameters into the program ranging from optimizer options to constants. By the far the most important parameter is the vector containing the number of objects to pass, followed by the relevant optimization variables (and their respective box constraints). An overview is provided in Table 6.2 below.

Object Vector This vector contains the object identification numbers. The length of this vector gives the number of legs in the transfer

\[ n_{\text{legs}} = \text{Length}(V_{\text{Obj}}) - 1 \]

and also the number of swing-bys

\[ n_{\text{swing -bys}} = n_{\text{legs}} - 1 \]

For example, the vector [3 4 3 2] represents an Earth Mars Earth Venus transfer and would consist of 3 legs and 2 swing-bys (at Mars and then at Earth).

Optimization Variables The number of variables that the optimizer must consider to arrive at an optimum point. Generally speaking, when there are more variables the problem becomes increasingly difficult to solve. The dimensionality of the problem refers to the total number of optimization variables. For this program the number of necessary variables can also be derived from the Object Vector by

\[ n_{\text{variables}} = 3(\text{Length}(V_{\text{Obj}}) - 1) + 1 \]

This is due to the fact that every transfer leg is determined by three optimization parameters:

- \( t_{\text{TOF}} \) The time of flight of this particular leg in the transfer.
- \( N \) The number of complete revolutions of this particular leg in the transfer.
- \( k_2 \) The winding parameter.

And one additional parameter, the initial time of departure, which is given once for the starting planet (usually Earth).

Box Constraints The user must input two vectors containing the lower and upper boundaries to search in. The vector lengths are determined by the number of optimization variables. For example, a single transfer from Earth to Mars would consist of a single leg with no swing-bys. Hence the number of optimization variables would be 4. A possible setup of boundaries would then be

Upper Boundary = \[ 0 \quad 50 \quad 0.01 \quad 0 \]
Lower Boundary = \[ 1000 \quad 500 \quad 1 \quad 2 \]

This searches for a transfer that starts between 0 and 1000 MJD2000 (Modified Julian Day since 01/01/2000, 12:00), has a transfer length between 50 and 500 days, has a standard \( k_2 \) range, and has between 0 and 2 complete revolutions around the Sun.

Spacecraft Parameters The user provides such information as solar panel area, efficiency, thrust efficiency, specific impulse etc. These may or may not be directly involved in the optimization (depending on user choice) but are certainly used for analysis after optimization has occurred.

Optimizer Settings The user provides optimizer settings to the program e.g. the display of generations and fitness values, the type of mutation, stopping parameters (such as tolerances and generation excesses), population settings, elitism, etc.

Constants Should the user wish to do so, constant parameters such as gravity at sea level and other gravitational constants may be changed or refined.

Solar Optimization If the design incorporates a transfer to an orbit (or any arbitrary point in space)
Implementation

Variables around the Sun three additional design parameters are passed on to the optimizer. Thus the number of variables becomes

\[ n_{\text{variables}} = 3(\text{Length}(V_{\text{obj}}) - 1) + 4 \]

These three parameters represent the position in a space around an object and are not to scale. The box constraints for these additional parameters vary from -1 to 1. Thus any point in a box with sides of 2 around the origin may be represented by a combination of these three parameters.

Table 6.2 An overview of the inputs a user is expected to provide to the program.
6.8.3 Flow Diagram

The diagram in Figure 6.15 shows the general process of the program, where the fitness is repeatedly evaluated by the optimizers in order to find a hopefully global minimum.

![Flow Diagram](image)

Figure 6.15 Flow diagram of the solution process involving a multi-leg optimization.

6.8.4 Parallel Processing

The Genetic Algorithm optimizer in Matlab supports parallel processing by splitting tasks such as computing fitness, performing genetic operations, and ranking individuals into separate threads. Significant gains in speed can be made by computing fitness values simultaneously in separate threads,
some amount of testing has shown that the speed increase from a single thread to four separate threads is slightly more than threefold. For instance, an optimization was brought down from 715 seconds to 236 seconds.
This chapter is an amalgam of topics such as validation of the software, sensitivity analysis, and optimizer tuning. There is also a section at the end of the chapter discussing perturbations, and why these may be neglected for a first-order approximation. Elements of the model should be tested and verified in order to ascertain the validity and usefulness of the model. Section 7.1 inspects the excess departure and arrival velocities (i.e. terminal velocities of the transfer) of an Earth-to-Mars transfer. Elements of the software (transfer shapes and time-of-flight curves) are compared to previously obtained results in section 7.2 to ensure validity of the model developed during this study. The results from an Earth-to-Mars transfer (obtained by the implementation of the model in GALOMUSIT) are compared to the results obtained during this study in section 7.3. Section 7.4 goes into the details of the robustness of the optimization and what can be done to aid the optimization process in finding the global optimum. Finally, section 7.5 demonstrates why the perturbations can be considered small enough to be neglected for an initial first-order analytical approximation.

7.1 TERMINAL VELOCITY (SENSITIVITY ANALYSIS)

One of the points of interest of the Exponential Sinusoids shape-based method is the terminal velocity and its behavior. This is inspected by performing an optimization for an Earth to Mars transfer, while placing maximum importance upon the arrival velocity by method of penalty.

An optimum transfer is found and plotted on the left in Figure 7.1. The 1500 kg spacecraft expends 138.2 kg of propellant and the fitness value including penalties is 139.9 kg. This would imply that the excess velocity is small, and indeed it is. The excess arrival velocity is only 25 m/s. The departure date is 10-05-2018 and the arrival date is 21-11-2019, resulting in a time of flight of 559.8 days. Further details are superfluous for this particular example. By holding all optimization parameters constant except for the time of flight (and thus by extension the arrival date), which is stepped through 40 days (in steps of 8) before and after the optimum time of flight that was found, the plot on the right is found in Figure 7.1.
Verification

Figure 7.1 Optimal Earth-Mars transfer (left) and multiple transfers where the time of flight is varied by 8 days.

An excess velocity is computed for each individual transfer (in the area of 40 days before and after the time of flight), the results of which are shown in Figure 7.2. When the time of flight is further decreased or increased the transfers begin to become invalid (i.e. no valid value for $k$ is found). To attain a valid transfer one would have to also change the other optimization parameters, making a comparison difficult if not impossible.

Figure 7.2 The magnitude of the excess arrival velocity as a function of time of flight.

It is immediately evident that the optimum shown in the figure coincides with the optimal time of flight calculated previously, demonstrating the effectiveness of the optimization. The change in the excess velocity magnitude is almost, but not entirely, linear. It is also interesting to see how the excess departure velocity evolves as this is also a design parameter for the transfer; this is shown in Figure 7.3.
Figure 7.3 The magnitude of the excess departure velocity as a function of time of flight.

The departure excess velocity becomes more optimal as the time of flight increases. Both departure and arrival excess velocity plots may be superposed to inspect the ‘total excess’ optimum. The plot of this is shown below in Figure 7.4.

Figure 7.4 Total, departure, and arrival excess velocity as a function of time of flight.

The optimum point at 559.8 days remains unmoved, perhaps a sign that the arrival excess velocity magnitude is the driving force for the total optimality of the excess velocities for a single leg transfer. To illustrate a potential weakness in the Lambert solution of the Exponential Sinusoid the excess departure and arrival velocities are given again, but now broken down into their components, in Figure 7.5 below.

Figure 7.5 x component (black), y component (blue), and z component (green) of the excess departure (left) and excess arrival (right) velocities.
The $x$, $y$, and $z$ components are shown in their respective colors (see caption). The $z$ component is almost constant in this particular region of the time of flight (and for these positions of both planets). This is due to Mars’ out of plane motion with respect to Earth’s in-plane motion (in the ecliptic). Since the Lambert solution describes a planar motion in the plane spanned by the Sun – Earth and the Sun – Mars vectors any difference in out-of-plane motion (the $z$ component of the velocity) must be introduced in the excess departure and arrival velocity vectors. This means that there will almost always be an out-of-plane excess departure (and arrival) velocity, depending on the positions of the involved planets. Luckily, the optimizer can be instructed to consider and weigh this penalty when it searches for an optimal solution.

It is possible to solve a shape based method, such as Exponential Sinusoids, with an out-of-plane motion. However, this form of solution also has its own particular drawbacks as described in, among others, *Lambert’s Problem for Exponential Sinusoids* [Izzo, 2006].

### 7.2 Previous Results (Validation)

An independent investigation is performed to the work in *Lambert’s Problem for Exponential Sinusoids* [Izzo, 2006], and also to verify the results in the thesis *Low-Thrust Orbits for Interplanetary Transfers and Implementation in GALOMUSIT* [Corradini, 2008].

#### 7.2.1 Verifying Izzo, 2006

*Lambert’s Problem for Exponential Sinusoids* [Izzo, 2006] contains a number of figures that will serve as a benchmark for the validity of the created code. The feasible exponential sinusoids of the classes $S_{1/2}[1, 5, \pi/2, 0]$ and $S_{1/4}[1, 5, \pi/2, 1]$ have been plotted in Figure 7.6, and show the same results as in the paper.

![Figure 7.6 The feasible exponential sinusoids in the classes $S_{1/2}[1, 5, \pi/2, 0]$ (left) and $S_{1/4}[1, 5, \pi/2, 1]$ (right).](image)

The numbers of the class $S_{1/2}[1, 5, \pi/2, 0]$ entail that $k_2$ is equal to $1/2$, the initial and final radii are 1 and 5, the transfer angle is $\pi/2$, and the number of complete revolutions is 0. The time-of-flight curves are also inspected for the curves in the family of $S_{1/12}[1, 1.5, \pi/2, N]$, where a curve is plotted for $N$ is equal to 0 through 5 (the time of flight is computed in seconds using a gravitational constant $\mu = 1$, and then normalized by dividing the time of flight by $2\pi$ [Izzo, 2006]). These results (Figure 7.7) also match the figure in [Izzo, 2006].
From the above figures we may conclude that the basis of the program is sound, having obtained results identical to the work in [Izzo, 2006].

The paper *Lambert’s Problem for Exponential Sinusoids* [Izzo, 2006] only shows time-of-flight curves that increase monotone with the initial flight path angle $\gamma_1$, giving the impression that there is at most one single intersection between time of flight and initial flight path angle (cf. Figure 7.7). In the next section, it will be demonstrated that there are cases of a monotone decreasing curve, and that there are even cases where a single time of flight may be attributed to two values of the initial flight path angle. Further results in [Corradini, 2008] can now be examined.

### 7.2.2 Verifying Corradini, 2008

This section will inspect the validity of the implementation in GALOMUSIT. [Corradini, 2008] identifies the fact that the time of flight curves are not always monotone increasing, but did not mention the cause for this behavior. To find this cause, let us now examine an example of monotone decreasing behavior. The normalized time of flight and trajectory shapes of the class $S_{9/10}[1, 1.5, \pi/2, 1]$ are plotted in Figure 7.8 below.
Verification

It can indeed be seen that this curve decreases in a monotone fashion, identical to the results in Stefano Corradini’s work [Corradini, 2008]. However, this curve is certainly a valid one, and not the result of some numerical aberration. This can be seen when a plot of the trajectory shapes is examined, as in the right of Figure 7.8, where four possible trajectories in this class (thus corresponding to four initial flight path angles) are shown. It can be seen that the smallest trajectory (prescribing a motion nearest to the Sun) corresponds to the larger initial flight path angle $\gamma_1$ (in this case also the maximum initial flight path angle) while the smallest shown $\gamma_1$ (in this case the minimum initial flight path angle) prescribes a much larger trajectory shape, which is not entirely shown in the plot. Thus it can be seen that when one generates trajectories with increasing initial flight path angle $\gamma_1$ that the trajectory shape actually becomes smaller, corresponding to a decreasing time of flight.

Now, an example where a single time of flight may correspond to two different initial flight path angles is presented. The normalized time of flight and trajectory shapes of the class $S_{v10}[1, 1.5, \pi/2, 3]$ are plotted in Figure 7.9 below.

![Normalized time of flight and trajectory shapes of the class $S_{v10}[1, 1.5, \pi/2, 3]$.](image)

Figure 7.9 Normalized time of flight and trajectory shapes of the class $S_{v10}[1, 1.5, \pi/2, 3]$.

This figure now shows a time-of-flight curve that is no longer monotone, and that has two asymptotes instead of one. Although this behavior seems strange at first, an inspection of the trajectory shapes yields the explanation. Three trajectory shapes are shown. While the shape with a high initial flight path angle (dashed line) and the shape with a low initial flight path angle (dotted line) are fairly large, the shape with the central flight path angle (solid line) is clearly the smallest, corresponding to a smaller time of flight. Figure 7.10 shows an additional two trajectory shapes, one with relatively high and one with relatively low $\gamma_1$, that have large shapes that extend much farther than the transfers in Figure 7.9. Naturally, these large shapes correspond to increased times of flight.
Verification

Figure 7.10 Two trajectory shapes, one with a relatively high initial flight path angle and one with a relatively low angle, of the class $S_{4/10}[1, 1.5, \pi/2, 3]$.

Another example of a case where a single time of flight may correspond to two different initial flight path angles is the class of shapes $S_{6/10}[1, 1.5, \pi/2, 2]$. The normalized time of flight and trajectory shapes of this class are plotted in Figure 7.11 below.

Figure 7.11 Normalized time of flight and trajectory shapes of the class $S_{6/10}[1, 1.5, \pi/2, 2]$.

The same behavior can be explained with the help of the trajectory shapes. Figure 7.11 above shows three shapes, one shape with an intermediate flight path angle (solid line), one with a higher value of the initial flight path angle (dashed line), and one with a lower value of the initial flight path angle (dotted line). Again the smaller shape with intermediate $\gamma_1$ is obviously the case with the smallest time of flight, while the other two shapes have longer times of flight.

All of the above time-of-flight curves are identical to the curves reported in [Corradini, 2008]. It has now been shown that these curves are not erroneous but an intrinsic characteristic of the time of flight as a function of the initial flight path angle.

7.3 Earth-Mars Transfer (Validation)

In this section the program is tested against the results of an Earth to Mars transfer obtained in Low Thrust Orbits for Interplanetary Transfers and Implementation in Galomusit [Corradini, 2008]. These results will primarily serve as an independent test to verify the implementation, but it will also be nice to compare performance. All results in the report were computed with a spacecraft mass of 1,000 kg and a
specific impulse of 3,000 s, so naturally the settings will be identically set for comparisons sake. Two types of optimization were performed, one with no constraints placed upon the terminal velocities and one with constraints. Let us begin by comparing the unconstrained model.

### 7.3.1 UNCONSTRAINED MODEL

[Corradini, 2008] found a best solution at 14.170 kg propellant use for the unconstrained model. On average the published results were between 14 and 15 kg of propellant. The launch date was limited to dates between the 1st or January 2015 and the 1st of January 2025. The results were obtained with an initial population of 1000 individuals. Unfortunately, no computation times were provided in the report. The details of the best result are provided in Table 7.1 directly below.

<table>
<thead>
<tr>
<th>Global fitness</th>
<th>Local Fitness</th>
<th>Time of flight</th>
<th>Global time</th>
<th>Local time</th>
<th>Excess Departure</th>
<th>Excess Arrival</th>
</tr>
</thead>
<tbody>
<tr>
<td>14.170 kg</td>
<td>N/A</td>
<td>39.67 d</td>
<td>?</td>
<td>N/A</td>
<td>16.106 km/s</td>
<td>14.287 km/s</td>
</tr>
</tbody>
</table>

Table 7.1 Best result for an Earth to Mars transfer using the unconstrained model from [Corradini, 2008].

Setting the spacecraft and search space parameters identical, and using a much smaller population size of 100 individuals (10 times less than above), gives us very similar results. Five optimization runs were performed, yielding the results shown in Table 7.2. The global and local fitness columns show the fitness value after global (Genetic Algorithm) and local (Nelder-Mead) optimization, respectively. The global and local time columns represent the computation times for the global and local optimization, respectively*

<table>
<thead>
<tr>
<th>Global fitness</th>
<th>Local Fitness</th>
<th>Time of flight</th>
<th>Global time</th>
<th>Local time</th>
<th>Excess Departure</th>
<th>Excess Arrival</th>
</tr>
</thead>
<tbody>
<tr>
<td>14.580 kg</td>
<td>14.419 kg</td>
<td>76.93 d</td>
<td>25 s</td>
<td>5 s</td>
<td>7.830 km/s</td>
<td>11.825 km/s</td>
</tr>
<tr>
<td>14.445 kg</td>
<td>14.412 kg</td>
<td>76.33 d</td>
<td>38 s</td>
<td>19 s</td>
<td>7.797 km/s</td>
<td>11.794 km/s</td>
</tr>
<tr>
<td>14.903 kg</td>
<td>14.422 kg</td>
<td>76.50 d</td>
<td>32 s</td>
<td>5 s</td>
<td>7.799 km/s</td>
<td>11.795 km/s</td>
</tr>
<tr>
<td>14.766 kg</td>
<td>14.442 kg</td>
<td>74.70 d</td>
<td>109 s</td>
<td>8 s</td>
<td>7.735 km/s</td>
<td>11.732 km/s</td>
</tr>
<tr>
<td><strong>13.546 kg</strong></td>
<td><strong>13.546 kg</strong></td>
<td>29.67 d</td>
<td><strong>99 s</strong></td>
<td><strong>18 s</strong></td>
<td><strong>20.772 km/s</strong></td>
<td><strong>21.129 km/s</strong></td>
</tr>
</tbody>
</table>

Table 7.2 Five results for a fully unconstrained (no penalties) Earth to Mars transfer.

The local optimum at around 14.4 kg propellant use proves to be an incredibly attractive local optimum for the optimizer to fall into. This is likely helped by the fact that the algorithm has a tolerance for errors of up to 1 day while looking for the appropriate time of flight. Because of this, around the 76 days mark the algorithm can no longer really distinguish between a 76.2 and a 76.3 day transfer. Nevertheless, a better result was found using 13.6 kg of propellant. Of course this improved result may just be down to slightly different ephemerides or the fact that the propellant use is solved using an ODE instead of an integral (that assumes constant mass during the transfer, cf. section 3.5). The transfer using 13.6 kg of propellant is shown in Figure 7.12.

---

* Computations performed on an Intel I7 920 @ 4 Ghz.
Figure 7.12 Unconstrained ‘optimal’ Earth to Mars transfer (left), mass evolution (top right), and thrust evolution (bottom right).

The a part of search space surrounding this area is plotted in Figure 7.13 below.

Figure 7.13 The search space of the unconstrained Earth to Mars transfer surrounding the found optimum of 13.6 kg.

The white dot in the left plot and the right vertical line in the right plot of Figure 7.13 above show the discovered optimum of 13.6 kg. The search space itself looks like a valley, where the floor of the valley is moving slightly downwards (i.e. becoming more optimal) along the time of flight in negative direction.

The astute reader will note the huge velocity mismatches (10+ km/s) for the unconstrained case. This is because the optimization process is basically seeking the shortest route where it can still fit an exponential shape with a valid initial flight path angle between the two planets. Any velocity mismatches are left unconsidered so these become extremely large (the problem of optimal propellant transfer could probably be approached by simply maximizing these velocity mismatches, as this will lead to minimum propellant use during actual transfer). Somewhat amusingly, a line would in fact be a better shape equation than the exponential sinusoid shape for this problem, prescribing an instant velocity change at the Earth with a coasting period using no propellant and another instant velocity change at Mars. We will now examine a much more realistic scenario, where the terminal velocities are of importance.
7.3.2 CONSTRANDED MODEL

The constrained model (where constraints by method of penalties are placed upon the boundary velocities of the transfer shape) is tested using the exact same parameters of specific impulse, spacecraft mass, launch dates, etc. The best results in [Corradini, 2008] are given in Table 7.3 below.

<table>
<thead>
<tr>
<th>Propellant</th>
<th>Time of Flight</th>
<th>Excess Departure</th>
<th>Excess Arrival</th>
</tr>
</thead>
<tbody>
<tr>
<td>200.732 kg</td>
<td>716.044 days</td>
<td>0.903 km/s</td>
<td>0.466 km/s</td>
</tr>
<tr>
<td>197.354 kg</td>
<td>712.365 days</td>
<td>0.913 km/s</td>
<td>0.362 km/s</td>
</tr>
<tr>
<td>215.350 kg</td>
<td>756.416 days</td>
<td>1.133 km/s</td>
<td>0.091 km/s</td>
</tr>
<tr>
<td>195.515 kg</td>
<td>682.224 days</td>
<td>0.984 km/s</td>
<td>0.214 km/s</td>
</tr>
</tbody>
</table>

Table 7.3 The best results obtained from [Corradini, 2008].

An effort is made to replicate these results. To study the sensitivity of the program to different penalty functions ten optimization results are given in Table 7.4. Each of these results is obtained using a different set of penalty functions (5 optimization runs are performed per set of penalty functions to avoid outliers). The global fitness column represents the fitness value (propellant mass and penalty value) after global optimization (Genetic Algorithm) and the local fitness column represents the fitness value after local optimization (Nelder-Mead Method). Note that these can be different from the corresponding propellant mass values, as penalty functions have now been introduced.

<table>
<thead>
<tr>
<th>Global fitness</th>
<th>Local fitness</th>
<th>Propellant Mass</th>
<th>Time of Flight</th>
<th>Global time</th>
<th>Local time</th>
<th>Excess Departure Velocity</th>
<th>Excess Arrival Velocity</th>
</tr>
</thead>
<tbody>
<tr>
<td>153.12</td>
<td>148.14</td>
<td>148.14 kg</td>
<td>559.11 d</td>
<td>21 s</td>
<td>7 s</td>
<td>2.08 km/s</td>
<td>6.9·10^{-14} km/s</td>
</tr>
<tr>
<td>233.64</td>
<td>216.36</td>
<td>148.14 kg</td>
<td>559.11 d</td>
<td>12 s</td>
<td>7 s</td>
<td>2.08 km/s</td>
<td>2.5·10^{-14} km/s</td>
</tr>
<tr>
<td>381.65</td>
<td>381.47</td>
<td>153.87 kg</td>
<td>581.14 d</td>
<td>45 s</td>
<td>22 s</td>
<td>0.42 km/s</td>
<td>0.87 km/s</td>
</tr>
<tr>
<td>606.18</td>
<td>606.17</td>
<td>179.66 kg</td>
<td>608.73 d</td>
<td>26 s</td>
<td>6 s</td>
<td>8.8·10^{-14} km/s</td>
<td>4.27 km/s</td>
</tr>
<tr>
<td>11894</td>
<td>10228</td>
<td>136.33 kg</td>
<td>472.8 d</td>
<td>13 s</td>
<td>9 s</td>
<td>0.499 km/s</td>
<td>1.627 km/s</td>
</tr>
<tr>
<td>232.25</td>
<td>202.87</td>
<td>113.06 kg</td>
<td>442.40 d</td>
<td>27 s</td>
<td>6 s</td>
<td>3.056 km/s</td>
<td>0.180 km/s</td>
</tr>
<tr>
<td>225.35</td>
<td>219.16</td>
<td>108.23 kg</td>
<td>441.43 d</td>
<td>28 s</td>
<td>7 s</td>
<td>2.579 km/s</td>
<td>0.270 km/s</td>
</tr>
<tr>
<td>342.21</td>
<td>298.08</td>
<td>113.10 kg</td>
<td>447.02 d</td>
<td>27 s</td>
<td>7 s</td>
<td>2.969 km/s</td>
<td>0.178 km/s</td>
</tr>
<tr>
<td>339.06</td>
<td>327.63</td>
<td>112.64 kg</td>
<td>463.42 d</td>
<td>13 s</td>
<td>11 s</td>
<td>2.476 km/s</td>
<td>0.268 km/s</td>
</tr>
<tr>
<td>338.36</td>
<td>328.78</td>
<td>151.93 kg</td>
<td>363.09 d</td>
<td>19 s</td>
<td>4 s</td>
<td>0.409 km/s</td>
<td>1.078 km/s</td>
</tr>
</tbody>
</table>

Table 7.4 Results using a constrained model for the Earth to Mars transfer.

The first two results have basically obtained the same values where the excess arrival velocity is essentially negligible. This is due to a heavy penalty function placed upon this variable. The third result was obtained using balanced penalty functions on both arrival and departure velocities, while the fourth result is the outcome of a heavy penalty placed upon the excess departure velocity. The other results are various combinations of lighter penalties resulting in less propellant use but higher excess velocities.

The results clearly show that it is possible to steer at least one of either excess velocities to an almost zero value, and that a balance of both can be achieved at slightly more than 1 km/s. Figure 7.14 shows both transfers with extremely low excess arrival velocity (left) and extremely low excess departure velocity (right).
It is worthwhile to mention that the left $xz$ plane plot in Figure 7.14 above shows the transfer in the plane of Mars’ orbit (and as a result being able to most effectively control the excess arrival velocity) while the right $xz$ plane plot shows the transfer to be in the plane of the Earth’s orbit (therefore minimizing excess departure velocity the most). Because the transfer shape itself is 2-dimensional it cannot account for the change in $z$ velocity between both planets in the thrust profile, and thus this difference can only be accounted for at the departure and arrival points. The thrust profiles for both these transfers are very similar and can be seen in Figure 7.15 below.

A part of the search space around the discovered optimal excess arrival velocity transfer is inspected in the following group of plots in Figure 7.16.
The above figure demonstrates the change a single linear penalty function can make to the search space, areas that were previously more optimal (e.g. around a launch date of 6600 MJD2000 and a transfer time of 200 days) have become less so, and vice versa. This same search space is plotted, with the propellant mass being replaced by the excess arrival velocity at Mars. The results are shown in Figure 7.17.
Figure 7.17 Excess arrival velocities surrounding the discovered optimum.

The plot shows the high values the excess arrival velocity can reach (up to roughly 20 km/s surrounding the discovered optimum).

Finally, we may conclude that the results obtained here in this section have improved upon the results listed in [Corradini, 2008]. When comparing the results that manage to lower excess arrival velocity the most the propellant use is improved from 215 kg to 148 kg, the time of flight is improved from 756 days to 559 days, and the already small excess arrival velocity of 91 m/s is reduced to effectively 0 m/s. The departure velocity however, has almost doubled from 1.13 km/s to 2.08 km/s.

7.4 OPTIMIZATION ROBUSTNESS (TUNING)

To test the robustness of the optimization a trial transfer consisting of 3 legs (and thus 2 swing-bys), that passes Venus and Mars and returns to Earth (the sequence is Earth-Venus-Mars-Earth), is examined. As the transfer consists of 3 legs the number of design parameters equals 10 (a single departure time design parameter and 3 legs, each described by 3 design parameters). The initial mass of the spacecraft is chosen to be 500 kg and the specific impulse is 5000 s. The search space boundaries allow for a departure date between 4,000 and 8,000 MJD2000 and each individual leg of the transfer may be between 100 and 2,000 days (with a maximum of 3 completed revolutions). The problem is constrained using penalty functions in order to move the solution towards reduced departure and arrival velocity mismatches.

The inspection is begun by completing five optimization runs with the goal to minimize propellant mass (with random seed numbers) using a population of only 100 individuals, the results of which are shown in Table 7.5. The columns Global fitness and Local fitness denote the fitness value that was found by the global - and local optimizer, respectively. The columns Global time and Local time denote the time the computation took to complete itself.
Verification

<table>
<thead>
<tr>
<th>Global fitness</th>
<th>Local fitness</th>
<th>Propellant mass [kg]</th>
<th>Time of flight [d]</th>
<th>Global time [s]</th>
<th>Local time [s]</th>
<th>Excess departure velocity [km/s]</th>
<th>Excess arrival velocity [km/s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>528.93</td>
<td>528.85</td>
<td>189.65</td>
<td>2308.2</td>
<td>100</td>
<td>86</td>
<td>2.34</td>
<td>1.05</td>
</tr>
<tr>
<td>557.68</td>
<td>555.48</td>
<td>199.38</td>
<td>2631.2</td>
<td>238</td>
<td>131</td>
<td>3.19</td>
<td>0.41</td>
</tr>
<tr>
<td>1018.68</td>
<td>603.19</td>
<td>221.31</td>
<td>1994.0</td>
<td>123</td>
<td>207</td>
<td>2.15</td>
<td>1.68</td>
</tr>
<tr>
<td>617.05</td>
<td>606.98</td>
<td>180.41</td>
<td>2391.4</td>
<td>399</td>
<td>183</td>
<td>3.45</td>
<td>0.86</td>
</tr>
<tr>
<td>639.90</td>
<td>639.87</td>
<td>181.06</td>
<td>2274.8</td>
<td>231</td>
<td>30</td>
<td>1.95</td>
<td>2.64</td>
</tr>
</tbody>
</table>

Launch Date Swing-by 1 Velocity Mismatch Swing-by 2 Velocity Mismatch

<table>
<thead>
<tr>
<th>Launch Date</th>
<th>Swing-by 1</th>
<th>Swing-by 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>MJD2000</td>
<td>GCD</td>
<td>GCD</td>
</tr>
<tr>
<td>4831.4</td>
<td>24-03-2013</td>
<td>6.37-10^{-13} km/s</td>
</tr>
<tr>
<td>7263.8</td>
<td>21-11-2019</td>
<td>1.07-10^{-5} km/s</td>
</tr>
<tr>
<td>4148.1</td>
<td>11-05-2011</td>
<td>3.92-10^{-9} km/s</td>
</tr>
<tr>
<td>7981.1</td>
<td>07-11-2021</td>
<td>1.44-10^{-9} km/s</td>
</tr>
<tr>
<td>5482.2</td>
<td>04-01-2015</td>
<td>2.56-10^{-9} km/s</td>
</tr>
</tbody>
</table>

Table 7.5 Optimization results for an Earth-Venus-Mars-Earth transfer using a population of 100 individuals.

The solutions found are quite varied, having different propellant masses, excess velocities, and launch dates. The third result is quite interesting, as the local optimization method was almost able to improve fitness twofold (from 1019 to 603). The other cases show a less drastic improvement. A population of 100 individuals is rather small to be able to obtain a nice spread covering 10 design parameters so we increase the number of individuals to 200, perform 5 additional optimization runs and observe the results, shown in Table 7.6 below.

<table>
<thead>
<tr>
<th>Global fitness</th>
<th>Local fitness</th>
<th>Propellant mass [kg]</th>
<th>Time of flight [d]</th>
<th>Global time [s]</th>
<th>Local time [s]</th>
<th>Excess departure velocity [km/s]</th>
<th>Excess arrival velocity [km/s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>338.12</td>
<td>338.05</td>
<td>199.27</td>
<td>2971.3</td>
<td>217</td>
<td>23</td>
<td>0.53</td>
<td>0.85</td>
</tr>
<tr>
<td>451.69</td>
<td>447.81</td>
<td>187.40</td>
<td>2232.0</td>
<td>156</td>
<td>56</td>
<td>0.84</td>
<td>1.75</td>
</tr>
<tr>
<td>516.55</td>
<td>492.56</td>
<td>228.13</td>
<td>2308.0</td>
<td>266</td>
<td>43</td>
<td>1.02</td>
<td>1.62</td>
</tr>
<tr>
<td>548.58</td>
<td>547.94</td>
<td>213.30</td>
<td>2442.8</td>
<td>128</td>
<td>59</td>
<td>2.21</td>
<td>1.14</td>
</tr>
<tr>
<td>593.30</td>
<td>592.15</td>
<td>225.32</td>
<td>2336.5</td>
<td>159</td>
<td>61</td>
<td>2.03</td>
<td>1.64</td>
</tr>
</tbody>
</table>

Launch Date Swing-by Velocity 1 Mismatch Swing-by 2 Velocity Mismatch

<table>
<thead>
<tr>
<th>Launch Date</th>
<th>Swing-by 1</th>
<th>Swing-by 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>MJD2000</td>
<td>GCD</td>
<td>GCD</td>
</tr>
<tr>
<td>7039.4</td>
<td>10-04-2019</td>
<td>4.46-10^{-13} km/s</td>
</tr>
<tr>
<td>5957.2</td>
<td>23-04-2016</td>
<td>2.52-10^{-8} km/s</td>
</tr>
<tr>
<td>4261.6</td>
<td>02-09-2011</td>
<td>178-10^{-4} km/s</td>
</tr>
<tr>
<td>7437.8</td>
<td>13-05-2020</td>
<td>4.44-10^{-15} km/s</td>
</tr>
<tr>
<td>6748.0</td>
<td>23-06-2018</td>
<td>3.01-10^{-7} km/s</td>
</tr>
</tbody>
</table>

Table 7.6 Optimization results for an Earth-Venus-Mars-Earth transfer using a population of 200 individuals.

It can be seen that average fitness has improved in this sample compared to the previous sample obtained using 100 individuals. The actual propellant use has not really been reduced, but rather the velocity mismatches at the start and end of the transfer. Let us now obtain one final sample (of 5 runs) by further increasing the number of individuals in the population to 400.
Table 7.7 Optimization results for an Earth-Venus-Mars-Earth transfer using a population of 400 individuals.

<table>
<thead>
<tr>
<th>Launch Date</th>
<th>Swing-by Velocity 1 Mismatch</th>
<th>Swing-by Velocity 2 Mismatch</th>
</tr>
</thead>
<tbody>
<tr>
<td>4956.7 MJD2000 28-07-2013 GCD</td>
<td>8.87·10^{-13} km/s</td>
<td>6.28·10^{-10} km/s</td>
</tr>
<tr>
<td>6718.0 MJD2000 24-05-2018 GCD</td>
<td>3.25·10^{-6} km/s</td>
<td>9.37·10^{-9} km/s</td>
</tr>
<tr>
<td>4923.8 MJD2000 25-06-2013 GCD</td>
<td>2.88·10^{-15} km/s</td>
<td>1.63·10^{-12} km/s</td>
</tr>
<tr>
<td>4786.4 MJD2000 07-02-2013 GCD</td>
<td>1.33·10^{-7} km/s</td>
<td>2.54·10^{-8} km/s</td>
</tr>
<tr>
<td>4513.7 MJD2000 11-05-2011 GCD</td>
<td>8.88·10^{-16} km/s</td>
<td>3.11·10^{-15} km/s</td>
</tr>
</tbody>
</table>

The results show that the average fitness has improved, and the best result from the previous sample has slightly improved.

It is quite possible that further improvements can be made by (1) increasing the population size even more, (2) adding more mutation individuals, (3) implementing a more aggressive mutation method where mutation is performed relatively more often further into the optimization (later generations), and (4) a different set of penalty functions.

It is important to set all stopping conditions to be relatively stringent to ensure the optimization will not prematurely stop before a good optimum has been found. Often with stringent stopping conditions, the Genetic Algorithm will spend quite some time improving the fitness in an already located optimum basin, which could be done much quicker by a local method. However, it is very difficult to find satisfactory values of the stopping conditions for a variety of problems such that the genetic algorithm stops at the right time and lets the local optimization method take over (in order to save computation time). Some improvement in certain cases was found when the percentage of individuals within the population that undergo mutation every generation was increased. Additionally, changing the shrink parameter (in order to maintain a stronger than default mutation strength as generations go by) of the mutation function seems to be of beneficial influence for most problems, such that the optimization will have more success finding better optima without increasing population size.

One can continue almost indefinitely with fine-tuning a Genetic Algorithm, trying to strike the desired balance between thoroughness and computation time, and between the various penalties imposed upon the design. For example, the above samples may have been more optimal if departure, and especially arrival, excess velocities had been punished more severely. This may have led to transfers that use more propellant but with improved excess velocities. Perhaps only increasing the arrival penalty would have given transfers with a tolerable departure excess, good propellant performance, and excellent arrival excess velocities. It is difficult to say without extensive experimentation.

Before moving on to an examination of the search space a graphical overview of the two best transfers is provided in Figure 7.18 below, where the left plots represent the transfer with 337 fitness and the right plots represent the transfer with 338 fitness.
Verification

Figure 7.18 EVME transfer of fitness 337 (left) and 338 (right) with transfer plot (top), mass evolution (center), and thrust profile (bottom).

In the mass evolution plots, the blue dots represent the spacecraft final mass as found by the optimizer, and the red dot represents the fitness value subtracted from the initial spacecraft mass (the vertical difference between the red and blue dots represent the sum of the penalty values). Both transfers were found at widely different launch dates, 28-07-2013 (left) and 10-04-2019 (right).

Both transfers share a similar shape, where the spacecraft goes significantly outside of Mars’ orbit when moving towards the Earth, presumably to better match the Earth’s velocity at arrival. Although the left transfer has a slightly better fitness, it could be argued that the right transfer is more optimal. It requires less propellant (roughly 20 kg), its thrust profile has slightly smaller extrema, and its arrival velocity mismatch is also slightly better (but its departure velocity mismatch is greater, and in addition its time of flight is almost 400 days greater).

The parameters of the transfer with fitness 337 are varied and stepped through to get an idea of the search space for this particular problem. Figure 7.19 shows how the optimizer sees the search space, complete with penalties. The space between each point in the figure is 10 days (i.e. the nodes on the grid are spaced at 10 day intervals).
Figure 7.19 The search space fitness based on the 337 fitness transfers with 10 design parameters, the above figures vary the parameters related to time (see each plot for details).

The excess arrival velocity is plotted in Figure 7.20, using the same grid as Figure 7.19.
Figure 7.20 The search space arrival velocity based on the 337 fitness transfers with 10 design parameters, the above figures vary the parameters related to time (see each plot for details).
Verification

Some of the search spaces are very uneven, especially the ones related to the departure date. When the lowest fitness on the grid with 10 day spacing is sought out we find – for the plots in Figure 7.19 from top to bottom, left to right – 635.7, 644.9, 600.3, 518.8, 381.9, and 416.7. Only one of these values is close to the found fitness of 337. We may therefore conclude that a much finer grid mesh would be necessary in order to find a suitable solution in this manner (using a simple grid search).

Three more plots that show the search space defined by each pair of $N$ and $k_2$, along with a plot of the excess arrival velocity as a function of the pair $N$ and $k_2$ that define the final leg, are given below in Figure 7.21.

Figure 7.21 The search space on the 337 fitness transfers with parameters related to $N$ and $k_2$ (see each plot for details).

These plots are much smoother than the others pertaining to the temporal parameters. The three strips on each plot correspond to each integer value of $N$.

Finally, it is interesting to see the effect (to the search space) of declaring transfers that exceed swing-by angle as invalid (by setting them to infinite). The first two plots in Figure 7.19 above are plotted once more from a top viewpoint on the left of Figure 7.22.
These plots show the drastic effect this action has upon the search space, where huge swaths of it become invalid (i.e. uncolored). Although the plots on the right of Figure 7.22 are more realistic (a transfer actually is completely invalid when a swing-by exceeds the maximum angle change a planets gravity can bring about), it is better to let the optimizer work on the search space on the left of Figure 7.22. This is because the optimization process has a much more difficult time converging when there are large discontinuities inside the search space. Generally speaking, it is unwise to set any penalties to infinite. A much better solution is to impose a heavy constant penalty or a very steep linear penalty (a constant penalty of 50,000 kg was used in this particular example).

7.5 PERTURBATIONS

While the analytical shaped-based method takes no perturbations (except thrust) into account at all, it is useful to inspect their magnitude in order to see which would be most important during further studies. An orbital perturbation is an alteration to the orbital path by a force, other than the gravitational force enacted by the primary point mass (or a perfectly radial symmetric sphere), that acts upon the orbiting
object. Perturbations can be caused by gravity field effects (objects are not perfect spheres in reality, see section 7.5.1). Other objects (that may be represented as point masses) of lesser gravitational influence can also be considered as a perturbation, e.g. the gravitational influence of Jupiter on an Earth orbiting satellite. These are referred to as third-body perturbations (section 7.5.2). Further perturbations can be caused by gravity field effects (objects are not perfect spheres in reality, see section 7.5.1), atmospheric effects (section 7.5.4), radiation (radiation emitted from one object enacts a pressure on other objects, see 7.5.3), electromagnetic effects (some objects possess an electromagnetic field which can influence the path of an electrically charged object, see 7.5.5), and relativistic effects (section 7.5.6). The various effects are investigated in further detail in this chapter. Finally, section 7.5.7 examines how these perturbations might influence the trajectory design.

7.5.1 Gravity Field

In reality, objects (such as planets) are not perfect spheres. In certain cases, irregularities in the gravity potential must be taken into account when determining the orbital path of a spacecraft.

**Qualitative Analysis**

The generic notation for the acceleration of an object in the n-body problem assumes the total mass of an object to be concentrated at the center of the object (i.e. a point mass). In reality, objects are neither perfect spheres nor point masses. Instead, their mass is distributed. The equation is adapted by considering a single point mass

$$\ddot{r} = -G \frac{M}{r^3} r$$

(7.1)

This acceleration can also be represented with the gradient of the corresponding gravity potential.

$$\ddot{r} = \nabla U, \text{ where } U = G M \frac{r}{r}$$

(7.2)

The expression for the potential can be generalized to an arbitrary mass distribution by summing up the contributions of every individual mass element (see Figure 7.23) [Montenbruck, et al., 2005].

$$U = \int \frac{dm}{|r - s|} = \int \frac{\rho(s) d^3s}{|r - s|}$$

(7.3)

where

$$\rho(s)$$ \text{ is the density at point } s \text{ inside the object.}

$$|r - s|$$ \text{ is the satellite’s distance from point } s
The integral is evaluated by expanding the inverse of the distance in a series of Legendre polynomials. For a more detailed derivation the reader is referred to [Montenbruck, et al., 2005]. After expansion in spherical harmonics the gravity potential can be written in spherical coordinates as

$$U = \frac{GM}{r} \sum_{n=0}^{\infty} \sum_{m=0}^{n} \left(\frac{R}{r}\right)^{n} P_{n,m}(\sin \delta)(C_{n,m} \cos(n\lambda) + S_{n,m} \sin(m\lambda))$$  \hspace{1cm} (7.4)

This result only holds for $r > s$, i.e. for points outside the object. The equation contains two coefficients, $C_{n,m}$ and $S_{n,m}$, which describe the dependence on the internal mass distribution of the object. In equation (7.4) the Legendre polynomial $P_{n,m}$ of degree $n$ and order $m$ is

$$P_{n,m}(u) = (1 - u^2)^{m/2} \frac{d^m}{du^m} P_n(u)$$  \hspace{1cm} (7.5)

Typical names for coefficients with certain $n$ and $m$ values are given below

- $S_{n,0}$: All $S_{n,0}$ terms are zero due to their definition.
- $C_{n,0}$: Coefficients with $m = 0$ are called zonal coefficients, as they describe the potential which is independent of longitude. The remaining zonal terms are often noted as $f_n = -C_{n,0}$.
- $m < n$: Terms with $m < n$ are named tesseral coefficients.
- $m = n$: Terms with $m = n$ are named sectorial coefficients.

Equation (7.4) can be rewritten to a more commonly used form in astrodynamics, with the help of the following Legendre polynomials

- $n = 0, m = 0 \rightarrow P_{n,m}(\sin \delta) = 1$
- $n = 1, m = 0 \rightarrow P_{n,m}(\sin \delta) = \sin \delta$
- $n = 1, m = 1 \rightarrow P_{n,m}(\sin \delta) = \cos \delta$

This leads to [Wakker, 2002]

$$U = \frac{\mu}{r} \left[ 1 - \sum_{n=2}^{\infty} \sum_{m=1}^{n} J_{n,m} \left(\frac{R}{r}\right)^{2} P_{n}(\sin \delta) \right]$$  \hspace{1cm} (7.6)

This form shows the gravity field separated into three parts. The first part corresponds to the point mass, the second part represents the zonal $J_n$ terms, and the third part shows the sectorial and tesseral $J_{n,m}$ terms.

The Legendre polynomial $P_n$ of degree $n$ is

$$P_n(u) = \frac{1}{2^n n!} \frac{d^n}{du^n} (u^2 - 1)^n$$  \hspace{1cm} (7.7)

This equation, although not shown, is also used in the derivation of equation (7.4). Substituting equation (7.4) into (7.2) and accounting for the point mass gives a perturbation acceleration of
This equation can be used to determine the size of perturbations due to irregularities in the gravity field.

**Quantitative Analysis**

Because the mission has a polar orbit around the Sun as its operating orbit it is worthwhile to investigate the Sun’s gravity field. The Sun has an equatorial bulge (additional mass around the Sun’s equator and a shortage of mass at the Polar Regions), which manifests itself in a zonal $J_2$ coefficient of $-0.226 \cdot 10^{-6}$ [Armstrong, et al., 1999]. Graphing the maximum (resulting in the worst case) magnitude of the perturbation (cf. Figure 7.24) shows us that it is certainly valid to neglect this influence for a first-order investigation.

![Figure 7.24](image)

Figure 7.24 The acceleration due to the Sun's zonal coefficient $J_2$ as a function of distance from the solar surface.

The other solar gravity potential coefficients are even smaller, so these may be safely neglected. The importance of irregularities in the Earth’s gravity potential are determined by investigating the most significant zonal coefficient, $J_3$, and the most significant sectorial coefficient, $J_{2,2}$. If these terms are already irrelevant, then the other coefficients are most certainly irrelevant. Plotting the maximum acceleration due to $J_2$ as a function of distance (cf. Figure 7.25) shows us that this acceleration rapidly diminishes.
Doing the same for $J_{2,2}$ (cf. Figure 7.26) shows an initially small perturbation which quickly decays.

Because of the parameters of the orbit insertion at the Earth (and the fact that the spacecraft will most likely not linger for long before beginning its interplanetary trajectory), and due to the small magnitudes of the Earth’s gravity field perturbations it is safe to assume that the irregularities in the Earth gravity field will be of negligible influence on the overall mission design (and $\Delta V$ budget). Near the Sun, even once the spacecraft is inserted into an operational polar orbit of 0.1 AU around the Sun the gravity field perturbation is insignificant, and thus neglected.

### 7.5.2 Third Body

An object’s orbital path is not only influenced by the primary mass (the mass that asserts the greatest gravitational pull), but also by secondary masses. These secondary masses may exert a smaller gravitational
pull. Nevertheless, an investigation is useful as these accelerations may perturb the movement of the spacecraft.

**Qualitative Analysis**

The following equation gives the acceleration of \( m_2 \) (e.g. a near-Earth satellite) relative to \( m_1 \) (e.g. the Earth), where the last term in the equation accounts for any perturbing bodies (e.g. the Moon and the Sun), shown again here as

\[
\ddot{r}_{12} = -\frac{G(m_1 + m_2)}{r_{12}^3} r_{12} - G \sum_{j=3}^{n} m_j \left( \frac{r_{j2}}{r_{j2}^3} - \frac{r_{j1}}{r_{j1}^3} \right) \tag{7.9}
\]

This equation gives the acceleration of \( m_2 \) (e.g. a near Earth satellite) relative to \( m_1 \) (e.g. the Earth), where the last term in the equation accounts for any perturbing bodies (e.g. the Moon and the Sun)

\[
\ddot{r}_{j2} = -G \sum_{j=3}^{n} m_j \left( \frac{r_{j2}}{r_{j2}^3} - \frac{r_{j1}}{r_{j1}^3} \right) \tag{7.10}
\]

Using this equation, accelerations can be calculated to determine the influence of secondary bodies upon the spacecraft.

**Quantitative Analysis**

Equation (7.10) can now be used to create worst case magnitudes of acceleration for a number of third bodies. Perturbations near Earth are investigated by calculating the size of the perturbations from the Moon, the Sun and Jupiter. The analysis is kept simple; use is made of mean distances in a straight line configuration of Sun, Earth, Moon, Jupiter and spacecraft.

![Figure 7.27 Third body perturbing accelerations as function of distance from the surface of the Earth.](image)

The figure shows us that the perturbations near Earth are small and for our purposes insignificant. Once inserted into the final operating orbit around the Sun the perturbations will be of even less importance as the Sun is far more massive than the Earth, so any perturbations from other planets will relatively be even smaller. Of course, for a more accurate model for an interplanetary mission from the Earth to the Sun...
certain periods of the trajectory will require a model that takes more than 2 bodies into account, such as when the gravity fields of the Sun and the Earth are similar in magnitude, and when swing-bys are performed. For our first order purposes however, it suffices to neglect these perturbations.

7.5.3 RADIATION

A spacecraft exposed to radiation will experience a pressure (and thus a force) due to the absorption and/or reflection of photons. In the solar system the main force of radiation is (of course) the Sun. The magnitude of the acceleration is not only dependent on the mass (such as the previous two perturbations), but also dependent on the surface area of the spacecraft.

**Qualitative Analysis**

The solar radiation pressure is determined by the solar flux [Montenbruck, et al., 2005]

\[
\Phi = \frac{\Delta \mathcal{E}}{A \Delta t}
\]  

(7.11)

where

- \(\Phi\) is the solar flux.
- \(\Delta \mathcal{E}\) is the energy that passes through the area \(A\) in the time interval \(\Delta t\).

A single photon has an impulse of

\[
p_{\gamma} = \frac{\mathcal{E}_{\gamma}}{c}
\]  

(7.12)

So the total impulse of an absorbing body is

\[
\Delta p = \frac{\Delta \mathcal{E}}{c} = \frac{\Phi}{c} A \Delta t
\]  

(7.13)

Thus, the body experiences a force of

\[
F = \frac{\Delta p}{\Delta t} = \frac{\Phi}{c} A
\]  

(7.14)

The pressure is the force per area

\[
\mathcal{P} = \frac{\Phi}{c}
\]  

(7.15)

A body will reflect a portion of the incoming radiation, and absorb the rest. It is important to note that the force vector is different for both cases. When radiation is absorbed the vector will align along the vector of the incoming radiation. When radiation is reflected the vector is perpendicular to the surface area. See Figure 7.28.
Figure 7.28: The force vector due to the solar radiation pressure for absorbing and reflecting surfaces.

The force vector for an absorbing surface can be deduced from Figure 7.28.

\[
F_{\text{abs}} = -P \cos(\alpha) A e
\]  

(7.16)

Likewise, using Figure 7.28, the force vector for a perfectly reflecting surface is

\[
F_{\text{refl}} = -2P \cos(\alpha) A \cos(\alpha) n
\]  

(7.17)

These equations may be combined to describe a body that reflects a fraction \(\varepsilon\) of the incoming radiation and absorbs the remaining energy \((1 - \varepsilon)\) [Montenbruck, et al., 2005].

\[
F = -P \cos(\alpha) A [(1 - \varepsilon) e + 2\varepsilon \cos(\alpha) n]
\]  

(7.18)

The acceleration experienced by the body is then

\[
\ddot{r} = -\frac{P}{m} \cos(\alpha) \left( (1 - \varepsilon) e + 2\varepsilon \cos(\alpha) n \right)
\]  

(7.19)

At 1 AU from the Sun the solar flux is on average (due to small variations in the solar cycle) [Montenbruck, et al., 2005]

\[
\Phi_\odot \approx 1.367 \text{ Wm}^{-2}
\]  

(7.20)

This yields a solar radiation pressure of

\[
P_\odot \approx 4.56 \cdot 10^{-6} \text{ Nm}^{-2}
\]  

(7.21)

Equation (7.19) can then be rewritten as a function of this constant pressure, by using the fact that this constant pressure is inversely proportional to the squared distance [Montenbruck, et al., 2005]

\[
\ddot{r} = -P_\odot \frac{r^2}{r^2} \frac{A}{m} \cos(\alpha) \left( (1 - \varepsilon) e + 2\varepsilon \cos(\alpha) n \right)
\]  

(7.22)

where

- \(P_\odot\) is the solar radiation pressure at 1 AU.
- \(r_\odot\) is the distance of 1 AU.
- \(A\) is the cross-section or area exposed to the solar radiation.
- \(m\) is the mass of the body.
- \(\varepsilon\) is the reflectivity coefficient (and \(1 - \varepsilon\) the absorption coefficient).
- \(e\) is the vector parallel to the solar radiation direction.
- \(n\) is the vector normal to the area \(A\).
\( \cos(\alpha) \) is the angle between vectors \( \mathbf{e} \) and \( \mathbf{n} \) \( (\cos(\alpha) = \mathbf{n}^T \mathbf{e}) \).

Further simplification is possible, for example, by assuming that the surface is normal to the Sun.

\[
\mathbf{r} = -P_\odot \frac{r_\odot^2 A}{r^5} (1 + \varepsilon)\mathbf{r} \quad (7.23)
\]

This simple model should suffice in order to acquire a first order estimation on the size of solar radiation perturbation.

Typical values of the reflectivity coefficient \( \varepsilon \) are shown in Table 7.8.

<table>
<thead>
<tr>
<th>Material</th>
<th>( \varepsilon )</th>
<th>( 1 - \varepsilon )</th>
<th>( C_R = 1 + \varepsilon )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solar Panel</td>
<td>0.21</td>
<td>0.79</td>
<td>1.21</td>
</tr>
<tr>
<td>High gain Antenna</td>
<td>0.30</td>
<td>0.70</td>
<td>1.30</td>
</tr>
<tr>
<td>Aluminum coated mylar solar sail</td>
<td>0.88</td>
<td>0.12</td>
<td>1.88</td>
</tr>
</tbody>
</table>

Table 7.8 Reflectivity, absorption and radiation pressure coefficient of satellite components [Montenbruck, et al., 2005].

In addition to direct solar radiation, the radiation from planets can also lead to a small pressure on spacecraft. Even for LEO satellites the Earth emitted radiation and optical albedo radiation (reflected solar radiation from the surface and atmosphere of the Earth) typically only account for 10% to 35% of the total amplitude of the radiation pressure acceleration [Montenbruck, et al., 2005]. Equations (7.22) and (7.23) are both easily adapted, by replacing the \( P_\odot \) and \( r_\odot \) parameters, to compute an approximation of their influence. At an altitude of 200 km the Earth albedo radiation flux is [Wakker, 2002]

\[
\Phi_{Albedo} \approx 400 \text{ Wm}^{-2} \quad (7.24)
\]

This leads to an albedo radiation pressure of

\[
P_{\odot Albedo} \approx 1.33 \cdot 10^{-6} \text{ Nm}^{-2} \quad (7.25)
\]

The infrared radiation flux, also at 200 km altitude, is [Wakker, 2002]

\[
\Phi_{IR} \approx 300 \text{ Wm}^{-2} \quad (7.26)
\]

This leads to an infrared radiation pressure of

\[
P_{\odot IR} \approx 1.00 \cdot 10^{-6} \text{ Nm}^{-2} \quad (7.27)
\]

The parameter \( r_\odot \) is simply changed to the reference altitude of 200 km. It is worth mentioning that although the acceleration decreases due to the inverse squared distance, this is partially compensated by the fact that (at these small altitudes) the illuminating surface section of the Earth increases with altitude [Montenbruck, et al., 2005].

**Quantitative Analysis**

The acceleration due to the solar radiation pressure is calculated for a spacecraft near Earth (cf. Figure 7.29), as well as for the Earth albedo radiation pressure (cf. Figure 7.30). Please note that the results are presented in acceleration mass per unit exposed area per reflectivity coefficient, meaning the results must be manipulated using the relevant spacecraft parameters to obtain the actual acceleration.
Figure 7.29 The solar radiation pressure acceleration near Earth as a function of distance from the Sun.

As can be seen, the resulting acceleration is very small, and for our purposes a spacecraft with typical mass, area (facing the Sun) and material may be safely neglected.

Figure 7.30 Earth albedo and infrared radiation acceleration as a function of distance from the surface of the Earth.

The sum of the Earth (emitted infrared and reflected from the Sun albedo) radiation results in even smaller values for the acceleration. This perturbation can safely be neglected for typical spacecraft parameters.
7.5.4 ATMOSPHERIC

When a spacecraft is moving through an atmosphere of a planet (e.g. a LEO satellite orbiting the Earth) it will encounter atmospheric forces. The dominant of these is drag, which results in an acceleration opposite to the path (relative to the atmosphere) of the spacecraft. Other contributions from the lift force, and other binormal forces, are so small in comparison to the drag force that these may be safely neglected [Montenbruck, et al., 2005]. Atmospheric drag can be an asset (e.g. re-entry) or a detriment (e.g. a LEO satellite) to a spacecraft.

Qualitative Analysis

The atmospheric drag may be modeled using the following relation

\[ \ddot{r} = -\frac{1}{2} C_D \frac{A}{m} \rho v_{atm}^2 \hat{v}_{atm} \]  

(7.28)

where

- \( C_D \) is the drag coefficient (dimensionless).
- \( v_{atm} \) is the velocity of the object relative to the atmosphere.
- \( A \) is the cross-section or area normal to the velocity \( v_{atm} \).
- \( m \) is the mass of the body.
- \( \rho \) is the atmospheric density around the object.
- \( \hat{v}_{atm} \) is the norm of the velocity \( v_{atm} \) (\( \hat{v}_{atm} = v_{atm} / |v_{atm}| \)).
The drag coefficient $C_D$ describes the interaction between atmosphere and spacecraft surface (material and shape). Obtained empirically, typical values for the drag coefficient $C_D$ lie between 1.5 and 3. The value is usually smaller at low altitudes than at higher altitudes.

**Quantitative Analysis**

Since the spacecraft will be inserted into GTO and will probably leave Earth as soon as possible, it is unlikely that the atmospheric drag will be of great importance. Atmospheric drag will only be significant when the spacecraft is moving through an atmosphere. During a mission the only times this would occur is when the spacecraft uses the atmosphere of a planet such as Venus to perform an aero gravity assist. The option of an aero gravity assist is not considered in this study.

7.5.5 **ELECTROMAGNETIC**

When an object is electrically charged, its orbital path is perturbed when passing through a magnetic field. Planets in possession of a global magnetic field are Mercury, the Earth, Jupiter, Saturn, Uranus and Neptune. These planets all possess a rotating metallic interior, necessary for the generation of such a large magnetic field [De Pater, et al., 2006]. The Sun also possesses a global magnetic field. As the mission’s goal is to reach a close orbit of the Sun, it is worthwhile to investigate the magnitude (and if great, the direction) of the perturbation in this case.

**Qualitative Analysis**

The interaction between magnetic field and charged orbiter results in a Lorentz force acting upon the orbiter [Wakker, 2002]

$$F_L = q \mathbf{V} \times \mathbf{B} \quad (7.29)$$

where

- $q$ is the electric charge of the orbiter.
- $\mathbf{V}$ is the orbiter velocity relative to the source of the magnetic field.
- $\mathbf{B}$ is the magnetic induction of the magnetic field.

The electric charge of the orbiter can be obtained from [Wakker, 2002]

$$q = U C \quad (7.30)$$

where

- $U$ is the voltage of the orbiter (relative to infinity).
- $C$ is the capacitance of the orbiter in Farad.

The capacitance may be approximated by assuming a spherical satellite [Wakker, 2002].

$$C = 1.1 \cdot 10^{-10} R_{Sat} \quad (7.31)$$

Where $R_{Sat}$ is the radius in meters of the spherical satellite.

Under the assumption of a magnetic dipole, the magnitude of the magnetic field strength is dependent on magnetic distance and latitude [Noomen, et al., 2003].
Verification

\[ B = B_0 \left( \frac{R}{r} \right)^3 \sqrt{1 + 3 \sin^2 \delta_m} \]  
\[ (7.32) \]

where

- \( B_0 \) is the magnetic field strength at the magnetic 'equator'.
- \( R \) is the radius of the magnetic field source.
- \( r \) is the distance between the charged orbiter and the magnetic field source.
- \( \delta_m \) is the magnetic latitude.

Combining equations (7.29) and (7.32) yields the magnitude of the acceleration in a magnetic field

\[ \dot{r} = \frac{UCVB_0}{m} \left( \frac{R}{r} \right)^3 \sqrt{1 + 3 \sin^2 \delta_m} \]  
\[ (7.33) \]

This equation can be used to obtain a first order approximation of the Lorentz forces experienced by the satellite as it moves through a magnetic field.

Quantitative Analysis

Equation (7.33) is used to investigate the acceleration per electric charge of a spacecraft orbiting at a circular velocity. The magnetic field strength at the magnetic equator is \( 3.076 \times 10^{-5} \) T [Williams, 2004], which is used in the calculation. Figure 7.32 shows the maximum acceleration (above the polar regions, \( \delta_m = 90 \)) per electric charge as a function of the distance from the surface of the Sun.

A satellite in a 500 km circular orbit with an \( R_{\text{sat}}/m \) value of 0.3 m/kg and a potential of -50 V experiences an acceleration of \( 3 \times 10^{-10} \) m/s² [Wakker, 2002]. Therefore, it is safe to say that this acceleration can be ignored for the purposes of our investigation. Perhaps of more interest is the acceleration the spacecraft experiences when close to the Sun. The magnetic field strength at the polar regions of the Sun is \( 2 \times 10^{-4} \) Tesla [Williams, 2004]. This acceleration per electric charge can then be plotted as a function of distance (cf. Figure 7.33).

\[ R_{\text{Earth}} < r < 5R_{\text{Earth}} \]
Near Earth

\[ \text{Near Earth} \]

Figure 7.32 The electromagnetic induced acceleration per electric charge as a function of the distance from the surface of the Earth.

A satellite in a 500 km circular orbit with an \( R_{\text{sat}}/m \) value of 0.3 m/kg and a potential of -50 V experiences an acceleration of \( 3 \times 10^{-10} \) m/s² [Wakker, 2002]. Therefore, it is safe to say that this acceleration can be ignored for the purposes of our investigation. Perhaps of more interest is the acceleration the spacecraft experiences when close to the Sun. The magnetic field strength at the polar regions of the Sun is \( 2 \times 10^{-4} \) Tesla [Williams, 2004]. This acceleration per electric charge can then be plotted as a function of distance (cf. Figure 7.33).
Using the same parameters as previously (an $R_{\text{Sun}}/m$ value of 0.3 m/kg and a potential of -50 V), a satellite circling the Sun at 0.1 AU distance experiences an acceleration of $1.6 \times 10^{-8}$ m/s$^2$. Although two orders of magnitude greater than for the Earth, this acceleration can still be neglected for a preliminary mission design.

7.5.6 RELATIVISTIC

On the 20th of March in 1916, Albert Einstein submitted his theory of relativity, known as the general theory of relativity, to *Annalen der Physik*. Since then, it has been the current description of gravity in modern physics. Describing gravity as a property of the geometry of space and time, it unifies special relativity (physical theory regarding measurement of movement in frames of reference, i.e. all motion is relative) and Newton’s law of universal gravitation.

**Qualitative Analysis**

Relativistic effects on the orbital path of a body should not be considered as a perturbation per se, but as a correction to the prediction provided by the classical model. The magnitude of the relativistic correction can be determined by adapting equation (C.2), the equation of motion for a 2-body system where one body has a negligible mass compared to the other [Wakker, 2002].

$$\frac{d^2 r}{d \theta^2} + r = \frac{\mu}{\hbar^2}$$  \hspace{1cm} (7.34)

This is a rewritten version of equation (C.2) where $r = r^{-1}$ and $\theta$ is the polar angle. The reader is referred to [Wakker, 2002] for full details. A term is added to approximate the relativistic effect [Wakker, 2002] to obtain

$$\frac{d^2 r}{d \theta^2} + r = \frac{\mu}{\hbar^2} + 3 \frac{\mu}{c^2} r^2$$  \hspace{1cm} (7.35)

A solution is approached by using the Method of Successive Approximations [Wakker, 2002].
Verification

\[ r = \frac{\mu}{h^2} [1 + e \cos(\theta - \omega)] \]

\[ + 3 \frac{\mu^3}{c^2 h^4} \left\{ \frac{1}{1 + \frac{1}{2} e^2 + e \theta \sin(\theta - \omega)} - \frac{1}{6} e^2 \cos 2(\theta - \omega) \right\} \]

The relativistic correction portion is identified above, divided into three terms:

1. A constant increase to \( r \)
2. A fluctuation, of which the amplitude continually increases with increasing polar angle \( \theta \).
3. An oscillation with constant amplitude.

The first term is a constant shift with a very small effect. The third term has a small amplitude. Only the second term is of importance when looking at a shift after a period of time has passed. Thus, the equation is greatly simplified to [Wakker, 2002]

\[ r = \frac{\mu}{h^2} [1 + e \cos(\theta - \omega) + \beta e \theta \sin(\theta - \omega)], \quad \beta = 3 \frac{\mu^2}{c^2 h^2} \]

Typical values of \( \beta \) are for planets: \( \beta < 9.6 \times 10^{-8} \) and for Earth orbiting satellites: \( \beta < 2.1 \times 10^{-9} \) [Wakker, 2002]. It can be stated that \( \beta \theta \ll 1 \), even for large values of \( \theta \). As such, for small periods

\[ \cos \beta \theta \approx 1, \sin \beta \theta \approx \beta \theta \]

Finally, the positional shift due to a relativistic correction imposed upon the equation of motion (formulated by classical mechanics) can be written, by combining the relations (7.37) and (7.38), and using \( r = r^{-1} \)

\[ r = \frac{h^2}{\mu [1 + e \cos(\theta - \omega - \beta \theta)]} \]

This equation provides us with the tool to quantitatively analyze the importance of the relativistic effect upon the orbital path.

Quantitative Analysis

It can be calculated that the orbital perigee shift due to the relativistic correction for an Earth satellite with a perigee of 500 km and apogee of 1000 km is 1280” per century [Wakker, 2002]. Looking at the relativistic corrections for the movement of the planets around the Sun also results in extremely small drifts. Thus, it may be concluded that the relativistic effects are extremely small, and that for the purposes of this study may be neglected.

7.5.7 Effects Analysis

Depending on the available thrust and mass of the spacecraft a certain acceleration can be achieved. This acceleration is of order \( 10^{-2} \) \( - 10^{-3} \) m/s. To ascertain the influence of the perturbations their order sizes for a typical spacecraft are listed in Table 7.9.

† As examples: For Mercury \( \beta \theta = 0.1 \) after 48,000 years. For a LEO satellite \( \beta \theta = 0.1 \) after 1,300 years.
The atmospheric and relativistic perturbations to the trajectory are omitted. The atmospheric perturbation can be neglected because the mission will be launched into GTO, and thus contact with the Earth’s atmosphere will be brief before the spacecraft spirals outwards away from the Earth. Atmospheric drag will only play a role during aero gravity assists, which are left unstudied in this report. Relativistic corrections, as mentioned previously, are small enough to be omitted from any calculations.

As Table 7.9 shows above, the electromagnetic perturbations may be ignored. Third-body perturbations near the Earth and Sun are negligible. When the spacecraft is in interplanetary space, a simple 2-body system of Sun and spacecraft should suffice to describe the motion accurately enough for our purposes. The perturbations due to gravity potential irregularities may also be neglected. The magnitude of these accelerations is simply too small to be of any consequence. Although the $J_2$ effect is of same order of magnitude as the thrust level, this magnitude is calculated for a polar orbit at close proximity to the Earth. In reality, the parameters of the GTO insertion orbit are neither polar nor always near the Earth (highly elliptical with an apogee of 35,700 km), making the zonal effect far smaller than it seems at first glance. The same arguments can be used for the $J_{2,2}$ effect. The perturbations of the Sun’s gravity field are small enough to be ignored. Finally, the acceleration due to radiation pressure can certainly be ignored at the Earth, and during the trajectory towards the Sun. At the final polar science orbit, the radiation pressure may begin to play a small role when it can account for a maximum of 1% of the available acceleration of the engine. This pressure will probably be even lower, as this estimate uses a very large area of the satellite and a relatively low mass. Most likely, using more realistic values of these parameters will result in a 0.1%, perhaps even 0.01%, magnitude of radiation pressure acceleration compared to the satellite’s low thrust propulsion.

From the above we may distill the fact that it is possible to construct an analytical shape-based model that provides a first-order approximation of an interplanetary transfer by only taking the Sun’s gravity and spacecraft thrust into account.
Verification
In this chapter the developed software is used to study a number of transfers, listed in the sections below. All transfers were initially computed under the assumption of an initial mass of 1,500 kg and a specific impulse of 5,000 s. Once a transfer has been obtained, one can play with these values to see the effect on the transfer parameters. Section 8.1 describes the simplest case, a direct transfer from Earth to the Sun with no gravity assists. It also shows why a polar transfer (inclined by 90° with respect to the ecliptic) is an unrealistic option. A gravity assist at Venus is introduced in section 8.2. Section 8.3 further increases the number of swing-bys at Venus to two. Section 8.4 studies the effect the introduction of a swing-by at the Moon has (in combination with a swing-by at Venus). Section 8.5 contains the most complex transfer, an Earth-Moon-Earth-Venus-Sun sequence (3 swing-bys in total). Section 8.6 shows the effect Jupiter can bring upon the inclination by using a swing-by at Jupiter. Finally, two solutions (at much lower initial masses than 1,500 kg) are offered in section 8.7. This section also compares the electric propulsion designs to the solar sailing design.

8.1 Earth-To-Sun Direct Transfer

This section will discuss the option of performing a direct Earth-to-Sun transfer, in both the polar plane (section 8.1.1) and ecliptic plane (section 8.1.2).

8.1.1 Polar Transfer

Perhaps the simplest case to investigate is the case where the spacecraft leaves the Earth and spirals inwards into the Solar System. The plane of motion is already rotated 90° along the axis spanned by the Earth (at date of departure) and Sun with respect to the ecliptic plane (plane of Earth’s motion about the Sun). In this way, no orbit plane changes have to be performed to insert the spacecraft into its final solar science orbit. To inspect the viability of such a transfer, an optimized design is provided in this section. The transfer is shown in Figure 8.1. Additional parameters (regarding search space and results) of the best transfer that was found (after 5 optimization runs) are provided in Table 8.1 and the relevant velocity and position vectors are given in Table 8.2.
Figure 8.1 Transfer Overview in the Sun Centered Ecliptic Reference Frame.

Relevant Transfer Parameters

- **Spacecraft**
  - Specific Impulse: 5000 s
  - Solar Panel Area: 33 m²
  - Panel Efficiency: 25%
  - Thruster Efficiency: 75%

- **Search Space**
  - Starting Date: 01-01-2000 12:00:00
  - Final Date: 09-09-2013 12:00:00
  - Time of Flight: 100 – 3000 Day
  - Winding Parameter: 0.01 – 1
  - # Revolutions: 0 – 5

- **Fitness Parameters**
  - Departure: 0.0263 (10-10-2000 12:37:52) MJD2000
  - Time of Flight: 360.121 Day
  - Arrival: 360.147 (26-12-2000 15:32:07) MJD2000
  - Winding Parameter: 0.01248
  - # Revolutions: 2

- **Mass**
  - Initial Mass: 1500 kg
  - Propellant Consumed: 436.64 kg
  - Final Mass: 1063.36 kg
  - Fitness Value: 602.72 kg
  - Penalty Value: 166.08 kg

*Table 8.1 Overview of transfer spacecraft parameters, search space size, final fitness parameters, and mass parameters.*
Position and Velocity Vectors

<table>
<thead>
<tr>
<th></th>
<th>Departure at Earth Position</th>
<th>Velocity of Earth at Departure</th>
<th>Spacecraft Departure Velocity</th>
<th>Excess Departure Velocity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Departure</td>
<td>$[-2.66 \ 14.47 \ 0] \cdot 10^7$ km</td>
<td>$[-29.79 \ -5.49 \ 0]$ km/s</td>
<td>$[0.68 \ -3.72 \ 29.93]$ km/s</td>
<td>$[30.46 \ 1.78 \ 29.93]$ km/s</td>
</tr>
<tr>
<td>Arrival</td>
<td>Science Orbit Insertion Position</td>
<td>$[1.03 \ -5.60 \ 59.57] \cdot 10^6$ km</td>
<td>$[8.47 \ -46.11 \ -4.47]$ km/s</td>
<td>$[8.51 \ -46.32 \ -4.49]$ km/s</td>
</tr>
<tr>
<td></td>
<td>Spacecraft Arrival Velocity</td>
<td>$[8.51 \ -46.32 \ -4.49]$ km/s</td>
<td>$[0.038 \ -0.208 \ -0.012]$ km/s</td>
<td></td>
</tr>
</tbody>
</table>

Table 8.2 Overview of the positions and velocities at departure and arrival.

The plot clearly shows the spacecraft spiraling inwards towards its 0.4 AU operating orbit and approximately inserting above the Solar North Pole. The transfer itself took just over 360 days to complete. It should be noted that the optimization process found this to be an ideal candidate due to the way penalties were set up. A large penalty was placed on excess arrival velocity in order to ensure that the spacecraft would be able to correctly insert itself into its science orbit while the excess departure velocity was left unpenalized. The total penalty value is therefore completely due to the arrival velocity of the spacecraft mismatching the circular velocity it needs. The spacecraft consumes 436.64 kg of propellant and arrives with a velocity excess (above circular velocity) of 211.8 m/s (see the above tables). The evolution of the spacecraft mass during transfer is shown in Figure 8.2 below.

![Figure 8.2 Mass evolution as a function of transfer duration.](image)

The red circle represents the final mass value, including penalties, found by the optimization while the blue circle represents the actual final mass, without penalties. The difference between the two circles here is the penalty imposed for the excess arrival velocity. We can examine the potential implications of the velocity mismatch at arrival by computing the escape velocity

$$v_{esc} = \sqrt{\frac{2\mu}{r}}$$  \hspace{1cm} (8.1)

and comparing this to the arrival velocity of the spacecraft. At a distance of 0.4 AU the escape velocity is 66.60 km/s. The magnitude of the velocity of the spacecraft itself is 47.31 km/s, which is 211.8 m/s in
excess of the circular velocity it requires. We can thus say with confidence that with an excess of 211.8 m/s (mostly in the $y$ direction) the spacecraft is definitely captured by the Sun in an almost circular orbit (eccentricity $e = 0.091$) and further maneuvers can be performed to fully circularize the orbit. The trajectory shape prescribes a certain thrust profile, shown below in Figure 8.3.

**Figure 8.3 Thrust evolution (in black) and maximum possible thrust (in blue) as a function of time.**

The figure depicts the evolution of the thrust and the (mirrored) thrust boundaries. The figure shows that the thrust is generally negative during the trajectory, this represents the ‘braking’ acceleration as the spacecraft moves inwards into the solar system. It also shows that the thrust exceeds the boundaries during the initial 150 days of the trajectory, but not by a substantial amount. This particular optimization process did not check for and punish designs for using higher than possible thrusts (an optional setting in the program). This is not a particularly pressing problem due to the fact that (1) the literature states that analytical representations can often go over their thrust boundaries but nonetheless serve as a good approximations for further investigation [Petropoulos, et al., 2004]. (2) In this case the overshoot is small and one could argue that simply improving solar panel area or efficiency (leading to a widening of the blue curves above) would be more practical. (3) It will be demonstrated that this particular trajectory design is not really viable anyway. Figure 8.4 below shows the same type of behavior for the acceleration.

**Figure 8.4 Acceleration evolution (in black) and maximum possible acceleration (in blue) as a function of time.**
Although not really seen here, the thrust boundaries become more forgiving as time passes due to the fact that the mass of spacecraft decreases, allowing for the same thrust to achieve a higher acceleration i.e. a higher acceleration can be provided at the same power level. This behavior is more clearly visible for longer transfers.

It will now be demonstrated why this transfer is not a viable option. When glancing at Table 8.2, the astute reader may have already noticed that the spacecraft excess departure velocity is enormous. This is not really surprising, as the spacecraft must cancel all of the forward planetary motion of the Earth and bring itself into an orbit that is 90° inclined with the plane of planetary motion. This leads to a required total velocity change at the Earth of 47.75 km/s! This number is more than 4 times the amount of velocity change required to bring a satellite from Earth into Low Earth Orbit, and therefore this option is not realistic.

8.1.2 ECLIPTIC TRANSFER

We now examine a second Earth to Sun direct transfer, but this time the plane of transfer lies within the ecliptic. This is achieved by telling the optimizer to heavily penalize the excess departure velocity at Earth and by setting the lower and upper boundaries of the third design parameter (i.e. the $z$ component) pertaining to the spherical coordinates around the Sun (cf. section 6.6) to zero. The resulting transfer (best of 5 optimization runs) can be viewed in Figure 8.5 below.

![Figure 8.5 Transfer Overview in the Sun Centered Ecliptic Reference Frame.](image)

The evolution of the vehicle mass as a function of time is shown in Figure 8.6.
Figure 8.6 Mass evolution as a function of transfer duration.

The evolution of the thrust as a function of time is shown in Figure 8.7 below.

Figure 8.7 Thrust evolution (in black) and maximum possible thrust (in blue) as a function of time.

The two tables below provide an overview of the relevant transfer parameters, position vectors, and velocity vectors.

### Relevant Transfer Parameters

<table>
<thead>
<tr>
<th>Spacecraft</th>
<th>Specific Impulse</th>
<th>5000 s</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Solar Panel Area</td>
<td>33 m²</td>
</tr>
<tr>
<td></td>
<td>Panel Efficiency</td>
<td>25 %</td>
</tr>
<tr>
<td></td>
<td>Thruster Efficiency</td>
<td>75 %</td>
</tr>
<tr>
<td>Search Space</td>
<td>Starting Date</td>
<td>4000 MJD2000</td>
</tr>
<tr>
<td></td>
<td>Final Date</td>
<td>8000 MJD2000</td>
</tr>
<tr>
<td></td>
<td>Time of Flight</td>
<td>100 – 3000 Day</td>
</tr>
<tr>
<td></td>
<td>Winding Parameter</td>
<td>0.01 – 1</td>
</tr>
<tr>
<td></td>
<td># Revolutions</td>
<td>0 – 5 -</td>
</tr>
<tr>
<td>Fitness Parameters</td>
<td>Departure</td>
<td>4261.26 (01-09-2011 18:20:35) MJD2000</td>
</tr>
<tr>
<td></td>
<td>Time of Flight</td>
<td>606.67 Day</td>
</tr>
<tr>
<td></td>
<td>Arrival</td>
<td>4867.93 (30-04-2013 10:22:08) MJD2000</td>
</tr>
</tbody>
</table>
Candidate Designs

| Winding Parameter | 0.01518 |
| # Revolutions     | -       |
| **Mass**          |         |
| Initial Mass      | 1500 kg |
| Propellant Consumed| 444.06 kg |
| Final Mass        | 1055.94 kg |
| Fitness Value     | 440.7 kg |
| Penalty Value     | -3.36 kg |

Table 8.3 Overview of transfer spacecraft parameters, search space size, final fitness parameters, and mass parameters.

Position and Velocity Vectors

<table>
<thead>
<tr>
<th><strong>Departure</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Departure at Earth Position</td>
<td>[14.09  -5.42  0] \times 10^7 km</td>
</tr>
<tr>
<td>Velocity of Earth at Departure</td>
<td>[10.20  27.69  0] km/s</td>
</tr>
<tr>
<td>Spacecraft Departure Velocity</td>
<td>[10.08  27.72  0] km/s</td>
</tr>
<tr>
<td>Excess Departure Velocity</td>
<td>[-0.12  0.02  0] km/s</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Arrival</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Science Orbit Insertion Position</td>
<td>[47.12  36.89  0] \times 10^6 km</td>
</tr>
<tr>
<td>Science Orbit Insertion Velocity</td>
<td>[0  0 -47.10] km/s</td>
</tr>
<tr>
<td>Spacecraft Arrival Velocity</td>
<td>[-28.83  37.56  0] km/s</td>
</tr>
<tr>
<td>Excess Arrival Velocity</td>
<td>[-28.83  37.56  47.09] km/s</td>
</tr>
</tbody>
</table>

Table 8.4 Overview of the positions and velocities at departure and arrival.

At the point of insertion the insertion velocity should be only directed in the z direction for a polar orbit (see Science Orbit Insertion Velocity above). As the spacecraft is transferred along the ecliptic its velocity is purely in the ecliptic (see Spacecraft Arrival Velocity above) and so the instantaneous change (the Excess Arrival Velocity listed above) would be enormous. Instead, the orbit plane is changed by way of Edelbaum’s method.

Let us first inspect the arrival velocity of the spacecraft in order to determine whether the spacecraft has been satisfactorily captured in a circular orbit (within the ecliptic plane). The velocity magnitude itself is 47.35 km/s. As mentioned previously the circular orbit velocity at 0.4 AU from the Sun is 47.10 km/s. The angular mismatch between the velocity of the spacecraft and the direction the velocity should be at that moment in the circular orbit is only 0.53°. This angle is obtained by constructing a vector normal to the positional vector of the science orbit insertion by

\[ \mathbf{n}_1 = \begin{bmatrix} c_x \\ -r_{SOI_x} c_y / r_{SOI_y} \\ 0 \end{bmatrix} \]  

(8.2)

where \( r_{SOI_x} \) and \( r_{SOI_y} \) signify the x and y components of the position at Science Orbit Insertion, and \( c_x \) is an arbitrary number. The cross product

\[ \mathbf{n}_2 = \mathbf{n}_1 \times \mathbf{r}_{SOI} \]  

(8.3)

is taken so that the newly created vectors \( \mathbf{n}_1 \) and \( \mathbf{n}_2 \) span a plane in which the velocity of the spacecraft should lie for it to be perfectly directed. The actual velocity of the spacecraft is mapped onto this plane by

\[ \mathbf{v}_{sc,m} = \frac{v_{sc} \cdot \mathbf{n}_1}{\mathbf{n}_1 \cdot \mathbf{n}_1} \mathbf{n}_1 + \frac{v_{sc} \cdot \mathbf{n}_2}{\mathbf{n}_2 \cdot \mathbf{n}_2} \mathbf{n}_2 \]  

(8.4)

The angle between this vector and the actual spacecraft velocity vector represents the mismatch in angle.
Candidate Designs

\[ \cos \alpha = \frac{v_{sc} \cdot v_{s, mapped}}{|v_{sc}| |v_{s, m}|} \]  \hspace{1cm} (8.5)

This angle is computed to be 0.53\(^\circ\). The velocity vector of the spacecraft for perfect circular velocity at insertion would thus be

\[ v_{sc, p} = \frac{v_{s, m}}{|v_{s, m}|} v_c \]  \hspace{1cm} (8.6)

The difference between this velocity and the actual spacecraft velocity at insertion is

\[ v_{sc, p} - v_{sc} = \begin{bmatrix} -0.194 \\ -0.475 \\ 0 \end{bmatrix} \text{ km/s} \]  \hspace{1cm} (8.7)

Looking at these numbers we may safely assume that the spacecraft arrival velocity is satisfactory, and that it is captured into a roughly circular orbit (eccentricity \( e = 0.0108 \)).

After arriving at the Sun the orbit plane must be rotated 90\(^\circ\) such that the orbit is polar. The original Edelbaum’s method (without any stepping) is used because it is the more mass efficient than the distance constrained adaptation (where the total transfer is performed in successive steps of smaller inclination change). An example obtainable thrust profile for this particular maneuver is shown in Figure 3.20 in section 3.7. For the specified spacecraft parameters the available thrust varies from 0.24 N to 2.16 N with a mean of 0.89 N. Edelbaum’s approach operates under the assumption of a constant thrust and this is set as the mean (0.89 N). Propellant use (this is independent of thrust) is a massive 883.6 kg, leaving the spacecraft with a final mass of 172.33 kg. The time it takes to complete this maneuver is 1207.03 days (at a constant thrust of 0.89 N). This means that the total transfer time between departure from Earth and insertion into the science orbit is 1813.7 days, giving an arrival date of 16\(^{th}\) September 2014. This data is summarized below in Table 8.5.

<table>
<thead>
<tr>
<th>Transfer Time</th>
<th>Departure 4261.26 (01-09-2011 18:20:35)</th>
<th>MJD2000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time of Flight</td>
<td>1813.69 Day</td>
<td></td>
</tr>
<tr>
<td>Arrival</td>
<td>5371.73 (16-09-2014 05:28:11) MJD2000</td>
<td></td>
</tr>
<tr>
<td>Mass</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Initial Mass</td>
<td>1500 kg</td>
<td></td>
</tr>
<tr>
<td>Propellant Transfer</td>
<td>444.06 kg</td>
<td></td>
</tr>
<tr>
<td>Mass after Transfer</td>
<td>1055.94 kg</td>
<td></td>
</tr>
<tr>
<td>Propellant Edelbaum</td>
<td>883.61 kg</td>
<td></td>
</tr>
<tr>
<td>Final Mass</td>
<td>172.33 kg</td>
<td></td>
</tr>
</tbody>
</table>

Table 8.5 Overview of total transfer times and masses.
8.2 EARTH-VENUS-SUN

In this section a transfer from Earth to the Solar Science Orbit utilizing a single swing-by at Venus will be studied. An optimization with 10 design parameters (1 departure date; 3 parameters per separate leg of the transfer: time of flight, winding parameter $k_2$, and number of completed revolutions $N$; and 3 parameters for the insertion position: $x$, $y$, and $z$) is performed, the result of which is shown below in Figure 8.8.

As can be seen above, the optimization process has arrived at an inclined final orbit (the obtained inclination change with respect to the ecliptic is $5.78^\circ$) for the spacecraft, this will help lower the propellant cost of the subsequent orbit plane change to a $90^\circ$ inclined orbit. Transfer parameters, positions, and velocities are given in the tables below.

**Relevant Transfer Parameters**

<table>
<thead>
<tr>
<th>Spacecraft</th>
<th>Specific Impulse</th>
<th>Solar Panel Area</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>5000 s</td>
<td>33 m²</td>
</tr>
</tbody>
</table>
Candidate Designs

<table>
<thead>
<tr>
<th>Search Space</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Panel Efficiency</td>
<td>25</td>
<td>%</td>
</tr>
<tr>
<td>Thruster Efficiency</td>
<td>75</td>
<td>%</td>
</tr>
<tr>
<td>Starting Date</td>
<td>4000</td>
<td>MJD2000</td>
</tr>
<tr>
<td>Final Date</td>
<td>8000</td>
<td>MJD2000</td>
</tr>
<tr>
<td>Time of Flight</td>
<td>100 – 3000</td>
<td>Day</td>
</tr>
<tr>
<td>Winding Parameter</td>
<td>0.01 – 1</td>
<td>-</td>
</tr>
<tr>
<td># Revolutions</td>
<td>0 – 2</td>
<td>-</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Fitness Parameters</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Departure</td>
<td>6069.49 (13-08-2016 23:43:00)</td>
<td>MJD2000</td>
</tr>
<tr>
<td>Time of Flight</td>
<td>1st 381.47 2nd 212.12</td>
<td>Day</td>
</tr>
<tr>
<td>Winding Parameter</td>
<td>1st 0.3446 2nd 0.4164</td>
<td>-</td>
</tr>
<tr>
<td># Revolutions</td>
<td>1st 1 2nd 1</td>
<td>-</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Mass</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial Mass</td>
<td>1500</td>
<td>kg</td>
</tr>
<tr>
<td>Propellant Consumed</td>
<td>403.02</td>
<td>kg</td>
</tr>
<tr>
<td>Final Mass</td>
<td>1096.98</td>
<td>kg</td>
</tr>
<tr>
<td>Fitness Value</td>
<td>9553.10</td>
<td>kg</td>
</tr>
<tr>
<td>Penalty Value</td>
<td>9150.08</td>
<td>kg</td>
</tr>
</tbody>
</table>

Table 8.6 Overview of transfer spacecraft parameters, search space size, final fitness parameters, and mass parameters.

Position and Velocity Vectors

<table>
<thead>
<tr>
<th>Departure</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Departure at Earth Position</td>
<td>[11.88  -9.41  0] · 10^7</td>
<td>km</td>
</tr>
<tr>
<td>Velocity of Earth at Departure</td>
<td>[18.00  23.25  0]</td>
<td>km/s</td>
</tr>
<tr>
<td>Spacecraft Departure Velocity</td>
<td>[17.89  23.21  0.001]</td>
<td>km/s</td>
</tr>
<tr>
<td>Excess Departure Velocity</td>
<td>[-0.12  -0.04  0.001]</td>
<td>km/s</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Swing-by</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Position of Venus at Swing-by</td>
<td>[2.45 · 10^7  1.05 · 10^6  3.26 · 10^3]</td>
<td>km</td>
</tr>
<tr>
<td>Velocity before Swing-by</td>
<td>[-34.56  7.93  -0.0005]</td>
<td>km/s</td>
</tr>
<tr>
<td>Velocity after Swing-by</td>
<td>[-32.92  8.73  3.45]</td>
<td>km/s</td>
</tr>
<tr>
<td>Excess Velocity Mismatch</td>
<td>3.61 · 10^-4</td>
<td>km/s</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Arrival</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Science Orbit Insertion Position</td>
<td>[-46.73  -37.18  3.75] · 10^6</td>
<td>km</td>
</tr>
<tr>
<td>Science Orbit Insertion Velocity</td>
<td>[-2.31  -1.84  -47.00]</td>
<td>km/s</td>
</tr>
<tr>
<td>Excess Arrival Velocity</td>
<td>[30.87  -36.99  43.29]</td>
<td>km/s</td>
</tr>
</tbody>
</table>

Table 8.7 Overview of the positions and velocities at departure and arrival.
The evolution of the mass as a function of time is shown in Figure 8.9.

\[ m \ (kg) \]

\[ \begin{array}{cccccc}
0 & 100 & 200 & 300 & 400 & 500 \\
1550 & 1400 & 1300 & 1200 & 1100 & 1000 \\
\end{array} \]

\[ t \ (Day) \]

**Figure 8.9** Mass evolution as a function of transfer duration.

The evolution of the thrust, as well as the maximum possible thrust, is shown as a function of time in Figure 8.10.

\[ T \ (N) \]

\[ \begin{array}{cccccc}
0 & 100 & 200 & 300 & 400 & 500 \\
0 & 1 & 2 & 3 & 4 & 5 \\
\end{array} \]

\[ t \ (Day) \]

**Figure 8.10** Thrust evolution (in black) and maximum possible thrust (in blue) as a function of time.

The velocity at the end of the transfer is inspected and compared to the circular velocity at that point in space. This is done in the same manner as described in section 8.1.2. The velocity magnitude is 48.34 km/s and is 1.89° out of plane. This leads to a vector distance between the perfect spacecraft circular velocity (at that point in space) and the actual spacecraft velocity of

\[
\mathbf{v}_{sc_f} - \mathbf{v}_{sc} = \begin{bmatrix} 29.05 \\ -36.88 \\ -3.71 \end{bmatrix} - \begin{bmatrix} 28.55 \\ -38.83 \\ -3.71 \end{bmatrix} = \begin{bmatrix} 0.493 \\ 1.943 \\ -0.004 \end{bmatrix} \ km/s
\]  

(8.8)

Looking at these numbers we may safely assume that the spacecraft arrival velocity is quite good (leading to an orbit around the Sun with eccentricity \( e = 0.053 \)) and will certainly facilitate capture of the spacecraft by the Sun.
Although the transfer already incorporates some change in inclination, this change must be brought up to 90° inclination. The propellant cost of this maneuver is determined using Edelbaum’s method. Before doing this, we must first know the angle of the plane change by obtaining the angle between the ecliptic and the transfer plane (the angle that has already been changed by the swing-by). The ecliptic plane is spanned by the x and y axis vectors ([1 0 0] and [0 1 0]) and the transfer plane is spanned by the position vectors at Venus \( \mathbf{r}_V \) and at orbit insertion \( \mathbf{r}_{SOI} \). The angle between the two cross products of these planes is the ecliptic angle change that has already been performed.

\[
\cos \alpha = \frac{(\mathbf{r}_V \times \mathbf{r}_{SOI}) \cdot (\mathbf{e}_z)}{|\mathbf{r}_V \times \mathbf{r}_{SOI}| |\mathbf{e}_z|}
\]  

(8.9)

During this computation we make sure that the vector \( \mathbf{r}_V \times \mathbf{r}_{SOI} \) is pointed in the positive z axis by performing a check on its z component and mirroring if it is negative. This angle \( \alpha \) represents the angle that has already been changed and thus the angle that must be changed is \( \beta = \frac{1}{2}\pi - \alpha \).

In this case angle \( \alpha \) is 5.78° (the already achieved inclination change) and so the total inclination change to be performed is 84.22°. The mean thrust of 0.89 N is used again (see section 8.1.2), leading to a transfer time of 1215.19 days. 907.64 kg of propellant is used to bring the spacecraft into its operational orbit. The final mass is 189.34 kg. The date of insertion into science orbit is the 27th of July 2021 (27-07-2021 18:49:00 Gregorian Calendar Date).

<table>
<thead>
<tr>
<th>Transfer Time</th>
<th>Departure</th>
<th>6069.49 (13-08-2016 23:43:00)</th>
<th>MJD2000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date of Swing-by</td>
<td>6450.96 (30-08-2017 11:02:24)</td>
<td>MJD2000</td>
<td></td>
</tr>
<tr>
<td>Arrival</td>
<td>7878.28 (27-07-2021 18:49:00)</td>
<td>MJD2000</td>
<td></td>
</tr>
<tr>
<td>Total Transfer Time</td>
<td>1808.80</td>
<td>Day</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Mass</th>
<th>Initial Mass</th>
<th>1500</th>
<th>kg</th>
</tr>
</thead>
<tbody>
<tr>
<td>Propellant Transfer</td>
<td>403.02</td>
<td>kg</td>
<td></td>
</tr>
<tr>
<td>Mass after Transfer</td>
<td>1096.98</td>
<td>kg</td>
<td></td>
</tr>
<tr>
<td>Propellant Edelbaum</td>
<td>907.64</td>
<td>kg</td>
<td></td>
</tr>
<tr>
<td>Final Mass</td>
<td>189.34</td>
<td>kg</td>
<td></td>
</tr>
</tbody>
</table>

Table 8.8 Overview of total transfer times and masses.

The results show there is some improvement by using a swing-by at Venus over performing none (cf. Table 8.5).
8.3 Earth-Venus-Venus-Sun

The Bepi Colombo mission used two subsequent swing-bys at Venus to slow the spacecraft down on its journey to Mercury. It turns out the model has an incredibly difficult time of finding the advantage in using two swing-bys at Venus. An optimization with 13 design parameters (1 departure date; 3 parameters per separate leg of the transfer: time of flight, winding parameter $k_2$, and number of completed revolutions $N$; and 3 parameters for the insertion position: $x$, $y$, and $z$) is performed, the result (best result of 5 optimization runs) of which is shown in Figure 8.11 below.

Figure 8.11 Plot of the Earth-Venus-Venus-Sun transfer in the x y plane (top) and in the x z plane (bottom).

The plot shows that the two swing-bys change the orbital plane of the transfer, but unfortunately not by much (inclination change with respect to ecliptic is 4.43°). Transfer parameters are given in Table 8.9, and positions and velocities are given in Table 8.10.
Candidate Designs

Relevant Transfer Parameters

<table>
<thead>
<tr>
<th>Spacecraft</th>
<th>Specific Impulse</th>
<th>$s$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solar Panel Area</td>
<td>33</td>
<td>$m^2$</td>
</tr>
<tr>
<td>Panel Efficiency</td>
<td>25</td>
<td>%</td>
</tr>
<tr>
<td>Thruster Efficiency</td>
<td>75</td>
<td>%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Search Space</th>
<th>Starting Date</th>
<th>4000 MJD2000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Final Date</td>
<td>8000 MJD2000</td>
<td></td>
</tr>
<tr>
<td>Time of Flight</td>
<td>100 – 2000</td>
<td>Day</td>
</tr>
<tr>
<td>Winding Parameter</td>
<td>0.01 – 1</td>
<td></td>
</tr>
<tr>
<td># Revolutions</td>
<td>0 – 3</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Time of Flight</td>
<td>1st</td>
<td>486.03 Day</td>
</tr>
<tr>
<td></td>
<td>2nd</td>
<td>959.62 Day</td>
</tr>
<tr>
<td></td>
<td>3rd</td>
<td>310.97 Day</td>
</tr>
<tr>
<td>Arrival</td>
<td>7259.87 (17-11-2019 08:52:22) MJD2000</td>
<td></td>
</tr>
<tr>
<td>Winding Parameter</td>
<td>1st</td>
<td>0.3673</td>
</tr>
<tr>
<td></td>
<td>2nd</td>
<td>0.4832</td>
</tr>
<tr>
<td></td>
<td>3rd</td>
<td>0.2201</td>
</tr>
<tr>
<td># Revolutions</td>
<td>1st</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>2nd</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>3rd</td>
<td>1</td>
</tr>
</tbody>
</table>

| Mass | Initial Mass | 1500 kg |
|      | Propellant Consumed | 792.88 kg |
|      | Final Mass | 707.12 kg |
|      | Fitness Value | 820.65 kg |
|      | Penalty Value | 113.53 kg |

Table 8.9 Overview of transfer spacecraft parameters, search space size, final fitness parameters, and mass parameters.

Position and Velocity Vectors

<table>
<thead>
<tr>
<th>Departure</th>
<th>Departure at Earth Position</th>
<th>$[-8.54 \ 12.00 \ 0] \cdot 10^7$ km</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Velocity of Earth at Departure</td>
<td>$[-24.76 \ -17.39 \ 0]$ km/s</td>
</tr>
<tr>
<td></td>
<td>Spacecraft Departure Velocity</td>
<td>$[-24.42 \ -16.82 \ 0.64]$ km/s</td>
</tr>
<tr>
<td></td>
<td>Excess Departure Velocity</td>
<td>$[0.34 \ 0.56 \ 0.64]$ km/s</td>
</tr>
<tr>
<td>Swing-by</td>
<td>Position of Venus at Swing-by 1</td>
<td>$[5.86 \cdot 10^7 \ 9.07 \cdot 10^8 \ -2.16 \cdot 10^6]$ km</td>
</tr>
<tr>
<td></td>
<td>Velocity before Swing-by 1</td>
<td>$[-28.70 \ 20.66 \ 0.25]$ km/s</td>
</tr>
<tr>
<td></td>
<td>Velocity after Swing-by 1</td>
<td>$[-31.97 \ 1.80 \ 2.09]$ km/s</td>
</tr>
<tr>
<td></td>
<td>Excess Velocity Mismatch 1</td>
<td>$6.23 \cdot 10^{-12}$ km/s</td>
</tr>
<tr>
<td></td>
<td>Position of Venus at Swing-by 2</td>
<td>$[-9.78 \cdot 10^7 \ 4.43 \cdot 10^7 \ 6.25 \cdot 10^6]$ km</td>
</tr>
<tr>
<td></td>
<td>Velocity before Swing-by 2</td>
<td>$[-17.06 \ -32.64 \ 0.54]$ km/s</td>
</tr>
<tr>
<td></td>
<td>Velocity after Swing-by 2</td>
<td>$[-15.56 \ -31.07 \ -1.70]$ km/s</td>
</tr>
<tr>
<td></td>
<td>Excess Velocity Mismatch 2</td>
<td>$4.40 \cdot 10^{-12}$ km/s</td>
</tr>
<tr>
<td>Arrival</td>
<td>Science Orbit Insertion Position</td>
<td>$[36.02 \ 47.73 \ 2.22] \cdot 10^6$ km</td>
</tr>
<tr>
<td></td>
<td>Science Orbit Insertion Velocity</td>
<td>$[1.05 \ 1.40 \ -47.06]$ km/s</td>
</tr>
<tr>
<td></td>
<td>Spacecraft Arrival Velocity</td>
<td>$[-40.08 \ 24.94 \ 3.04]$ km/s</td>
</tr>
<tr>
<td></td>
<td>Excess Arrival Velocity</td>
<td>$[-41.13 \ 23.54 \ 50.10]$ km/s</td>
</tr>
</tbody>
</table>

Table 8.10 Overview of the positions and velocities at departure and arrival.
The evolution of the mass as a function of time is shown in Figure 8.12.

![Figure 8.12 Mass evolution as a function of transfer duration.](image)

The evolution of the thrust, as well as the maximum possible thrust, is shown as a function of time in Figure 8.21.

![Figure 8.13 Thrust evolution (in black) and maximum possible thrust (in blue) as a function of time.](image)

The velocity at the end of the transfer is inspected and compared to the circular velocity at that point in space. This is done in the same manner as described in section 8.1.2. The velocity magnitude is 47.29 km/s and is 5.00° out of plane. This leads to a vector distance between the perfect spacecraft circular velocity (at that point in space) and the actual spacecraft velocity of

$$v_{scp} - v_{sc} = \begin{bmatrix} -37.58 \\ 28.21 \\ 3.19 \end{bmatrix} - \begin{bmatrix} -40.8 \\ 24.94 \\ 3.04 \end{bmatrix} = \begin{bmatrix} 2.50 \\ 3.27 \\ 0.15 \end{bmatrix} \text{ km/s} \quad (8.10)$$

This indicates that the spacecraft is inserted into a relatively circular orbit (with eccentricity $e = 0.009$) around the Sun.

The transfer incorporates a small change in inclination of 4.43°, but this must be increased further up to a 90° inclination. The propellant cost of this maneuver is once again determined using Edelbaum’s method.
Candidate Designs

The method of finding this angle is shown in section 8.2. The total inclination change to be performed is 85.57° and a mean thrust of 0.89 N is used (cf. section 8.1.2). This gives a transfer time of 789.65 days. 586.81 kg of propellant is used to bring the spacecraft into its operational orbit. This leaves the spacecraft with a final mass of 120.34 kg. The date of insertion into science orbit is the 15th of January 2022.

| Transfer Time | Departure | 5503.25 (13-08-2016 23:43:00) | MJD2000 |
| Date of Swing-by 1 | 5989.28 (25-05-2016 18:50:47) | MJD2000 |
| Date of Swing-by 2 | 6948.91 (10-01-2019 09:43:23) | MJD2000 |
| Arrival | 8049.52 (15-01-2022 00:36:39) | MJD2000 |
| Total Transfer Time | 2546.28 | Day |

| Mass | Initial Mass | 1500 | kg |
| Propellant Transfer | 792.88 | kg |
| Mass after Transfer | 707.15 | kg |
| Propellant Edelbaum | 586.81 | kg |
| Final Mass | 120.34 | kg |

Table 8.11 Overview of total transfer times and masses.

This transfer is not a particularly attractive one. The final mass is not very good, and certainly not better than the candidate utilizing only 1 swing-by at Venus. Additionally, the arrival velocity at the Sun is certainly not the best we have encountered. There are some brighter points though - the departure velocity at Earth is quite low (0.92 km/s) and the velocity mismatches at both swing-bys are extremely small (in the order of 1·10^{-12} km/s).
8.4 Earth-Moon-Venus-Sun

The transfer from Earth to the Solar Science Orbit utilizing a single swing-by at Venus in section 8.2 is modified by adding an additional lunar swing-by. This increases the number of design parameters from 10 to 13 (1 departure date; 3 parameters per separate leg of the transfer: time of flight, winding parameter $k_2$, and number of completed revolutions $N$; and 3 parameters for the insertion position: $x$, $y$, and $z$). The best result (of 5 optimization runs) is shown below in Figure 8.14.

![Plot of the Earth Moon Venus Sun transfer in the x y plane (top) and in the x z plane (bottom).](image)

The plot shows that the swing-by at Venus brings almost no change (0.35°) to the orbital plane of the transfer. Transfer parameters, positions, and velocities are given in the tables below.

### Relevant Transfer Parameters

<table>
<thead>
<tr>
<th>Spacecraft</th>
<th>Specific Impulse</th>
<th>$s$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solar Panel Area</td>
<td>33</td>
<td>$m^2$</td>
</tr>
<tr>
<td>Panel Efficiency</td>
<td>25</td>
<td>%</td>
</tr>
<tr>
<td>Thruster Efficiency</td>
<td>75</td>
<td>%</td>
</tr>
</tbody>
</table>
### Candidate Designs

<table>
<thead>
<tr>
<th>Search Space</th>
<th>Starting Date</th>
<th>Final Date</th>
<th>MJD2000</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>4000</td>
<td>8000</td>
<td>MJD2000</td>
</tr>
<tr>
<td>Time of Flight</td>
<td>100 – 1000</td>
<td>Day</td>
<td></td>
</tr>
<tr>
<td>Winding Parameter</td>
<td>0.01 – 1</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td># Revolutions</td>
<td>0 – 2</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>

| Time of Flight        | 1st 84.60 2nd 302.71 3rd 209.62 | Day      |
| Arrival               | 7219.51 (08-10-2019 00:16:08) | MJD2000  |
| Winding Parameter     | 1st 0.6719 2nd 0.3895 3rd 0.4131 | -        |
| # Revolutions         | 1st 0 2nd 1 3rd 1 | -        |

| Time of Flight        | 596.93        | Day        |         |
| Date of Swing-by 1    | 6707.18 (13-05-2018 16:23:05) | MJD2000  |
| Date of Swing-by 2    | 7009.89 (12-03-2019 09:23:09) | MJD2000  |
| Arrival               | 8471.03 (12-03-2023 12:39:30) | MJD2000  |
| Time of Flight        | 596.93        | Day        |         |
| Edelbaum Transfer     | 1251.52       | Day        |         |
| Total Transfer Time   | 1848.45       | Day        |         |

| Mass                  | Initial Mass  | 1500       | kg      |
| Propellant Transfer   | 403.27        | kg         |         |
| Mass after Transfer   | 1096.73       | kg         |         |
| Propellant Edelbaum   | 917.19        | kg         |         |
| Final Mass            | 179.54        | kg         |         |

Table 8.12 Overview of transfer spacecraft parameters, search space size, final fitness parameters, and mass parameters.

### Position and Velocity Vectors

| Departure             | Departure at Earth Position | \([-12.72\ 7.53\ 0]\) \cdot 10^7 | km     |
| Velocity of Earth at Departure | \([-15.67\ -25.75\ 0]\) | km/s          |
| Spacecraft Departure Velocity | \([-15.56\ -25.78\ 0.0085]\) | km/s          |
| Excess Departure Velocity | \([0.104\ -0.030\ 0.0085]\) | km/s          |
| Arrival               | Science Orbit Inertion Position | \([47.62\ 36.24\ 0.024]\) \cdot 10^6 | km     |
| Science Orbit Inertion Velocity | \([0.015\ 0.014\ -47.10]\) | km/s          |
| Spacecraft Arrival Velocity | \([-27.67\ 39.61\ -2.93]\) | km/s          |
| Excess Arrival Velocity | \([-27.68\ 39.60\ 46.80]\) | km/s          |
| Swing-by              | Position of Moon at Swing-by 1 | \([-9.0\ 10^7\ -1.2\ 10^8\ 4.3\ 10^4]\) | km     |
| Velocity before Swing-by 1 | \([23.37\ -18.28\ 0.001]\) | km/s          |
| Velocity after Swing-by 1 | \([23.92\ -16.95\ 0.25]\) | km/s          |
| Excess Velocity Mismatch 1 | \(8.43 \cdot 10^{-10}\) | km/s          |
| Position of Venus at Swing-by 2 | \([-3.1\ 10^7\ -1.0\ 10^8\ 3.6\ 10^5]\) | km     |
| Velocity before Swing-by 2 | \([34.06\ -9.76\ 0.28]\) | km/s          |
| Velocity after Swing-by 2 | \([32.51\ -10.26\ -0.18]\) | km/s          |
| Excess Velocity Mismatch 2 | \(3.48 \cdot 10^{-2}\) | km/s          |

Table 8.13 Overview of the positions and velocities at departure and arrival.
The evolution of the mass as a function of time is shown in Figure 8.15.

![Figure 8.15 Mass evolution as a function of transfer duration.](image)

The evolution of the thrust, as well as the maximum possible thrust, is shown as a function of time in Figure 8.16.

![Figure 8.16 Thrust evolution (in black) and maximum possible thrust (in blue) as a function of time.](image)

The velocity at the end of the transfer is inspected and compared to the circular velocity at that point in space. This is done in the same manner as described in section 8.1.2. The velocity magnitude is 48.31 km/s and is 2.34° out of plane. This leads to a vector distance between the perfect spacecraft circular velocity (at that point in space) and the actual spacecraft velocity of

\[
\mathbf{v}_{scp} - \mathbf{v}_{sc} = \begin{bmatrix} -28.52 \\ 37.48 \\ -0.29 \end{bmatrix} - \begin{bmatrix} -27.67 \\ 39.61 \\ -0.29 \end{bmatrix} = \begin{bmatrix} -0.86 \\ -2.13 \\ 6.37 \cdot 10^{-3} \end{bmatrix} \text{ km/s} \quad (8.11)
\]

This indicates that the spacecraft will be able to be inserted satisfactorily in its orbit, with an eccentricity of 0.057, around the Sun.

The transfer changes the inclination by a miniscule amount of 0.35° (the method of finding this angle is shown in section 8.2), and so this must be further brought up to 90° inclination. The total inclination
change to be performed is 89.65° and a mean thrust of 0.89 N is used (cf. section 8.1.2). This gives a transfer time of 1252 days. A massive 917 kg of propellant is used to bring the spacecraft into its operational orbit. This leaves the spacecraft with a final mass of 179.5 kg. The date of insertion into science orbit is the 12th of March 2023.

**Other Runs**

Table 8.14 provides the 5 previously mentioned runs (the results from the first row, indicated in blue, represent the transfer described above), where all 5 use the same penalty functions. All of them are quite similar except for the first, which has a longer transfer time but an improved excess arrival velocity. The column *Local Improvement* signifies the fitness in percentage the local optimizer was able to improve from the global fitness.

<table>
<thead>
<tr>
<th>Local Improvement</th>
<th>Final Mass</th>
<th>Time of flight</th>
<th>Global time</th>
<th>Local time</th>
<th>Excess Departure Velocity</th>
<th>Excess Arrival Velocity</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.41 %</td>
<td>179.54 kg</td>
<td>1848.45 d</td>
<td>124 s</td>
<td>89 s</td>
<td>0.11 km/s</td>
<td>2.30 km/s</td>
</tr>
<tr>
<td>1.01 %</td>
<td>177.51 kg</td>
<td>2260.74 d</td>
<td>123 s</td>
<td>88 s</td>
<td>2.42·10^{-5} km/s</td>
<td>0.61 km/s</td>
</tr>
<tr>
<td>12.73 %</td>
<td>178.25 kg</td>
<td>1834.03 d</td>
<td>129 s</td>
<td>101 s</td>
<td>0.10 km/s</td>
<td>2.69 km/s</td>
</tr>
<tr>
<td>4.20 %</td>
<td>179.09 kg</td>
<td>1847.83 d</td>
<td>128 s</td>
<td>83 s</td>
<td>0.11 km/s</td>
<td>2.16 km/s</td>
</tr>
<tr>
<td>4.76 %</td>
<td>175.35 kg</td>
<td>1823.63 d</td>
<td>125 s</td>
<td>96 s</td>
<td>0.12 km/s</td>
<td>3.63 km/s</td>
</tr>
</tbody>
</table>

Comparing these results to the results found in section 8.2 (the same transfer without lunar swing-by) we may say that the found solutions are actually worse than those without lunar swing-by, with slightly lower arrival masses (179 kg here compared to 189 kg final mass for the Earth-Venus-Sun transfer).
This transfer tries to partially emulate Bepi Colombo’s transfer, where a lunar swing-by is used to go inwards to 0.85 AU, the spacecraft then travels outwards to 1.15 AU before coming back to Earth for a second swing-by. This is then followed by two swing-bys at Venus before travelling further inwards towards the Earth. By employing swing-bys at the Moon and the Earth the mass that can be inserted at Mercury is 180 kg higher [Campagnola, et al., 2009]. The transfer studied here also performs a lunar swing-by and a swing-by at Earth, but only one swing-by at Venus (because of the poor performance demonstrated in section 8.3) and of course, the target is not Mercury but a circular orbit at 0.4 AU from the Sun. The best obtained result (of 5 runs) is shown in Figure 8.17.

The plot shows that the swing-by at Venus brings almost no change (0.31°) to the orbital plane of the transfer. Transfer parameters, positions, and velocities are given in the tables below.
Candidate Designs

 Relevant Transfer Parameters

<table>
<thead>
<tr>
<th>Spacecraft</th>
<th>Specific Impulse</th>
<th>5000</th>
<th>s</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solar Panel Area</td>
<td>33</td>
<td>m²</td>
<td></td>
</tr>
<tr>
<td>Panel Efficiency</td>
<td>25</td>
<td>%</td>
<td></td>
</tr>
<tr>
<td>Thruster Efficiency</td>
<td>75</td>
<td>%</td>
<td></td>
</tr>
</tbody>
</table>

Search Space

| Starting Date | 4000 | MJD2000 |
| Final Date | 8000 | MJD2000 |
| Time of Flight | 10 – 1000 | Day |
| Winding Parameter | 0.01 – 1 |
| # Revolutions | 0 – 2 |

Fitness Parameters

| Departure | 7820.78 (31-05-2021 06:42:35) | MJD2000 |
| Time of Flight 1st | 423 | Day |
| Arrival | 9058.49 (19-10-2024 23:41:08) | MJD2000 |
| Winding Parameter 1st | 0.86 |
| # Revolutions 1st | 1 |

Transfer Time

| Departure | 7820.78 (31-05-2021 06:42:35) | MJD2000 |
| Date of Swing-by 1 | 8243.80 (28-07-2022 07:16:05) | MJD2000 |
| Date of Swing-by 2 | 8443.39 (12-02-2023 21:16:48) | MJD2000 |
| Date of Swing-by 3 | 8812.44 (16-02-2024 22:43:29) | MJD2000 |
| Arrival | 10271.52 (15-02-2028 00:36:59) | MJD2000 |
| Time of Flight | 1237.71 | Day |
| Edelbaum Transfer | 1213.03 | Day |
| Total Transfer Time | 2450.75 | Day |

Mass

| Initial Mass | 1500 | kg |
| Propellant Transfer | 437.19 | kg |
| Mass after Transfer | 1062.81 | kg |
| Propellant Edelbaum | 888.88 | kg |
| Final Mass | 173.92 | kg |

Table 8.15 Overview of transfer spacecraft parameters, search space size, final fitness parameters, and mass parameters.

Position and Velocity Vectors

| Departure | Departure at Earth Position | [-5.17 -14.26 0] · 10⁷ km |
| Velocity of Earth at Departure | [27.52 -10.28 0] km/s |
| Spacecraft Departure Velocity | [27.59 -10.31 0.04] km/s |
| Excess Departure Velocity | [0.07 -0.03 0.04] km/s |
| Arrival | Science Orbit Insertion Position | [43.36 41.22 0.065] · 10⁶ km |
| Science Orbit Insertion Velocity | [0.37 0.35 -47.09] km/s |
| Spacecraft Arrival Velocity | [-35.40 32.01 -0.33] km/s |
| Excess Arrival Velocity | [-35.77 31.66 46.77] km/s |

| Swing-by | Position of Moon at Swing-by 1 | [8.7 · 10⁷ -1.2 · 10⁸ 1.7 · 10⁵] km |
| Velocity before Swing-by 1 | [1.24 0.17 0.20] km/s |
| Velocity after Swing-by 1 | [0.98 0.79 0.08] km/s |
| Excess Velocity Mismatch 1 | 2.54 · 10⁻¹¹ km/s |
| Position of Earth at Swing-by 2 | [-1.2 · 10⁸ 8.7 · 10⁷ 0] km |
| Velocity before Swing-by 2 | [-17.61 -24.31 0.10] km/s |
| Velocity after Swing-by 2 | [-17.58 -23.69 -0.16] km/s |
| Excess Velocity Mismatch 2 | 0.246 km/s |
| Position of Venus at Swing-by 3 | [-1.6 · 10⁷ -1.1 · 10⁸ -5.3 · 10⁵] km |
Velocity before Swing-by 3 \[34.88 \ -6.42 \ 0.08\] km/s
Velocity after Swing-by 3 \[33.78 \ -6.38 \ 0.39\] km/s
Excess Velocity Mismatch 3 0.284 km/s

Table 8.16 Overview of the positions and velocities at departure and arrival.

The evolution of the mass as a function of time is shown below in Figure 8.18.

\[
m \ (\text{kg})
\]

\[
\begin{array}{cccc}
0 & 1000 & 1400 & \text{t (Day)} \\
1100 & 1200 & 1300 & 1400 \\
1500 & & & \\
\end{array}
\]

Figure 8.18 Mass evolution as a function of transfer duration.

The evolution of the thrust, as well as the maximum possible thrust, is shown as a function of time in Figure 8.19 below.

\[
T \ (\text{N})
\]

\[
\begin{array}{cccc}
0 & -3 & -1 & 1 \ (\text{N}) \\
0 & 1000 & 800 & 600 & 400 & 200 & 0 \ (\text{Day}) \\
\end{array}
\]

Figure 8.19 Thrust evolution (in black) and maximum possible thrust (in blue) as a function of time.

The velocity at the end of the transfer is inspected and compared to the circular velocity at that point in space. This is done in the same manner as described in section 8.1.2. The velocity magnitude is 47.72 km/s and is 4.32° out of plane. This leads to a vector distance between the perfect spacecraft circular velocity (at that point in space) and the actual spacecraft velocity of

\[
\vec{v}_{sc_p} - \vec{v}_{sc} = \begin{bmatrix} -32.45 \\ 34.13 \\ -0.28 \end{bmatrix} - \begin{bmatrix} -35.39 \\ 32.01 \\ -0.32 \end{bmatrix} = \begin{bmatrix} 2.94 \\ 2.12 \\ 0.04 \end{bmatrix} \ km/s
\]

(8.12)
This indicates that the spacecraft will be able to be inserted satisfactorily into an almost circular orbit (with eccentricity $e = 0.027$) around the Sun.

The transfer is only able to change the inclination by $0.31^\circ$ (the method of finding this angle is shown in section 8.2), and so this must be further brought up to $90^\circ$ inclination. The total inclination change to be performed is $89.69^\circ$ and a mean thrust of $0.89$ N is used (cf. section 8.1.2). This gives a transfer time of 1213 days. $889$ kg of propellant is used to bring the spacecraft into its operational orbit. This leaves the spacecraft with a final mass of $173.93$ kg. The date of insertion into science orbit is the 15th of February 2028.

**Other Runs**

Table 8.17 provides the 5 previously mentioned runs (the results from the first row, indicated in blue, represent the transfer described above). All of them have quite low values for excess departure velocity. This design is not able to improve upon the more basic model featuring a single swing-by at Venus, where more mass is brought into final orbit. The column *Local Improvement* signifies the fitness in percentage the local optimizer was able to improve from the global fitness.

<table>
<thead>
<tr>
<th>Local Improvement</th>
<th>Final Mass</th>
<th>Time of Flight</th>
<th>Global Time</th>
<th>Local Time</th>
<th>Excess Departure Velocity</th>
<th>Excess Arrival Velocity</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.40 %</td>
<td>173.92 kg</td>
<td>2450.74 d</td>
<td>342 s</td>
<td>290 s</td>
<td>0.09 km/s</td>
<td>3.63 km/s</td>
</tr>
<tr>
<td>0.00 %</td>
<td>150.49 kg</td>
<td>2697.13 d</td>
<td>399 s</td>
<td>121 s</td>
<td>0.18 km/s</td>
<td>1.60 km/s</td>
</tr>
<tr>
<td>0.00 %</td>
<td>160.87 kg</td>
<td>2711.13 d</td>
<td>375 s</td>
<td>120 s</td>
<td>0.05 km/s</td>
<td>6.38 km/s</td>
</tr>
<tr>
<td>1.94 %</td>
<td>161.49 kg</td>
<td>2710.59 d</td>
<td>299 s</td>
<td>141 s</td>
<td>0.11 km/s</td>
<td>8.89 km/s</td>
</tr>
<tr>
<td>0.03 %</td>
<td>157.77 kg</td>
<td>2687.12 d</td>
<td>211 s</td>
<td>104 s</td>
<td>0.27 km/s</td>
<td>9.04 km/s</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Launch Date</th>
<th>Swing-by Velocity Mismatch 1</th>
<th>Mismatch 2</th>
<th>Mismatch 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>31-05-2021</td>
<td>$2.54 \cdot 10^{-11}$ km/s</td>
<td>0.25 km/s</td>
<td>0.28 km/s</td>
</tr>
<tr>
<td>18-02-2013</td>
<td>$3.22 \cdot 10^{-11}$ km/s</td>
<td>1.52 $\cdot 10^{-9}$ km/s</td>
<td>0.09 km/s</td>
</tr>
<tr>
<td>23-06-2015</td>
<td>$7.09 \cdot 10^{-9}$ km/s</td>
<td>$8.54 \cdot 10^{-11}$ km/s</td>
<td>$4.55 \cdot 10^{-10}$ km/s</td>
</tr>
<tr>
<td>06-10-2018</td>
<td>$4.67 \cdot 10^{-11}$ km/s</td>
<td>0.04 km/s</td>
<td>0.57 km/s</td>
</tr>
<tr>
<td>09-10-2018</td>
<td>$6.74 \cdot 10^{-5}$ km/s</td>
<td>$1.12 \cdot 10^{-3}$ km/s</td>
<td>0.92 km/s</td>
</tr>
</tbody>
</table>

Table 8.17 Optimization results for an Earth-Moon-Earth-Venus-Sun transfer.

Comparing these results to the results found in section 8.2 (the Earth-Venus-Sun transfer) we may say that the obtained solutions here are worse, with slightly lower arrival masses ($174$ compared to $189$ kg for the Earth-Venus-Sun transfer).
8.6 EARTH-JUPITER-SUN

It is possible to use the enormous mass of Jupiter to enact a large orbit plane change. For example, Ulysses was able to gain an 80° inclination change with respect to the ecliptic by performing a gravitational slingshot at Jupiter, this slingshot was entirely ballistic (no rocket motors were used) [ESA, 2010]. The possibility of such a large inclination change is explored in this section. A transfer using a swing-by at Jupiter (best result from 5 runs using identical penalty functions) is shown below in Figure 8.20.

![Figure 8.20 Plot in x y plane (left) and x y z (right) of the transfer from Earth to Solar orbit with a swing-by at Jupiter.](image)

The figure shows that the swing-by at Jupiter is able to bring about quite a large inclined change of 52.37° to the orbit of the spacecraft; this will help in lowering the propellant cost of the subsequent orbit plane change to a 90° inclined orbit. Additional transfer parameters, positions, and velocities are given in the tables below.

### Relevant Transfer Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Spacecraft</strong></td>
<td></td>
</tr>
<tr>
<td>Specific Impulse</td>
<td>5000 s</td>
</tr>
<tr>
<td>Solar Panel Area</td>
<td>33 m²</td>
</tr>
<tr>
<td>Panel Efficiency</td>
<td>25%</td>
</tr>
<tr>
<td>Thruster Efficiency</td>
<td>75%</td>
</tr>
<tr>
<td><strong>Search Space</strong></td>
<td></td>
</tr>
<tr>
<td>Starting Date</td>
<td>4000 MJD2000</td>
</tr>
<tr>
<td>Final Date</td>
<td>12000 MJD2000</td>
</tr>
<tr>
<td>Time of Flight</td>
<td>100 – 4000 Day</td>
</tr>
<tr>
<td>Winding Parameter</td>
<td>0.01 – 1</td>
</tr>
<tr>
<td># Revolutions</td>
<td>0 – 5</td>
</tr>
<tr>
<td><strong>Fitness Parameters</strong></td>
<td></td>
</tr>
<tr>
<td>Departure</td>
<td>10921.3 (25-11-2029 19:04:23) MJD2000</td>
</tr>
<tr>
<td>Time of Flight</td>
<td>1st 514.61 Day 2nd 2548.61 Day</td>
</tr>
<tr>
<td>Arrival</td>
<td>13984.5 (16-04-2038 00:14:54) MJD2000</td>
</tr>
<tr>
<td>Winding Parameter</td>
<td>1st 0.0243 2nd 0.2370</td>
</tr>
<tr>
<td># Revolutions</td>
<td>1st 0 2nd 1</td>
</tr>
<tr>
<td><strong>Transfer Time</strong></td>
<td></td>
</tr>
<tr>
<td>Departure</td>
<td>10921.3 (25-11-2029 19:04:23) MJD2000</td>
</tr>
<tr>
<td>Time of Flight</td>
<td>3063.22 Day</td>
</tr>
<tr>
<td>Edelbaum Transfer</td>
<td>397.21 Day</td>
</tr>
<tr>
<td>Total Transfer Time</td>
<td>3460.42 Day</td>
</tr>
<tr>
<td>Date of Swing-by</td>
<td>11435.9 (24-04-2031 09:41:39) MJD2000</td>
</tr>
<tr>
<td>Arrival</td>
<td>14150.3 (28-09-2038 19:12:57) MJD2000</td>
</tr>
</tbody>
</table>
Candidate Designs

<table>
<thead>
<tr>
<th>Mass</th>
<th>Initial Mass</th>
<th>kg</th>
</tr>
</thead>
<tbody>
<tr>
<td>Propellant Transfer</td>
<td>808.27</td>
<td>kg</td>
</tr>
<tr>
<td>Mass after Transfer</td>
<td>664.84</td>
<td>kg</td>
</tr>
<tr>
<td>Propellant Edelbaum</td>
<td>407.07</td>
<td>kg</td>
</tr>
<tr>
<td>Final Mass</td>
<td>257.77</td>
<td>kg</td>
</tr>
</tbody>
</table>

Table 8.18 Overview of transfer spacecraft parameters, search space size, final fitness parameters, and mass parameters.

Position and Velocity Vectors

**Departure**
- Departure at Earth Position: $[6.52, 13.25, 0] \cdot 10^7$ km
- Velocity of Earth at Departure: $[-27.21, 13.04, 0]$ km/s
- Spacecraft Departure Velocity: $[-32.21, 15.34, -1.04]$ km/s
- Excess Departure Velocity: $[-5.00, 2.30, -1.04]$ km/s

**Arrival**
- Science Orbit Insertion Position: $[-11.57, -58.68, 1.91] \cdot 10^6$ km
- Science Orbit Insertion Velocity: $[-0.29, -1.47, -47.07]$ km/s
- Spacecraft Arrival Velocity: $[29.05, -7.11, -38.70]$ km/s
- Excess Arrival Velocity: $[29.34, -5.64, 8.37]$ km/s

**Swing-by**
- Position of Jupiter at Swing-by: $[-13.98, -78.04, 0.64] \cdot 10^7$ km
- Velocity before Swing-by: $[8.05, -11.47, 0.35]$ km/s
- Velocity after Swing-by: $[7.61, -7.40, -9.98]$ km/s
- Excess Velocity Mismatch: $6.32 \cdot 10^{-2}$ km/s

Table 8.19 Overview of the positions and velocities at departure and arrival.

The evolution of the mass as a function of time is shown below in Figure 8.21.

![Mass evolution as a function of transfer duration.](image-url)

Figure 8.21 Mass evolution as a function of transfer duration.
The evolution of the thrust, as well as the maximum possible thrust, is shown as a function of time in Figure 8.22.

![Figure 8.22 Thrust evolution (in black) and maximum possible thrust (in blue) as a function of time. The left image shows the complete graph while the right image shows an enlarged view of a part of the left image.](image)

The thrust level plots show that the black line is often outside of area marked by the blue lines, this shows us that the power system in its current implementation is having trouble providing the necessary power to the propulsion system. Larger solar panels (with side effect of leading to a greater mass), a more efficient engine, or a different power source (a radioisotope thermoelectric generator for instance) could help to alleviate the situation. Alternatively, a transfer featuring more revolutions would lessen the thrust level but come at the cost of both transfer time and propellant cost. It should be noted that this approximation serves as an initial analysis and that the literature states that an imperfect approximation can form the basis for further analysis [Petropoulos, et al., 2004].

The velocity at the end of the transfer is inspected and compared to the circular velocity at that point in space. This is done in the same manner as described in section 8.1.2. The velocity magnitude is 48.92 km/s and only 0.14° out of plane. This leads to a vector distance between the perfect spacecraft circular velocity (at that point in space) and the actual spacecraft velocity of

$$\mathbf{v}_{\text{scp}} - \mathbf{v}_{\text{sc}} = \begin{bmatrix} 27.99 \\ -6.73 \\ -37.27 \end{bmatrix} - \begin{bmatrix} 29.05 \\ -7.11 \\ -38.71 \end{bmatrix} = \begin{bmatrix} -1.059 \\ 0.382 \\ 1.438 \end{bmatrix} \text{ km/s} \tag{8.13}$$

The velocity difference is almost entirely magitudinal, and the velocity magnitude of 48.92 km/s is well under the escape velocity at this distance from the Sun (66.6 km/s). We may therefore safely conclude that the spacecraft is inserted into an almost circular orbit (eccentricity $e = 0.079$) at 0.4 AU satisfactorily, but will need further maneuvering to circularize the orbit.

This swing-by already performs an inclination change of 52.37° (determined by using the method demonstrated previously in section 8.2) with respect to the ecliptic, this means that a further 37.63° change is necessary to bring the spacecraft into its actual polar orbit. The propellant cost of this maneuver is once again determined by using Edelbaum’s method. A mean thrust of 0.89 N is used (see section 8.1.2), leading to a transfer time of 397.21 days. 407.07 kg of propellant is expended during this maneuver to bring the spacecraft into its operational orbit. The final mass is 257.77 kg.
**Other Runs**

Table 8.21 provides the results of the 5 other runs using identical penalty functions. The solution described above is highlighted in blue.

<table>
<thead>
<tr>
<th>Launch Date (MJD2000)</th>
<th>Time of Flight (day)</th>
<th>Final Mass (kg)</th>
<th>Excess Departure Velocity (km/s)</th>
<th>Mismatch Polar (km/s)</th>
<th>Mismatch Circle (km/s)</th>
<th>Out of plane angle (°)</th>
<th>Swing-by Mismatch (km/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10921.29</td>
<td>3460.42</td>
<td>257.77</td>
<td>5.602</td>
<td>31.030</td>
<td>1.826</td>
<td>0.14</td>
<td>0.063</td>
</tr>
<tr>
<td>6133.88</td>
<td>2827.62</td>
<td>289.13</td>
<td>5.729</td>
<td>29.431</td>
<td>3.69</td>
<td>0.21</td>
<td>1.087</td>
</tr>
<tr>
<td>10524.58</td>
<td>4110.64</td>
<td>242.03</td>
<td>6.421</td>
<td>33.069</td>
<td>1.441</td>
<td>1.01</td>
<td>0.041</td>
</tr>
<tr>
<td>6137.68</td>
<td>2863.86</td>
<td>283.89</td>
<td>7.725</td>
<td>31.283</td>
<td>3.969</td>
<td>2.12</td>
<td>0.067</td>
</tr>
<tr>
<td>6931.24</td>
<td>3168.90</td>
<td>248.00</td>
<td>5.058</td>
<td>32.064</td>
<td>1.721</td>
<td>0.22</td>
<td>0.061</td>
</tr>
</tbody>
</table>

Table 8.20 Optimization results for an Earth-Jupiter-Sun transfer.

By experimenting with the penalty functions further results were found (cf. Table 8.22), all possessing a particular characteristic that makes the unattractive (indicated by the numbers highlighted in red). This illustrates the fact that the right selection of penalty functions is critical for successfully finding a good solution.

<table>
<thead>
<tr>
<th>Launch Date (MJD2000)</th>
<th>Time of Flight (day)</th>
<th>Final Mass (kg)</th>
<th>Excess Departure Velocity (km/s)</th>
<th>Mismatch Polar (km/s)</th>
<th>Mismatch Circle (km/s)</th>
<th>Out of plane angle (°)</th>
<th>Swing-by Mismatch (km/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>6129.56</td>
<td>2677.58</td>
<td>561.52</td>
<td>5.579</td>
<td>7.628</td>
<td>3.535</td>
<td>0.09</td>
<td>5.867</td>
</tr>
<tr>
<td>10128.77</td>
<td>5663.91</td>
<td>279.48</td>
<td>15.093</td>
<td>28.163</td>
<td>2.785</td>
<td>2.76</td>
<td>0.055</td>
</tr>
<tr>
<td>6566.18</td>
<td>3560.66</td>
<td>325.38</td>
<td>12.013</td>
<td>24.695</td>
<td>2.324</td>
<td>0.58</td>
<td>0.072</td>
</tr>
<tr>
<td>5882.84</td>
<td>1924.75</td>
<td>558.06</td>
<td>31.367</td>
<td>15.323</td>
<td>8.785</td>
<td>1.53</td>
<td>0.077</td>
</tr>
</tbody>
</table>

Table 8.21 A selection of characteristics of a number of Earth-Jupiter-Sun Transfers.

It is therefore necessary to do a number of test runs (for each different transfer) to get a right balance of penalty functions that is able to obtain a good solution. After this balance has been achieved, a number of runs with the acquired penalty function set can be run to attempt to find an optimal solution.

Five optimization runs were also performed with a lunar swing-by (Earth-Moon-Jupiter-Sun), but were unable to further improve performance. Table 8.22 shows the best result that was found (the last column shows the value of the swing-by mismatch at the Moon before the slash and the value of the swing-by mismatch at Jupiter after the slash).

<table>
<thead>
<tr>
<th>Launch Date (MJD2000)</th>
<th>Time of Flight (day)</th>
<th>Final Mass (kg)</th>
<th>Excess Departure Velocity (km/s)</th>
<th>Mismatch Polar (km/s)</th>
<th>Mismatch Circle (km/s)</th>
<th>Out of plane angle (°)</th>
<th>Swing-by Mismatch (km/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>7947.13</td>
<td>4567.86</td>
<td>181.46</td>
<td>0.386</td>
<td>45.857</td>
<td>1.548</td>
<td>0.28</td>
<td>1.946/0.052</td>
</tr>
</tbody>
</table>

Table 8.22 Best found Earth-Moon-Jupiter-Sun Transfer.

Although the excess departure velocity is now incredibly small, the final mass has suffered. In addition, the swing-by at the Moon has a velocity magnitude mismatch of 1.95 km/s. An overview of transfer is shown in Figure 8.23.
Figure 8.23 x y z plot of Earth Moon Jupiter Sun Transfer.

8.7 BEST SOLUTIONS

Table 8.23 contains basic information on the best solutions the method was able to identify. More detailed information on these solutions is available in the previous sections of Chapter 8. The letters under the column ‘Transfer’ denote the order of planets the spacecraft passes ($E$ = Earth, $J$ = Jupiter, $M$ = Moon, $S$ = Sun, $V$ = Venus).

<table>
<thead>
<tr>
<th>Transfer</th>
<th>Final Spacecraft Mass [%]</th>
<th>Transfer Time [Days]</th>
<th>Excess Departure Velocity [km/s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>$ES$</td>
<td>11.5</td>
<td>1813</td>
<td>0.12</td>
</tr>
<tr>
<td>$EVS$</td>
<td>12.6</td>
<td>1808</td>
<td>0.13</td>
</tr>
<tr>
<td>$EMVS$</td>
<td>12.0</td>
<td>1848</td>
<td>0.11</td>
</tr>
<tr>
<td>$EVVS$</td>
<td>8.0</td>
<td>2546</td>
<td>0.92</td>
</tr>
<tr>
<td>$EVEMS$</td>
<td>11.6</td>
<td>2451</td>
<td>0.09</td>
</tr>
<tr>
<td>$EJS$</td>
<td>17.2</td>
<td>3460</td>
<td>5.60</td>
</tr>
<tr>
<td>$EMJS$</td>
<td>12.1</td>
<td>4568</td>
<td>0.39</td>
</tr>
</tbody>
</table>

Table 8.23 Overview of investigated transfers.

In the table above, the final spacecraft mass percentage signifies the percentage of the initial mass of the spacecraft that remains after insertion into its science orbit. Thus, for any initial mass a corresponding final mass can be computed using this percentage value. For example, a 1,000 kg spacecraft performing the EJS transfer would be able to insert 172 kg of mass into science orbit. This means that more than $4/5$th of the spacecraft is propellant!

All of the transfers in Chapter 8 were run with a starting vehicle mass of 1,500 kg, but once the transfer has been found any starting mass may be selected to run through it again (the shape of the transfer itself is independent of mass). Of course, some of the mass dependent variables, such as thrust (the acceleration profile is the same but the spacecraft mass has changed, leading to a different thrust profile) and time of flight (a smaller initial mass leads to a quicker inclination change when using Edelbaum’s method), will change (the acceleration profile is the same but the spacecraft mass has changed, leading to a different...
Candidate Designs

thrust profile). By experimentation and varying the starting mass a linear relation between starting mass and final mass is found. Hence, one is safely able to simply take the percentage (from the table above) from any starting mass as the valid final mass. To strengthen this claim, a number of final spacecraft masses are plotted against their corresponding final masses for the Earth-Venus-Sun transfer in Figure 8.24.

![Figure 8.24 The linear relation between spacecraft initial mass and final mass for a given transfer shape.](image_url)

In this figure, the blue circles represent the samples obtained by rerunning a transfer with different initial mass, and the black line is a linear fit between these points.

As shown in section 2.3 the payload mass to match is 5 kg, and this gives a spacecraft dry mass between 16.7 and 33.3 kg. Taking an additional 20% margin (margin for unaccounted aspects) gives us a final spacecraft dry mass of 20 to 40 kg. From the table, the two best strategies are pursued, an outward strategy using a gravity assist at Jupiter (EJS), and an inward strategy with a gravity assist at Venus (EVS).

For the outward strategy the initial spacecraft would have a dry mass of 20 to 40 kg and thus the initial spacecraft mass at departure would be between 116 kg and 233 kg. The time of flight would be 3460 days, or 9.48 years. For the inward strategy the initial spacecraft mass at departure would be between 159 kg and 318 kg, with a total time of flight of 1808 days (4.95 years). It is debatable which strategy would be best, and therefore we will inspect both more closely in the following section. Additionally, a selection of figures showing parts of the search space of the Earth-Jupiter-Sun and Earth-Venus-Sun transfers is provided to the reader in Appendix E.

8.7.1 EARTH-VENUS-SUN

For the inward strategy the worst case of 318 kg is assumed. The transfer occurs as described in section 8.2, but with two changes. First, the starting mass is reduced from 1500 to 318 kg and the solar panel area is reduced to 8 m² (more details on this later).
As can be seen above, the optimization process has arrived at an inclined final orbit (the obtained inclination change with respect to the ecliptic is 5.78°) for the spacecraft, this will help lower the propellant cost of the subsequent orbit plane change to a 90° inclined orbit. Transfer parameters, positions, and velocities are given in the tables below.

Relevant Transfer Parameters

<table>
<thead>
<tr>
<th>Spacecraft</th>
<th>Specific Impulse</th>
<th>Solar Panel Area</th>
<th>Panel Efficiency</th>
<th>Thruster Efficiency</th>
<th>Time</th>
<th>Date of Swing-by</th>
<th>Arrival at 0.4 AU</th>
<th>Inclination Change</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>5000 s</td>
<td>8 m²</td>
<td>25 %</td>
<td>75 %</td>
<td>Departure</td>
<td>6069.49 (13-08-2016 23:43:00)</td>
<td>MJD2000</td>
<td>6663.07 (30-03-2018 13:46:23)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Time of Flight</td>
<td>1st 381.47 2nd 212.12</td>
<td>Day</td>
<td>MJD2000</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6450.96 (30-08-2017 11:02:24)</td>
<td>MJD2000</td>
<td></td>
</tr>
</tbody>
</table>

Figure 8.25 Plot of the transfer in the x y plane (top) and in the x z plane (bottom).
Candidate Designs

Arrival 7435.94 (11-05-2020 10:44:27) MJD2000

<table>
<thead>
<tr>
<th>Total Transfer Time</th>
<th>1366.46 Day</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mass</td>
<td></td>
</tr>
<tr>
<td>Initial Mass</td>
<td>318.0 kg</td>
</tr>
<tr>
<td>Propellant Transfer</td>
<td>85.4 kg</td>
</tr>
<tr>
<td>Mass after Transfer</td>
<td>232.6 kg</td>
</tr>
<tr>
<td>Propellant Edelbaum</td>
<td>192.4 kg</td>
</tr>
<tr>
<td>Final Mass</td>
<td>40.1 kg</td>
</tr>
</tbody>
</table>

| Table 8.24 Overview of transfer spacecraft parameters, search space size, final fitness parameters, and mass parameters. |

Position and Velocity Vectors

<table>
<thead>
<tr>
<th>Departure</th>
<th>Departure at Earth Position [11.88 \ -9.41 \ 0] \times 10^7 km</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Velocity of Earth at Departure [18.00 \ 23.25 \ 0] km/s</td>
</tr>
<tr>
<td></td>
<td>Spacecraft Departure Velocity [17.89 \ 23.21 \ 0.001] km/s</td>
</tr>
<tr>
<td></td>
<td>Excess Departure Velocity [-0.12 \ -0.04 \ 0.001] km/s</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Swing-by</th>
<th>Position of Venus at Swing-by [2.45 \times 10^7 \ 1.05 \times 10^8 \ 3.26 \times 10^3] km</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Velocity before Swing-by [-34.56 \ 7.93 \ -0.0005] km/s</td>
</tr>
<tr>
<td></td>
<td>Velocity after Swing-by [-32.92 \ 8.73 \ 3.45] km/s</td>
</tr>
<tr>
<td></td>
<td>Excess Velocity Mismatch [3.61 \times 10^{-4}] km/s</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Arrival</th>
<th>Science Orbit Insertion Position [-46.73 \ -37.18 \ 3.75] \times 10^6 km</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Science Orbit Insertion Velocity [-2.31 \ -1.84 \ -47.00] km/s</td>
</tr>
<tr>
<td></td>
<td>Spacecraft Arrival Velocity [28.55 \ -38.83 \ -3.71] km/s</td>
</tr>
<tr>
<td></td>
<td>Excess Arrival Velocity [30.87 \ -36.99 \ 43.29] km/s</td>
</tr>
</tbody>
</table>

| Table 8.25 Overview of the positions and velocities at departure and arrival. |

Although at first glance the excess arrival velocity seems to be huge, this is the difference between the spacecraft velocity when being inserted at 5.78° and the velocity it must have in order to be instantaneously inserted into a 90° inclination (with respect to the ecliptic plane).

The evolution of the mass as a function of time is shown below in Figure 8.26.

![Figure 8.26 Mass evolution as a function of transfer duration for the Earth-Venus-Sun transfer.](image)

It should be noted that this graph follows the mass of the spacecraft until the Edelbaum inclination change occurs, at 232.6 kg. The evolution of the thrust, as well as the maximum possible thrust, is shown as a function of time in Figure 8.27.
The solar panels were sized at 8 m² to exactly allow for the power necessary to the propulsion. In reality the solar panels would have to have a larger surface area (or be of higher quality) to account for degradation. Additionally, other systems would also require power to operate simultaneously.

The velocity at the end of the transfer (at 0.4 AU) is inspected and compared to the required circular velocity at that point in space (the method is described in section 8.1.2). The velocity magnitude is 48.34 km/s and is 1.89° out of plane. This leads to a vector distance between the perfect spacecraft circular velocity (at that point in space) and the actual spacecraft velocity of

$$\vec{v}_{e-0} = \vec{v}_e - \vec{v}_0 = \begin{bmatrix} 29.05 \\ -36.88 \\ -3.71 \end{bmatrix} - \begin{bmatrix} 28.55 \\ -38.83 \\ -3.71 \end{bmatrix} = \begin{bmatrix} 0.493 \\ 1.943 \\ -0.004 \end{bmatrix} \text{ km/s} \tag{8.14}$$

Looking at these numbers we may safely assume that the spacecraft arrival velocity is quite good and will certainly facilitate capture of the spacecraft by the Sun. The orbit around the Sun is almost circular and has an eccentricity of 0.053.

It should be noted that the velocity discrepancy in equation (8.14) is not translated into an actual amount of propellant in the mass budget. It is assumed that an actual transfer would have the spacecraft arrive with the correct velocity for a circular orbit (by design and by performing navigational corrections along the before insertion).

The angle of the final orbit plane with respect to the ecliptic is 5.78° and so the total inclination change to be performed is 84.22°. Under the assumption of a constant thrust of 0.3 N (taken as the maximum value from Figure 8.27) the propellant cost is 192.4 kg and will take almost 773 days to complete.

### 8.7.2 Earth-Jupiter-Sun

It is possible to use the enormous mass of Jupiter to enact a large orbit plane change. For example, Ulysses was able to gain an 80° inclination change with respect to the ecliptic by performing a gravitational slingshot at Jupiter, this slingshot was entirely ballistic (no rocket motors were used) [ESA, 2010]. For the outward strategy the worst case of 233 kg initial mass is assumed. The transfer occurs as described in
section 8.6, but with two changes. The starting mass is reduced from 1500 to 233 kg and the solar panel area is reduced slightly from 33 to 30 m².

The figure shows that the swing-by at Jupiter is able to bring about quite a large inclined change (52.37°) to the orbit of the spacecraft, this helps in lowering the propellant cost of the subsequent orbit plane change to a 90° inclined orbit. Additional transfer parameters, positions, and velocities are given in the tables below.

Relevant Transfer Parameters

<table>
<thead>
<tr>
<th>Spacecraft</th>
<th>Specific Impulse</th>
<th>5000</th>
<th>s</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solar Panel Area</td>
<td>30</td>
<td>m²</td>
<td></td>
</tr>
<tr>
<td>Panel Efficiency</td>
<td>25</td>
<td>%</td>
<td></td>
</tr>
<tr>
<td>Thruster Efficiency</td>
<td>75</td>
<td>%</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Time</th>
<th>Departure</th>
<th>10921.3 (25-11-2029 19:04:23)</th>
<th>MJD2000</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Time of Flight</td>
<td>1st 514.61 2nd 2548.61</td>
<td>Day</td>
</tr>
<tr>
<td></td>
<td>Date of Swing-by</td>
<td>11435.9 (24-04-2031 09:41:39)</td>
<td>MJD2000</td>
</tr>
<tr>
<td></td>
<td>Arrival at 0.4 AU</td>
<td>13984.5 (16-04-2038 00:14:54)</td>
<td>MJD2000</td>
</tr>
<tr>
<td></td>
<td>Inclination Change</td>
<td>277.64</td>
<td>Day</td>
</tr>
</tbody>
</table>

| Mass | Initial Mass | 233.0 | kg |
|      | Propellant Transfer | 129.7 | kg |
|      | Mass after Transfer | 103.3 | kg |
|      | Propellant Edelbaum | 63.2 | kg |
|      | Final Mass | 40.0 | kg |

| Total Transfer Time | 3340.86 | Day |

Table 8.26 Overview of transfer spacecraft parameters, search space size, final fitness parameters, and mass parameters.

Position and Velocity Vectors

<table>
<thead>
<tr>
<th>Departure</th>
<th>Departure at Earth Position</th>
<th>[6.52 13.25 0] · 10⁷</th>
<th>km</th>
</tr>
</thead>
<tbody>
<tr>
<td>Velocity of Earth at Departure</td>
<td>[-27.21 13.04 0]</td>
<td>km/s</td>
<td></td>
</tr>
<tr>
<td>Spacecraft Departure Velocity</td>
<td>[-32.21 15.34 -1.04]</td>
<td>km/s</td>
<td></td>
</tr>
<tr>
<td>Excess Departure Velocity</td>
<td>[-5.00 2.30 -1.04]</td>
<td>km/s</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Swing-by</th>
<th>Position of Jupiter at Swing-by</th>
<th>[-13.98 -78.04 0.64] · 10⁷</th>
<th>km</th>
</tr>
</thead>
<tbody>
<tr>
<td>Velocity before Swing-by</td>
<td>[8.05 -11.47 0.35]</td>
<td>km/s</td>
<td></td>
</tr>
</tbody>
</table>

Figure 8.28 Plot in x y plane (left) and x y z (right) of the transfer from Earth to Solar orbit with a swing-by at Jupiter.
### Table 8.27 Overview of the positions and velocities at departure and arrival.

<table>
<thead>
<tr>
<th></th>
<th>km/s</th>
</tr>
</thead>
<tbody>
<tr>
<td>Velocity after Swing-by</td>
<td>[7.61 -7.40 -9.98]</td>
</tr>
<tr>
<td>Excess Velocity Mismatch</td>
<td>$6.32 \cdot 10^{-2}$</td>
</tr>
<tr>
<td>Arrival Science Orbit Insertion Position</td>
<td>$[-11.57 -58.68 1.91] \cdot 10^6$ km</td>
</tr>
<tr>
<td>Arrival Science Orbit Insertion Velocity</td>
<td>$[-0.29 -1.47 -47.07]$ km/s</td>
</tr>
<tr>
<td>Spacecraft Arrival Velocity</td>
<td>$[29.05 -7.11 -38.70]$ km/s</td>
</tr>
<tr>
<td>Excess Arrival Velocity</td>
<td>$[29.34 -5.64 8.37]$ km/s</td>
</tr>
</tbody>
</table>

The evolution of the mass as a function of time is shown below in Figure 8.29.

**Figure 8.29** Mass evolution as a function of transfer duration.

The evolution of the thrust, as well as the maximum possible thrust, is shown as a function of time in Figure 8.30.

**Figure 8.30** Thrust evolution (in black) and maximum possible thrust (in blue) as a function of time.

The large solar panel surface area of 30 m² is most likely too optimistic for a design featuring a spacecraft mass of only 233 kg. Alternatively, a different power source such as a radioisotope thermoelectric generator for instance, may be a more realistic design for a vehicle going out to distances of 5 AU.
Candidate Designs

The velocity at the end of the transfer is inspected and compared to the circular velocity at that point in space. The velocity magnitude is 48.92 km/s and only 0.14° out of plane. This leads to a vector distance between the perfect spacecraft circular velocity (at that point in space) and the actual spacecraft velocity of

\[
\mathbf{v}_{scp} - \mathbf{v}_{sc} = \begin{bmatrix} 27.99 \\ -6.73 \\ -37.27 \end{bmatrix} - \begin{bmatrix} 29.05 \\ -7.11 \\ -38.71 \end{bmatrix} = \begin{bmatrix} -1.059 \\ 0.382 \\ 1.438 \end{bmatrix} \text{ km/s} \quad (8.15)
\]

The velocity difference is almost entirely magnitudinal, and the velocity magnitude of 48.92 km/s is well under the escape velocity at this distance from the Sun (66.6 km/s). We may therefore safely conclude that the spacecraft is inserted into an almost circular orbit (eccentricity \( e = 0.079 \)) at 0.4 AU satisfactorily, but will need further maneuvering to circularize the orbit.

The velocity discrepancy in equation (8.15) is not translated into an actual amount of propellant in the mass budget. It is assumed that an actual transfer would have the spacecraft arrive with the correct velocity for a circular orbit (by design and by performing navigational corrections along the before insertion).

This swing-by already causes an inclination change of 52.37° (determined by using the method demonstrated previously in section 8.2) with respect to the ecliptic, this means that a further 37.63° change is necessary to bring the spacecraft into its actual polar orbit. The propellant cost of this maneuver is determined by using Edelbaum’s method. A constant thrust of 0.2 N (taken from the maximum value in Figure 8.30) is used, leading to a transfer time of 278 days.

8.7.3 COMPARISON

We now possess two analytical solutions to bring a continuous low thrust spacecraft into solar polar orbit at a distance of 0.4 AU, both bringing 40 kg of spacecraft mass (5 kg payload included) into final science orbit. The most important characteristics of both transfers are given in the table below.

<table>
<thead>
<tr>
<th></th>
<th>Spacecraft Mass</th>
<th>Solar panel Area</th>
<th>Transfer Time</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>EVS</strong></td>
<td>318 kg</td>
<td>8 m²</td>
<td>1367 Days</td>
</tr>
<tr>
<td><strong>EJS</strong></td>
<td>233 kg</td>
<td>30 m²</td>
<td>3341 Days</td>
</tr>
</tbody>
</table>

Table 8.28 Critical characteristics of the EVS and EJS transfers that deliver 40 kg of spacecraft mass into science orbit.

It is questionable whether a spacecraft performing a swing-by at Jupiter will be able to sustain and propel itself on solar power alone. If not, it would require a radioisotope thermoelectric generator to power itself. This a technology that has only ever been used on NASA spacecraft, and which use would bring about environmental and political discussion. The design of the EVS transfer may require more propellant, but the power system is of smaller size (and suitable for solar polar) and the transfer time is much reduced by almost 2000 days. Based on this information there is a preference for the \(\text{EVS} \) transfer.

The key question here is whether this transfer can stand up to a design using solar sails to power itself. The answer is not a resounding yes; the solar sailing spacecraft weighs roughly 100 kg less at 204 kg but spends 850 days more in transit (a transfer time of 2219 days) [Garot, 2006]. However, as electric propulsion technology develops, reaching higher levels of thrust and specific impulse, preference will shift towards a spacecraft using that technology for this particular mission.
Fitness functions can be used for comparison between the options. Solely basing preference on a single fitness function for two fundamentally designs (solar sailing and electric propulsion) is perhaps a little simplistic but it is an easy and quick method to provide a first estimate. One such fitness function is the fitness function, from equation (5.5), \( J_1 = (m_{\text{prop}} - m_0) \cdot t^{-1} \). Another fitness function (that takes payload mass instead of propellant mass into account) is \( J_2 = m_0 \cdot t \cdot m_{\text{pay}}^{-1} \) (for both functions it holds that smaller values of \( J \) are more beneficial than larger values). The results from both are shown in Table 8.29.

<table>
<thead>
<tr>
<th></th>
<th>( J_1 )</th>
<th>( J_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Electric Propulsion Venus Swing-by</td>
<td>-2.93 ( \times ) 10^{-2}</td>
<td>86,941</td>
</tr>
<tr>
<td>Electric Propulsion Jupiter Swing-by</td>
<td>-1.20 ( \times ) 10^{-2}</td>
<td>155,690</td>
</tr>
<tr>
<td>Solar Sailing</td>
<td>-9.19 ( \times ) 10^{-2}</td>
<td>90,535</td>
</tr>
</tbody>
</table>

Table 8.29: Comparison of fitness values for both electric propulsion designs and the solar sailing design.

The table shows that for the function \( J_1 \) the solar sailing option scores the best and that for function \( J_2 \) the Earth-Venus-Sun transfer with electric propulsion scores slightly better than the solar sailing option. The Earth-Jupiter-Sun transfer with electric propulsion is crippled by its total transfer time, giving it the worst \( J \) values in both cases.
Candidate Designs
The purpose of this thesis was to explore the possibility of a continuous low thrust transfer to a solar polar orbit (at a distance of 0.4 AU) using analytical methods. A design is sought that is able to bring as much payload in as little time as possible into final science orbit, while fulfilling various constraints (e.g. arrival velocity). This design is then compared to a design using solar sailing that inserts a spacecraft into an identical orbit.

**Design**

Two strategies were identified. One of them is an outward strategy that performs a swing-by at Jupiter, using its mass to achieve a significant inclination change, before proceeding inwards into the solar system again. The second is an inward strategy, where a swing-by at Venus is used to alter velocity and perform a small inclination change in order to help the spacecraft arrive at the final operational orbit. Both designs deliver a 40 kg spacecraft into orbit, carrying 5 kg of payload (the design using solar sailing also brings 5 kg of payload into the operational science orbit). The following table provides the basic parameters of spacecraft mass (propellant included) and transfer time of all three designs (cf. Chapter 8 for more details).

<table>
<thead>
<tr>
<th>Design</th>
<th>Spacecraft Mass</th>
<th>Transfer Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Electric Propulsion Venus Swing-by</td>
<td>318 kg</td>
<td>1367 Days</td>
</tr>
<tr>
<td>Electric Propulsion Jupiter Swing-by</td>
<td>233 kg</td>
<td>3341 Days</td>
</tr>
<tr>
<td>Solar Sailing</td>
<td>204 kg</td>
<td>2219 Days</td>
</tr>
</tbody>
</table>

*Table 9.1 Comparison chart of all the two electric propulsion candidates and solar sailing design.*

The electric propulsion design with a swing-by at Jupiter is least attractive, as it suffers from a significantly longer transfer time than the other two designs, and is also heavier than the solar sailing design. The decision between the other two designs is not so clear-cut. The electric propulsion design is more than 100 kg heavier but manages to reduce transfer time by more than 2 years. Although fitness functions by no means tell the entire story, they can help judge the available options. Fitness functions \( J_1 = (m_{prop} - m_0) \cdot t^{-1} \) (taking propellant mass, initial mass, and transfer time into account) and \( J_2 = m_0 \cdot t \cdot m_{pay}^{-1} \) (taking payload mass, initial mass, and transfer time into account). Both functions \( J \) are minimized i.e. smaller values of \( J \) are superior. The results from both are shown in Table 9.2.
Conclusions & Recommendations

The table shows that for the function $J_1$ the solar sailing option scores the best and that for function $J_2$ the Earth-Venus-Sun transfer with electric propulsion scores slightly better than the solar sailing option. The Earth-Jupiter-Sun transfer with electric propulsion is crippled by its total transfer time, giving it the worst $J$ values in both cases.

It should be noted here that the solar sailing spacecraft spends 424 days escaping the ‘geocentric phase’ of the transfer [Garot, 2006], and that this could be partially or completely avoided by performing a maneuver using a chemical rocket at departure. Removing these 424 days from the total transfer time still makes it slower by more than a year when compared to the electric propulsion option. Both spacecraft mass and operation time influence cost to the design. Although cost is not directly mentioned in this report, the transfer is optimized in such a way to minimize cost indirectly (cost is reduced when the trajectory design allows for more payload in a smaller transfer time, using less propellant). There are a number of issues to be looked at before one can fully decide which is best. For instance, is a 40 kg spacecraft at 0.4 AU, complete with thermal systems, power, and propulsion, a feasible final mass? Fortunately, because this is an analytical approximation, and since this approach has difficulty with multiple gravity assists, it is likely that an improvement with regards to the final mass can be made. The solar sailing design may be lighter, but is a mechanically complex spacecraft with many moving parts. Additionally, it has never been flown in space before. Meanwhile, electric propulsion is being used to great success in a range of missions. This makes a solar sailing design a much riskier proposition, and thus electric propulsion certainly has the edge here regarding development and risk. Because both spacecraft are of different nature, the total cost for both designs could be very different. It is hard to say which would be cheaper, although electric propulsion certainly enjoys a higher degree of completed research and development and an overall less complex design. Furthermore, there is ongoing development in electric propulsion, allowing engines to reach ever higher thrusts and exhaust velocities. The former will lead to shorter transfer times while the latter will allow for more mass-efficient designs.

**Approach**

To tackle the aforementioned task a modular software tool has been written in MATLAB language that is able to seek solutions by connecting subsequent low thrust transfer shapes through an optimization process. The connection points are the locations of objects where a swing-by is performed. In principle, the program allows one to implement as many subsequent shapes as they please, where the dimensionality of the problem increases by three with each newly introduced shape.

The transfer shapes themselves are modeled by solving Lambert’s problem for the exponential sinusoid shape. All further information, such as the acceleration profile, follows from this shape. The advantage of this method is that it is analytical and satisfies some of the boundary constraints (departure and arrival points) by definition. The disadvantage of this approach is that the acceleration profile is continuous and generally nonzero; whereas thrust profiles from actual designs specify the engine turning off and on (this is more optimal than continuous thrust).
Implemented optimizers are a Monte Carlo optimizer, the Genetic Algorithm from the MATLAB Toolbox, and a local Nelder-Mead method (this method was chosen because it does not require derivatives, which would have to be found numerically). The optimization itself is strongly guided by penalty functions, these shape the search space in order to steer the design to attain optimal solutions that possess certain desirable attributes. One must be careful with penalty functions as their incorrect use can lead to convergence problems during optimization. The local optimizer can improve the fitness value by up to 10% in some cases, dependent on the settings of the global optimizer. Using a local optimizer has the potential to greatly improve the design with minimal computational effort.

The goal of this model and optimization implementation is to quickly and easily find a good approximation for an optimal low thrust transfer, find launch windows, and aid in finding the best route.

The current implementation of an analytical model of a low thrust transfer is an improvement over GALOMUSIT in accuracy, user friendliness, and features. It correctly computes propellant use based on a varying mass model instead of a constant mass model. It is also able to perform more than one leg (in theory as many legs as one would like). In addition, it is much more user friendly with a simple user input file and a modular nature. In addition, the reason behind the shapes of the time-of-flight curves has been identified (bathtub etc.)

**Further Conclusions**

1. *The plane change is critical to the design*. The plane change takes up a large portion of the propellant budget and transfer time. Edelbaum’s analytical approach provides a basic overview of transfer time and propellant mass but using a more accurate model that could handle the constraint of distance would definitely be helpful.

2. *Terminal velocities*. Linear penalty functions, if properly balanced, seem to work well at reducing departure and arrival velocities. Especially for single leg transfers between two planets, one is able to reduce the excess arrival velocity to very low values.

3. *Smaller simulations can be more efficient*. Multiple smaller simulations perform (up to a certain degree) really well compared to larger simulations. For example, it can often be much quicker to perform 10 global Genetic Algorithm computations with 100 individuals as a population than 1 computation with 1000 individuals. The 10 runs might have a large diversity in results but together will often achieve the same result as the single larger computation in much less time. As the dimension of the problem increases this technique becomes less and less effective. In any case, it is certainly wise to run many smaller simulations first in order to gain a grasp of the influence of the penalty functions so that one can correctly implement these in a more comprehensive computation in order to find good solutions.

4. *Visualization is extremely helpful*. It is very helpful to visualize - interim or final - results to get a grasp on a particular problem. A good example being the time of flight as a function of initial flight path angle curves, where a collection of various transfer plots allow us to quickly see that monotone descending and bathtub time of flight curves are completely valid. Often enough, it is sufficient to glance at a plot of a multiple gravity assist transfer before having seen any numbers to determine whether a particular transfer is reasonable or not.
6. *Penalty functions are tricky to implement.* It takes some amount of effort to craft a balanced set of penalty functions for a particular problem. This difficulty isn’t helped by the fact that this balance changes whenever changes are made to spacecraft initial mass, the search space, and especially when the basic arrangement of swing-bys is changed (i.e. which planets to use for a swing-by and in what order). Experimentation is the key here, but there are some helpful tips. Infinite penalties should never be used to punish something that is invalid, but should simply be heavily penalized with a constant. If there are valid areas in the search space the optimizer should be able to find these quickly enough. Otherwise (with infinite values) the optimizer can have immense trouble converging. This was extremely noticeable when experimenting with asteroids; because of their low mass a swing-by angle change is impossible and the optimizer will not even be able to find an approximate with an infinite penalty on the angle.

7. *Monte Carlo effectiveness varies on the problems complexity.* A Monte Carlo optimization (in order to obtain an initial population for the Genetic Algorithm) did not show benefit when performing small optimizations using only 4 design parameters. However, for more complex problems the implemented model benefits from an initial Monte Carlo run to form a population. In addition, it can provide a valuable search space reduction when confronted with large search spaces.

8. *Local optimization was very useful.* The local search algorithm can bring very good improvements for very low computational cost. Dependent on the settings of the Genetic Algorithm the gains in fitness can be very significant. Although this report shows almost no results where the local optimizer made significant gains, it has sometimes improved fitness by a factor of 1.5. This especially comes to the fore when the Genetic Algorithm is set more for computational speed than thoroughness.

9. *Lambert’s solution for exponential sinusoids is two-dimensional.* For a transfer between two objects there is no out of plane thrust along the transfer. This means that any planar difference between the two objects must be accounted for by the excess velocities at the start and end of the transfer. This is not too bad when solving for a transfer between planets that are relatively coplanar (e.g. a transfer between Earth and Mars). Also, a transfer with a swing-by planet works relatively well. The mass of the planet usually allows for the swing-by to change the plane of the transfer sufficiently for this to not be a problem. When confronted with lower masses however, such as asteroids in inclined orbits, this becomes a significant problem. Swing-bys are no longer able to perform plane changes and thus a solution becomes difficult, if not impossible, to find. The following figure demonstrates the problem schematically. If the starting point of the transfer lies within plane 1 and the arrival point within plane 2, then the transfer will be departure velocity optimal when the entire transfer lies within plane 1, and be arrival velocity optimal when the entire transfer lies within plane 2.

\[ \text{Plane 1} \]
\[ \text{Plane 2} \]

\[ \varepsilon \]

*Figure 9.1 Two orbital planes illustrating the out of plane component of a transfer.*
10. Simple numerical methods offer satisfactory performance. Both time of flight and mass evolution can be integrated with sufficient accuracy using simple single step methods such as Euler and 4th order Runge-Kutta. During optimization a surprisingly small number of steps already leads to mass values that the optimizer can compare well enough to identify/rank which candidates are best. Afterwards, the solution can still always be processed once more using an integrator with a much finer mesh of points for a higher degree of accuracy.

11. The shape is independent of mass. The shape of the transfer lays down all further parameters of the transfer. Once a transfer shape has been obtained with certain spacecraft specifications one can recalculate the whole transfer in a matter of seconds with a different set of specifications. There are some issues one must be aware of, for example when having performed an optimization where thrust excesses are penalized and then subsequently changing spacecraft propulsion parameters, thereby leading to a transfer with a too demanding thrust profile.

12. A constraint placed on distance to the Sun is usually unnecessary. This constraint can be disregarded for increased computational speed as all encountered transfers tended to avoid getting closer to the Sun in any case. Experimentation showed that transfers passing the Sun very closely and then moving outwards again are seen as sub optimal in the model, and so the optimizer steers away from such solutions.

13. More mutation during the optimization is beneficial. It is suspected that more mutation throughout the optimization (especially at later generations where by default the mutation seems to taper off too quickly) would have helped to find better optimum points more easily. Thankfully, the Genetic Algorithm within MATLAB offers extensive options for the user to play around with (even allowing one to write their own mutation function).

14. Penalties that make an individual invalid (e.g. set to infinity) should be avoided at all costs. When applying such penalties, the search space quickly becomes much less navigable. A search space with large empty holes with no associated fitness has an incredibly detrimental effect on the convergence behavior of a global search algorithm. The system simply ends up with too many individuals that it has no way to effectively rank. It is much more effective to penalize swing-by angle errors and mass overruns (where the propellant exceeds the initial spacecraft mass) by either constants of large magnitude (several times the initial spacecraft mass) or steep linear curves with a large constant.

15. Switching at the right moment from global to local search can heavily reduce computation time. Large gains in computational speed can be gained if one can successfully determine when the best global basin has been found and then switching over to a local search algorithm. Of course, this is very difficult to implement well, and the situation changes with each transfer. Additionally, in most cases one will tend towards a safer strategy that is more comprehensive at the cost of speed. Nevertheless, with sufficient fine-tuning a particular problem can first be inspected several times using quick optimizations to identify global basins and then run more thoroughly.

16. Real low thrust spacecraft transfers tend to have more revolutions than the ones shown in this report. This is hard to model with the current implementation as there are no coasting arcs, meaning the engine will almost always be using some amount of propellant at any point in the transfer. Transfers with many
revolutions therefore often possess a too high propellant usage to be considered a satisfactory solution by the model.

17. Patching exponential sinusoid shapes to each other using gravity assists does not accurately predict the advantages of swing-bys. Adding multiple swing-bys to a problem does not guarantee an improvement at all, in some cases performance actually deteriorates. This is a limitation of the model, a more involving model with separate thrust and coasting arcs would almost certainly have a better performance.

18. Lambert’s solution for exponential sinusoids suffers from the limitation of having two-dimensional transfer planes. This is less of a problem for objects of large mass in relatively coplanar orbits but becomes especially problematic for transfers towards objects in inclined orbits. However, using a swing-by to perform small plane changes can alleviate this problem somewhat, but this only works for heavier objects such as planets. When considering a single leg transfer the shape based approach, especially when using an analytical Lambert’s solution, suffers from the lack of an out of plane component. As a result, this further increases the difficulty of matching velocities at departure and arrival.

Recommendations

1. A more complex analytical model would be desirable. Instead of using only a thrust arc shape (such as an exponential sinusoid) for each leg the total arc could be divided into a succession of thrust and coasting arcs. As a consequence, this model would no longer apply Lambert’s solution to the shape but rather employ the original solution proposed in Petropoulos’ paper for targeting objects (the transfer is adapted based on how close it is to the object when it intersects its orbit). If coded in a modular fashion, this would have the extra benefit of being able to easily replace the shape representing the thrust arcs in order to experiment with shapes such as the inverse polynomial. Gravity assists could be implemented as described in this report. Experimentation and some form of implementation of out of plane thrust to achieve three-dimensional transfers would also be a beneficial to the model (and more easily achieved if Lambert’s solution were to be abandoned), this would also likely partially reduce excess velocities at departure and arrival. Such a model would cost more computation time, due to the higher number of design parameters. However, this should be offset by the extra fidelity this model would provide. For example, the model would be able to emulate a transfer that has the engine thrust set to off and on, this is more optimal than the continuous thrust profile that a single thrusting arc can provide. The optimization would still be guided by penalty functions that steer the design towards particular qualities desired by the designer. An analytical approach to the design of a solar polar mission would likely benefit from this model, hopefully giving improved propellant use values.

2. The possibility to achieve better performance with other shapes using multiple gravity assists. The above model (mentioned under 1.) would have the benefit of easily being able to switch thrust shapes to identify what works well.

3. Solar light pressure. Solar pressure may become an issue for particular configurations of the spacecraft (low mass, high surface area). Modeling this would allow for the investigation of a small spacecraft with small mass and relatively large solar panels, where the spacecraft propels itself by both electric propulsion and solar pressure.
4. A more detailed dynamic model. If the model described in the first recommendation is able to provide good initial approximations a more detailed full dynamic model, taking into account third-body forces and other perturbations, can be made to numerically integrate and optimize a transfer. Especially the solar pressure is an interesting perturbation to study. Perhaps certain configurations of spacecraft, for example electric propulsion spacecraft with small mass and large solar panel area, can use this perturbation to their advantage during transfer. This is certainly worth exploring in more detail. This model would require more computation time but it allows for a much more accurate model of the transfer when compared to analytical methods.

5. A more detailed orbit plane changing model. It is suspected that Edelbaum’s approach is likely too simplistic, and as the orbit plane change maneuver is mission critical (especially for a strategy that does not involve Jupiter) a more detailed numerical method using control theory should be made to study this particular problem. In addition, the way in which the orbital elements such as distance are currently constrained, by performing multiple subsequent maneuvers, is crude and likely far from optimal. There are a number of papers, using control theory and numerical methods, that study this particular problem (unconstrained and constrained) for Earth orbiting satellites. These could, with some adaptation, serve as a basis for an implementation by a future thesis student.

6. The obtained results. It is likely that there is quite some room for improvement left regarding the results contained within this report. If these results can be further improved by a more detailed model, and if the mass budgets listed in Space Mission Analysis & Design are sufficiently accurate for a spacecraft operating at 0.4 AU, then it can probably be safely concluded that a design featuring electric propulsion is superior to one featuring solar sails.

7. Search Space Reductions. There seems to me to be a more efficient way of reducing the search space to find the areas of interest. The shape-based method often yields large portions of the search space that are infeasible (this leads to search space pictures with many ‘islands’ of valid values surrounded by a ‘sea’ of invalid values). It would be worthwhile to inspect the possibility of identifying these valid spaces (e.g. by using some kind of boxing algorithm) and only performing optimization in these spaces. In this way, it could be possible that a local method would already suffice to search through these smaller valid spaces.
Conclusions & Recommendations


Bibliography


Bibliography
A constant thrust can be modeled by rewriting the equation of motion for a two body problem by superposing an acceleration term $a_T$.

$$\ddot{r} + \frac{\mu}{r^3} r = a_T$$  \hspace{1cm} (A.1)

It is useful to rewrite the equation of motion in polar coordinates to obtain a radial and tangential part. We begin by defining some parameters in Figure A.1 shown below.

**Figure A.1 Plane Polar Coordinates.**

From Figure A.1 it can be easily seen that the Cartesian components of the polar vectors $e_r$ and $e_\theta$ are

$$e_r = \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix}, \quad e_\theta = \begin{bmatrix} -\sin \theta \\ + \cos \theta \end{bmatrix}$$  \hspace{1cm} (A.2)

The position vector can be written as

$$r = r \cdot e_r$$  \hspace{1cm} (A.3)

The velocity vector follows from this as

$$\frac{dr}{dt} = \dot{r} e_r + r \dot{e}_r$$  \hspace{1cm} (A.4)

where the time derivative of the radial vector is obtained by differentiating equation (A.2), which yields

$$\dot{e}_r = \begin{bmatrix} -\sin \theta \\ + \cos \theta \end{bmatrix} \dot{\theta} = \dot{\theta} e_\theta$$  \hspace{1cm} (A.5)

By differentiating the position vector once again the acceleration is
\[
\frac{d^2 \mathbf{r}}{dt^2} = \ddot{\mathbf{r}} \mathbf{e}_r + \dot{\mathbf{r}} \dot{\mathbf{e}}_r + (\ddot{r} \dot{\theta} + r \ddot{\theta}) \mathbf{e}_\theta + r \dot{\theta} \dot{\mathbf{e}}_\theta 
\]  
(A.6)

where the time derivative of the tangential vector is obtained by differentiating equation (A.2), which yields

\[
\dot{\mathbf{e}}_\theta = \begin{bmatrix} -\cos \theta \\ -\sin \theta \end{bmatrix} \dot{\theta} = -\dot{\theta} \mathbf{e}_r 
\]  
(A.7)

The relation for then acceleration is then rewritten, using equations (A.5) and (A.7), as

\[
\ddot{\mathbf{r}} = (\ddot{r} - r \dot{\theta}^2) \mathbf{e}_r + (r \dddot{\theta} + 2 \ddot{\theta}) \mathbf{e}_\theta 
\]  
(A.8)

This can be substituted, together with equation (A.3), into the original form of the equation of motion, equation (A.1), giving

\[
(r \dddot{\theta} + 2 \ddot{\theta}) \mathbf{e}_r + (r \dddot{\theta} + 2 \ddot{\theta}) \mathbf{e}_\theta + \frac{\mu}{r^2} \mathbf{e}_r = 0 
\]  
(A.9)

Since \( \mathbf{e}_r \) and \( \mathbf{e}_\theta \) are orthogonal we can separate the equation into the radial component of the equation of motion

\[
\ddot{r} - r \dot{\theta}^2 + \frac{\mu}{r^2} = 0 
\]  
(A.10)

and the tangential component of the equation of motion

\[
r \dddot{\theta} + 2 \ddot{\theta} = 0 
\]  
(A.11)

It is interesting to note that the tangential part can be written as

\[
r \dddot{\theta} + 2 \ddot{\theta} = \frac{1}{r} \frac{d}{dt} (r^2 \dot{\theta}) = \frac{1}{r} \frac{d}{dt} (r \cdot \dot{r}) = \frac{1}{r} \frac{d}{dt} (h) = 0 
\]  
(A.12)

This verifies the derivation because the specific angular moment \( h \) is constant; therefore its derivative is zero.

To model the effect of a constant thrust it is useful to define a thrust vector relative to either the radial or tangential unit vector. The thrust vector can be written in terms of the unit vectors \( \mathbf{e}_r \) and \( \mathbf{e}_\theta \) as

\[
\mathbf{a}_T = a_T \sin \alpha \mathbf{e}_r + a_T \cos \alpha \mathbf{e}_\theta 
\]  
(A.13)

The set of tangential and radial components of the equation of motion can now be modified to include the thrusting term [Wakker, 2002]

\[
\ddot{r} - r \dot{\theta}^2 + \frac{\mu}{r^2} = a_T \sin \alpha 
\]  
(A.14)

\[
r \dddot{\theta} + 2 \ddot{\theta} = a_T \cos \alpha 
\]
APPENDIX B

Derivation of a rotation matrix for a rotation about an arbitrary axis defined by two points in a three dimensional space.

Note: The three dimensional coordinates will be written using four homogeneous coordinates \((x, y, z, t)\) in order to accommodate for translations (omitting translations would allow for regular 3x3 rotation matrices).

Translation Matrix

This transformation moves the point \(P_1(a, b, c)\) to the origin. The product \(T_p \cdot v\) is essentially equivalent to \<-a, -b, -c, 0> + v\). The transformation for \(P_1\) can be written as

\[
T_{P_1} = \begin{bmatrix}
1 & 0 & 0 & -a \\
0 & 1 & 0 & -b \\
0 & 0 & 1 & -c \\
0 & 0 & 0 & 1
\end{bmatrix}
\]  \hspace{1cm} (B.1)

Rotation Matrix

The matrices for rotation by \(\alpha\) about the x-axis, \(\beta\) about the y-axis, and \(\gamma\) about the z-axis are provided here without further derivation (with the additional fourth dimension).

\[
R_x(\alpha) = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & \cos\alpha & -\sin\alpha & 0 \\
0 & \sin\alpha & \cos\alpha & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

\[
R_y(\beta) = \begin{bmatrix}
\cos\beta & 0 & \sin\beta & 0 \\
0 & 1 & 0 & 0 \\
-\sin\beta & 0 & \cos\beta & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]  \hspace{1cm} (B.2)

\[
R_z(\gamma) = \begin{bmatrix}
\cos\gamma & -\sin\gamma & 0 & 0 \\
\sin\gamma & \cos\gamma & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

The general rotation depends on the order of the individual rotations. For example the following matrix describes a rotation about \(x\), then \(y\), and finally \(z\).

\[
R_xR_yR_z = \begin{bmatrix}
\cos\beta \cos\gamma & \cos\beta \sin\gamma & \cos\alpha \cos\gamma \sin\beta - \cos\alpha \sin\gamma & \cos\alpha \cos\gamma \sin\beta + \sin\alpha \sin\gamma & 0 \\
\cos\beta \sin\gamma & \cos\alpha \cos\gamma + \sin\alpha \sin\beta \sin\gamma & -\cos\alpha \gamma + \cos\alpha \sin\beta \sin\gamma & -\cos\alpha \sin\beta + \sin\alpha \sin\gamma & 0 \\
-\sin\beta & \cos\beta \sin\gamma & \cos\alpha \cos\beta & 0 & 0 \\
0 & 0 & \cos\beta \sin\alpha & 0 & 1
\end{bmatrix}
\]

Other rotations are achieved by changing the order of the individual rotation matrices.
Appendix B

Transforming a vector to the z-axis

Here we move a rotation vector \(<u,v,w>\) to the z-axis (this fails if the vector is parallel to the z-axis). The matrix to rotate a vector about the z-axis to the xz plane is

$$R_{xz} = \begin{bmatrix} u/\sqrt{u^2 + v^2} & v/\sqrt{u^2 + v^2} & 0 & 0 \\ -v/\sqrt{u^2 + v^2} & u/\sqrt{u^2 + v^2} & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{B.3}$$

The matrix to rotate a vector in the xz plane to the z-axis is

$$R_{xz2z} = \begin{bmatrix} w/\sqrt{u^2 + v^2 + w^2} & 0 & -\sqrt{u^2 + v^2}/\sqrt{u^2 + v^2 + w^2} & 0 \\ 0 & 1 & 0 & 0 \\ \sqrt{u^2 + v^2}/\sqrt{u^2 + v^2 + w^2} & 0 & w/\sqrt{u^2 + v^2 + w^2} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{B.4}$$

Rotations about the origin

The point \((x,y,z)\) is rotated about the vector \(<u,v,w>\) by the angle \(\theta\). The matrix to rotate about the origin is the product

$$R_{xz}^{-1} R_{xz2x}^{-1} R_x(\theta) R_{xz2x} R_{xz} \tag{B.5}$$

Writing this out leads to

$$\begin{bmatrix} u^2 + (v^2 + w^2)\cos \theta & uv(1 - \cos \theta) - w\sqrt{u^2 + v^2 + w^2}\sin \theta & uv(1 - \cos \theta) + w\sqrt{u^2 + v^2 + w^2}\sin \theta & 0 \\ uv(1 - \cos \theta) + w\sqrt{u^2 + v^2 + w^2}\sin \theta & u^2 + v^2 + w^2 & 0 & 0 \\ uv(1 - \cos \theta) - w\sqrt{u^2 + v^2 + w^2}\sin \theta & 0 & u^2 + v^2 + w^2 & 0 \\ uv(1 - \cos \theta) + w\sqrt{u^2 + v^2 + w^2}\sin \theta & 0 & 0 & u^2 + v^2 + w^2 \end{bmatrix}$$

Multiplying this matrix by the point to be rotated \((x,y,z)\) we get the rotated point

$$\begin{bmatrix} u(ux + vy + wz) + (x(v^2 + w^2) - u(vy + wz))\cos \theta + \sqrt{u^2 + v^2 + w^2}(-wy + vz)\sin \theta \\ v(ux + vy + wz) + (y(u^2 + w^2) - v(ux + wz))\cos \theta + \sqrt{u^2 + v^2 + w^2}(wx - uz)\sin \theta \\ w(ux + vy + wz) + (z(u^2 + v^2) - w(ux + vy))\cos \theta + \sqrt{u^2 + v^2 + w^2}(-vx + uy)\sin \theta \\ 0 \end{bmatrix}$$

Rotation about an arbitrary line

An orientation axis must be provided in order to define positive and negative angles of the rotation. If the axis of rotation is given by two points \(P_1 = (a,b,c)\) and \(P_2 = (d,e,f)\) then we define the oriented vector of
Appendix B

rotation as \( <u,v,w> = <d,a,e-b,f-c> \). The matrix for rotation about an arbitrary line is then given by the product

\[
T_p^{-1} R_z^{-1} R_x^{-1} (\theta) R_x R_z T_p
\]

This is the rotation operator for rotations about the line through points \( P_1 \) parallel to \( <u,v,w> \) by the angle \( \theta \). Writing this product out yields

\[
\begin{bmatrix}
\frac{u^2 + (v^2 + w^2) \cos \theta}{L} \\
\frac{v(1 - \cos \theta) - w \cos \theta}{L} \\
\frac{w(1 - \cos \theta) + u \cos \theta}{L} \\
\frac{(v^2 + w^2) \cos \theta}{L} \\
\frac{w(u^2 + v^2) \cos \theta}{L} \\
\frac{a(v^2 + w^2) - u(bv + cw) + (uv + cw - u^2 + w^2) \cos \theta + (bw - cw) \sin \theta}{L}
\end{bmatrix}
\]

where \( L = \sqrt{u^2 + v^2 + w^2} \). Multiplying this matrix times the point to be rotated \((x,y,z)\) gives us the rotated point

\[
\begin{bmatrix}
a(x^2 + w^2) + u(-bv - cw + ux + vy + wz) + (-a(x^2 + w^2) + u(bv + cw - vy - wz) + (v^2 + w^2)x) \cos \theta + \sqrt{u^2 + v^2 + w^2}(-cv + bw - wy + vz) \sin \theta \\
b(u^2 + w^2) + v(-au - cw + ux + vy + wz) + (-b(u^2 + w^2) + v(au + cw - ux - wz) + (u^2 + w^2)y) \cos \theta + \sqrt{u^2 + v^2 + w^2}(cu - aw + wx - uz) \sin \theta \\
c(u^2 + v^2) + w(-au - bv + ux + vy + wz) + (-c(u^2 + v^2) + w(au + bv - ux - vy) + (u^2 + v^2)z) \cos \theta + \sqrt{u^2 + v^2 + w^2}(-bu + av - vx + uy) \sin \theta \\
1
\end{bmatrix}
\]

Arbitrary Rotation Function

We may write a function dependent on 10 variables for the point \((x,y,z)\) about the line through \((a,b,c)\) parallel to \( <u,v,w> \) by the angle \( \theta \). See Figure B.1 below for a graphical presentation. The function is

\[
f(x, y, z, a, b, c, u, v, w, \theta) = \begin{bmatrix}
a(x^2 + w^2) + u(-bv - cw + ux + vy + wz) + ((x - a)(x^2 + w^2) + u(bv + cw - vy - wz) + (v^2 + w^2)x) \cos \theta + \sqrt{u^2 + v^2 + w^2}(-cv + bw - wy + vz) \sin \theta \\
b(u^2 + w^2) + v(-au - cw + ux + vy + wz) + ((x - b)(u^2 + w^2) + v(au + cw - ux - wz) + (u^2 + w^2)y) \cos \theta + \sqrt{u^2 + v^2 + w^2}(cu - aw + wx - uz) \sin \theta \\
c(u^2 + v^2) + w(-au - bv + ux + vy + wz) + ((x - c)(u^2 + v^2) + w(au + bv - ux - vy) + (u^2 + v^2)z) \cos \theta + \sqrt{u^2 + v^2 + w^2}(-bu + av - vx + uy) \sin \theta \\
1
\end{bmatrix}
\]

This function can be used to rotate Exponential Sinusoid shapes such that they are placed correctly in the three dimensional space ensuring precise three dimensional plots in addition to accurate information regarding the velocity, position, thrust, etc anywhere on the trajectory.

![Figure B.1 Rotation of point \((x,y,z)\) around vector \(<u,v,w>\) with origin \((a,b,c)\) by angle \(\theta\).](image-url)
Assuming that there are two masses that are perfectly spherical (or point masses) and that there are no internal or external forces acting upon the system other than the gravitational forces of the two bodies the expression for $N$ bodies may be reduced to [Bate, et al., 1971]

$$\ddot{r} = -\frac{G(M + m)}{r^3} \cdot r$$  \hspace{1cm} (C.1)

Usually one of the bodies is a spacecraft and as such, has an insignificant mass compared to the other body. In this case the equation of motion can be further simplified to [Bate, et al., 1971]

$$\ddot{r} = -\frac{GM}{r^3} \cdot r \rightarrow \ddot{r} + \frac{\mu}{r^3} \cdot r = 0$$  \hspace{1cm} (C.2)

where $\mu$ is the gravitational parameter.

### C.1 CONSTANTS OF MOTION

#### Conservation of mechanical energy

A gravitational field is ‘conservative’, meaning an object moving under gravity’s influence never loses energy. It merely exchanges one type of energy, kinetic energy, for another, potential energy. The energy constant of motion can be derived by multiplying equation (C.2) by $r$.

$$r \cdot \dot{r} + \dot{r} \cdot \frac{\mu}{r^3} \cdot r = 0$$  \hspace{1cm} (C.3)

Rewriting the left term into scalar velocities

$$v \cdot \dot{v} + \frac{\mu}{r^3} \cdot r \cdot \dot{r} = 0$$  \hspace{1cm} (C.4)

By using the fact that $v \cdot \dot{v} = \frac{d}{dt} \left(\frac{v^2}{2}\right)$ and $\frac{\mu}{r^3} \cdot \dot{r} = \frac{d}{dt} \left(-\frac{\mu}{r}\right)$

$$\frac{d}{dt} \left(\frac{v^2}{2}\right) + \frac{d}{dt} \left(-\frac{\mu}{r}\right) = 0$$  \hspace{1cm} (C.5)

The integration introduces a constant

$$\frac{d}{dt} \left(\frac{v^2}{2} + c - \frac{\mu}{r}\right) = 0$$  \hspace{1cm} (C.6)
Because the time rate of this expression is zero it can be integrated to

$$\mathcal{E} = \frac{v^2}{2} + c - \frac{\mu}{r}$$  \hspace{1cm} (C.7)

The value of the constant $c$ depends on the zero reference of the potential energy. This is arbitrary, so by setting the reference for the potential energy at infinity we set the constant to zero. The only result of this is that the potential energy of the satellite is always negative ($-\frac{\mu}{r}$). The specific mechanical energy of the spacecraft is then [Bate, et al., 1971]

$$\mathcal{E} = \frac{v^2}{2} - \frac{\mu}{r}$$  \hspace{1cm} (C.8)

**Conservation of angular momentum**

It takes a tangential component of force acting upon an object to change its angular momentum when it is in a rotational motion about a center of rotation (in this case, the body with more mass in the center of the reference frame). The derivation is started by cross-multiplying (C.2) by $\mathbf{r}$.

$$\mathbf{r} \times \ddot{\mathbf{r}} + \mathbf{r} \times \frac{\mu}{r^3} \mathbf{r} = 0$$  \hspace{1cm} (C.9)

The cross multiplication of two identical vectors is zero ($\mathbf{a} \times \mathbf{a} = 0$), leaving us with

$$\mathbf{r} \times \ddot{\mathbf{r}} = 0$$  \hspace{1cm} (C.10)

With the knowledge that $\frac{d}{dt}(\mathbf{r} \times \dot{\mathbf{r}}) = \dot{\mathbf{r}} \times \dot{\mathbf{r}} + \mathbf{r} \times \ddot{\mathbf{r}}$

$$\frac{d}{dt}(\mathbf{r} \times \dot{\mathbf{r}}) = 0$$  \hspace{1cm} (C.11)

Integrating leads to a constant [Bate, et al., 1971]

$$\mathbf{h} = \mathbf{r} \times \mathbf{v}$$  \hspace{1cm} (C.12)

Therefore it has been demonstrated that the specific angular moment $\mathbf{h}$ of a satellite remains constant along its orbit. $\mathbf{h}$ is the cross product of $\mathbf{r}$ and $\mathbf{v}$ and so it must always be perpendicular to both, which means that $\mathbf{r}$ and $\mathbf{v}$ must always remain in the same plane. This plane is referred to as the orbital plane.

**C.2 KEPLERIAN ORBITS**

To find the position of the body the equation of motion, equation (C.2), must be integrated. Cross multiplying with $\mathbf{h}$ leads to a form that can be integrated.

$$\ddot{\mathbf{r}} \times \mathbf{h} = \frac{\mu}{r^3} (\mathbf{h} \times \mathbf{r})$$  \hspace{1cm} (C.13)

The left side can be rewritten as

$$\frac{d}{dt}(\dot{\mathbf{r}} \times \mathbf{h}) = \dot{\mathbf{r}} \times \mathbf{h} + \dot{\mathbf{r}} \times \dot{\mathbf{h}} = \dot{\mathbf{r}} \times \dot{\mathbf{h}}$$  \hspace{1cm} (C.14)
Using the fact that $\mathbf{r} \cdot \dot{\mathbf{r}} = r \dot{r}$ and expanding $\mathbf{h}$

$$\frac{\mu}{r^3} (\mathbf{h} \times \mathbf{r}) = \frac{\mu}{r^3} (\mathbf{r} \times \mathbf{v}) \times \mathbf{r} = \frac{\mu}{r^3} [\mathbf{v}(\mathbf{r} \cdot \mathbf{r}) - \mathbf{r} (\mathbf{r} \cdot \mathbf{v})] = \frac{\mu}{r} \mathbf{v} - \frac{\mu \dot{r}}{r^2} \mathbf{r}$$ (C.15)

Rewriting the result into a time rate of change of a vector

$$\frac{\mu}{r} \mathbf{v} - \frac{\mu \dot{r}}{r^2} \mathbf{r} = \frac{d}{dt} \left( \frac{\mathbf{r}}{r} \right)$$ (C.16)

Equation (C.13) can then be rewritten as

$$\frac{d}{dt} (\mathbf{r} \times \mathbf{h}) = \frac{d}{dt} \left( \frac{\mathbf{r}}{r} \right)$$ (C.17)

Integrating this yields

$$\dot{\mathbf{r}} \times \mathbf{h} = \frac{\mu}{r} \mathbf{r} + \mathbf{B}$$ (C.18)

where $\mathbf{B}$ is the vector constant of integration. Multiplying this with $\mathbf{r}$ gives us a scalar equation

$$\mathbf{r} \cdot \dot{\mathbf{r}} \times \mathbf{h} = \mathbf{r} \cdot \frac{\mu}{r} \mathbf{r} + \mathbf{r} \cdot \mathbf{B}$$ (C.19)

Using the mathematical rules $\mathbf{a} \cdot \mathbf{b} \times \mathbf{c} = \mathbf{a} \times \mathbf{b} \cdot \mathbf{c}$ and $\mathbf{a} \cdot \mathbf{a} = a^2$

$$\mathbf{h} = \mu \mathbf{r} + r \mathbf{B} \cos \theta$$ (C.20)

where $\theta$ is the angle between the constant vector $\mathbf{B}$ and the radius vector $\mathbf{r}$. Isolating for $\mathbf{r}$ yields

$$\dot{r} = \frac{h^2 / \mu}{1 + (B / \mu) \cos \theta}$$ (C.21)

This equation is similar to the formula of a conic section written in polar coordinates [Bate, et al., 1971]

$$\mathbf{r} = \frac{p}{1 + e \cos \theta}$$ (C.22)

where

$p$ is the geometrical constant of the conic, i.e. the ‘semi-latus rectum’.

$e$ is the eccentricity, determining the shape of conic section.

$\theta$ is the polar angle.

The similarity between equations (C.21) and (C.22) verifies Kepler’s laws. This similarity permits the inclusion of all orbital motion along any conic section path, not just ellipses. Figure C. shows the different types of section through a cone, representing different types of orbital motion.
From the mathematical definition of a conic section it can be shown that (except for a parabola)

$$e = \frac{c}{a}$$ \hspace{1cm} (C.23)

and that [Bate, et al., 1971]

$$p = a(1 - e^2)$$ \hspace{1cm} (C.24)

The distance from the prime focus to the periapsis can be found by substituting a polar angle of 0° and equation (C.24) into the polar equation of a conic section, equation (C.22) [Bate, et al., 1971].

$$r_p = \frac{p}{1 + e} = a(1 - e)$$ \hspace{1cm} (C.25)

The distance between prime focus and apoapsis can be found in a similar manner, by substituting a polar angle of 180° [Bate, et al., 1971].

$$r_a = \frac{p}{1 - e} = a(1 + e)$$ \hspace{1cm} (C.26)

One final simple relation, that holds for all the conic orbits, for the energy can be derived. From equation (C.21) and (C.22) we know that

$$p = \frac{h^2}{\mu}$$ \hspace{1cm} (C.27)

And by the fact that the specific angular moment, $h$, is constant

$$h = r_p v_p = r_a v_a$$ \hspace{1cm} (C.28)
By rewriting the energy equation (C.8) for periapsis conditions and substituting from equation (C.28).

\[ \mathcal{E} = \frac{v^2}{2} - \frac{\mu}{r} = \frac{h^2}{r_p^2} - \frac{\mu}{r_p} \] (C.29)

Then, substituting equation (C.24) into equation (C.27) and isolating for \( h \).

\[ h^2 = \mu a (1 - e^2) \] (C.30)

This equation is then substituted into equation (C.29) and reduced to [Bate, et al., 1971]

\[ \mathcal{E} = -\frac{\mu}{2a} \] (C.31)

This simple relation holds for all conic orbits. The shape of the conic section depends on the value of the eccentricity. When the eccentricity is 0 the shape of the section is a circle \((r = p)\). For eccentricity values of \(0 < e < 1\) the shape of the conic section is an ellipse. When the eccentricity is exactly 1 the section is a parabola. And finally, for eccentricities exceeding 1 the section is a hyperbola. Some qualities of the various conic sections are given below in Table C.9.3.

<table>
<thead>
<tr>
<th>Conic</th>
<th>Total Energy, ( \mathcal{E} )</th>
<th>Semi-major Axis, ( a )</th>
<th>Eccentricity, ( e )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Circle</td>
<td>&lt; 0</td>
<td>&gt; 0</td>
<td>= 0</td>
</tr>
<tr>
<td>Ellipse</td>
<td>&lt; 0</td>
<td>&gt; 0</td>
<td>0 &lt; e &lt; 1</td>
</tr>
<tr>
<td>Parabola</td>
<td>= 0</td>
<td>( \infty )</td>
<td>= 1</td>
</tr>
<tr>
<td>Hyperbola</td>
<td>&gt; 0</td>
<td>&lt; 0</td>
<td>&gt; 1</td>
</tr>
</tbody>
</table>

*Table C.9.3 Orbit properties for the four conic sections [Bate, et al., 1971].*

The conic sections will be discussed in more detail below.

**The Elliptical Orbit**

From the geometry of an ellipse we can determine

\[ r_p + r_a = 2a \] (C.32)

From equation (C.23) the eccentricity can be written as a function of the apoapsis and periapsis

\[ e = \frac{r_a - r_p}{r_a + r_p} \] (C.33)

The velocity at a given \( r \) is acquired by substituting equation (C.31) into equation (C.8) [Bate, et al., 1971]

\[ v = \sqrt{\frac{2}{\mu}(\frac{1}{r} - \frac{1}{a})} \] (C.34)

The velocity at periapsis is maximal, and can be written as a function of semi-major axis \( a \) and eccentricity \( e \) using equations (C.25) and (C.26).
Appendix C

\[ v_p = \sqrt{\frac{\mu \left(1 + e\right)}{a \left(1 - e\right)}} \]  
(C.35)

The same can be performed for the apoapsis.

\[ v_a = \sqrt{\frac{\mu \left(1 - e\right)}{a \left(1 + e\right)}} \]  
(C.36)

The escape velocity for an ellipse entails the velocity necessary to change the orbit from a closed ellipse to an open parabola. The parabola has a semi-major axis value of infinity \((a = \infty)\), and substituting this into equation (C.34) gives the following expression for the escape velocity [Bate, et al., 1971]

\[ v_{esc} = \sqrt{\frac{2\mu}{r}} \]  
(C.37)

The final remaining derivation is that of the orbital period. The tangential component of the velocity can be written as \(r \dot{\theta}\). Substituting this into the scalar equivalent of equation (C.12) yields

\[ h = r \cdot r \cdot \dot{\theta} = \frac{r^2 d\theta}{dt} \]  
(C.38)

This can be rewritten to

\[ dt = \frac{r^2}{h} d\theta \]  
(C.39)

From basic calculus we know that the differential area swept out by a radius vector as it moves along an angle is

\[ dA = \frac{1}{2} r^2 d\theta \]  
(C.40)

This is shown graphically in Figure C.2.

![Figure C.2 Differential element of area](image)

The differential element of area can be substituted into equation (C.39) leading to

\[ dt = \frac{2}{h} dA \]  
(C.41)

Integrating this for one full orbital revolution yields
This can then be rewritten by using equations (C.23), (C.24), (C.27) and the fact that $a^2 = b^2 + c^2$ for an ellipse [Bate, et al., 1971].

$$T = \frac{2\pi a b}{h}$$  \hspace{1cm} (C.42)

$$T = \frac{2\pi}{\sqrt{\mu}} \sqrt{\frac{a^3}{\mu}} = 2\pi \sqrt{\frac{a^3}{\mu}}$$  \hspace{1cm} (C.43)

This shows that the period is only dependent on the size of the semi-major axis $a$. It also proves Kepler’s third law, which was empirically obtained.

**The Circular Orbit**

The circular orbit could be said to be a special case of the elliptical orbit where the apoapsis and periapsis altitude are identical. This gives a constant circular velocity of [Bate, et al., 1971]

$$v_c = \frac{\sqrt{\mu}}{a}$$  \hspace{1cm} (C.44)

**The Parabolic Orbit**

The parabolic orbit has an eccentricity of $e = 1$, making it the borderline case between an open and a closed orbit. It does not occur in nature (although some comets are in an almost parabolic orbit). The speed of an object in a parabolic orbit is merely the escape velocity, equation (C.37).

**The Hyperbolic Orbit**

A hyperbolic orbit is a conic section with an eccentricity greater than $e = 1$. The hyperbola has two branches, although only one of them is relevant in astrodynamics. The branch wrapped around the focus represents an attraction between two bodies, while the other branch represents a repulsion between them (for the same focus). For a parabola, the velocity of an object will reach zero at an infinite distance from the main body, this can be seen by substituting an infinite distance into equation (C.37). Seeing as how an object in a hyperbolic trajectory has a greater than escape velocity it would be expected that the object would attain a certain finite velocity at an infinite distance from the main body. This residual speed at infinite distance is named the hyperbolic excess speed. This speed can be calculated from the energy equation written for two separate points, a point near the main body and at infinite distance from it.

$$\mathcal{E} = \frac{v^2}{2} - \frac{\mu}{r} = \frac{v_{\infty}^2}{2} - \frac{\mu}{r_{\infty}}$$  \hspace{1cm} (C.45)

This simplifies to

$$v_{\infty}^2 = v^2 - v_{esc}^2$$  \hspace{1cm} (C.46)

The velocity is smallest at infinite distance and greatest at periapsis. The velocity at periapsis can be calculated using a slightly adapted equation (C.35). Of course, a distance at infinity is purely theoretical. Once an object is far enough from the main gravitational body it has, in a practical sense, escaped. A

---

‡ The semi-major axis is negative so the equations includes a minus prior to the semi-major axis term.
sphere of influence can be defined around any gravitational body, objects outside this sphere can then be said to have escaped from it.
APPENDIX D

Overview of functions and programs created in the course of this thesis.

Exponential Sinusoid Analysis

test_exposin.m Program to construct groups of exponential sinusoid shapes based on arbitrary points in a three dimensional space.

func_exposin.m Function responsible for the calculation of the group of valid exponential sinusoids between the specified objects. The function not only determines shape but provides vectors storing many of the parameters such as position, velocity, thrust, acceleration, current mass etc.

func_tofexposin.m Function responsible for the calculation of the time of flight based on provided exponential sinusoid parameters.

func_mfexposin.m Function responsible for the calculation of the total fuel use (in kg) based on provided exponential sinusoid parameters and initial mass.

func_exposinsplot.m Function responsible for producing plots of the trajectory shape, fuel use, etc.

Planechanging

planechanging Program used to explore the evolution of orbital elements and to find the transfer time and propellant use for a continuous low thrust circle-to-circle orbit plane change transfer.

planechanging_in_steps This program is identical to the previous program but performs a single orbit plane change in parts in order to constrain the eccentricity of the maneuver.

Perturbation Analysis

perturbations_earth Program used for the analysis of perturbations in the surroundings of the Earth.

perturbations_sun Program used for the analysis of perturbations as a function of distance from the Sun.

Gravity Assist Analysis

gavityassistanalysis Program used to inspect the energy and angle/direction change for a variety of planets.

Multi-leg Optimization Program

ga_exposin_multileg_run User input file to start a multi-leg optimization run.

ga_exposin_multileg Main fitness function that produces the total mass use of the transfer for an individual in the optimization population.

func_eph Ephemeris file that produces a Cartesian position and velocity vector when given an object number (e.g. 5, which corresponds to Jupiter) and date.

func_eph_asteroids Supporting ephemeris function for asteroids.

func_eph_moon Supporting ephemeris function for the Earth’s Moon.
Appendix D

func_eph_neo
Supporting ephemeris function for near earth objects.

func_eph_planets
Supporting ephemeris function for the planets in the solar system.

func_oneleg
Calculates the mass use during one particular leg (between two objects) of the total trajectory. An individual of an Earth Venus Mercury trajectory would call on this function twice as the trajectory consists of two legs.

func_toorbitsun
Similar to func_oneleg above, but incorporates a transfer from an object to an orbit around the Sun.

func_oneleg_penalty
Also similar to func_oneleg, but incorporates an additional loop to check whether engine specifications are being exceeded during flight. If this is the case, the design is penalized accordingly.

func_swingby
Determines the differences in excess velocities at the swing-by object, and if the angle change of the orbit plane falls within acceptable parameters. The difference in excess velocity is penalized by imposing a proportionate penalty mass use. If the plane change falls outside of acceptable parameters the penalty is such that the individual is no longer considered fit.

func_swingbydata
Contains planetary gravitational constant and radius, used for the computation of swing-bys.

func_findtof
Function that finds the correct flight path angle when provided with a time of flight. Based on the shape the function will find either a single or two valid flight path angles.

func_tofexposin
Function responsible for the calculation of the time of flight based on provided exponential sinusoid parameters.

func_mfexposin
Function responsible for the calculation of the total fuel use (in kg) based on provided exponential sinusoid parameters and initial mass.

func_orbitperiod
Function that provides the orbit period of the chosen body (e.g. when Earth is selected 365 days is returned). This function is purely aesthetic, and ensures that trajectory plots show the correct orbit paths of selected planets.

func_planetplot
Another purely aesthetic function that provides the color and size of a selected planet for plotting purposes.

func_ga_plot
Function responsible for producing plots of the trajectory shape, fuel use, etc.

func_excarrvelpen
Function that computes the velocity mismatch of a spacecraft when it arrives at its destination. Used for imposing penalties upon the design.

func_excdepvelpen
Function that computes the velocity mismatch of a spacecraft when it departs from its origin. Used for imposing penalties upon the design.

func_massuse
Function that computes propellant use based on parameters. This function can be called by various integrators for use during optimization.

func_monte_carlo
Function that can call one of the fitness functions and repeatedly samples the searchspace. Best candidates can be stored and used as an initial population for the Genetic Algorithm.

func_penalty
File containing all the penalty function shapes that the user has specified.

Fitness Functions

ga_exposin_multileg
Standard Fitness Function File

ga_exposin_multileg_ap
Fitness function file that allows for the penalization of a transfer exceeding specified engine capabilities.

ga_exposin_multileg_mt
Fitness function that allows for the optimization of mass per time.

ga_exposin_multileg_mt
Fitness function that allows for the optimization of mass per time, and also allows for the checking of exceeding engine capabilities.

ga_exposin_multileg_sun
Fitness function used for transfers to an orbit around the Sun.
Appendix D

Terminal Velocity Analysis

excessarrival This program takes an existing single leg transfer solution and varies design parameters slightly to inspect the evolution in velocity mismatches.

Searchspace Exploration

exposin_searchspace This program takes an existing solution and samples the searchspace

Supporting Functions

func_rotate Rotation function, cf. Appendix B.
func_kep2cart Function that converts Keplerian orbital elements to Cartesian coordinates.

Function Block Overview

The picture below shows a basic flow of the functions and supporting functions used for the multi-leg optimization program.

Figure D.1 Function Block Overview.
A selection of search space figures of the Earth Jupiter Sun and Earth Venus Sun transfers.

**APPENDIX E**

**Search Space Images of the Earth Jupiter Sun transfer.**
Search Space Images of the Earth Venus Sun transfer.
Search Space Images of the Earth Venus Sun transfer.