MASTER'S THESIS

Development of a Trajectory Modeling Software for Spacecrafts in Earth Orbit as well as Interplanetary Transfers

Ishan Basyal
2013

Master of Science (120 credits)
Space Engineering - Space Master

Luleå University of Technology
Department of Computer Science, Electrical and Space Engineering

LULEÅ UNIVERSITY OF TECHNOLOGY

European Space Master Program Round 7

Master Thesis (Final Version)

Development of a Trajectory Modeling Software for Spacecrafts in Earth Orbit as well as Interplanetary Transfers

Ishan Basyal

September 2013

Supervisors: Mr. Uwe Bruege
Mr. Tobias Lutz

Abstract

Trajectory modeling is one of the most important aspects of any mission design. The trajectory should be able to propagate the S/C to the final destination while optimizing the flight duration, the total change in velocity and also the total launch mass. The Spacecraft Trajectory Optimizer (STO) tool described in this report first solves the Gauss Lambert problem and generates initial departure and arrival conditions which can also be expressed as porkchop plots. These initial conditions are then used as input to optimize the flight steps which are based on a patched conic approximation with the elliptical transfer with respect to the Sun and the hyperbolic transfers at the departure and arrival planet's sphere of influence. The tool is completely based on MATLAB 2007 or later and uses ODE45 for trajectory propagation and FMINCON with *Active-set* algorithm for optimization. The results obtained in house were compared with four Mars Sample return orbits calculated at ESOC and there is a very good correlation between the required change in velocities and transfer duration for e.g. Orbit case: $O22S$, ESOC values: total $\Delta V = 3.946 - 4.119$ $[km/s]$, $TOF = 329 - 342$ $[days]$ & STO values: $\Delta V = 3.986$ $[km/s]$ & $TOF = 335$ $[days]$. The in house data was also used as an input in the System Tool Kit (a professional trajectory calculation software) for modeling an interplanetary trajectory to Mars and the S/C arrived at Mars without any optimization. Therefore, even though the STO does not have all the capabilities of a professional software it can be used for preliminary mission analysis as it offers quite accurate results for interplanetary transfers.

Contents

Acknowledgements

First and foremost I would like to offer my sincerest gratitude to my supervisors, Mr. Tobias Lutz and Mr. Uwe Bruege for their support, patience, encouragement and their trust in allowing me the room to work on my own, through out my thesis. Without them this thesis would not have been possible and I could not wish for better or more supportive supervisors. Additionally I would also like to extend my gratitude to the On-board S/W & GNC department (TEA41) Bremen for making my stay in Bremen enjoyable and also helping me learn the German language.

At the Université Paul Sabatier (UPS) Toulouse III, my special thanks goes to Prof. Christophe Peymirat for his support and advice during the semester in Toulouse. I would also like to acknowledge and thank Ms. Maude Perier Camby and Héléne Perea for their assistance with all the issues academic as well as administrative and making my stay in Toulouse a pleasant one. I would also like to thank Prof. Peter Von Ballmoos and Prof. Dominique Toublanc at UPS for their support.

At the Luleå University of Technology I would first like to thank Prof. Victoria Barabash for her continuous support in the last two years. My thanks also goes to Prof. Johnny Ejemalm for his support during the thesis. I would also like to specially thank Ms. Anette Snällfot-Brändström and Ms. Maria Winnebäck for always being there to help with all the bureaucratic and non-bureaucratic problems I had during the Spacemaster program.

The last two years in the Spacemaster program have been very important for my personal and professional development and I would like to thank the amazing people I met in Spacemaster, in the M2PTSI program in Toulouse and during my thesis in Bremen. Thank you for a truly great experience and it has been a pleasure knowing you all.

I would also like to thank Prof. Joachim Vogt and Prof. Eveline Gottzein who despite their busy schedule always find time for me for any advice I seek and have supported me in every way possible in the last few years. This thesis would also not be possible without the love and support of my host family in Bremen, Wiburg and Gert, who gave me a home away from home.

Last, but by no means least, I would like to my express my deepest gratitude to my grand mother, my parents and my sister. I am forever grateful for your unending love and support and for instilling within me the love for creative and scientific pursuits. Thank you for everything and without you none of this would be possible.

List of Figures

List of Tables

Abbreviations

AGI	Analytical Graphics Inc.
ANSI	American National Standards Institute
CK	C-matrix Kernel
DCM	Direction Cosine Matrix
EADS	European Aeronautic Defence and Space Company
ECC	Earth Course Correction
EK	Events Kernel
ESA	European Space agency
ESOC	European Space Operations Center
FMINCON	Find Minimum of Constrained Nonlinear Multivariable Function
FORTRAN	Formula Translation
GLO	Gauss Lambert Optimizer
GLS	Gauss Lambert Solver
GUI	Graphical User Interface
IK	Instruments Kernel
JPL	Jet Propulsion Laboratory
JUICE	Jupiter Icy Moon Explorer Mission
MATLAB	Matrix Laboratory
MCC	Mars Course Correction
MICE	SPICE toolkit for MATLAB
MOI	Mars Orbital Insertion

MREP	Mars Robotic Exploration Program
NAIF	Navigation and Ancillary Information Facility
NASA	National Aeronautics and Space Administration
ODE113	Adams Moulton Predictor Method
ODE45	Fourth Order Runge Kutta Method
PCK	Planet C-matrix Kernel
RAAN	Right Ascension of the Ascending Node
S/C	Spacecraft
SPICE	Spacecraft Planet Instrument C-matrix Events
SPK	Spacecrafts Planets Kernel
SQP	Sequential Quadratic Programming
STK	System Tool Kit
STO	Spacecraft Trajectory Optimizer
TEI	Trans Earth Injection
TMI	Trans Mars Injection
TOF	Time Of Flight

Chapter 1

Introduction

The master thesis was performed at the EADS Astrium *Space Transportation* site in Bremen as a course requirement for the *European Space Master* course. The trajectory modeling project is being developed as an in-house orbital analysis tool comparable to the System Tool Kit software developed by AGI. The software aims to simulate satellite and spacecraft trajectories orbiting the Earth as well as travelling to the celestial bodies in the Solar system. As the license for the STK software is very expensive, the in-house software will be used for initial trajectory, delta velocity and time of transfer analysis and towards the final phases of the mission analysis the STK software will be used to check the integrity of the in-house data and also for accurate simulation of the orbital parameters. The following sections describe the necessity of having such a software and also the expected improvements on the existing software.

1.1 Motivation

After a stagnation of over forty years there has been a renewed interest in human space exploration in the recent years namely due to the announcements of companies such as Planetary resources, SpaceX and Mars One. These ambitious companies aim not only to populate planet Mars but also aim to mine asteroids for resources [23] [26] [25]. Apart from these private companies the governmental organizations such as NASA and ESA also have planned projects such as the JUICE [8], Asteroid Retrieval Initiative [2] and so on.

The first step in achieving these missions lies in the proper calculation of the launch window, the time of transfer, the required fuel for velocity change and also the trajectory. As is evident from the proposed human mission to Mars, the challenges do not lie mostly in the hardware but rather in the round trip time. A long space travel exposes the astronauts to radiation, psychological trauma and also other factors such as muscle atrophy, osteoporosis, slowing of the cardiovascular system so finding the optimum trajectory is the basis of all deep space exploration projects.

An orbital transfer tool to the Moon was already developed prior to this thesis. The system uses the Optimization toolbox within MATLAB and the solvers used are ODE45 and FMINCON

to optimize the trajectory to the Moon. An additional feature in this initial orbital transfer tool is a toolbox named "Lunatool" which finds a launch window to the moon and then optimizes the trajectory in four steps. Even though this tool is quite useful it does not work for any other system other than the Sun, the Earth and the Moon, so extra features have to be added to the program to make it useful for various missions within the Solar system and the features added during the thesis will be discussed in the following section.

1.2 Objectives

The following objectives were outlined as the possible improvements during the master thesis.

1. Improve analysis capability of the STO software.
    - Develop a feature to generate porkchop plots for preliminary mission analysis for objects within the Solar system.
    - Develop a tool to find launch windows for interplanetary transfers including non-Hohmann transfers.
    - Develop a tool to correctly propagate and optimize Type I transfers less than 180^o) and Type II(transfers less than 360^o) trajectories using the patched conic approximation.
2. Improve the GUI and the graphics of the software.
    - Change the plane from the ecliptic to the Earth equatorial for S/C orbiting the Earth and also include the axial tilt of the Earth.
    - Update the planetary database to include other planets and also the Martian and Jovian Moons.
    - Check for other integration methods that may be more suitable than ODE45.
    - Compare the performance of Quaternions over DCM to check for increased computation speed or for better results and replace DCM's if necessary.
    - Try to resolve plot function issues i.e. the generated graphs do not have labels so try to make them more easy to understand.
3. Cross check the tool by performing an interplanetary mission analysis and compare the results with professional sources and softwares like the STK.

1.3 Report Layout

The report has been organized mainly into three parts. The first part of the report the *Theoretical Background* deals with the basic concepts of Astrodynamics and also with the optimization toolbox as well as the integrators built within MATLAB. This section aims to provide a general overview so that the STO and simulations of the trajectories can easily be understood.

The second part of the report deals with defining the operation of the software from generating the ephemeris, defining the flight steps and the operation of the toolboxes and their constraints and concepts. Then the case study of a simulated *Sample return mission* to Mars is discussed and along with the importance of such a mission and its objectives.

The third part of the report compares the results obtained from the STO software with the ones obtained at European Space Operations Center for integrity and also the validity of calculated data by checking it in STK. Then finally the conclusion of the report deals with the achieved improvements, the application of this software for trajectory calculation for other missions and also the possible future improvements.

Chapter 2

Theoretical Background

This chapter deals with the basic formulae required to establish the basis of the trajectory calculations. The Keplerian elements have first been explained along with the notions such as escape velocity, and also the types of possible orbits. Then the conic section approach to solving astrodynamical problems has been explained along with the different integration methods such as ODE45 and ODE113 have been thoroughly illustrated along with the FMINCON function.

2.1 Fundamentals of Orbital Dynamics

Since humans started looking up towards the sky, they have been fascinated with the stars, the planets and space in general. In the old times the Earth was supposed to be the center of the universe and everything else revolved around it. However in the 16^{th} century this view was challenged by Copernicus and the helio-centric definition of the Solar system came into existence. Then in the beginning of the 17^{th} century Johannes Kepler came up with the three basic laws to define the motion of planets around the solar system and these laws with Newton's law of gravitation have come to be the basis of all orbital dynamics. [12] The following sections describe all the basic principles in detail.

2.1.1 Kepler's Laws of Orbital Dynamics

The three laws governing the motion of celestial bodies orbiting the Sun or any central body are as follows [12]:

1. Every planet moves in an elliptical orbit, with the Sun at one focus of the ellipse.
2. The radius vector drawn from the Sun to any planet sweeps out equal areas in equal times.
3. The squares of the periods of revolution of the planets are proportional to the cubes of the semi-major axes of their orbits.

The Keplerian laws can most easily be derived by considering a two body problem and it was first solved by Sir Isaac Newton. If the central body is considered to be much larger than the body orbiting it, then the mass of the other body can be neglected. In addition to the

negligible mass the force exerted on the satellite always points towards the center of the central body because the applied force is always anti-parallel to the position vector thus eliminating any acceleration perpendicular to the plane, hence its orbit is always confined to a plane at all times.

Assuming the two body approximations mentioned above, Newton's laws of gravitation define the acceleration of the orbiting body to be given by:

$$\ddot{\mathbf{r}} = -\frac{G\,M}{r^3}\mathbf{r} \tag{2.1}$$

Then if a cross product of equation 2.1 and the position vector of the orbiting body is taken, the right hand side reduces to zero because of the vector property of cross product which states that a cross product of a vector with itself is always zero. The left hand side of equation [2.1] can then be written as

$$\mathbf{r} \times \ddot{\mathbf{r}} = \mathbf{r} \times \ddot{\mathbf{r}} + \dot{\mathbf{r}} \times \dot{\mathbf{r}} = \frac{d}{dt}(\mathbf{r} \times \dot{\mathbf{r}}) \tag{2.2}$$

As the time derivative of $\mathbf{r} \times \dot{\mathbf{r}}$ equals zero, the quantity must be a constant and can be denoted as $\mathbf{h}$, which is known as the angular momentum per unit mass or the specific angular momentum. So if the orbiting object's motion is taken to be linear over a small time step δt the Kepler's second law can then be derived as follows:

$$\Delta A = \frac{1}{2}|\mathbf{r} \times \dot{\mathbf{r}}\ \Delta t| = \frac{1}{2}|\mathbf{h}|\Delta t \tag{2.3}$$

Where, ΔA is the area swept by the radius vector in the interval Δt and since $\mathbf{h}$ has been found to be a constant from equation 2.2 the radius vector sweeps equal areas in equal time intervals and $\mathbf{h}$ is also known as the *areal velocity*.

Using the equation 2.1 and $\mathbf{h}$, other orbital properties can also be found. Multiplying $\mathbf{h}$ with both the sides of equation 2.1 and solving yields:

$$\mathbf{h} \times \ddot{r} = -G\,M\frac{d}{dt}\left(\frac{\mathbf{r}}{r}\right) \tag{2.4}$$

Then integrating equation 2.4 results in

$$\mathbf{h} \times \dot{r} = -G\,M\left(\frac{\mathbf{r}}{r}\right) - \mathbf{A} \tag{2.5}$$

Where $\mathbf{A}$, called the *Laplace vector* is a constant of integration that is determined by the initial position and velocity of the orbiting body. Due to vector properties, $\mathbf{A}$ also lies in the orbital plane and multiplying equation 2.4 with $\mathbf{r}$ results in a new parameter known as the *true anomaly* ν which is the angle between $\mathbf{A}$ and $\mathbf{r}$.

$$\begin{aligned}(\mathbf{h} \times \dot{\mathbf{r}}) \cdot \mathbf{r} &= -G\,M\,r - \mathbf{A} \cdot \mathbf{r} \\ h^2 &= G\,M\,r + A\,r\,cos\nu\end{aligned} \tag{2.6}$$

If the left hand side of the equation 2.5 is simplified using the identity $(\mathbf{a} \times \mathbf{b})\, \dot{\mathbf{c}} = -(\mathbf{c} \times \mathbf{b})\, \dot{\mathbf{a}}$ two auxiliary quantities; the parameter and the eccentricity of the conic can be defined as:

$$p = \frac{h^2}{G\,M} \tag{2.7}$$

$$e = \frac{A}{G\,M} \tag{2.8}$$

From the preceding equations 2.8 & 2.7 the distance of the orbiting body r can be found from the reference direction $\mathbf{A}$. The path in the orbital plane is dependent on the eccentricity and equation 2.9 illustrates this relation.

$$r = \frac{p}{1 + e\,cos\nu} \tag{2.9}$$

Depending upon the value of the true anomaly the distance from the center varies from r_{min} for a true anomaly equal to zero and r_{max} can range upto infinity for parabolic as well as hyperbolic orbits. From the maximum and minimum values of r the semi-major axis of the orbit can be defined as follows:

$$a = \frac{1}{2}\,(r_{min} + r_{max}) = \frac{1}{2}\left(\frac{p}{1+e} + \frac{1}{1-e}\right) = \frac{p}{1-e^2} = \frac{h^2}{G\,M\,(1-e^2)} \tag{2.10}$$

In orbital terms the true anomaly is measured from the point of closest approach i.e. r_{min} which is also known as the perigee and the farthest point r_{max} is known as the apogee. The eccentricity of the orbit defines its shape and size and it will be discussed in detail in 2.2.

The third Kepler's law can be derived from the Energy relation of the orbit. If the equation 2.5 is squared, $|\dot{r}|$ is replaced by v and the inverse of equation 2.10 is used the *vis-viva* law is obtained:

$$v^2 = G\,M\left(\frac{2}{r} - \frac{1}{a}\right) \tag{2.11}$$

The equation 2.11 is equivalent to the total energy of the orbit and it can also be observed that the total energy of the orbit is dependent only upon the semi-major axis and not the eccentricity of the orbit. For elliptical orbits the energy is negative, for parabolic orbits the energy is equal to zero whereas for a hyperbolic orbit the energy is positive which implies the orbiting object reaches infinity.

Proceeding with the *vis-viva* equation the time dependence of motion in orbit and the orbital period can be described not only for circular but also for elliptical orbits, which is also the third Kepler law.

To derive the third law the *Eccentric Anomaly* of the orbit has to be defined. The eccentric anomaly for an orbit assuming a two dimensional case is defined via the following figure 2.1 and equations:

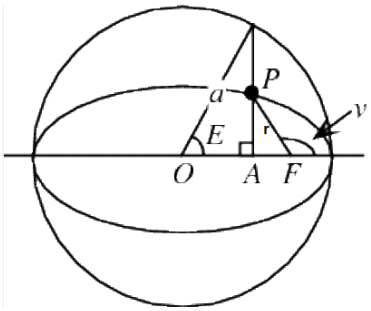


Figure 2.1: Figure illustrating the eccentric anomaly [37]

$$\begin{aligned}
&\hat{x} = r\ cos\ \nu = a(cos\ E - e) \\
&\hat{y} = r\ sin\ \nu = a\sqrt{(1-e^2)}\ sin\ E \\
&i.e.\ in\ Polar\ co\text{-}ordinates:\ \ r = a(1 - e\ cos\ E)
\end{aligned} \tag{2.12}$$

From the corresponding Cartesian equations of the polar distance in equation 2.12 the areal velocity **h** can be expressed as a function of the eccentric anomaly as follows:

$$\begin{aligned}
&h = \hat{x} \cdot \dot{\hat{y}} - \hat{y} \cdot \dot{\hat{x}} \\
&= a(cos(E) - e) \cdot a\sqrt{1-e^2}\ cos(E)\ \dot{E} + a\sqrt{1-e^2}\ sin(E) \cdot a\ sin(E)\ \dot{E} \\
&= a^2\sqrt{1-e^2}\dot{E}\ (1 - e\ cos(E))
\end{aligned} \tag{2.13}$$

And using the relation $h = \sqrt{G\ M\ a(1-e^2)}$ the eccentric anomaly of the orbit can in turn be expressed with respect to the mean motion in the orbit:

$$\begin{aligned}
&(1 - e\ cos\ E)\ \dot{E} = n \\
&where,\ mean\ motion:\ \ n = \sqrt{\frac{G\ M}{a^3}}
\end{aligned} \tag{2.14}$$

Integrating the equation 2.14 with respect to time, the final equation for the eccentric anomaly and its relation to the time since perigee passage is i.e. the *mean anomaly* is obtained. *Mean anomaly* is an increasing quantity that is used to describe the orbit.

$$\begin{aligned}
&E(t) - e\ sin\ E(t) = n\ (t - t_p) \\
&M = n\ (t - t_p) \\
&Instantaneous\ Mean\ anomaly:\ \ M = M_o + n\ (t - t_o)
\end{aligned} \tag{2.15}$$

So finally the time relation between orbital motion can be generated from the equation 2.15. A complete rotation is the total change of the mean anomaly with 2π and as it is proportional

to the inverse of the mean motion n, the Kepler's third law for orbits with eccentricities less than one is represented as follows:

$$T = \frac{2\pi}{n} = 2\,\pi\sqrt{\frac{a^3}{G\,M}} \tag{2.16}$$

Note: Unless otherwise stated all the Kepler's laws were derived from Montenbruck et. al. [22].

2.1.2 Orbital Parameters

The position of any object in orbit is defined with the help of six orbital parameters, which are also known as Keplerian elements. Three Keplerian elements *eccentricity*, *semi-major axis* and *true/mean anomaly* have already been derived in the preceding subsection 2.1.1, however orbital elements have not yet been mentioned.

1. **Eccentricity:** The parameter determining the shape of the orbit is known as the eccentricity. It is a property of the conic section that ranges from a circle to a hyperbola. For a circle the eccentricity equals zero, for an ellipse it is less than one, for a parabola it is equal to one and finally for a hyperbola it is greater than one.

2. **Semi-major axis:** Depending upon the conic the semi-major axis is defined as the half of the sum of periapsis and apoapsis distance for eccentricites less than one. The semi-major axis for eccentricities greater than or equal to one will be described in detail in the section 2.2.

3. **True/Mean Anomaly:** The quantity describing the time since the perigee passage is known as the true anomaly and it is denoted by μ. The true anomaly ranges from 0 to 2π. However the better representation could be in terms of the mean anomaly as it is always referenced to some epoch to correctly illustrate the position of the orbiting body.

4. **Inclination:** The angle between the orbital plane of the orbiting body (small) and the bigger central body is known as the inclination of the small body. Inclination is measured locally and is relevant according to the reference body. For e.g. the plane of rotation of the Earth around the Sun is known as the ecliptic however the inclination of a satellite is measured according to the Earth while taking into consideration of the equatorial plane and not the ecliptic so it like all the other Keplerian elements is a localized quantity with only the immediate central body in consideration. The convention of representing the inclination of an orbit is with an italicized i and the inclination is measured from the line of nodes i.e. the line where the orbital and the equatorial planes cross over. The inclination of an orbit ranges from 0 to π. This relation will be illustrated in the figure 2.2.

5. **Right Ascension of the Ascending node (RAAN):** The angle between the vernal equinox and the point on the orbit at which the orbiting body crosses the equator from the South to the North is known as the right ascension of the ascending node. The RAAN is represented as Ω and it ranges from 0 to $2\,\pi$.

6. **Argument of perigee:** The angle between the direction of the ascending node i.e. when the orbiting body crosses from the South to the North and the direction of the perigee is known as the argument of perigee and it is represented as ω. The argument of perigee ranges from 0 to π.

The following figure 2.2 illustrates all the orbital parameters.

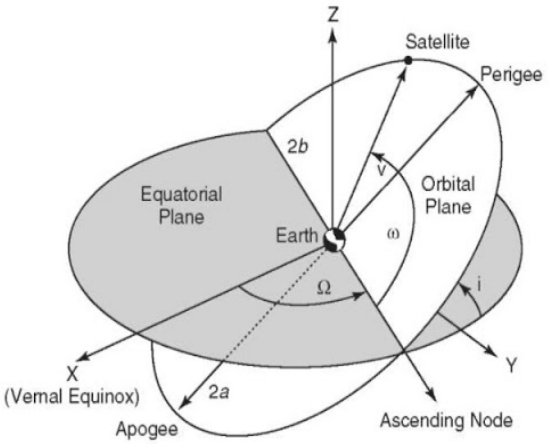


Figure 2.2: Figure illustrating the Orbital elements of a Keplerian orbit [24]

2.1.3 Sphere of Influence

The force of gravitation is always present however depending upon the position in the orbit, the gravitational attraction from different bodies differs in magnitude. From Newton's laws it is known that the gravitational force of attraction is inversely proportional to the square of the distance between the center of the bodies. The multi-body problem can be simplified by making an assumption that when a body is sufficiently close enough to one central body compared to the other bodies the forces of attraction from other bodies can effectively be neglected. This greatly simplifies the problem into just a two body problem as assumed while calculating the Keplerian laws in section 2.1.1. Thus the distance at which the influence of other bodies can be neglected is calculated as follows:

$$Sphere\ of\ influence = a\ \left(\frac{m}{M}\right)^{\frac{2}{5}} \tag{2.17}$$

Where, a is the semi-major axis of the body in consideration for e.g. Earth in a heliocentric frame. Then m represents the mass of the smaller orbiting body for e.g. the Earth and M is the mass of the central body for e.g. the Sun [35].

2.1.4 Escape Velocity

As mentioned earlier in 2.1.3 every object has a sphere of influence. However with sufficient energy this gravitational attraction can be overcome and an object can be free of gravitational

attraction of the central body. So for a body orbiting around a central planet the energy required to exit its sphere of influence is given by:

$$V_{esc} = \sqrt{\frac{2\,G\,M}{r}} \tag{2.18}$$

Where M is the mass of the central body and r is the distance between them. Once a body is supplied with energy equal to or above the escape velocity V_{esc} it is no longer in an orbit around the planet but rather escapes in either a parabolic or a hyperbolic trajectory. So for example to reach Mars the spacecraft should have a velocity at least greater than the escape velocity of the Earth [35].

2.2 Orbital Types and Characteristics

The orbit around a central body is defined according to the eccentricity. The energy of the orbit is also dependent upon the eccentricity thus it is important to understand the various orbits which are crucial in relation to the mission requirements. The different types of orbits and their properties have been explained in detail below.

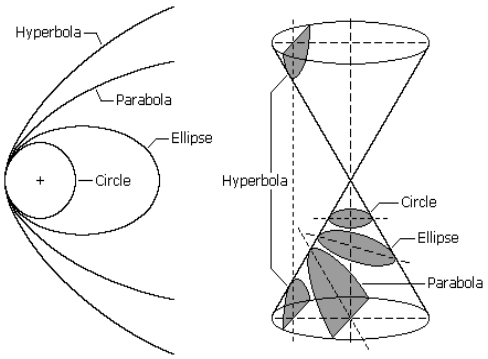


Figure 2.3: Figure illustrating the different types of conic sections [11]

In equation 2.11 the vis-viva law was derived and it was mentioned that it is equivalent to the energy law i.e. the sum of the kinetic as well as potential energy. An object's kinetic and potential energy is given as:

$$E_{kin} = \frac{1}{2} m\, v^2 \tag{2.19}$$

$$E_{pot} = -\frac{G\,M\,m}{r} \tag{2.20}$$

And the total energy is always constant during motion, so:

$$E_{tot} = \frac{1}{2} m\, v^2 - \frac{G\,M\,m}{r} = -\frac{1}{2}\,\frac{G\,M\,m}{a} \tag{2.21}$$

So the total energy of any orbit is only dependent upon the semi-major axis and not the eccentricity of the orbit [22]. The properties of the different orbits has been presented in table 2.1

Table 2.1: A table containing the properties of different orbit types [11]

Conic Section	Eccentricity	Semi-major Axis	Energy
Circle	0	$= \ radius$	< 0
Ellipse	$0 < e < 1$	> 0	< 0
Parabola	1	$infinity$	0
Hyperbola	> 1	< 0	> 0

The three types of orbits used in the STO software are circular - for initial coastal phase, elliptical for the transfer phase in case of simple transfers using the *Lunatool* or the transfer phase around the Sun in the patched conic approximation. And the hyperbolic orbit for the departure and arrival of sections in the patched conic approximation i.e. the *Gauss Lambert Optimizer*. As the basics of finding the orbital elements of a circular/elliptical orbits has already been mentioned in section 2.1 only a brief description is presented here with the equations for calculating the orbital elements. The circular orbit parameters can be obtained by replacing the eccentricity equal to zero and the semi-major axis equal to the radius from the equations for the elliptical orbit.

2.2.1 The Elliptical Orbit

The elliptical orbit is the most used orbit ranging from launching the satellites, to transfers within the solar system. The general properties of an elliptical orbit and the energy have previously been mentioned, so assuming a typical launch scenario as presented in the figure 2.4 the following equations help in determining the orbital elements [11].

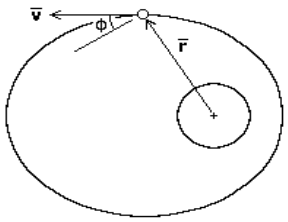


Figure 2.4: Figure illustrating a typical elliptical transfer orbit [11]

$$\left(\frac{R}{r}\right) = \frac{-C \pm \sqrt{C^2 - 4(1-C)(-cos^2\phi)}}{2(1-C)} \qquad where\ C = \frac{2GM}{rv^2} \tag{2.22}$$

$$e = \sqrt{\left(\frac{rv^2}{GM} - 1\right)^2 cos^2\phi + sin^2\phi} \qquad where\ \phi\ \ is\ the\ flight\ path\ angle \tag{2.23}$$

$$tan\,\nu = \frac{\left(\frac{rv^2}{GM}\right) cos\phi\, sin\phi}{\left(\frac{rv^2}{GM}\right) cos^2\phi - 1} \tag{2.24}$$

The quadratic equation 2.22 has two solutions and they correspond to the perigee and apogee radius. So for an elliptical orbit the semi-major axis can then be calculated by taking the half of the sum of perigee and apogee or directly from equation 2.11. Once the semi-major axis, the eccentricity and the true anomaly of an orbit are found, the flight path angle and the position of the orbiting body can recursively be calculated as:

$$\phi = arctan\left(\frac{e\,sin\,\nu}{1 + e\,cos\,\nu}\right) \tag{2.25}$$

$$r = \frac{a(1-e^2)}{1 + e\,cos\nu} \tag{2.26}$$

In the STO software however the orbital elements are calculated from the position and velocity vectors of the central and the orbiting body and a detailed description is given in the subsection 2.4.1.

2.2.2 The Hyperbolic Orbit

In the STO tool the solver can be alternated between a normal approach that takes into account the gravitational attraction of all the bodies present or a patched conic approach where the sphere of influence is calculated and only the gravitational influence of the central body is taken into account. In the patched conic approach the velocity of the spacecraft is higher than the V_{esc} of the central reference body in the departure and arrival phases but less than the V_{esc} of the central reference body in the transfer phase. For e.g. in the transfer to Mars in the patched conic approach the S/C is in hyperbolic trajectory with respect to the Earth and Mars but elliptical with respect to the Sun. As the property of a hyperbolic orbit is different to that of circular and elliptical orbits it has been explained in detail.[1]

A typical hyperbolic orbit is presented in figure 2.5. From the figure it can be observed that a hyperbola has two arms and the arms are asymptotic to the the intersecting straight lines. If the central body is assumed to be on the left focus, the right arm of the hyperbola can be disregarded as the gravitational force is not repulsive. As a hyperbola is also a conic section, the eccentricity (eqn. 2.27) can be defined using the property of the directrix as:

$$e = \frac{c}{a} \tag{2.27}$$

The path of a S/C arriving on a hyperbolic trajectory is turned by an angle equal to the angle of the asymptotes δ when it encounters a central gravitational body and this turning angle is related to the geometry of the hyperbola (2.28) as follows:

$$sin\left(\frac{\delta}{2}\right) = \frac{1}{e} \tag{2.28}$$

[1] A detailed description can be found in Braeunig, R.A. [11] from where this subsection has been almost completely adapted unless stated otherwise.

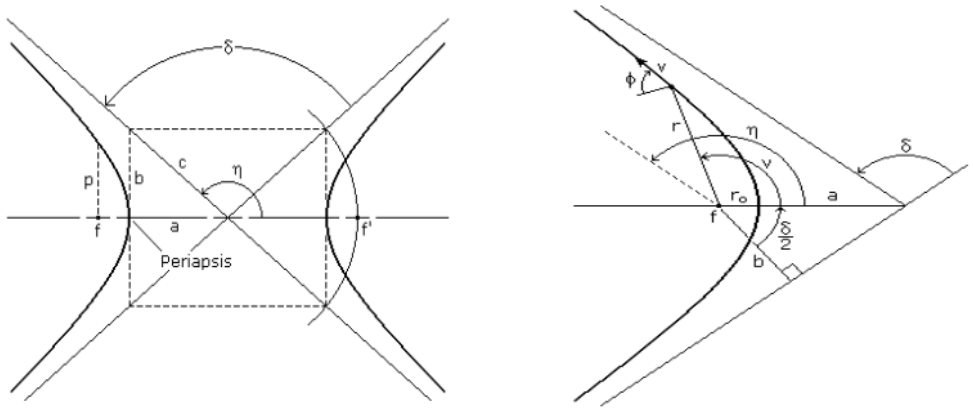


Figure 2.5: Figure illustrating the properties of a hyperbolic orbit [11]

If the radial distance, the instantaneous velocity and the flight path angle at a certain time are known, the eccentricity and the semi-major axis for a hyperbolic orbit can be calculated using equations 2.23 and 2.11. From these values the true anomaly at that position is calculated with equation 2.29:

$$\nu = arccos\left(\frac{a(1-e^2)-r}{e\cdot r}\right) \tag{2.29}$$

As the true anomaly ranges from 0 to 2π whereas *arccos* only generates values between 0 and π the flight path angle ϕ is used to determine the quadrant, so if ϕ is positive ν should be positive and vice-versa. If there is no gravitational deflection between the S/C and the central body a new parameter called the *impact parameter* b (eqn. 2.30) is introduced and it gives the distance of closest approach.

$$b = \frac{-a}{tan\left(\frac{\delta}{2}\right)} \tag{2.30}$$

However if there is a gravitational deflection the S/C and the central body are separated by a perigee distance (eqn: 2.31)

$$r_p = a(1-e) \tag{2.31}$$

The parameter of the conic can be calculated using the equation 2.10. After obtaining the true anomaly the radius vector, flight path angle and the velocity can be calculated in a similar way as the elliptical transfer using the equations 2.26, 2.25 and 2.11. The time of flight was calculated using the eccentric anomaly in subsection 2.1.1 so a similar parameter called the *hyperbolic eccentric anomaly, F* (eqn. 2.32) can be introduced:

$$cosh\, F = \frac{e + cos\,\nu}{1 + e\, cos\,\nu} \tag{2.32}$$

Then from F the time of flight in a hyperbolic orbit is derived as shown in equation 2.33:

$$t - t_o = \sqrt{\frac{(-a)^3}{GM}}[(e\ sinh\ F - F) - (e\ sinh\ F_o - F_o) \tag{2.33}$$

The orbital elements of a hyperbolic orbit are also determined in a similar way to the elliptical orbit, so the detailed description has been presented in the subsection 2.4.1. The hyperbolic departure and arrival conditions have been explained in the subsection 2.4.2.

2.3 Orbital Manoeuvres

The act of modifying the orbital parameters of an object in orbit is known as orbital manoeuvre. Any changes in the orbital elements is dependent upon either the magnitude or direction change of orbital velocity. For simplification reasons the manoeuvre is always taken as an impulse velocity change and this can be assumed to be a realistic estimate because with most of the current propulsion systems the orbital period is much larger than the actual propulsion period. So any manoeuvre to change the orbit of a space vehicle must occur at the intersection of the initial and final orbit e.g. raising the apogee of an orbit. Whereas if there is no intersection of orbits e.g. transfer from the Earth to Mars an intermediate transfer orbit that intersects both the initial and final orbits has to be used and it results in at least two burns [11].

Depending upon the requirements there are various types of orbital manoeuvres which result in coplanar transfer, non-coplanar, bi-elliptic transfer and so on. As the details of these transfers can easily be found in various textbooks such as *Fundamentals of Astrodynamics and Applications., Vallado, D.A.,* the only transfers that were used in the program for generating data have been thoroughly explained in this section. The focus of the program was to study both the time efficient and the energy efficient transfers, so the most obvious choices i.e. Hohmann transfer − energy efficient and One tangent burn − time efficient transfers were studied during the thesis.

2.3.1 Hohmann Transfer

The most energy efficient orbital transfer between two circular orbits was first suggested by Walter Hohmann in 1925 A.D. The author proposed a theory which suggested the minimum change in velocity transfer could be achieved between orbits by using two tangential burns. In the Hohmann transfer there are two burns, first at the initial departure orbit with a flight path angle equal to zero and the final at the destination after travelling a 180^o in the transfer ellipse. The property of this transfer allows tangential burns at departure and arrival thus excludes both parabolic as well as hyperbolic orbits. Another property of this transfer is that the transfer orbit between two circular orbits is elliptical whereas the one between two elliptical orbits might either be circular or elliptical depending upon the geometry of the initial and final orbits [35].

In the Hohmann transfer the departure position is taken as the perigee of the transfer ellipse whereas the arrival point is the apogee of the ellipse, therefore the semi-major axis is easily defined.

$$a_{HT} = \frac{r_A + r_B}{2} \tag{2.34}$$

Then using the equation 2.16 the time of transfer is determined which is half of the total period because the transfer ellipse terminates at a true anomaly of 180 degrees:

$$T_{trans} = \pi \sqrt{\frac{a_{HT}^3}{G\ M}} \tag{2.35}$$

The geometry of the basic Hohmann transfer i.e. between two circular orbits is illustrated in the figure 2.6

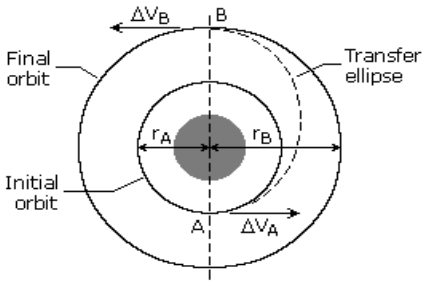


Figure 2.6: Figure illustrating the Hohmann transfer between two circular orbits [11]

For a Hohmann transfer between two circular orbits as presented in 2.6 the following algorithm can be established to solve for all the various required velocity changes [11]:

$$V_A = \sqrt{\frac{G\ M}{r_A}} \qquad \textit{velocity at point A} \tag{2.36}$$

$$V_B = \sqrt{\frac{G\ M}{r_B}} \qquad \textit{velocity at point B} \tag{2.37}$$

$$V_{HT_A} = \sqrt{G\ M\ \left(\frac{2}{r_A} - \frac{1}{a_{HT}}\right)} \qquad \textit{transfer orbit initial velocity} \tag{2.38}$$

$$V_{HT_B} = \sqrt{G\ M\ \left(\frac{2}{r_B} - \frac{1}{a_{HT}}\right)} \qquad \textit{transfer orbit final velocity} \tag{2.39}$$

$$\Delta V_A = V_{HT_A} - V_A \qquad \textit{initial delta V} \tag{2.40}$$

$$\Delta V_B = V_B - V_{HT_B} \qquad \textit{final delta V} \tag{2.41}$$

$$\Delta V_{tot} = \Delta V_A + \Delta V_B \qquad \textit{total delta V} \tag{2.42}$$

As it can be seen from the equations above two crucial elements have been disregarded in this analysis. The first being the gravitational attraction of the other bodies that lie in the circular orbits. For small objects like satellites around the Earth the mass can be neglected however when orbital transfers between planets is planned the calculations only taking the mass of the

Sun as the central body do not lead to accurate results. The second problem with the above mentioned equations is that the planetary orbits are not circular but rather elliptical, so even though these values can be taken as a rough estimate for basic mission planning the following steps must be checked for elliptical orbits [35]:

1. The initial and final orbits have to be co-axially aligned to satisfy the tangential burn condition as well as the true anomaly of each orbit to be either $\pm 180^o$ *or* 0^o. Even though the coaxial alignment is required for the circular case it is a critical requirement for the elliptical orbits.

2. As the orbits are elliptical the velocities are different at both the orbits so equations 2.38 and 2.39 should always be used.

3. The semi-major axes of the initial and final orbits from the trajectory equation always have to be specified i.e. $+ve$ for apogee and $-ve$ for perigee (where the symbols have their usual meanings).

$$a = r\left(\frac{1 + e\,cos\,\nu}{1 - e^2}\right) = \frac{r}{1 \pm e} \tag{2.43}$$

2.3.2 One Tangent Burn

Even though the Hohmann transfer is the most energy efficient it has a big drawback regarding the flight time. For any complete transfer, half of the transfer ellipse has to be traversed. So when faster transfers are required, a trade off between the flight time and the departure velocity has to be done. The ideal case for shortest transfer is to use ΔV that approaches infinity however it is not practical so a realistic ΔV within the available engineering and propulsion constraints has to be found and one of the possible solutions is to use the *One tangent burn*. The one tangent burn differs from the Hohmann not only in terms of energy but it also makes use of intermediate parabolic or hyperbolic orbits. The basic principle of a one tangent burn is illustrated in the picture below:

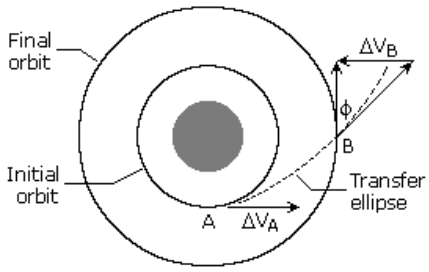


Figure 2.7: Figure illustrating the One tangent transfer between two circular orbits [11]

The only similarity between Hohmann and One tangent burn is that the initial and final orbits must either be circular or coaxial elliptic. The most important feature for a one tangent burn however is the precise knowledge of the true anomaly to locate the position for the final

non-tangential burn. If the transfer is between two elliptical orbits then the true anomaly is identical to the final or initial true anomaly because for this transfer to happen the ellipses have to be co-axially aligned [35].

The easiest way to achieve a one tangent transfer is to select a semi-major axis larger than the one used for hohmann transfer. In the one tangent burn the first burn of the transfer orbit is perpendicular to the initial orbit whereas the second burn takes place at the intersection of the transfer orbit and the desired final orbit where the intersection angle is equal to the flight path angle of the transfer orbit. The transfer orbit can thus be defined by specifying the size of the transfer orbit, angular change of the transfer or the time required to complete the transfer for e.g. a semi-major axis larger than the one of a hohmann transfer $a_{tx} = 1.5 \times a_{HT}$ can be taken and the new orbital parameters such as the travelled angular distance, required velocity transfer and the time of flight can be calculated using the following equations [11]:

$$e_{tx} = 1 - \frac{r_A}{a_{tx}} \qquad \textit{transfer ellipse eccentricity} \tag{2.44}$$

$$\nu = arccos\left[\frac{\left(\frac{a_{tx}\,(1-e^2)}{r_B} - 1\right)}{e}\right] \qquad \textit{true anomaly at second burn} \tag{2.45}$$

$$\phi = arctan\left(\frac{e\,sin\nu}{1 + e\,cos\nu}\right) \qquad \textit{flight path angle at intersection} \tag{2.46}$$

$$\Delta V_B = \sqrt{V_{txB}^2 + V_B^2 - 2\,V_{txB}\,V_B\,cos\phi} \qquad \textit{final delta V} \tag{2.47}$$

$$E = arctan\left(\frac{\sqrt{1-e^2}\,sin\nu}{e + cos\nu}\right) \qquad \textit{eccentric anomaly} \tag{2.48}$$

$$TOF = (E - e\,sinE)\sqrt{\frac{a_{tx}^3}{G\,M}} \qquad \textit{flight time, E in radians} \tag{2.49}$$

All the equations calculated for the Hohmann transfer except equation 2.41 also has to be used with a new semi-major axis a_{tx} to find the velocities at the points A and B of the transfer ellipse.

2.3.3 Orbital Plane Changes

The launch position and time of a spacecraft largely determines its orbital parameters therefore the launch windows are very important. However with the STO tool not much attention is paid to all the orbital parameters but just the RA. If the RA condition is satisfied the launch is possible for any situation and the inclination and other parameters are optimized by the software using the FMINCON function.

The orbital planes of celestial bodies in the solar system do not lie in the same plane so the orbit plane change manoeuvre as well as the inclination change to fit the final conditions has

to be done. Since the ODE45 integrator function just integrates the position from launch until arrival and the inclination and orbit plane angles are calculated at each step and optimized with the final value an optimum orbital plane change manoeuvre in the true sense does not take place so only a trivial description of the orbital plane change manoeuvre is given.

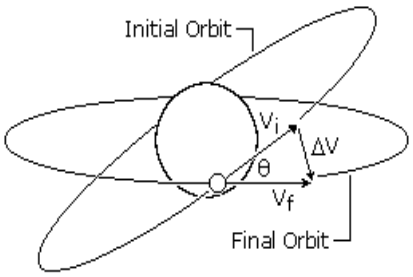


Figure 2.8: Figure illustrating the orbital plane change [11]

Orbital plane changes are conducted by varying the direction of the velocity vector. For an orbiting body the velocity direction is always tangential to the orbit, so for any plane changes a ΔV component perpendicular to the orbital plane or the initial velocity vector is required. For a simple plane change where all the orbital parameter except the inclination remain the same the law of cosines results in the equation 2.50 if the initial and final velocity vectors are equal.

$$\Delta V = 2\, V_i\, sin\left(\frac{\theta}{2}\right) \tag{2.50}$$

Whereas for plane changes where all the orbital parameters are different and also the initial and final velocities are not same the total required change in velocity is calculated as:

$$\Delta V = \sqrt{V_i^2 + V_f^2 - 2\, V_i\, V_f\, cos\, \theta} \tag{2.51}$$

From equations 2.51 and 2.50 it can be seen that plane change manoeuvres are very costly and for e.g. a simple plane change of 60^o already requires a delta V that is equal to the original velocity [11], so the orbital manoeuvres should be carried out at the apogee as the initial velocity at that point is a minimum. In the case of the STO toolboxes the inclination or the orbital plane change in the transfer ellipse is done after the apogee minimization to reduce propulsion consumption.

2.4 Orbit Determination and Trajectory Calculation

One way of determining the orbits of objects travelling in space is with the help of state vectors i.e. at least two position and velocity vectors of the body at a certain time interval as shown in earlier subsections 2.3.1 and 2.3.2. Another way to determine the orbital parameters of a body is by taking at least three right ascension and declination angle measurements. The former is known as the Lambert or Laplacian type orbit determination but as the velocity information is

obtained by interpolating positional measurements (via. radar, laser) it can lead to errors for longer tracking arcs. Whereas the latter is known as the Gaussian orbit determination and was first formulated by Karl Friedrich Gauss to find the location of Ceres [12]. The orbit determination of a body depends upon the available data and usually these two methods are combined to get the orbital parameters and also the possible transfer orbits between two points.

The path described by any moving object as a function of time is known as a trajectory. Every object moving in a potential field traces a path and it can be described by differential equations as done in section 2.1 with the two body approximation. As the initial values are known, the differential equation can be integrated over time to get the trajectory. The first part of this section deals with orbital determination assuming only the central mass. The second part deals with the the mass/body interaction depending upon the sphere of influence.

2.4.1 The Gauss Lambert Problem

As the name suggests, the *Gauss Lambert Problem* combines both the properties of the orbital determination types to generate a transfer orbit with a central body for e.g. a transfer from the Earth to Mars with the Sun as the central body. The basic orbit determination principle using this method is to take two position and velocity vectors at different times and construct a new conic usually an ellipse to connect these two points. The figure 2.9 illustrates the basics of this method.

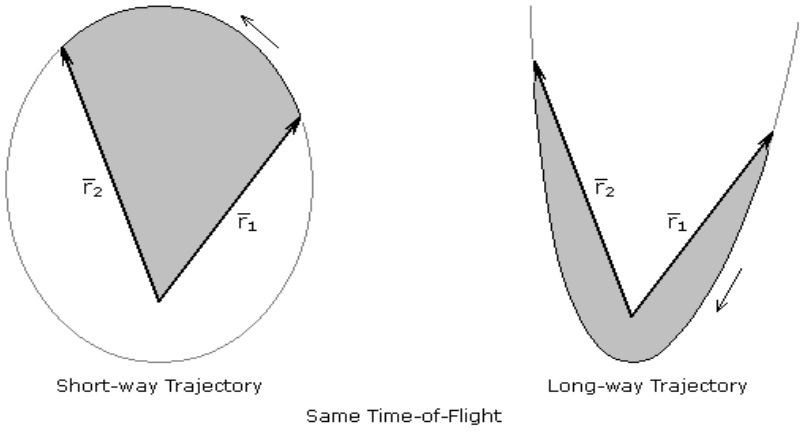


Figure 2.9: Figure illustrating the long and short way transfers in the Gauss Lambert Problem [10]

From the figure 2.9 it can be seen that for two position vectors in space there are two possible trajectories either with the total true anomaly change less than or greater than π. The transfer with the total true anomaly change less than π is known as Type I transfer and the one with $\nu > \pi$ is known as Type II transfer.

The two position vectors can be assumed to be the position vectors of the Earth and Mars with respect to the Sun. The vectors are generated from the MICE toolkit using the ephemeris

and the time of flight is user defined. Infinite number of orbits connecting these two points are possible however there are only two unique solutions for a certain time of flight, one for each type of transfer. From the figure it can also be observed that the transfer plane is uniquely defined however there is a pitfall for a true anomaly equaling π i.e. $\mathbf{r_1}$ and $\mathbf{r_2}$ are collinear and in opposite directions. In such a case the transfer orbit cannot be defined and a unique solution for the departure velocity V_1 and arrival velocity V_2 cannot be found. In case of transfers with true anomaly equalling 0 or 2π a unique solution is possible even if the orbit is a degenerate conic [10]

There are various algorithms to solve the Gauss Lambert problem such as the *Universal Variable Algorithm*, *Lambert Battin Method* and so on. The most stable of these is the Lambert Battin method as there are corrections to derive an unique solution even with a true anomaly change of 180^o but as it takes a lot of computation time and a lot of variables to implement it was decided that the original Gauss solution to the Lambert problem was sufficient for the master thesis. The original solution works by determining the eccentric and hyperbolic anomalies and can be solved using the f and g functions [35].

The f and g relations can be expressed as[2]:

$$\mathbf{r_2} = f\mathbf{r_1} + g\mathbf{v_1} \tag{2.52}$$

$$\mathbf{v_1} = \frac{\mathbf{r_2} - f\mathbf{r_1}}{g} \tag{2.53}$$

$$\mathbf{v_2} = \dot{f}\mathbf{r_1} + \dot{g}\mathbf{v_1} \tag{2.54}$$

$$\tag{2.55}$$

Where the f and g parameters are defined as:

$$f = 1 - \frac{r_2}{p}(1 - cos\,\Delta\nu) = 1 - \frac{a}{r_1}(1 - cos\,\Delta E) \quad p \rightarrow \textit{semi-latus rectum} \tag{2.56}$$

$$g = \frac{r_1\, r_2\, sin\Delta\nu}{\sqrt{GM\, p}} = t - \sqrt{\frac{a^3}{GM}}(\Delta E - sin\Delta E) \tag{2.57}$$

$$\dot{f} = \sqrt{\frac{GM}{p}} tan\frac{\Delta\nu}{2}\left(\frac{1 - cos\Delta\nu}{p} - \frac{1}{r_1} - \frac{1}{r_2}\right) = \frac{-\sqrt{GM\, a}}{r_1 \cdot r_2} sin\Delta E \tag{2.58}$$

$$\dot{g} = 1 - \frac{r_1}{p}(1 - cos\Delta\nu) = 1 - \frac{a}{r_2}(1 - cos\Delta E) \tag{2.59}$$

In the equations 2.56, 2.57 and 2.58 there are four known variables and three unknowns (p, a, ΔE). As there are three unknowns and three equations the variables can easily be determined but as the equations are transcendental in nature an iteration has to be done:

[2]The original algorithm has been adapted from Braeunig,R., [10] as it presents the solution in a very concise and simple way, so all the equations have been derived using this reference unless stated otherwise.

1. Take a trial value for p, a and ΔE from guesses or for e.g. assuming a transfer orbit similar to the one tangent burn case.

2. Then compute the two remaining unknown variables using equations 2.56 and 2.58

3. Using the values obtained above solve equation 2.57 for time.

4. If the values do not agree, then adjust the trial values and repeat the iteration.

There are various ways to get the correct answer for the trial value and one of the methods is to use the *p-iteration* technique in which the trial value for the parameter is guessed and then trial values for a and ΔE are calculated as inputs for the iteration. The steps for the *p-iteration* technique are:

1. Define three constants:

$$k = r_1\, r_2(1 - cos\Delta\nu) \qquad r_1 = |\mathbf{r_1}|\ \&\ r_2 = |\mathbf{r_2}| \tag{2.60}$$

$$l = r_1 + r_2 \tag{2.61}$$

$$m = r_1\, r_2(1 + cos\Delta\nu) \tag{2.62}$$

2. Take a trial value of p and determine a

$$a = \frac{m\, k\, p}{(2m - l^2)p^2 + 2\, k\, l\, p - k^2} \tag{2.63}$$

3. After the first two steps the semi-major axis has to be checked to see if the orbit is hyperbolic or elliptic using equations 2.56 through 2.58. If a is positive the eccentric anomaly ΔE can be obtained from equations 2.56 & 2.58. If the orbit is hyperbolic the ΔF i.e. corresponding anomaly in a hyperbolic orbit is obtained from equation 2.64

$$cosh\Delta F = 1 - \frac{r_1}{a}(1 - f) \tag{2.64}$$

4. The time of flight is then determined with the help of the following equations:

$$t = g + \sqrt{\frac{a^3}{GM}}(\Delta E - sin\Delta E) \qquad when,\ a > 0 \tag{2.65}$$

$$t = g + \sqrt{\frac{(-a)^3}{GM}}(sin\Delta F - \Delta F) \qquad when,\ a > 0 \tag{2.66}$$

5. The values of p obtained from these steps correspond to two parabolic orbit passing through the two position vectors $\mathbf{r_1}$ and $\mathbf{r_2}$. The two values of p can then be written as equations:

$$p_1 = \frac{k}{l + \sqrt{2\,m}} \tag{2.67}$$

$$p_2 = \frac{k}{l + \sqrt{2\,m}} \tag{2.68}$$

$$\tag{2.69}$$

The limits for the values of p are as follows:

- If $\nu < \pi$ the value of p lies between p_1 and infinity.
- If $\nu > \pi$ the value of p lies between zero and p_2.

Note: In the STO software the value of p is found within the limits p_1 and p_2.

6. Then iterations are done to find the optimum value of p. For the STO, the linear interpolation technique is used. Time of flights corresponding to p at n and $n-1$ are calculated and then a new value is obtained as shown in equation 2.70.

$$p_{n+1} = p_n + \frac{(t - t_n)(p_n - p_{n-1})}{t_n - t_{n-1})} \tag{2.70}$$

Note: In the STO two thousand iterations are done to find the required TOF.

7. After finding the required time of flight, equation 2.59 has to be used to find $\dot{g}$. Then the departure and arrival velocities corresponding to the points $\mathbf{r_1}$ and $\mathbf{r_2}$ can be found from equations 2.53 & 2.54.

The original Gauss-Lambert solution is easy to implement however it has two drawbacks. First, it fails when the two position vectors are collinear and in opposite directions and second, it uses two different equations depending upon the orbit type.

Once the Gauss problem has been solved the position and velocity vectors are obtained in the central body's equatorial frame e.g. heliocentric ecliptic orbit. The orbital elements of the body in orbit can then be calculated as follows [10]

1. Calculate the *specific angular momentum* $\mathbf{h}$ which is perpendicular to the orbital plane by taking the cross product of the position and velocity vector.

$$\mathbf{h} = \mathbf{r} \times \mathbf{v} \tag{2.71}$$

2. Then calculate the nodal vector $\mathbf{n}$ which is perpendicular to both the orbital plane and central body's equatorial plane since it is a cross product of the normal vectors of both planes it has to lie at their intersection i.e. the line of nodes. The situation can easily be understood from equation 2.72 and figure 2.10.

$$\mathbf{n} = \mathbf{z} \times \mathbf{h} \tag{2.72}$$

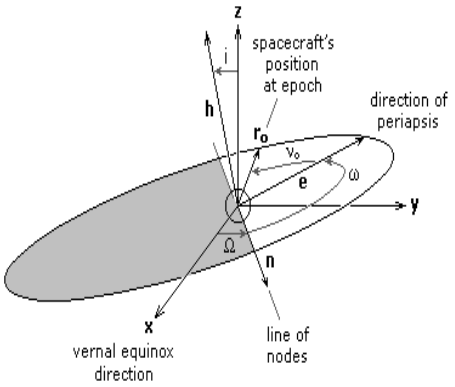


Figure 2.10: Figure illustrating the derivation of orbital parameters from the specific angular momentum and nodal vector [10]

3. The eccentricity vector is obtained from equation 2.73 and it points from the focus toward perihelion with a magnitude equal to the eccentricity ($|\mathbf{e}|$) of the orbit.

$$\mathbf{e} = \frac{1}{GM}\left[\left(v^2 - \frac{GM}{r}\right)\mathbf{r} - (\mathbf{r}\cdot\mathbf{v})\mathbf{v}\right] \tag{2.73}$$

4. The semi major axis is calculated using the equation 2.11.

5. The inclination i of the orbit is the angle between $\mathbf{z}$ and $\mathbf{h}$ and is found as:

$$cos\, i = \frac{h_z}{h} \tag{2.74}$$

6. The RAAN (Ω) is the angle between $\mathbf{x}$ and $\mathbf{n}$ and as it ranges from 0 to 2π the correct quadrant has to be assigned. The angle is obtained from equation 2.75 and if $n_y > 0$ then Ω is less than 180^o.

$$cos\, \Omega = \frac{n_x}{n} \tag{2.75}$$

7. The argument of periapsis is the angle between $\mathbf{n}$ and $\mathbf{e}$ and is found from equation 2.76 and if $e_z > 0$ then ω is less than 180^o.

$$cos\, \omega = \frac{\mathbf{n}\cdot\mathbf{e}}{n\, e} \tag{2.76}$$

8. Finally the true anomaly is obtained from equation 2.77 and is less than 180^o when $\mathbf{r}\cdot\mathbf{v} > 0$.

$$cos\, \nu_o = \frac{\mathbf{e}\cdot\mathbf{r}}{e\, r} \tag{2.77}$$

2.4.2 The Patched Conic Approximation

The Gauss Lambert solution is the first step in determining a trajectory around any central body and also for interplanetary transfers even though it gives a good estimate it doesn't correctly mimic the real world because it doesn't take into consideration various perturbations that arise from the geometry of the central body e.g. the J2 term due to the oblateness of the Earth and also the gravitational attraction of the non-central bodies. Without the use of advanced software such as the STK which have all the perturbation parameters included and are computationally intensive a very accurate solution cannot be achieved. Nonetheless the Gauss-Lambert solution can be improved by dividing the entire trajectory for e.g. in interplanetary transfers to three different zones depending upon the sphere of influence of the bodies in question. In an interplanetary transfer to Mars the departure and arrival phases can be approximated as hyperbolic and the transfer phase is elliptical.

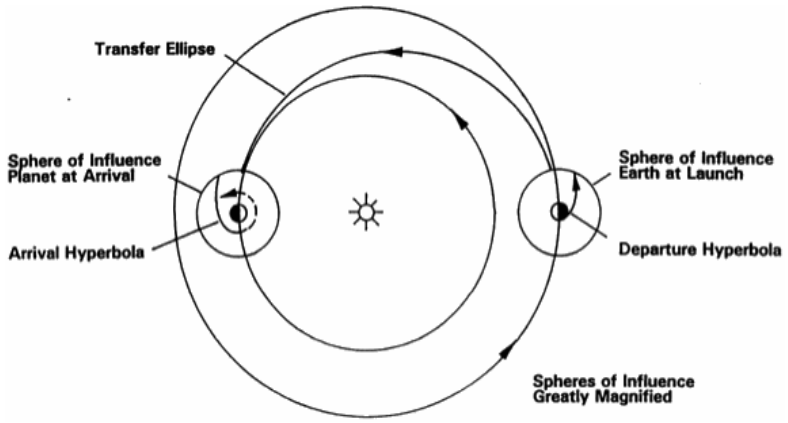


Figure 2.11: Figure illustrating the patched conic transfer [13]

As the elliptical orbit has already been introduced in 2.2.1 only the hyperbolic arrival and departure is covered in this section.

2.4.2.1 The Hyperbolic Departure

Any spacecraft designed for interplanetary missions has to overcome the escape velocity of the Earth and once this is achieved the S/C is no longer in a closed conic but either in a parabolic or a hyperbolic path compared to the initial central body for e.g. the Earth. The V_{dep} values obtained in the Gauss problem are relative to the central body for e.g. the Sun in case of Earth-Mars transfer so a few extra calculations need to be done to determine the actual departure and arrival orbital parameters and velocities. Earth-Mars transfer using the patched conic approximation is used here to illustrate the concept.

The departure velocity V_{dep} is obtained from solving the gauss problem. The first step to finding the hyperbolic velocity at departure is to calculate the relative velocity of the S/C with respect to the planet using equation 2.78

$$V_{S/P} = |\mathbf{V_{dep}} - \mathbf{V_{planet}}| = \sqrt{V_{S/Px}^2 + V_{S/Py}^2 + V_{S/Pz}^2} \tag{2.78}$$

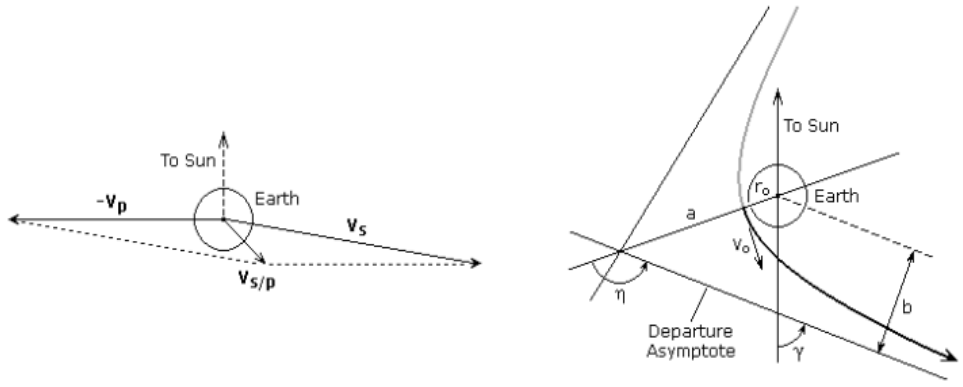


Figure 2.12: Figure illustrating the hyperbolic departure [10]

If the S/C only has a velocity equal to the escape velocity, it leaves the central body in a parabolic path but if the change in velocity is greater than the escape velocity, the S/C leaves in a hyperbolic path, as there is still a velocity component greater than zero at infinity. If the manoeuvre is impulsive and the velocity at burnout is V_{bo} the velocity at infinite is 2.79:

$$V_{inf}^2 = V_{bo}^2 - V_{esc}^2 \tag{2.79}$$

Even though the term infinite velocity is used it should be understood as the instantaneous velocity when the object escapes the sphere of influence of the central body. Assuming $V_{S/P} \approx V_{inf}$ the injection velocity from the surface of the Earth is found from equation 2.80 and if the launch takes place from a parking orbit with a perigee altitude r_p the required ΔV is found from equation 2.81.

$$V_{injection} = \sqrt{v_{inf}^2 + \frac{2GM}{r}} \quad r \rightarrow \textit{planet radius} \tag{2.80}$$

$$\Delta V = V_{injection} - \sqrt{\frac{GM}{r_p}} \tag{2.81}$$

A small error in the injection velocity can have a very big difference in the hyperbolic excess velocity for e.g. for an interplanetary Hohmann transfer to Mars one percent error in injection velocity results in 15% error in hyperbolic excess velocity [10]. In addition the launch also has to take place at a certain zenith angle for the departure asymptote to be parallel to the hyperbolic excess velocity as illustrated in figure 2.12. The zenith angle is obtained from the dot product of $\mathbf{r}$ and $V_{S/P}$ as 2.82:

$$\gamma = arccos\left(\frac{r_x\, v_x + r_y\, v_y + r_z\, v_z}{r\, v}\right) \tag{2.82}$$

The process to determine the orbital properties of a hyperbolic orbit has already been described in sub-section 2.2.2.

2.4.2.2 The Hyperbolic arrival

As the name already suggests the arrival of a S/C with the velocity solutions derived from the Gauss problem is always hyperbolic relative to the destination. The Gauss problem is solved using the position vector of the planet after a certain time of flight so the entry is always an impact at the center. So if the mission requires the object to arrive at a miss distance d the miss distance has to be added or subtracted to the destination's position vector [10]. The process of avoiding collision with the central body is known as B-plane targeting. The B-plane can be thought of as a planar co-ordinate system that is perpendicular to the trajectory plane and has the destination central body as the focus of the arrival hyperbola. The figure 2.13 describes the arrival scenario with a miss distance

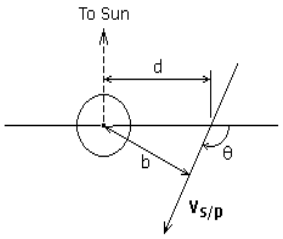


Figure 2.13: Figure illustrating the hyperbolic arrival in a patched conic transfer [10]

and assuming the arrival $V_{inf} = V_{arr} - V_{destination}$ the following sets of equations describe the process to calculate the orbital parameters [10].

$$d_x = \frac{-d\, r_y}{\sqrt{r_x^2 + r_y^2}} \quad miss\ distance\ x-component \tag{2.83}$$

$$d_y = \frac{-d\, r_x}{\sqrt{r_x^2 + r_y^2}} \quad miss\ distance\ y-component \tag{2.84}$$

$$\theta = arccos\left(\frac{d_x\, V_{arr_x} + d_y\, V_{arr_y}}{d\, V_{arr}}\right) \quad arrival\ angle \tag{2.85}$$

$$a = -\frac{-GM}{V_{inf}^2} \quad semi\text{-}major\ axis \tag{2.86}$$

$$e = \sqrt{1 + \frac{b^2}{a^2}} \quad eccentricity \tag{2.87}$$

The the other orbital parameters can be derived as described in subsection 2.4.1.

2.5 MATLAB Optimization Toolbox and Solvers

The optimization toolbox is a feature within MATLAB which contains various solvers and optimization algorithms that enable finding the maximum or minimum of problems and also helps

to fit models to data [31]. The algorithms solve constrained and unconstrained continuous and discrete problems and the functions can be linear, non-linear, quadratic, multi-variable and so on depending upon the version [33]. The toolbox is used in the STO software to minimize either the flight duration, or the velocity change or the change in mass of the S/C. The optimization function used is $FMINCON$ and the solver that was used is the the *active-set*. The final data of the trajectory was calculated using the $ODE45$ solver but $ODE113$ was also analysed to check for simulation speed and accuracy. The toolbox offers many features and solvers and more detailed information can be found in the MATLAB webpage [32], so only relevant information has been presented in this report.

2.5.1 The FMINCON function

The FMINCON finds the minimum of problems specified as:

$$min\ f(x) such\ that = \begin{cases} c(x) \leq 0 \\ ceq(x) = 0 \\ A \cdot x \leq b \\ Aeq \cdot x = beq \\ lb \leq x \leq ub, \end{cases}$$

Where $f(x)$ is a function that returns a scalar, b and beq are vectors, A and Aeq matrices, $c(x)$ and $ceq(x)$ are functions that return vectors and $f(x)$, $c(x)$ and $ceq(x)$ can be non-linear functions. The variables x, lb and ub are defined as either vectors or matrices.

The function starts with an initial value and attempts to find a constrained minimum of a scalar function with several variables in other words also known as *constrained non-linear optimization* [30]. The general syntax for minimizing a problem using FMINCON is:

$$[x, fval, exitflag] = fmincon(fun, x0, A, b, Aeq, beq, lb, ub, nonlcon, options) \tag{2.88}$$

Where the variables mean the following [38]

- **fun:** the objective function to be minimized
- **x0:** the initial starting point
- **A,b:** if mentioned the minimum of x within the linear inequality $A \times x \leq b$ is found.
- **Aeq,beq:** the function fun is minimized to within the linear inequalities $Aeq \times x = beq$ as well as $A \times x \leq b$, if there are no inequalities then A and b should be defined as empty matrices
- **lb,ub:** the lower and upper bound on the design variables, such that x always lies between $lb \leq x \leq ub$, if there are no inequalities $Aeq = [\,]$ and $beq = [\,]$
- **nonlcon:** the minimization is subjected to non-linear inequalities $c(x)$ or equalities $ceq(x)$ which are defined in $nonlcon$, the optimization is done such that $c(x) \leq 0$ and $ceq(x) = 0$, if there are no bounds lb and ub should be set as empty matrices

- **options:** the definition of optimization parameters which are defined using *optimset*
- **fval:** returns the value of the objective function at the solution x
- **exitflag:** returns a value that describes the exit condition of the optimization

The detailed description of the FMINCON function can be found in the website from MATLAB [30] and [38]. The medium scale *active-set algorithm* is used in the STO. The principle behind $active-set$ algorithms states that *if a minimizer on each working surface is found during each iteration within the defined active-set region and there is a decrease in the value of f(x) at each iteration the algorithm terminates after finitely many iterations* [15]. So if there is a solution that satisfies the constraints within the given feasible region then the algorithm terminates. The active-set was used as it requires the least computation time. A detailed description of all the available algorithms in MATLAB and also their properties can be found in [27] and [15]. The only problem however with active-set is that if the initial supplied values are very far away from the optimum values the computation can take a long time and sometimes even fail if the answer returned is infinity. In the STO, as a Gauss problem is calculated the initial points are very close to the optimum values, so it is a good algorithm to use.

2.5.2 ODE Integrators

The equations of motion in a Keplerian orbit are subject to perturbations from different sources such as gravitational attraction from different bodies, oblateness of the central body and so on. Therefore all the perturbations have to be taken into account to correctly predict the trajectory of a body. One simple way to do this is to express the equations of motion of the object including all the perturbations and then integrate them, also known as Cowell's formulation. In this method all the perturbations can be linearly added assuming a two body problem [1] and the state vector update in this method is expressed in equations 2.89, 2.90 and 2.91.

The equation accounting perturbation, for a two body problem is expressed as:

$$\ddot{\mathbf{r}} = -\frac{\mu}{r^3}\mathbf{r} + \mathbf{a}_{perturbed} \tag{2.89}$$

Where $\mathbf{r}$ is the position vector, μ is the standard gravitational parameter and $\mathbf{a}_{perturbed}$ is the acceleration produced from the perturbing forces. As the equation 2.89 is second order it is converted into first order equations as follows, if $\mathbf{X}$ is taken as the state vector.

$$\mathbf{X} = \begin{pmatrix} \mathbf{r} \\ \mathbf{v} \end{pmatrix} \tag{2.90}$$

$$\dot{\mathbf{X}} = \begin{pmatrix} \mathbf{v} \\ -\frac{\mu}{r^3}\mathbf{r} + \mathbf{a}_{perturbation} \end{pmatrix} \tag{2.91}$$

The final equation 2.91 can then be integrated using the inbuilt integrators in MATLAB. The most common integrator for trajectory problems is the ODE45 as it is both fast and has a high accuracy. The integration of the trajectory was also done with ODE113 because after ODE45, it is the only inbuilt non-stiff solver reliable enough even though it is computationally intensive.

2.5.2.1 ODE45

The Runge Kutta method is one of the most widely used integration methods. The ODE45 solver inbuilt in MATLAB is an explicit fourth order Runge-Kutta method [29]. As it requires the function value at just the preceding time step it is a one step solver. The mathematics behind the fourth order Runge Kutta methods is mentioned below. If the differential equation is expressed as:

$$\dot{y} = f(t, y), \ \ with \ \ y(t_0) = y_0 \tag{2.92}$$

The algorithm for solving the 4^{th} order RKM for a step size $h > 0$ is as follows:

$$y_{n+1} = y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \tag{2.93}$$

$$t_{n+1} = t_n + h \tag{2.94}$$

Where the updated position is given by 2.93 and the time update is provided by 2.94. The $k_{\#}$ parameters are increments whose weighted average is taken in the final update of the position. The different k's have different meanings and are described below.

$$k_1 = hf(t_n, y_n) \tag{2.95}$$

$$k_2 = hf(t_n + \frac{1}{2}h, y_n + \frac{1}{2}k_1) \tag{2.96}$$

$$k_3 = hf(t_n + \frac{1}{2}h, y_n + \frac{1}{2}k_2) \tag{2.97}$$

$$k_4 = hf(t_n + h, y_n + k_3) \tag{2.98}$$

The first increment is done by first taking the slope at the beginning of the interval with y_n in a manner similar to the Euler method. The second one is incremented by taking a slope at the midpoint of the points y_n and $y_n + \frac{1}{2}k_1$, the third increment uses the slope of the midpoint of the interval with $\frac{1}{2}k_2$ and the final increment is done using the end of the interval $y_n + k_3$. As this is a weighted average and as the interval is divided into four sections the ODE45 is quite robust and accurate. The 4^{th} order RKM has a final error of $O(h^5)$. [36]. More information regarding the solver ODE45 and the implementation process can be found in the Matlab webpage [29].

2.5.2.2 ODE113

ODE113 is a non-stiff multi-step solver. It is based on the Adams Moulton Predictor corrector method and it normally needs solutions at many preceding time points to compute the current solution, so even though it is more efficient than ODE45 at stringent error tolerances it is slower and expensive to evaluate [28].

The total number of function evaluations in the single step methods can be reduced by storing the values from the previous steps. Thus integration using the stored values gives rise to the multi-step methods. Multistep methods are more efficient for differential equations defined by complicated functions and a lot of arithmetic operations. So, assuming that the approximate

values η_j of the solution $y(t_n)$ at equidistant times with time step h is known the differential equation 2.92 can be integrated to obtain the expression:

$$\mathbf{y}(t_{n+1}) = \mathbf{y}(t_n) + \int_{t_n}^{t_n+h} \mathbf{f}(t, \mathbf{y}(t))dt \tag{2.99}$$

However the integral cannot be evaluated because it depends upon the unknown solution $\mathbf{y}(t)$ of the differential equation so the problem is circumvented by introducing the integrand by a polynomial $\mathbf{p}(t)$ which interpolates some values to the previous time that are already known according to the initial assumption leading to equation 2.100:

$$\eta_{n+1} = \eta_n + \int_{t_n}^{t_n+h} \mathbf{p}(t)dt \tag{2.100}$$

The increment function in multistep solvers is then defined as equation 2.101:

$$\phi = \frac{1}{h} \int_{t_n}^{t_n+h} \mathbf{p}(t)dt \tag{2.101}$$

The mathematical derivation of the ODE113 is quite extensive and combines other methods such as the Adams Bashforth, so only the algorithm to calculate the final result is presented in the report. A detailed derivation can be found in [22] and the general syntax and manual can be found in [28]. The algorithm to calculate the final function value using the ODE113 [22]

1. Predictor step, an initial estimated solution is generated at time t_{n+1} using the Adams Bashforth formula:
$$\eta_{n+1}^p = \eta_n + h\phi_{AB} \tag{2.102}$$

2. Evaluation step, the corresponding function value is found:
$$\mathbf{f}_{n+1}^p = \mathbf{f}(t_{n+1}, \eta_{n+1}^p) \tag{2.103}$$

3. Corrector step, the Adams Moulton formula is used to find a better estimate.
$$\eta_{n+1} = \eta_n + h\phi_{AM}(\mathbf{f}_{n+1}^p) \tag{2.104}$$

4. Final evaluation step, the function value is updated to the new time step.
$$\mathbf{f}_{n+1} = \mathbf{f}(t_{n+1}, \eta_{n+1}) \tag{2.105}$$

Chapter 3

Software Design and Layout

The STO software is built on MATLAB platform and works on MATLAB versions 2007 and later. The software makes use of the optimization toolbox present within MATLAB to find the most efficient flight steps for a specified manoeuvre. The parameters that are optimized in the software are the time of flight, change in velocity or the change in mass, depending upon the preference of the user. The constraints are the orbital elements and the range can be specified while setting up the simulation scenario. This chapter describes the various features added to the STO GUI in detail. The ephemeris for the software were generated using MICE, the MATLAB version of the NASA SPICE toolkit. The old STO software had the capability to setup a simulation scenario i.e. add celestial bodies & flight-steps and also optimize elliptical and circular trajectories within the sphere of influence of the Earth. An elliptical transfer to the Moon using the Lunatool was also possible. After the conclusion of the thesis the STO software has these capabilities which were non-existent in the earlier version. **Note:** The scenario setup and general layout of the STO was however kept the same as no changes were required.

- Transfers within any two celestial bodies within the Solar system is possible whereas before only the transfer to the Moon was possible.
- STO now supports non-Hohmann transfers whereas the previous version only worked for elliptical Hohmann transfers.
- Launch windows can be calculated for any given date with the *get_positions.m* function within the Gauss Lambert Optimizer function.
- Porkchop plots for interplanetary and Lunar transfers can be generated.
- SPICE database has been updated to include Martian and Jovian moons.
- Optimizer also supports hyperbolic transfers and uses Patched conic approximation whereas the previous version only worked with orbits with eccentricity less than one and only within the sphere of influence of the Earth.
- The representation of S/C in Earth equatorial reference frame and the celestial bodies in ecliptic frame is possible. Previously all the bodies were only represented in the ecliptic frame and the axial tilt of the Earth was ignored.

The description of the various functions is however given in the Appendix B (6).

3.1 NASA Spice Toolkit

The NAIF under NASA's planetary science division has built an information system named "SPICE" which is a complete collection of ephemeris information for not only planets but also for space borne payload. SPICE aids in planning, interpreting, modeling scientific observations and also planned missions. The SPICE data sets are known as *kernels* and they contain navigation and other ancillary information for most bodies in the Solar system. The SPICE data sets are classified according to the information they contain and can easily be understood as follows [6]

- **S:** Ephemeris of the spacecrafts and represented as a function of time (SPK).
- **P:** The emphemeris of planets, natural satellites, asteroids or the location of any body as a function of time is also represented in the **SPK**. However another part of the **P** ephemeris also includes the physical, dynamical and cartographic constants of target bodies such as size/shape specification, orientation of the spin axis and prime meridian and is available as extension (PCK).
- **I:** The kernel containing the description of the instrument's parameters such as the field of view size, shape and orientation. Available under (IK).
- **C:** The time tagged pointing or the orientation angles for a spacecraft bus or a spacecraft structure over which a payload is present. The **C** pointing kernel may also include the angular rate data for that structure. Available as: (CK).
- **E:** The either planned events or unanticipated events that occur during a mission lifetime are recorded under the events kernel **E**. The three components of the (EK) are the science plans, sequences and notes.

In addition to the aforementioned features of the SPICE data set, there are also other features such as the frame kernel **(FK)**, spacecraft clock **(SCLK)**, leap seconds **(LSK)** and digital shape model kernel **(DSK)** however as they were never used for the required task of the thesis, they have not been mentioned here. If a complete description of the SPICE data sets is required the reader can find all the documentation under the NASA's *The SPICE concept* webpage.[1]

The SPICE data set includes a SPICE toolkit that is a library of portable subroutines which read the kernel files and return the parameters such as the range, the position and velocity vectors of the bodies of interest and so on. The original SPICE toolkit was written in ANSI FORTRAN 77, however it has been translated to other languages and is now available in C, IDL and MATLAB. The MATLAB version of the spice toolkit was integrated into the STO to generate the ephemeris of all the bodies. The SPICE toolkit was selected for this purpose because of the following reasons [4]:

[1] http://naif.jpl.nasa.gov/naif/spiceconcept.html

- Works on different platforms if it is compiled in one.
- Updated tool-kits are released about every 10-15 months.
- The code is tested by NAIF before it is released to users.
- Backward compatibility of all the new toolkits.
- Well documented and free source code.
- In built exception handling for most of the invalid inputs.
- Computations are done in double precision numbers.
- Kernels can be separated so only the required ones can be loaded and used.
- New kernels can easily be added without compatibility problems.

A flow chart explaining the operation principle of the NAIF Spice toolkit is presented as figure 3.1.

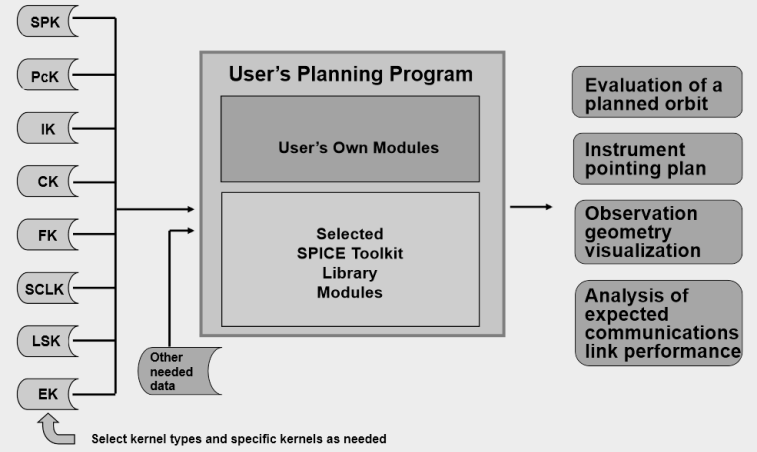


Figure 3.1: A flowchart explaining the operation principle of the NAIF SPICE toolkit [4]

The SPICE kernel for MATLAB is known as MICE and after integrating new kernels for Martian and Jovian moons the final celestial body can be accessed through the main STO GUI and the bodies can be selected from the drop downlist.

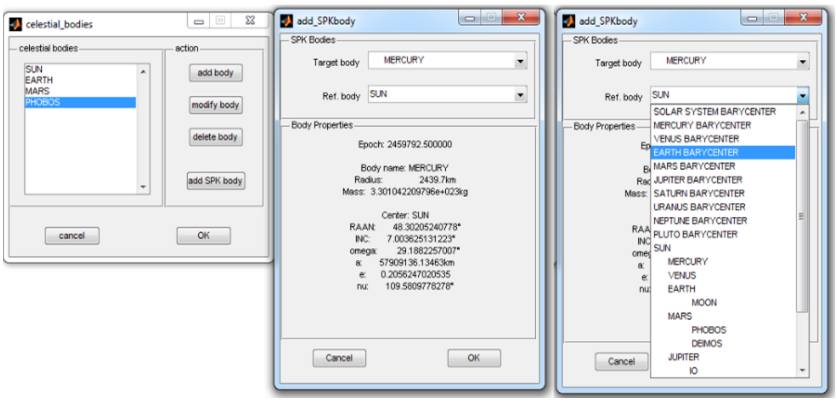


Figure 3.2: Figure showing the steps of adding celestial bodies to the simulation [4]

3.2 Spacecraft and Flight Step Definition

After adding the required celestial bodies as shown in figure 3.2 a number of spacecrafts can be added to the software and the initial orbit and the desired final orbit can also be entered. These input parameters are taken by the STO software for initializing the Gauss problem [2.4.1] as well as the required final conditions in the optimizer. The initial and final orbital conditions consist of user defined apogee, perigee, RAAN, the argument of perigee, inclination and the true anomaly whereas the input for the flight steps consists of defining the duration, the central body, the in-plane/out of plane angle, the specific impulse and the thrust of the available propulsion system. The flight steps also support multi-stage rockets, so a jettison mass for each flight step can also be specified. The flight steps can then be optimized into three parameters: change in velocity, change in time of flight or change in mass. The detailed description of the software features is given in Appendix B (6). Nevertheless two figures 3.3 and 3.4 describing the S/C initial condition definition and flight steps with different parameters have been presented to give a quick overview of the STO software.

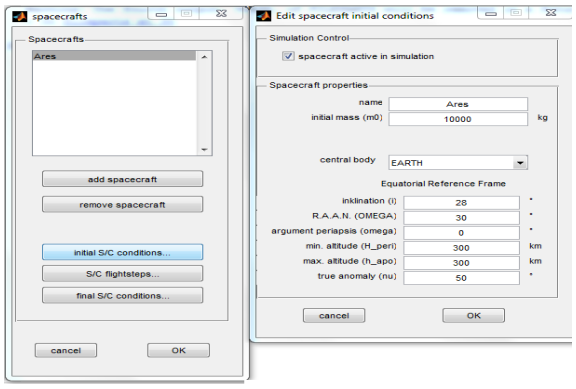


Figure 3.3: STO initial condition definition GUI window

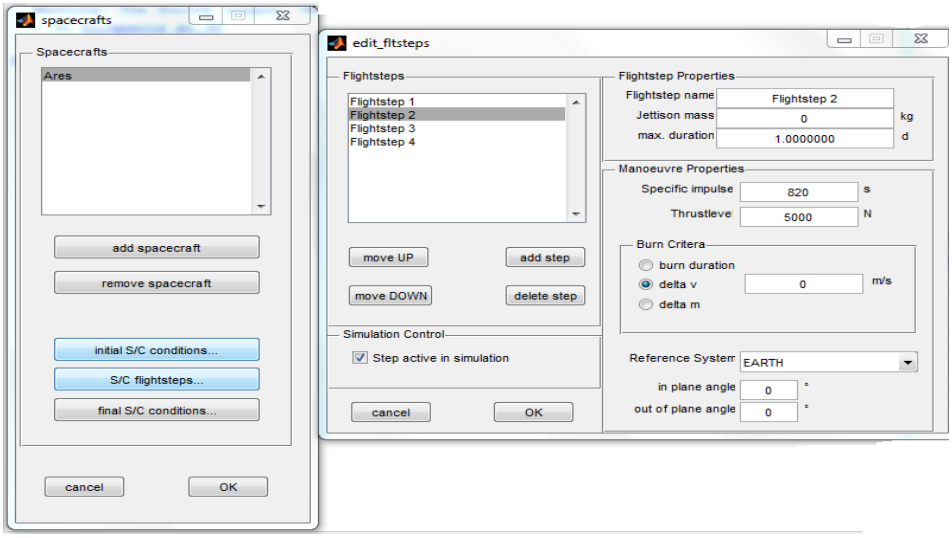


Figure 3.4: STO initial flight step definition GUI window

3.3 Software Features

The STO software was edited over the course of the master thesis and a few features were added. The main features currently included in the software that are capable of optimizing the trajectories and also calculating the orbital parameters as well as departure and arrival conditions have been mentioned in this section. Other changes like adding the equatorial view for the Earth, updates of the SPICE kernels to include the Martian and Jovian moons and other changes have not been explained as they are not the most important additions.

3.3.1 Luna Tool

The *Luna Tool* is an optimization routine included in the STO software and it was already existing at the beginning of the master thesis. As the name suggests the tool is only specialized to calculate trajectories from the Earth to the Moon and also the trajectory can be modelled to return to Earth however no direct launch from the Moon is possible but rather a swing by manoeuvre at the Moon. The tool only calculates the launch angle (close to zero degrees) and the TOF is calculated using the Hohmann manoeuvre. The initial conditions are setup using the steps described in figures 3.2, 3.3 and 3.4. Once the scenario with four flight steps has been setup with the Earth as the center for the first three flight steps and the Moon as the final the Luna Tool can be selected from the drop down menu and the software starts optimizing the trajectory and also the selected optimization parameter. Once the optimization is complete a trajectory leading to the Moon is plotted in the GUI and a sample is presented in figure 3.5.

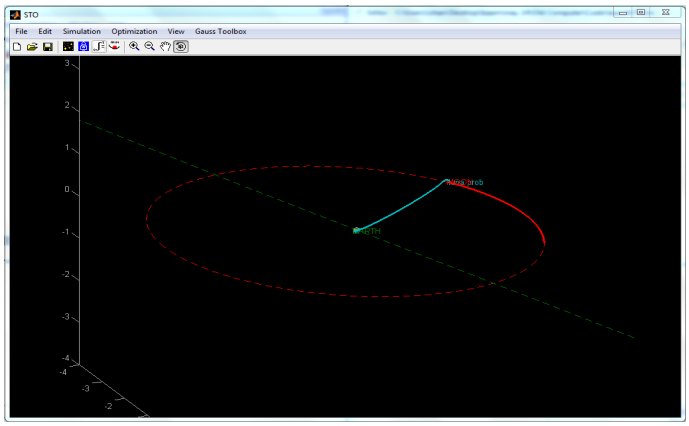


Figure 3.5: Figure showing the optimized trajectory from the Earth to the Moon using the Luna Tool optimization

3.3.2 Gauss Lambert Solver

The Gauss Lambert Solver was developed in addition to the STO software to make easy quick guesses to the launch times where the required change in velocity is the minimum. The program uses the Gauss Lambert algorithm mentioned in section 2.4 and then uses the contour generating code developed by Eagle, D. [14] which was freely available in the Matlab online repository. As this additional software is used as the first estimate for the flight conditions, the JPL ephemeris (MICE data) was used as it is i.e. the planets are referred to as numbers rather than with their names for e.g. Earth is defined as 3, Mars as 4. All the planets including Pluto are numbered from 1 until 9, the Moon is numbered at 10, the Sun as 11 and the Solar System Barycentre is numbered as 12.

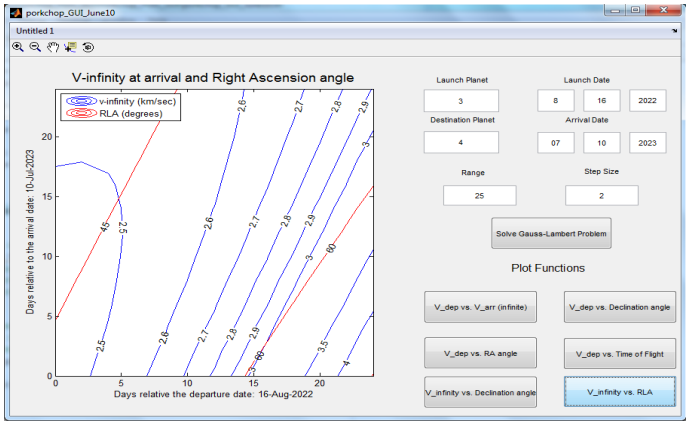


Figure 3.6: Figure illustrating the layout of the Gauss Lambert Solver

Even though the software only supports these bodies right now, new ephemeris files can be added to give porkchop plots for new scenarios for e.g. if the central body is switched to Mars, the porkchop plot for transfers from Phobos to Deimos should also be possible albeit it requires

a small code change. The GUI of the software is presented in figure 3.6 with the departure and arrival velocities for an interplanetary mission to the Mars.

3.3.3 Gauss Lambert Optimizer

The Gauss Lambert optimizer can be used to optimize any trajectory within the solar system. As this optimizer works on the patched conics, a special attention has to be paid while defining the scenario. For e.g. for simulating an interplanetary orbit from the Earth to Mars the Sun should be described with respect to the Solar System Barycenter, whereas the Earth and Mars with respect to the Sun. The central body should be described first then the launch planet and finally the destination. However if Mars is described before the Earth in this scenario the escape and infinite departure velocities are calculated with respect to Mars. After defining the bodies the required simulation date should be selected and the GLO should be run. After running the GLO a small window appears asking the user for search range for days, tolerance angle and the desired time of flight and once these values are entered the software tries to find the initial launch conditions.

If no error is made in defining the sequence of launch and arrival bodies the optimizer first integrates the positions of all the bodies to the end of user defined time of flight. Then the position vector of the launch planet at the beginning of TOF and the position vector of the destination planet at the end of TOF is used as an input for solving the Gauss Lambert problem and the infinite departure velocity, right ascension and declination of the departure hyperbola are found. After this the first flight step is activated and a launch window in which a flight path angle equalling the reference RA is sought.

The RA in GLO is calculated from the vernal equinox and the FPA is calculated counter-clockwise from the x-axis. As all the celestial bodies are represented in Ecliptic J2000 co-ordinate system and the S/C is also represented in the same co-ordinate system the assumption of equating the RA and the flight path angle is valid. One point to note however is that for S/C originating at the Earth, the Earth Mean Equator with epoch of J2000 is used, so even though the x-axis coincides with the vernal equinox, the z-axis is tilted to the ecliptic therefore, the RA is the same however the declination angle has to be added to the axial tilt of the Earth.

Once the launch windows within the requested range of days has been calculated another window appears with the list of possible departure dates, time and angles within the defined tolerance. The user can select one of these options and this defines the total time of the coastal phase and initial departure conditions of the S/C. Then the GLO runs through the four flight steps that are *hard coded* to return the requested optimization parameter i.e. ΔV, Δt or Δm for each of the flight steps. In the second flight step the required ΔV calculated relative to the launch planet to achieve the hyperbolic trajectory is included into the ODE45 integrator and the orbit is propagated to the desired destination. The second step aims at hitting the target with the full flight time.

After finding an optimized solution in the second flight step, the third step begins to divide

the trajectory into two sections i.e. patched conics. The TOF for the hyperbolic departure and elliptical transfer to the destination is calculated using the velocity of the S/C at the beginning of transfer i.e. departure planet's sphere of influence radius is divided by the departure velocity to find the TOF for the hyperbolic phase and the time remaining from the total TOF minus the coastal phase and the hyperbolic departure phase is used for the elliptical transfer phase. In the third flight step a trajectory correction manoeuvre which requires a very small ΔV is also included. Once the S/C arrives at the sphere of influence of the destination the final deceleration manoeuvre is initiated. In the fourth and final flight step the hyperbolic arrival velocity is optimized (decelerated) to put the S/C in the desired orbit, predefined in the final conditions. If the final orbit is not properly optimized an additional flight step can be added to correct the orbit. Once the solution is optimized the result is also plotted in the STO GUI. Since the case study deals with the *Gauss Lambert Optimizer* a detailed description of the mission cases and the obtained solutions is presented in the case study section 4. The workings of the GLO can be observed in figure 3.7.

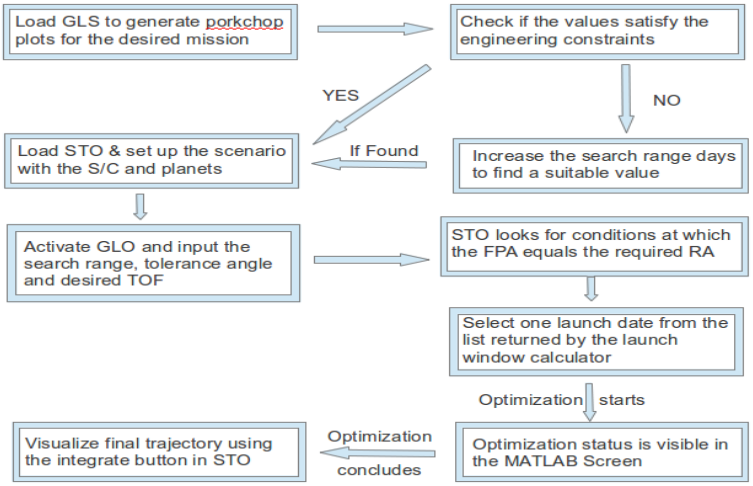


Figure 3.7: Figure illustrating the working principle of the GLO

3.4 Trajectory Optimization Concept and Constraints

The first step in running the optimization tool is to first find the ideal launch conditions for a certain time of flight using the Gauss Lambert solver [3.3.2] and then set up the scenario in the STO software with these launch conditions i.e. the orbital parameters, the engine specifications i.e. the specific impulse, thrust, change in velocity and the desired number of flight steps. Then the optimization can be done in two ways. The first being the *hard coded* optimizers such as the *Luna tool* and the *Gauss Lambert Optimizer* which use four flight steps. These predefined optimization tools require at least four flight steps so they might not always be used for e.g. if only one flight step optimization is required. The STO tool also incorporates this feature with a separate window to define the optimization parameters and also the upper and lower bound for the variables. A typical parameter definition window where individual flight steps can be

edited and also different optimization parameters can be selected is presented in figure 3.8. This feature is mostly useful for simulating the transfers to the Geo-stationary transfer orbits and so on.

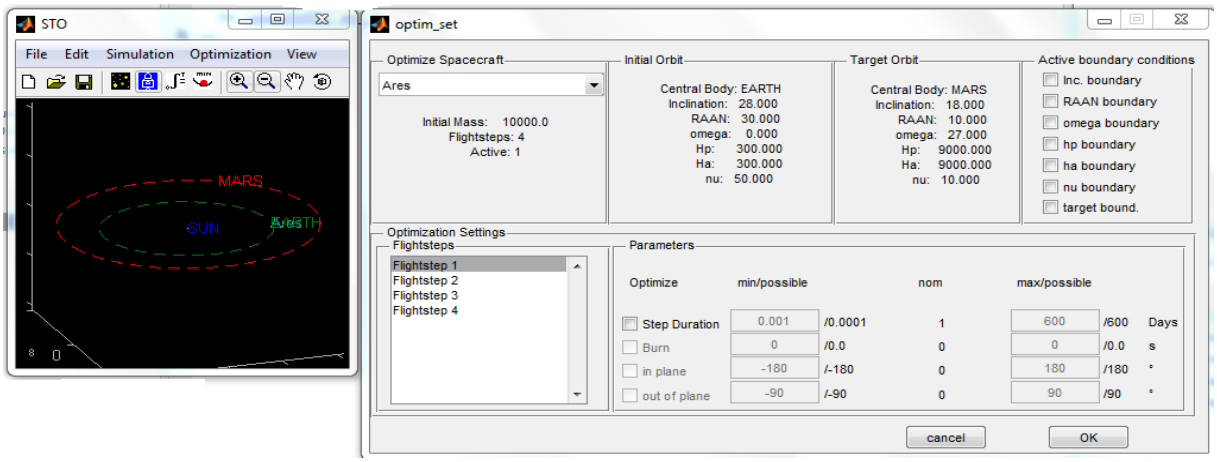


Figure 3.8: Figure showing the *non-hard coded* optimization parameter specification window where individual flight steps can be customized

Even though an example for the interplanetary mission to Mars has been presented here the customization of the parameters and bounds for all the defined flight steps is valid for any configuration.

Chapter 4

Case Study: Mars Sample Return Mission

Since the beginning of the space age Mars has been the clear favourite for interplanetary exploration. The landscape is dotted with structures that are similar to gullies and deltas found on Earth which suggests the possible presence of water on Mars in not so distant past. The presence of water could also suggest the presence of alien life and clues to how it disappeared from the surface of the planet. The Viking missions and the recent missions such as Opportunity and Curiosity attest to the continuing interest in Mars. Even though a direct sample return from Mars would help in analysis of the geo-chemistry, the presence of water and also the history of Mars, the gravitational attraction of Mars poses a huge restriction on the required launch capabilities of a direct sample return mission. Therefore for the proof of concept, missions to Martian moons are planned as a less energetically expensive way of returning the samples and understanding Mars. Such a mission would enable in understanding not only Mars but also the moons that exhibit peculiar characteristics.

Phobos and Deimos are two Martian moons and a sample from them is interesting due to the following hypothesis as even their origin is not clearly known [20]:

- Captured Asteroids → Primitive
- Mars Accretion remnants → Primitive
- Mars Impact Ejecta → Evolved
- Reaccreted Captured Asterioids → Primitive + Evolved
- Reaccreted Mars Accretion Remnants → Primitive + Evolved
- Reaccreted Mars Impact Ejecta → Highly evolved

And in addition to these hypotheses there is also no knowledge on, if these two moons are actually related to each other since they exhibit different properties. Studying these two or either one of these moons can help us in understanding their nature and origin, clues to the early solar

system formation processes, information about Mars as they hold the dust and meteoric activity on Mars, evolution of Mars and also possible resources such as water, Martian regolith and so on. The moons could also hold clues to the primitive body formation and evolution of our solar system as the planets were forming from proto-planets.

These two moons exhibit different spectral characteristics at the old and newly exposed areas. The old areas which are almost global exhibit red spectra that match D-type asteroids [21]. D-type asteroids are primarily made up of primitive material and did not experience any extensive heating [16] so they could have organic rich silicates, carbon and also ice, which could resemble the dirty/icy satellites orbiting Jupiter and Saturn [17]. The newly exposed areas in these moons give off blue spectra which could suggest dehydrated carbonaceous chondrites that is consistent with anhydrous primitive asteroid of non-ferric type [20]. The newly exposed surfaces also exhibit some features of the C-type asteroids that are carbonaceous chondrites with significant amount of volatiles such as water and carbon minerals that form at low temperatures [17]. During remote sensing operations not much diffusion has been seen from the asteroids [21] so unless a sample is returned from these moons the mystery behind their origin, their composition and the clue about their formation and Mars cannot be found as the spectral analysis has reached an impasse.

These moons are also interesting because a sample return from them could be as good as one from Mars, since they contain ejecta from Mars and could be done in a more energy efficient way compared to a sample collection at the surface of Mars. As Phobos is the one closest to Mars and holds more dust ejected from Mars it has been preferred over Deimos for the case study. Phobos was also preferred as various space agencies are also developing a sample return mission for e.g. ESA's *Phootprint mission* that aims to return soil and rock sample from Phobos in 2024. The conditions described by ESA have been taken and the results obtained from the STO software and ESOC have been compared to see the deviation of values obtained from the in-house software. The Orbital characteristics of Phobos obtained from the NASA webpage [5] are presented in table 4.1 and the image of Phobos with the *blue* and *reddish* hue is presented in figure 4.1.

Table 4.1: A table containing the Orbital Characteristics of Phobos [5]

Periapsis	9,234 km
Apoapsis	9,518 km
Orbital velocity	7,696.7 km/hr
Eccentricity	0.0151
Inclination	1.075
Escape velocity	41 km/hr

ESA's Phootprint mission is the continuation of Mars exploration programs after the Exo-

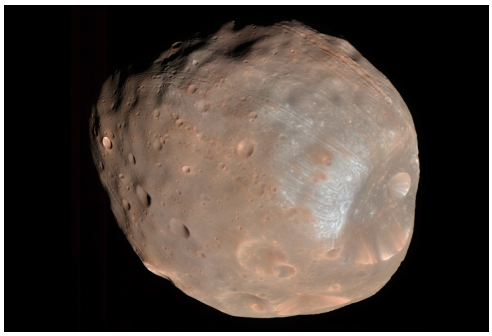


Figure 4.1: Figure illustrating the blue and reddish hue seen on Martian moon Phobos [3]

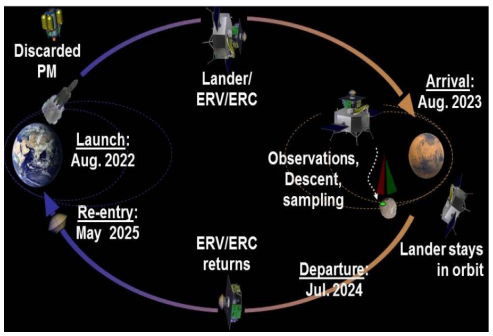


Figure 4.2: Figure illustrating the overall sample return mission from Mars [7]

mars missions under the Mars Robotic Exploration Program . ESA aims to launch the *Phootprint* mission in 2022 as the planetary alignment offers the minimum energy transfer. The *Phootprint* mission is centered on a Earth re-entry sample return with a sample capture and processing. A slide illustrating the *Phootprint* mission has been presented in figure 4.3.

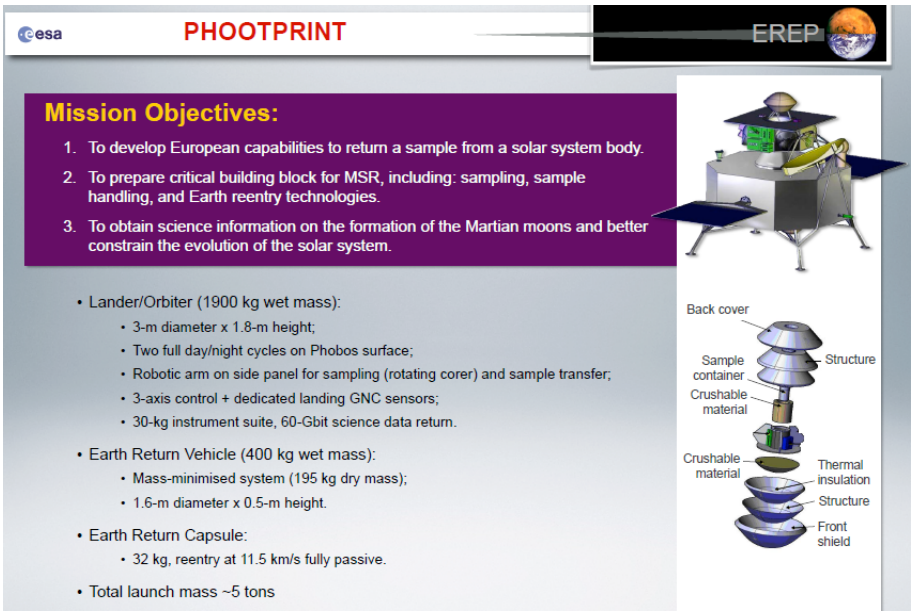


Figure 4.3: Figure showing the details of ESA's Phootprint mission [34]

The STO software uses the weight constraints, the launch dates, the time of flight decided by ESA to analyse the mission. For a rough estimate in the STO solver the final conditions of the S/C are identical to the orbital elements of Phobos to put it the same orbit around Mars. However in reality a B-plane targeting as mentioned in subsection 2.4.2.2 is done and an additional manoeuvre is required for orbital insertion to Phobos. Nevertheless for simplicity the orbital elements have been made identical. Once the sampling has taken place a new scenario has to be setup in the STO tool for the return mission. It can either be done in one optimization

step i.e. using the interface mentioned in 3.4 to have the Sample return in a target bound orbit to the Earth or a similar four flight step method can also be implemented to have the sample return optimized to an orbit. Nevertheless the sample return according to ESA is a passive one, only a target bound manoeuvre from Phobos should be sufficient for sample recovery. The simulations done in the STO as well as STK software have been presented in the next chapter. Even though the missions are planned to return a sample from Phobos, all the calculations are done with respect to Mars as Phobos has a negligible gravity and mission analysis directly from Phobos would be of no use.

Note: The launch to Mars transfer takes place from a 300 km circular parking orbit around the Earth whereas the return leg takes place from a 500 km circular parking orbit at Mars [18].

Chapter 5

Results and Discussion

This chapter first explains the improvements that were made during the thesis to the already existing STO software and then moves on to compare the student generated data with the one obtained from professional resources such as ESOC and STK.

The first change done to the STO software was the update of the SPICE database with the inclusion of Martian and Jovian moons (please refer to figure 3.2). After the update the Gauss Lambert Solver and Optimizer were developed to incorporate not only Hohmann transfers to the Moon but to the bodies within the Solar system whose ephemeris were updated in the SPICE database. Once this was done the minimum departure and arrival velocities as well as the departure ascension and declination angles could be calculated for any two bodies. These parameters then could be used to simulate trajectories within the Solar system and they incorporate short as well as long travel times. However the total revolution for every transfer has been restricted to Type I (less than 180^o) and Type II (less than 360^o) orbits, complete revolutions have been excluded in the STO. Nevertheless this option can be changed by changing the parameter *nrev*; total number of revolutions from zero to any desired value in the file *get_positions.m* which generates the launch and departure position vectors.

The plane change scenarios were also calculated with quaternions however as only an ecliptic to equatorial transformation was desired the DCM was not replaced with quaternions as it would require extensive code modification for not a significant improvement in the software. The orbital conditions for the S/C can now be entered in the Earth's equatorial reference frame and they are displayed in the Earth centred frame and conserve the axial tilt of the Earth.

To illustrate the difference between the representation of a S/C in the original software and the updated version a trajectory around the Earth with an inclination of 28.5^o (i.e. inclination of Cape Canaveral) was done and the simulation was run for 0.2 days. The date selected for the trajectory simulation was the vernal equinox (March 20, 2010). The updated version shows the real life depiction of the trajectory (ref. fig. 5.2)whereas the original depiction in ecliptic frame is illustrated in figure 5.1.

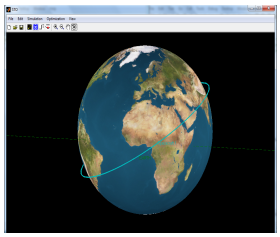


Figure 5.1: Figure illustrating the original depiction of a S/C around the Earth in STO

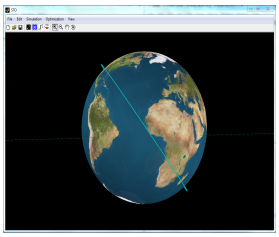


Figure 5.2: Figure illustrating the updated representation of a S/C around the Earth in STO

The difference in performance between the ODE45 and ODE113 with different algorithms within FMINCON was also analysed during the thesis. One trajectory problem was optimized using the FMINCON function and the total number of iterations (Iter), the function counts (F-count) and the final value of the objective function (f(x)) was compared for three different algorithms and the results have been presented in table 5.1. As the performance of ODE45 was better than that of ODE113 and the algorithm *active-set* was equally good as the others no modification was done.

Table 5.1: A table comparing the performance of ODE45 vs. ODE113 and different algorithms within the FMINCON optimizer

Integrator	**Algorithm**	**Iter**	**F-count**	**f(x)**
ODE45	Active-set	35	311	5470.67
	SQP	35	311	5470.67
	Interior-point	35	311	5470.67
ODE113	Active-set	43	394	5473.7
	SQP	43	394	5473.7
	Interior-point	43	394	5473.7

The Gauss lambert solver, Gauss Lambert Tool and the System Tool Kit were used to generate results presented in this section. The mission details were obtained from the mission analysis document received from ESOC. The document presents mission analysis for the next eight years and a general description for the orbit and mission is given. Even though it is a good document for preliminary analysis a more thorough analysis cannot be done because the launch time, the flight time between manoeuvres and the exact arrival and departure dates have not been presented in the paper from ESOC. So, the launch and arrival dates were selected around the middle of the range presented in the paper. These dates were then entered into the Gauss Lambert solver to generate the rough estimate. Since there were similarities between user generated and ESOC's values, Gauss Lambert optimizer was used with the launch date and the flight time to generate precise values which were then used in the STK solver as inputs. Two departure cases that fall within the dates 2022 to 2024 AD with return legs in 2024 and 2026AD

were analysed and have been presented here.

5.1 Departure Trajectory: Earth to Mars

As both the STK and STO softwares need two different scenarios to arrive and depart from Mars the orbital sequence has also been analysed in a similar fashion. The initial weight of the S/C is five tonnes.

5.1.1 Mission Analysis Orbit Case: O22S

The first trajectory to be analysed was the **CASE:O22S**. The porkchop plots for this mission using the ESOC's launch date and the TOF were created using the GLS as the first step of mission analysis and the plots for departure and arrival velocities as well as the declination angle have been presented in figures 5.3 and 5.4.

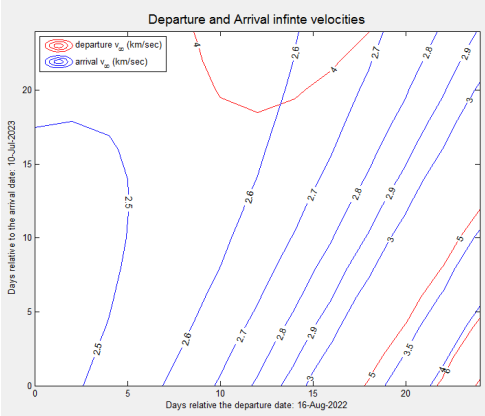


Figure 5.3: A porkchop plot of infinite departure and arrival velocities for orbit CASE:O22S

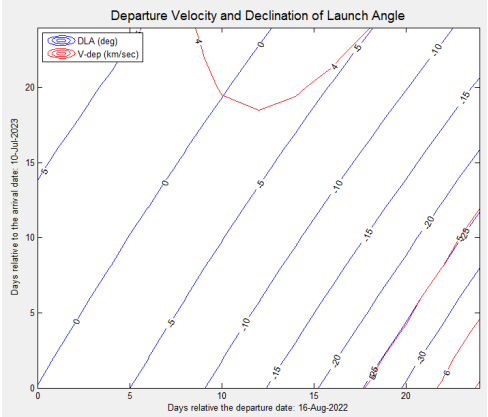


Figure 5.4: A porkchop plot of departure velocity and declination angle for orbit CASE:O22S

Once the porkchop plots for the possible departure and arrival dates were generated, the GLO was used to generate the accurate departure and arrival conditions and the hyperbolic right ascension and declination angles required by the orbit. These values were then used by the FMINCON optimizer to generate the exact conditions for arrival at Mars. One point to be noted though is that the document from ESOC does not mention a certain launch time and also the right ascension angle, so a coastal phase was inserted in the STO to find the right conditions for launch as the resulting error from even a minor difference in launch angle can be huge in interplanetary distances. The flight path angle of the S/C was made equal to the hyperbolic RA angle and the launch was done with the declination at this point with a deep space manoeuvre estimated by the optimizer in the third flight step to correct this error. The launch conditions are presented in tables 5.2 and 5.3 and the non-optimized values from GLO are compared with that from ESOC in table 5.4.

Table 5.2: A table containing the initial orbital parameters of the interplanetary S/C at Earth

Parameter	**Value**
Inclination	5^o
RAAN	20^o
Argument of Perigee Ω	10^o
True Anomaly ν	10^o
Eccentricity e	0
Semi-major axis [a]	6678 [km]

Table 5.3: A table containing the flightsteps and engine specifications of the interplanetary S/C

Flightstep	**Specific Impulse (Isp) [s]**	**Thrust [N]**
Coastal phase	N/A	N/A
Trans Mars Injection (TMI)	950	2500
Mars Course Correction (MCC)	850	1500
Mars Orbital Injection (MOI)	650	2000

Table 5.4: Comparison of orbital transfer parameters calculated at ESOC (Case: O22S) [18] with STO

	ESOC	**STO**
Launch Epoch	2022/8/16 - 2022/9/5	2022/8/26
Escape Velocity [km/s]	3.5	3.9865
Escape Declination [deg]	0	N/A
Escape Right Ascension [deg]	N/A	78.7807
Deep Space Manoeuvre [m/s]	446-619	N/A
Arrival Epoch	2023/7/10 - 2023/8/13	2023/07/21
Hyperbolic Arrival Velocity [km/s]	2.092 - 2.442	2.5 - 3
Impulsive Mars Orbital Insertion(MOI) [m/s]	535-687	N/A
Transfer Duration [days]	329 - 342	335

Note: The N/A in the tables represent unavailable values from ESOC's document and also from STO which before optimization only uses a one burn transfer.

Comparing the values obtained at ESOC and the ones obtained with the STO tool the similarities in the magnitude of the values albeit few differences can be seen. The orbital parameters obtained at ESOC are with the most optimum launch conditions, transfer durations, final B-plane targeting and manoeuvres so the ΔV values are less. Nevertheless, the total ΔV at least in the unoptimized case which does not take into account the required plane changes, the different manoeuvres and is basically a one burn transfer the sum of the initial ΔV at Earth and the deep space manoeuvre in data from ESOC almost equals the initial ΔV required in the STO tool. This value either decreases or increases with the FMINCON optimizer depending upon the state of the S/C after the coastal phase. The final trajectory obtained from the STO tool appears almost identical (fig. 5.6) to the one obtained from ESOC (fig. 5.5).

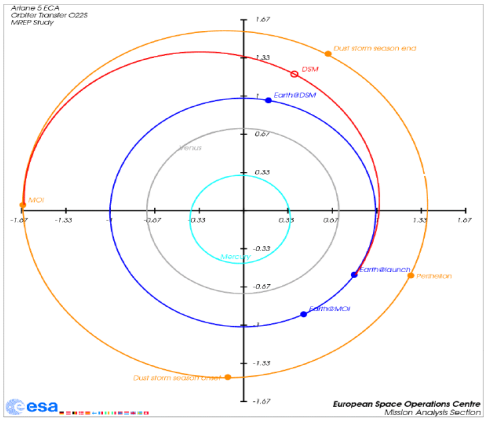


Figure 5.5: CASE: O22S - trajectory to Mars as calculated at ESOC [18]

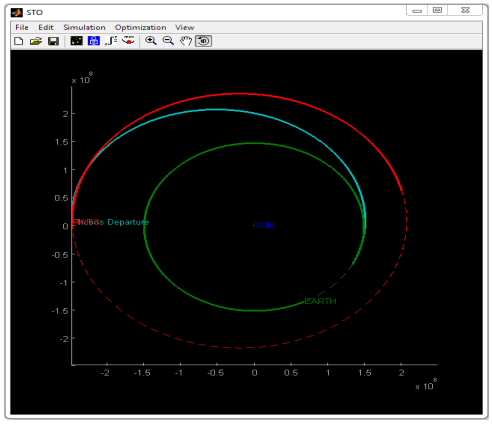


Figure 5.6: CASE: O22S - the final trajectory optimized with the STO software

From the table 5.5 it can be seen that the values after optimization are higher than the previously calculated estimates. This arises due to the fact that the STO does not account for the initial declination of the S/C after the coastal phase and the orbital insertion at Mars is done at an altitude of 9000 km. The increase in MOI value is also attributed to the increased departure velocity (TMI) calculated by the optimizer as a more energetic launch requires a higher delta V for deceleration and orbital insertion.

Table 5.5: Optimized orbital parameters of orbit (Case: O22S) as calculated with the STO software

Flightsteps	**TOF [Days]**	**Delta V [m/s]**
Coastal phase	0.4612159	N/A
TMI	3.46928	5760.68
MCC	333.7855	10.0204
MOI	0.2	-2198.68

The boundary conditions for the optimizer were target body's inclination, perigee, apogee

and true anomaly and the parameters optimized were flight duration, delta V and the in-plane and out of plane angles. The in and out of plane angles are calculated in the inertial frame and are most significant in the elliptical transfer phase and were found to be -170.441^o and 5.1937^o.

5.1.2 Mission Analysis Orbit case: O24S

The second departure mission analysed was case *O24S*. The initial conditions for the S/C were kept the same and data was generated.

Table 5.6: Comparison of the orbital transfer parameters calculated at ESOC (Case: O24S) [18] and STO

	ESOC	**STO**
Launch Epoch	2024/9/20 - 2024/10/10	2024/9/30
Escape Velocity [km/s]	3.55	3.8426
Escape Declination [deg]	0	0
Escape Right Ascension [deg]	N/A	100.06
Deep Space Manoeuvre [m/s]	40 - 195	N/A
Arrival Epoch	2025/7/23 - 2025/8/15	2025/07/23
Hyperbolic Arrival Velocity [km/s]	2.653 - 2.687	2.8121
Impulsive Mars Orbital Insertion(MOI) [m/s]	788 - 805	N/A
Transfer Duration [days]	306 - 310	306

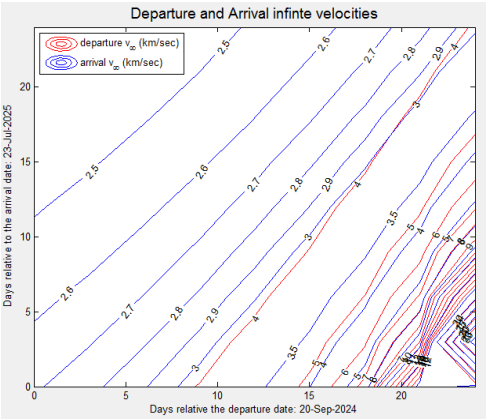


Figure 5.7: A porkchop plot of infinite departure and arrival velocities for orbit CASE:O24S

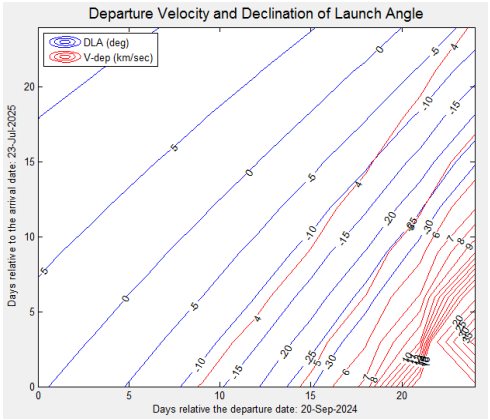


Figure 5.8: A porkchop plot of departure velocity and declination angle for orbit CASE:O24S

The same parameters were selected for boundary conditions and for flight step optimization. The in plane and out of plane angles obtained for the elliptical phase respectively are -9.941^o

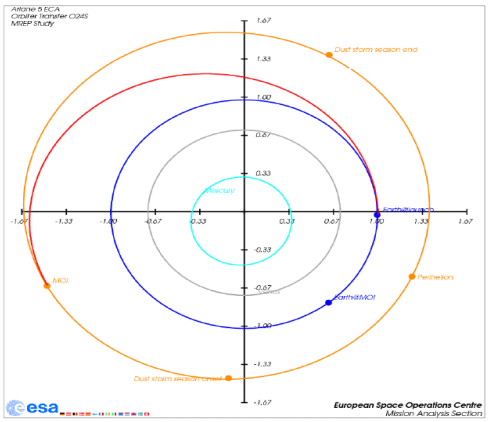


Figure 5.9: CASE: O24S - trajectory to Mars as calculated at ESOC [18]

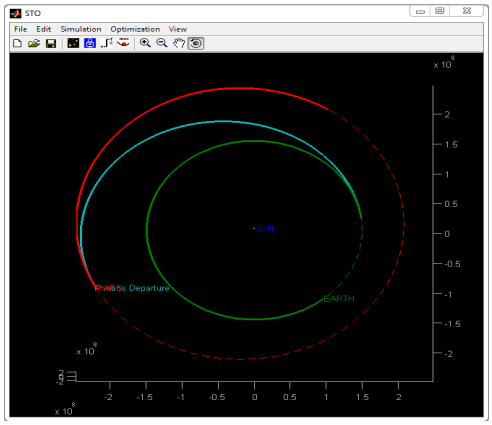


Figure 5.10: CASE: O24S - final trajectory optimized with the STO software

Table 5.7: A table containing the optimized orbital parameters of orbit (Case: O24S) as calculated with the STO software

Flightsteps	TOF [Days]	Delta V [m/s]
Coastal phase	0.0875	N/A
TMI	4.1815	5474.17
MCC	301.8137	35.1091
MOI	0.2	637.428

and 83.9932°. From table 5.7 the change in initial values can easily be observed. The departure velocity is higher but the orbital insertion velocity is lower in this case. The difference in values also arises from the setup of the optimizer and error arising from it. A more elaborate description of the FMINCON and possible errors from it is discussed in the conclusion.

5.2 Arrival Trajectory: Mars to Earth

Unlike the departure trajectory the return leg of the mission is much simpler as only an entry altitude is the required parameter for the S/C to enter the Earth. This leg of the mission is an impact trajectory at the Earth. For this leg no predefined latitude, longitude requirement has been provided in the document from ESOC and the calculations in the paper are also based on simple arrival trajectories, so the optimization in the return leg was done for only three flight steps rather than the four flight steps for the departure to Mars. As the calculations at ESOC were based on a return trajectory from Mars starting at a circular orbit at 500 km altitude, the same parameters were used in the STO tool to simulate the return conditions.

5.2.1 Mission Analysis Orbit case: R24S1

The first mission to be analysed for the return leg was the orbit case $R24S1$ as it is the first possible return trajectory for orbit $O22S$. The remaining mass of the S/C which was around 2.5 tonnes was used as the initial mass of the S/C. The value calculated for the Phootprint mission as presented in 4.3 is a mass of around 1.9 tonnes. Since the orbital correction, insertion and landing in Phobos is not done with the STO tool as well as the data obtained from ESOC the mass value appears to be a bit higher. Nonetheless, it is a reasonable result for the performance of the STO tool. The initial orbital parameters of the S/C and the flight steps/engine setup for the return leg are presented in tables 5.8 and 5.9.

Table 5.8: A table containing the initial orbital parameters of the interplanetary S/C at Mars

Parameter	**Value**
Inclination	1.075^o
RAAN	0^o
Argument of Perigee Ω	0^o
True Anomaly ν	0^o

Table 5.9: A table containing the flightsteps and engine specifications of the interplanetary S/C

Flightstep	**Specific Impulse (Isp) [s]**	**Thrust [N]**
Coastal phase	N/A	N/A
Trans Earth Injection (TEI)	950	2500
Earth Course Correction (ECC)	550	1500

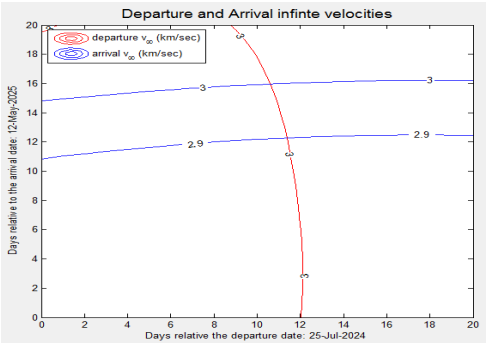


Figure 5.11: Orbit CASE:R24S1 infinite departure and arrival velocities porkchop plot

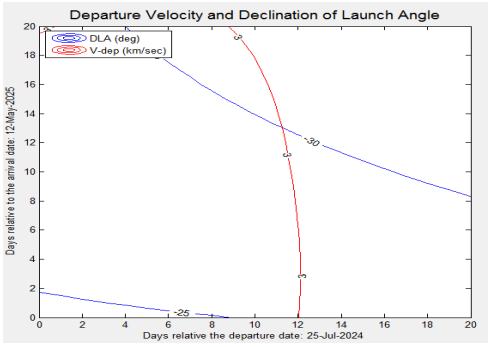


Figure 5.12: Orbit CASE:R24S1 departure velocity and declination angle porkchop plot

Once the crude estimate was made with the porkchop plots, refined values were generated with the GLO. These values were then compared with the ones obtained from ESOC orbit case $R24S1$ for the return leg of the transfer and are presented in table 5.10.

Table 5.10: A table comparing the interplanetary orbital transfer parameters calculated at ESOC (Case: R24S1) [18] to the ones obtained before optimization in STO

	ESOC	**STO**
Launch Epoch	2024/7/25	2024/7/25
Escape Velocity [km/s]	2.935	2.1458
Escape Declination [deg]	9.9	-3.5505
Escape Right Ascension [deg]	N/A	110.04
Deep Space Manoeuvre [m/s]	0	N/A
Arrival Epoch	2025/5/12	2025/05/12
Hyperbolic Arrival Velocity [km/s]	2.810	2.6074
Transfer Duration [days]	291	291

The resulting orbit from STO is then compared with the official one from ESOC for similarities (ref. fig.5.13 and 5.14)

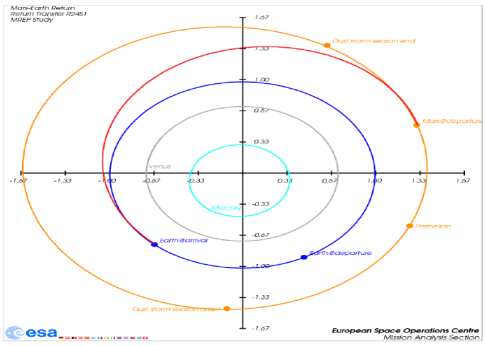


Figure 5.13: CASE: R24S1 trajectory to Earth as calculated at ESOC [18]

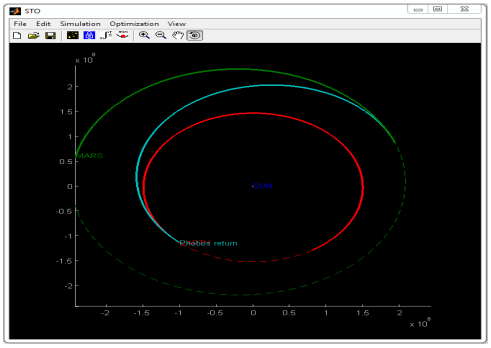


Figure 5.14: CASE: R24S1 the final trajectory as optimized with the STO software

The boundary conditions and the constraints were kept the same as in the Earth departure conditions. After optimizing the flight steps the values obtained are presented in table 5.11.

The in plane and out of plane angles for the elliptical phase of the return orbit $R24S1$ were found to be -93.4201^o and 85.6548^o respectively.

Table 5.11: Case: R24S1 STO software optimized orbital parameters

Flightsteps	TOF [Days]	Delta V [m/s]
Coastal phase	0.3695	N/A
TEI	4.4904	2419.11
ECC	283.2748	161.6650

5.2.2 Mission Analysis Orbit Case: R26S

Another sample return mission corresponding to the departure orbit $O24S$ is the return leg starting at Mars on 2026 A.D. The sample return in this case was also analysed with the same initial conditions of the S/C as for the orbit $R24S1$. The porkchop plots for this return leg are presented in figures 5.15 and 5.16.

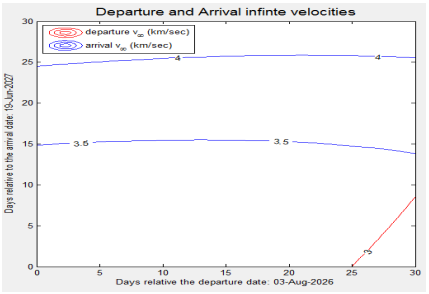


Figure 5.15: CASE:R26S infinite departure and arrival velocities porkchop plot

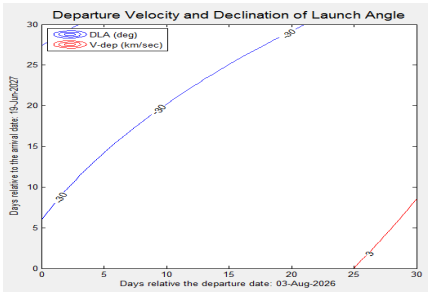


Figure 5.16: CASE:R26S departure velocity and declination angle porkchop plot

Table 5.12: A table comparing the interplanetary orbital transfer parameters calculated at ESOC (Case: R26S) [18] to the ones obtained before optimization in STO

	ESOC	STO
Launch Epoch	2026/08/03	2026/08/03
Escape Velocity [km/s]	2.677	2.0445
Escape Declination [deg]	4.6	-20.1357
Escape Right Ascension [deg]	N/A	325.0540
Deep Space Manoeuvre [m/s]	0	N/A
Arrival Epoch	2027/6/19	2027/06/19
Hyperbolic Arrival Velocity [km/s]	3.093	2.9626
Transfer Duration [days]	320	320

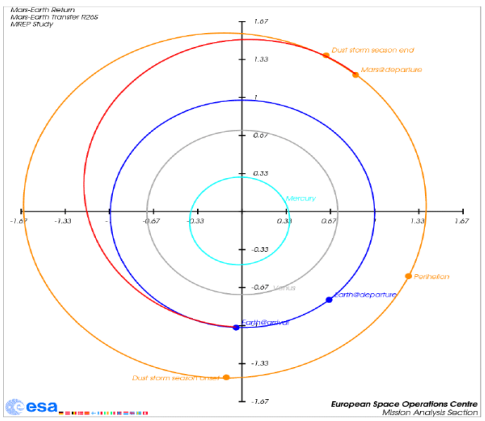


Figure 5.17: CASE: R26S trajectory to Earth as calculated at ESOC [18]

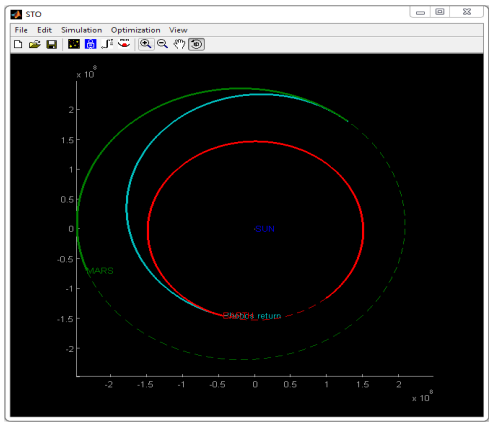


Figure 5.18: CASE: R26S the final trajectory as optimized with the STO software

With the similar boundary conditions as in the orbit case $R24S$, the flight steps were optimized and the values obtained are presented in table 5.13 and the resulting orbits are compared in figures 5.17 and 5.18.

Table 5.13: A table containing the optimized orbital parameters of orbit (Case: R26S) as calculated with the STO software

Flightsteps	**TOF [Days]**	**Delta V [m/s]**
Coastal phase	0.8877	N/A
TEI	3.8122	2258.86
ECC	310.105	408.901

The in plane and out of plane angles for the elliptical phase of the return orbit $R24S1$ were found to be -48.4395^o and -60.105^o respectively.

From the analysis of the departure and return trajectories to Mars i.e. four orbits $O22S$, $O24S$, $R24S1$ and $R26S$ the similarities in the values obtained at ESOC and that with the STO software are easily visible. Even though they are not the same they fall within the same range so the STO software can be used as the estimate for all the missions. The discrepancy between the values also arises from the unavailability of the actual launch conditions for these missions. The mission analysis at ESOC does not contain the RA values, the exact launch and arrival times, the engine properties and also the TOF after which the deep space manoeuvre phase is activated so only a good estimate could be entered into the STO for trajectory calculation.

Regarding the porkchop plots for the four orbit cases two points have to be noted. The first being the departure v infinity and $V - dep$ are identical quantities but have been labelled

differently. Another being the lack of clear minima available in the porkchop plots as expected. The porkchop plots in this case were generated only for the time range that was available in the mission analysis from ESOC , so they are used here more for verification of values rather than finding the minimum energy regions. The porkchop plots represented also just compare departure and arrival infinite velocities and also the departure declination and departure velocity. This was done to check ESOC's values and also to check if the declination angles mentioned in the mission analysis were also verified with the porkchop plots. For e.g. for the orbit case O22S the declination angle of zero degrees for a departure velocity of around 4 km/s occurs around August 26, which was the date taken for the GLO.

In addition to the comparison with ESOC's mission analysis the STO software was also compared with AGI STK to see if it offers the best initial conditions for more detailed mission analysis. The RA, square of infinite velocity (C3 energy) and declination values obtained with the STO were entered in the STK and the trajectory obtained without optimization for the orbit case $O22S$ is presented in figure 5.19 and 5.20 and other values also produced similar results.

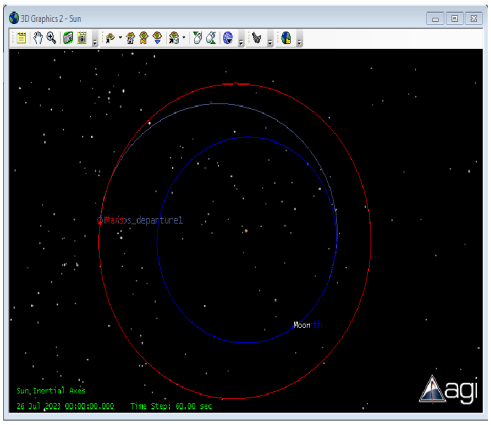


Figure 5.19: Unoptimized orbit obtained in STK with initial data obtained with STO [9]

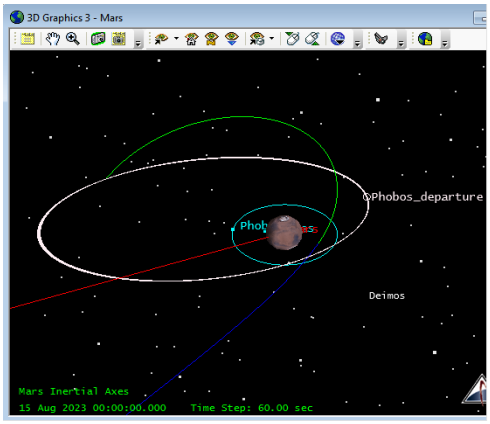


Figure 5.20: Orbital insertion and plane change manoeuvre at Mars with STO values [9]

The difference in departure and arrival velocities can be attributed to the RA and declination angle which were different from ESOC. The optimized TOF from the STO tool was also always found to be less than the ones from ESOC at least for the return orbits. This is due to the optimization being concluded in just three flight steps rather than four. However, four flight steps can be included to minimize the ΔV if the required inclination at arrival and other arrival conditions are known. The values also differ because ESOC considers the J2 perturbation of the Earth, the perturbation at Mars and also the atmospheric influence for the arrival at Mars whereas the STO does not consider these effects but only the perturbations arising from the bodies in question i.e. in this case the Sun, Earth, Mars and the S/C. The perturbation generated by each body on other bodies is however considered and lineraly added to the acceleration component using the Cowell method as mentioned in the section 2.5.2.

As previously mentioned the Gauss Lambert problem is calculated while only considering the position of the planets at the beginning and the end of time. The launch however takes place from a parking orbit and the position of the S/C is not exactly co-incident with the RA calculated from the Gauss Lambert problem. Therefore a coastal phase is inserted to find the launch condition which corresponds to the required hyperbolic departure condition but this already leads to an error as the planets move within this time and the departure conditions have already changed in the meantime. The FMINCON optimizer tries to modify the coastal phase and target the S/C to the destination so this also generates some error however the point to be noted is that the GLS gives a good estimate for the initial conditions and makes it easy for the optimizer to find a complete solution for all the flight steps in a couple of minutes. The change in the ΔV after optimization can also be attributed to this error correction.

After comparison with the values from ESOC and STK it can be said that the STO tool is quite effective and good enough for preliminary mission analysis. The initial conditions of the S/C can be defined, the launch windows can be calculated, flight step optimization with either the duration, ΔV or the mass can be done, if entered, the software also accounts for jettisoned mass so a multi-stage rocket engine can also be simulated, the SPICE kernel can be updated to include other bodies and the tool works for trajectories to and from any two planetary bodies. Thence the STO can be thought of as a valuable tool for orbital analysis within the solar system and the values obtained from STO can be used in professional softwares like STK for a more detailed analysis. Thus, all in all it is a good tool for preliminary analysis of orbits.

Chapter 6

Conclusion

The main goal of the master thesis was to increase the capability of the existing STO software. The final STO looks similar to the original but has many additional capabilities. The positions of the S/C in Earth orbit are shown in the Earth mean equator and equinox of J2000 reference frame or the Earth Centered Inertial frame and the axial tilt of the Earth and the real inclination of the S/C is observed in the GUI unlike the original where everything was represented in the ecliptic.

The initial questions about the benefits of using a more accurate integrator and the benefits of using quaternions instead of DCM were also answered. The fourth order Runge Kutta (ODE45)was found to be faster and also accurate in comparison to the Adams Moulton Predictor Corrector (ODE113) method as shown in table 5.1. Regarding plane changes only one rotation was required so the DCM was not replaced with quaternions as it would not offer an added benefit and an extensive modification of the code was required.

The initial software had a limited SPICE database and did not include the Martian and Jovian moons where exploration missions in the near future have been planned. The SPICE kernel was updated to include other planetary barycentres and also their moons. In the original STO the celestial objects could only be defined according to the Solar System Barycentres but in the updated version the definition of bodies is also possible with respect to the barycentre of the Earth and Mars. This is useful if the orbital analysis is to be done for objects within the planetary sphere of influence without considering the influence from the Sun, for e.g. previously the Moon could not be described without defining the Sun as the reference for the Earth but now the centre of the Earth can be defined as Earth Barycentre and the Moon can be described with its centre as the Earth.

The initial software was only capable of finding and optimizing Hohmann transfer trajectories to the Moon. The launch windows calculated in this method had stringent requirements and sometimes a huge coastal phase spanning a few days was required. The launch window for interplanetary missions was therefore calculated by solving the Gauss Lambert problem and the right ascension and declination of the asymptote of the departure hyperbola was used as

the parameter to define the launch window. This modification helps in generating launch windows within the user defined search range usually one day. The right ascension of the desired hyperbolic asymptote is equated with the S/C's flight path angle calculated from the vernal equinox (x-axis) and the user can choose between the first occurrence or the best occurrence which is defined according to the tolerance in angle which by default has been set to $\pm\ 3^o$. The declination requirement is not considered in the launch window but rather optimized with the FMINCON function in the second flight step. These changes to the structure of the software enable calculation of departure and return trajectories between any two bodies within the solar system.

To aid preliminary mission analysis a separate tool, the Gauss Lambert Solver was also developed. The tool also uses the same ephemeris file as the STO tool and can generate porkchop plots defining the infinite departure and arrival velocities, the right ascension and declination at departure and arrival and also the time of flight of the interplanetary transfer. The search range and the precision of the data is user defined as shown in figure 3.6 therefore the user can select any departure and arrival planets and also the precision for the plots. One point to be noted however is the validity of the ephemeris file DE421, which only has data until 2050 A.D, so for both the STO as well as the Gauss Lambert Solver no data can be generated after 2050 A.D.

The data obtained from STO was compared with the ones obtained at the European Space Operations Centre for the examining the accuracy of the in house generated data. Even though the exact initial launch conditions were not known and some guesses on the right ascension angle, actual time of flight, declination angle and the time elapsed until the deep space manoeuvre had to be guessed the results obtained with STO are very similar to the one obtained at ESOC. For e.g. the Gauss Lambert Optimizer which solves the trajectory problem with one tangent burn had a required ΔV of $3.9865 km/s$ for the orbital case $O22S$ and the ΔV required in the orbit calculated at ESOC was $3.5 km/s$ for the initial burn and $0.446\ -\ 0.619 km/s$ for the deep space manoeuvre, so in total the required ΔV was almost the same. The optimized value was a bit higher because the STO does not consider impulsive burn and also the correction for the declination angle and the slightly changed conditions at the end of the coastal phase are considered.

The STO also has a graphical feature to analyse the change of mass and the evolution of orbital parameters. The changes in mass and distances for all the orbital cases have been presented in Appendix A (6). As a feature that enables adding titles and labels was found in the original STO software, a default labelling of the axes and title generation was not activated as the current option also has *hold on* features, so different properties can be analysed at once and having default titles could be misleading, so it has been left to the user to properly label the axes and give a meaningful title.

Since the STO now has many features that enable a proper orbital analysis it can be thought of as a complete software in its own regard for preliminary analysis nevertheless there is always room for improvement and some more changes and features can be included into the software to make it more user friendly and accurate. One issue with the current version of the software

is the use of FMINCON which is a gradient based algorithm and sometimes can be stuck in a local minima thus avoiding the global minima and the required orbital convergence. The orbits calculated in the report were already known to be global minima from the porkchop plots and also from the report from ESOC, so this problem was not encountered. However if a short time of flight is desired the software can encounter this problem and even though the trajectory propagates towards the destination the required orbital insertion is never achieved. One solution to this problem could be the use of genetic algorithms that use pattern search and thus avoid stopping at local minima unless a global minima is found. This could greatly enable the STO tool to calculate trajectories for any planetary configuration and any time of flight while avoiding convergence problems.

Another improvement could be the inclusion of the Gauss Lambert Solver into the STO, as an additional tab. This would facilitate its use and also enable the user to do the orbital analysis in the same programme rather than having to launch two different softwares. Once this is done the numbering of the planets could also be replaced with strings, so the Earth could be selected from the drop down menu rather than entering 3 into the departure planet box. These changes would make the software more user friendly and complete.

With the added features and capabilities the STO is more powerful than the previous version and accurate enough for basic orbital analysis. However more features such as porkchop plots for transfers to Jovian moons, the update of the ephemeris to include all the bodies in the solar system, a more accurate launch window calculation considering the declination of the hyperbolic asymptote and the axial tilts of the planets can also be included in the future to further improve the capabilities of the software. The second version of the software developed during the thesis has nevertheless reached its objectives and these changes have been left for the future.

Bibliography

[1] M.S.K. Adnan, R. Razali, Md. Azlin, and Md. Said. A study of perturbation effect on satellite orbit using cowell's method. http://www.dlr.de/Portaldata/49/Resources/dokumente/archiv4/IAA-B4-1380P.pdf. accessed on 25/06/2013.

[2] National Aeronautics and Space Administration. Future missions. www.nasa.gov/missions/future/. accessed on 16/06/2013.

[3] National Aeronautics and Space Administration. Gallery:phobos. http://solarsystem.nasa.gov/multimedia/gallery/PHOBOS_732X520.jpg. accessed on 25/08/2013.

[4] National Aeronautics and Space Administration. Navigation and ancillary information facility: An overview of spice. http://naif.jpl.nasa.gov/pub/naif/toolkit_docs/Tutorials/pdf/individual_docs/04_spice_overview.pdf. accessed on 25/06/2013.

[5] National Aeronautics and Space Administration. Solar system exploration: Phobos facts and figures. http://solarsystem.nasa.gov/planets/profile.cfm?Object=Mar_Phobos&Display=Facts&System=Metric. accessed on 16/08/2013.

[6] National Aeronautics and Space Administration. The spice concept. http://naif.jpl.nasa.gov/naif/spiceconcept.html. accessed on 25/06/2013.

[7] European Space Agency. European robotic exploration programme. http://www.belspo.be/belspo/space/doc/euPolicy/2012_07_03/EREP.pdf. accessed on 25/08/2013.

[8] European Space Agency. Missions. http://sci.esa.int/science-e/www/area/index.cfm?fareaid=71. accessed on 16/06/2013.

[9] Inc. Analytical Graphics. Stk systems tool kit version 10, 2013.

[10] Robert A. Braeunig. Interplanetary flight. http://www.braeunig.us/space/index.htm. accessed on 25/06/2013.

[11] Robert A. Braeunig. Orbital mechanics. http://www.braeunig.us/space/index.htm. accessed on 25/06/2013.

[12] Vladimir A. Chobotov. *Orbital Mechanics*. Americal Institute of Aeronautics and Astronautics Inc., third edition, 2002.

[13] Michael J. Croteau. Earth-moon libration point 2: A gateway to the far side of the moon and deep space exploration. http://ccar.colorado.edu/asen5050/projects/projects_2012/croteau/. accessed on 10/07/2013.

[14] David. Eagle. Interplanetary porkchop plots. http://www.mathworks.de/matlabcentral/fileexchange/39248-interplanetary-pork-chop-plots. accessed on 14/05/2013.

[15] Herbert. Halpern. Non-linear programming: Active-set method. https://math.uc.edu/~halpern/Nonlinearprogramming/hofile/Activesetmethodnlpho.pdf. accessed on 03/08/2013.

[16] Takahiro. Hiroi, Michael E. Zolensky, and Carle M. Pieters. The tagish lake meteorite: A possible sample from a d-type asteroid. http://planetary.brown.edu/pdfs/2724.pdf. accessed on 16/08/2013.

[17] David. Hughes. Asteroids. http://www.astronomy-education.com/index.php?page=114. accessed on 16/08/2013.

[18] M. Khan. Mars sample return orbiter: Mission analysis guidelines. *SRE-PAP/MREP/MOR-MAG*, 1, 2010.

[19] Jet Propulsion Laboratory. Jpl planetary and lunar ephemerides. http://ssd.jpl.nasa.gov/?planet_eph_export. accessed on 14/08/2013.

[20] Pascal. Lee. Phobos and deimos sample return: Importance, challenges and strategy. http://www.lpi.usra.edu/meetings/sssr2011/presentations/lee.pdf. accessed on 14/08/2013.

[21] Pascal. Lee. Solar system sample return mission (2011): Phobos and deimos sample return. http://www.lpi.usra.edu/meetings/sssr2011/pdf/5044.pdf. accessed on 14/08/2013.

[22] Oliver. Montenbruck and Eberhard. Gill. *Satellite Orbits: Models, Methods and Applications*. Springer-Verlag Berlin Heidelberg, first edition, 2002.

[23] Mars One. Mission. www.mars-one.com/en/mission. accessed on 16/06/2013.

[24] The-Crankshaft Publishing. Precision orbit determination for earth observation systems. http://what-when-how.com/space-science-and-technology/precision-orbit-determination-for-earth-observation-systems/. accessed on 20/06/2013.

[25] Planetary Resources. Our vision and mission. www.planetaryresources.com/mission/. accessed on 16/06/2013.

[26] SpaceX. Updates. www.spacex.com/news. accessed on 16/06/2013.

[27] Inc. The Mathworks. Constrained non-linear optimization algorithms: R2013a documentation. http://www.mathworks.de/de/help/optim/ug/constrained-nonlinear-optimization-algorithms.html. accessed on 21/07/2013.

[28] Inc. The Mathworks. Ode113: R2013a documentation. http://www.mathworks.de/de/help/matlab/ref/ode45.html#bti6n8p-45. accessed on 03/08/2013.

[29] Inc. The Mathworks. Ode45: R2013a documentation. http://www.mathworks.de/de/help/matlab/ref/ode45.html#bti6n8p-40. accessed on 03/08/2013.

[30] Inc. The Mathworks. Optimization toolbox: Fmincon r2013a documentation. http://www.mathworks.de/de/help/optim/ug/fmincon.html. accessed on 21/07/2013.

[31] Inc. The Mathworks. Optimization toolbox for use with matlab. http://www.cs.ubc.ca/~murphyk/Software/CRF/MatlabOptimizationToolbox.pdf. accessed on 10/07/2013.

[32] Inc. The Mathworks. Optimization toolbox: R2013a documentation. http://www.mathworks.de/de/help/optim/. accessed on 21/07/2013.

[33] Inc. The Mathworks. Optimization toolbox user's guide r2011b. http://cda.psych.uiuc.edu/matlab_programming_class_2012/optim_tb.pdf. accessed on 10/07/2013.

[34] J.L. Vago, M. Coradini, ExoMars, and EREP Teams. Exomars: Mars exploration programme. http://www.lpi.usra.edu/meetings/marsconcepts2012/plenary/0.3.Vago.pdf. accessed on 25/08/2013.

[35] David A. Vallado. *Fundamentals of Astrodynamics and Applications*. Microcosm Press, third edition, 2013.

[36] C. J. Vosenek. Implementing a fourth order runge kutta method for orbit simulation. http://spiff.rit.edu/richmond/nbody/OrbitRungeKutta4_fixed.pdf. accessed on 03/08/2013.

[37] Eric W. Weisstein. Eccentric anomaly. http://mathworld.wolfram.com/EccentricAnomaly.html. accessed on 25/06/2013.

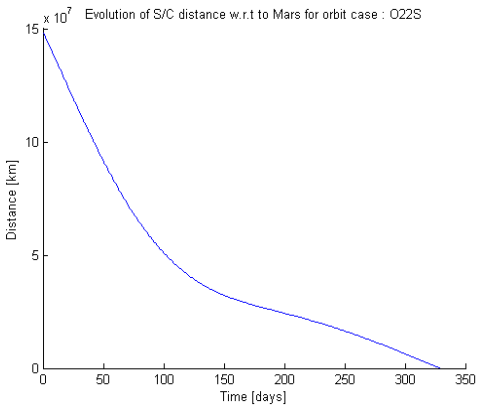
[38] Chris D. Weizmann. Fmincon (optimization toolbox). www.weizmann.ac.il/matlab/toolbox/optim/fmincon.html. accessed on 21/07/2013.

[39] R. Wuseni. Entwicklung eines softwaretools zur optimierung von erde-mond transferbahnen: Diplomarbeit. Institut für Raumfahrtsysteme Universität Stuttgart, 2010.

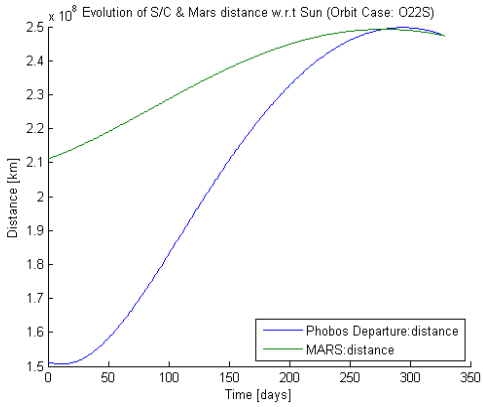
Appendix A

Detailed analysis of the four orbit cases presented in Results and Discussion 5.

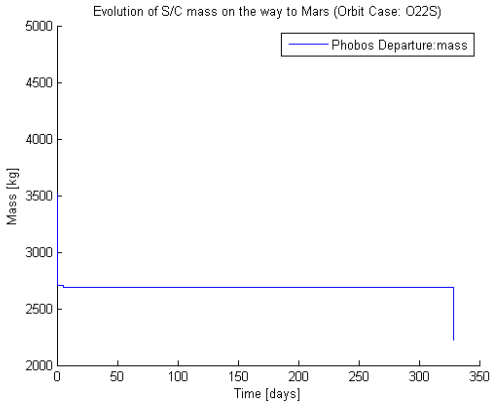
Analysis: Orbit Case - O22S



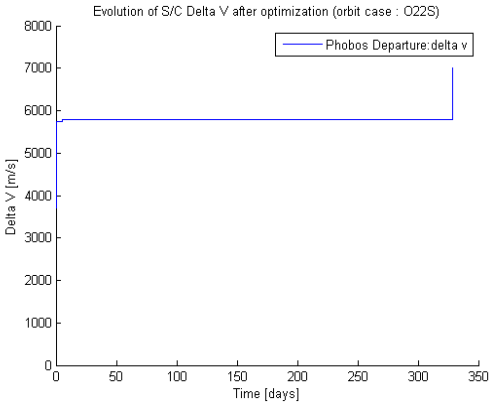
S/C distance evolution during the transfer to Mars



S/C and Mars distance evolution during the whole trip

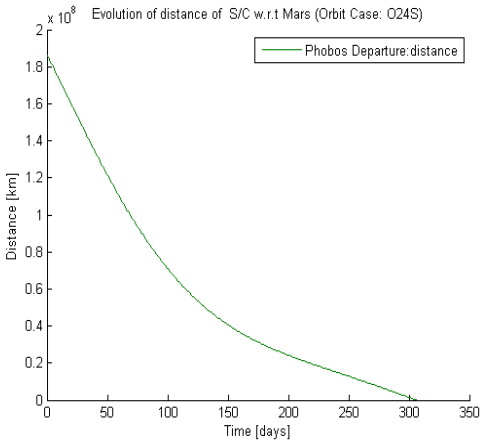


S/C mass evolution during the transfer to Mars

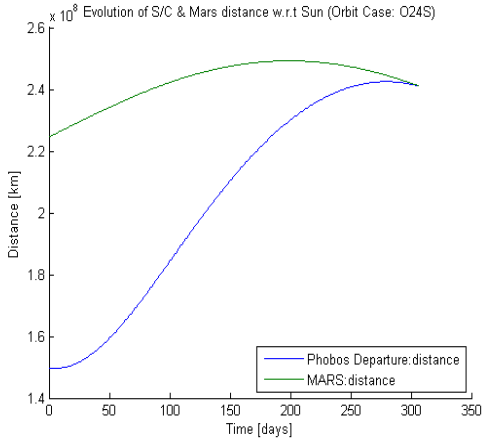


ΔV change with the initial burn, deep space manoeuvre and the final burn

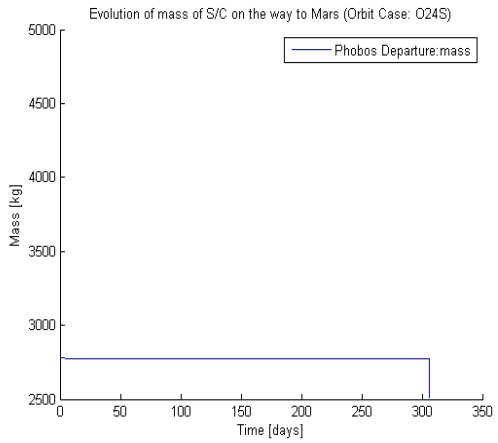
Analysis: Orbit Case - O24S



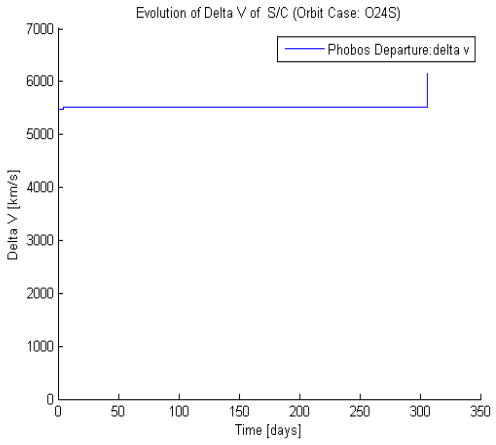
S/C distance evolution during the transfer to Mars



S/C and Mars distance evolution during the whole trip

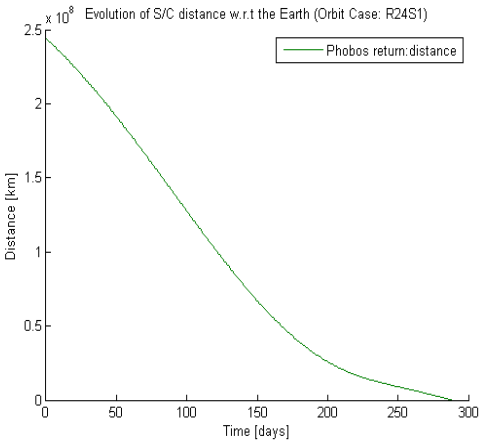


S/C mass evolution during the transfer to Mars

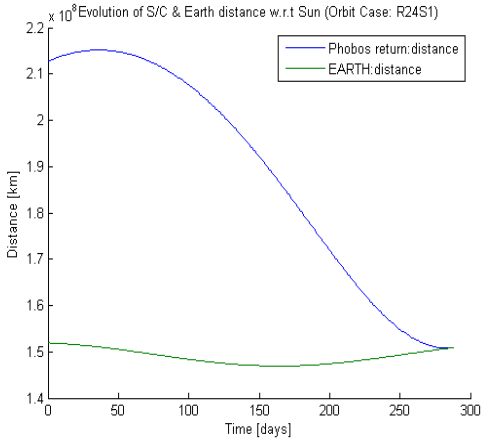


ΔV change with the initial burn, deep space manoeuvre and the final burn

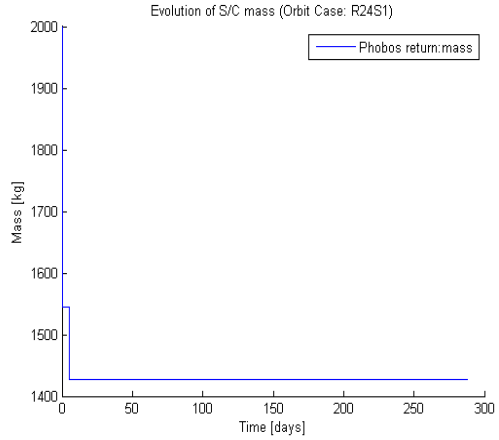
Analysis: Orbit Case - R24S1



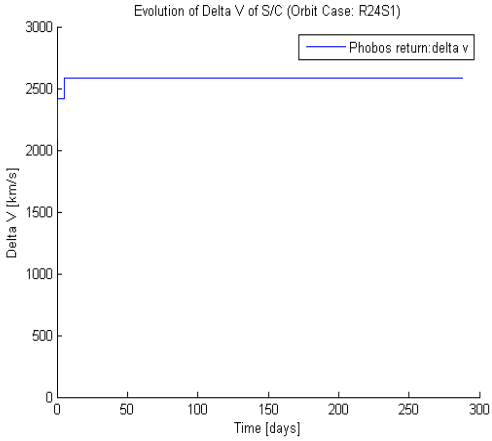
S/C distance evolution during the transfer to Earth



S/C and Earth distance evolution during the whole trip

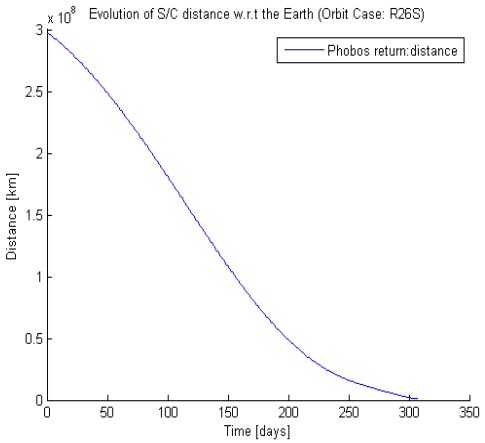


S/C mass evolution during the flight back to Earth

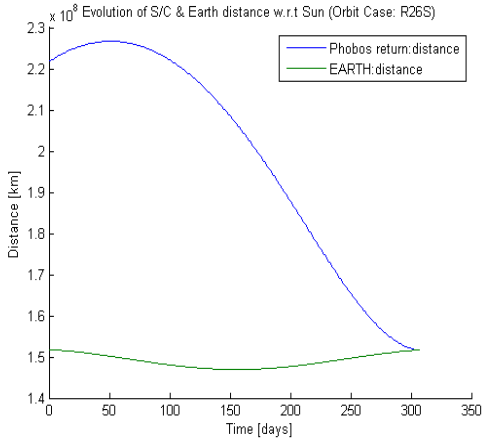


ΔV change with the initial burn, deep space manoeuvre and the final burn

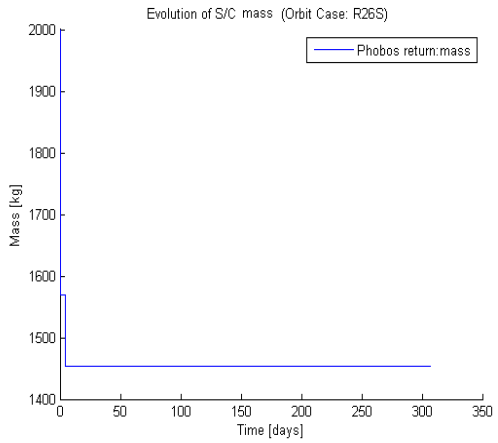
Analysis: Orbit Case - R26S



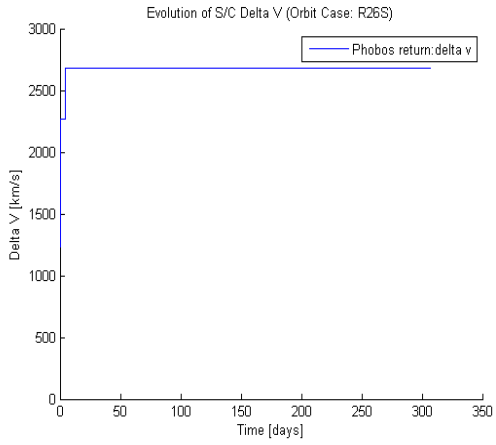
S/C distance evolution during the transfer to Earth



S/C and Earth distance evolution during the whole trip



S/C mass evolution during the flight back to Earth



ΔV change with the initial burn, deep space manoeuvre and the final burn

Appendix B

Detailed analysis of the STO software as prepared by Mr. Robert Wuseni. [39]

General Overview

Implementation

Besides an interface for celestial constellation data, the entire software is written for and operates from MATLAB R2007b or later. The MATLAB calls to the routines for accessing the celestial constellation data are redirected to a MEX library. All saved or exported data consists of structures, which are bundled in a mat-file, the standard MATLAB export format for data fields.

Reference Systems

Ecliptic Co-ordinates

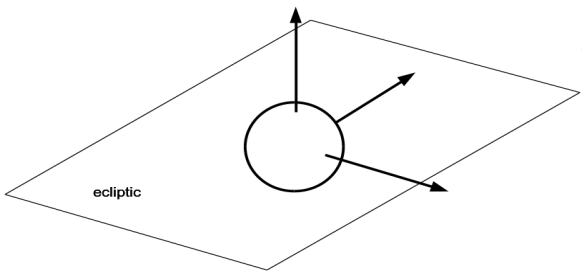
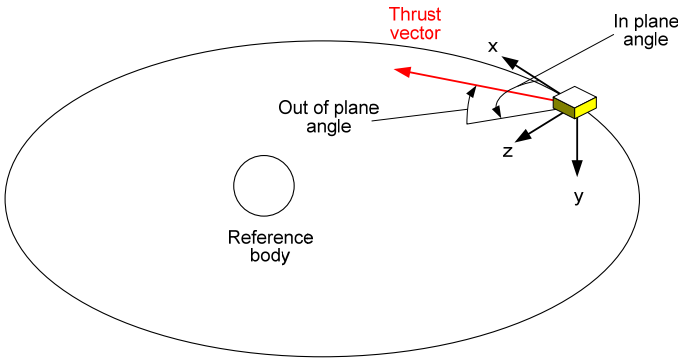


Figure showing the general layout of the Ecliptic co-ordinate system used in the STO

For the definition of all orbits, the ecliptic and mean equinox of the J2000 epoch is used. There are no other coordinate systems, to define the orbital elements in. This is not possible, since there is also no obliquity of the celestial bodies' aces, in which other coordinate systems usually are defined (For example the Earth equatorial system). So all orbital elements have to be entered in this system and are also returned in this system by the program.

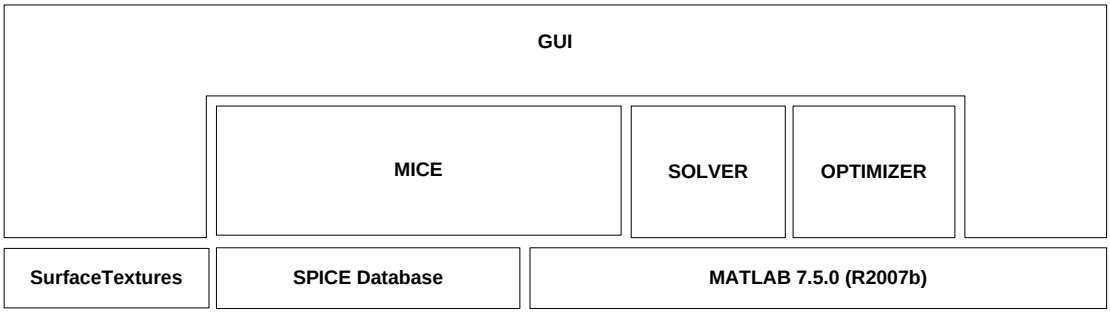
Trajectory coordinate system



For the definition of the orientation of manoeuvres, a relative local trajectory coordinate system is used. A reference body is required, to define this system. The systems x-axis points in the direction of the spacecrafts velocity vector with respect to the reference body. The z-axes is orthogonal to x and points towards the reference body (but not necessarily exactly onto the body). The y-axis completes to a right-handed coordinate system.
The in plane thrust angle is measured in the x-z-plane from the x-direction towards z. It ranges from -180° to 180°. The out of plane angle is measured from the x-z-plane towards the negative y-direction. It ranges from -90° to 90°.

Software design

Software modules



The software is grouped into modules. Each module consists of a set of m-files, which have been written to fulfil a specific task. The module description is listed in the table below:

Module	Description
GUI	Contains all functions, which are required for the graphical user interface.
MICE	Interface and conversion functions for the celestial body data interface
SOLVER	State propagation functions
OPTIMIZER	Optimization functions

Additional functions, which are used by all modules, are also grouped in the same way. But since they don't have a specific task, to deal with, they can't be seen as a module.

File structure

All executable MATLAB files are contained in the "code" folder. Here they are sorted by function into fife subfolders. Each folder corresponds to one of the four software function blocks. General support functions, which are used by several software blocks, are located in an additional "bin" folder. A sixth folder ("data") contains surface textures for the plotting of planets and all database files, which are required for fetching celestial constellations as a function of time. The subfolders and their contents are listed in the table below.

Folder	Content description
`bin`	- conversion functions for orbital elements - conversion functions for time bases - plot routines - export routines - functions for SPICE database access
`data`	- planet surface textures in jpeg format - SPICE kernel summary text file - SPICE database files
`GUIs`	Contains all MATLAB figure files and m-files for the STO graphical user interface
`mice`	MATLAB implementation of the Navigation and Ancillary Information Facility (NAIF) planetary database toolkit "SPICE". For more information on SPICE, please refer to the SPICE documentation on http://naif.jpl.nasa.gov/naif/spiceconcept.html
`OPTIMIZER`	Contains all optimization functions
`SOLVER`	Contains all propagation functions

Data structures

The communication between the different modules is done by passing data structures that contain all information required. The content of these structures is described in this chapter.

Settings structure

All settings and options related to displaying, solver accuracy, optimization goals and data handling are stored in the settings structure. Also physical constants are stored here. The structure contains the following Variables:

Variable name	Possible Values	Description
`gravconst`	0.0 to +inf [N(m/kg)^2]	Gravitational constant. The value entered here is used for: 1.) If SPICE data is used as body property source, its mass is calculated from its GM with this variable as factor. 2.) The GMs of all bodies are calculated before each simulation with this variable

jd	-4000.0 to +inf [days]	Stores the Julian Date of the begin of the simulation
searchrange	0.0 to +inf [days]	Search range for lunar transfer windows
searchangle	0.0 to 360 [°]	Allowed tolerance for lunar transfer window search. This is the maximum allowed angle between the perigee for an ideal Hohmann transfer and the closest approach of a spacecraft to this position.
windowoption	'next', 'best', 'userselect'	Type of option to choose for a lunar transfer. The possible options are: next: The first valid window will be used. Best: The windows with the smallest tolerance angel will be used. userselect: The user will be asked to chose a window from a list.
inistep	>0.0 to < maxstep > [days]	Initial time step size for the solver
maxstep	>0.0 to +inf [days]	Maximum allowed time step size for the solver
relerror	>0.0 to +inf [-]	Tolerated relative solver error
abserror	>0.0 to +inf [km, km/d]	Tolerated absolute solver error
spice	All valid file paths [text]	Path to a SPICE file or to a text file, that contains a list of SPICE files in the SPICE script notation
ref	0 to number of bodies (celestial bodies + spacecrafts) [-]	The body, whose number is set in this variable, is the central body for plotting.
plotorbits	0 / 1 [-]	Switch for plotting initial orbits
plottrajec	0 / 1 [-]	Switch for plotting trajectories
plotnames	0 / 1 [-]	Switch for plotting body names
plotplanets	0 / 1 [-]	Switch for plotting planet
plotmode	0 / 1 [-]	Switch for type of range selection: 0: manual select of plot range 1: automatic scaling of plot range
plotrange	0 to +inf [km]	Value for manual selected plot range
time	0 to max. simulated timestep [days]	Time offset form begin of simulation for plotting
opti	1 to number of spacecrafts [-]	The trajectory of the spacecraft, whose number is selected by 'opti', will be optimized in a following optimizer call.
frames	1 to +inf [-]	Number of frames, which are used for AVI export.
bound_inc	0 / 1 [-]	Optimizer switch for targeting the desired inclination at the target body
bound_raan	0 / 1 [-]	Optimizer switch for targeting the desired argument of the ascending node at the target body
bound_omega	0 / 1 [-]	Optimizer switch for targeting the desired argument of perigee at the target body
bound_a	0 / 1 [-]	Optimizer switch for targeting the desired perigee altitude at the target body

bound_e	0 / 1 [-]	Optimizer switch for targeting the desired apogee altitude at the target body
bound_nu	0 / 1 [-]	Optimizer switch for targeting the desired true anomaly at the target body
bound_tar	0 / 1 [-]	Optimizer switch for minimizing the distance to the target body; When this switch is activated, all others should be deactivated. Although this won't cause any numerical problems, there is a conflict in the desired output. Use this mode only for an initial targeting of a body.

Bodydata structure

The orbital elements and physical properties of all natural bodies at the initial state of the simulation are stored in this structure. The bodies are arranged as an array, whose elements hafe to following properties:

Variable name	Possible Values	Description
activate	False/ True	Indicates, if the defined body shall be used in the simulation.
name	[text]	This field contains the name of the body.
radius	0 to +inf [km]	This field contains the radius of the body
center	[text]	Name of the body, to which the orbital elements refer to. This body can be previously defined body or the inertial reference frames centre.
mass	0 to +inf [kg]	Mass of the body in kg
inc	0 to 180 [°]	Inclination of the orbital plane
raan	0 to 360 [°]	Right ascension of ascending node
omega	0 to 360 [°]	Argument of perigee
a	-inf to +inf [km]	Semi-major axis of orbit
ecc	0 to +inf [-]	Eccentricity of the orbit
nu	0 to 360 [°]	True anomaly of orbit

scdata_init structure

The initial state of all entered spacecrafts is defined by the use of orbital elements, which define the spacecraft position and velocity with respect to a central body. This body has to be defined in the 'bodydata' structure. The spacecraft data is arranged as an array of structures with the following properties:

Variable name	Possible Values	Description
activate	False/ True	Indicates, if the defined spacecraft shall be taken into account for the simulation.
name	[text]	Spacecraft name
center	[text]	Reference body for spacecraft orbit. This

		body has to be defined in the
mass	0 to +inf [kg]	Spacecraft initial mass
inc	0 to 180 [°]	Inclination of orbital plane
raan	0 to 360 [°]	Right ascension of ascending node
omega	0 to 360 [°]	Argument of perigee
hp	0 to +inf [km]	Perigee altitude
ha	0 to +inf [km]	Apogee altitude
nu	0 to 360 [°]	True anomaly of orbit

Scdata_final structure

For the optimization a target orbit is required. This orbit is entered in the structure array 'scdata_final'. The structure contains the following fields:

Variable name	Possible Values	Description
center	[text]	Central body for orbit targeting
inc	0 to 180 [°]	Inclination of target orbit
raan	0 to 360 [°]	Right ascension of ascending node
omega	0 to 360 [°]	Argument of perigee
hp	0 to +inf [km]	Perigee altitude
ha	0 to +inf [km]	Apogee altitude
nu	0 to 360 [°]	True anomaly

Fltsteps structure

The manoeuvre sequence of each spacecraft is split up into arranged by in time steps. Each time step has a specified length and can contain data for an orbital manoeuvre. The time steps are stored in the m*n array 'fltsteps', where m represents the number of spacecrafts and n is the maximum number of flight steps for a single spacecraft. Each flight step structure consist of the following data fields:

Variable name	Possible Values	Description
used	True/ False	This flag is used to compensate for the different amounts of flight steps of the different spacecrafts, since the total length of the flight step field is dictated by the spacecraft with the most flight steps. If a flight step exists for a spacecraft, it's "used" flag has to be set to "true".
activate	True/ False	Indicates, if a manoeuvre is performed, or if entered values are discarded and the step is regarded as a coasting phase.
name	[text]	Name of this flight step
jmass	0 to +inf [kg]	At the beginning of each flight step a certain amount of mass can be jettisoned. The value of this mass can be entered here.
duration	0 to 600 [days]	Nominal duration of a flight step.

isp	0 to +inf [s]	Specific impulse of the used engine in units of seconds
thrust	0 to +inf [N]	Thrust level of the used engine
burn	0 to +inf [m/s] / [s] / [kg]	This field contains sizing information for the specified manoeuvre. It contains one of the three following redundant information: 1.) Burn duration for the manoeuvre 2.) Delta v of the manoeuvre 3.) Change in mass for the manoeuvre The type of information is specified in the "bunits" field.
bunits	1, 2 , 3	Specifies the type of information required for the engine burn: 1.) Burn duration 2.) Delta v 3.) Delta m
ref	[text]	Name of the body of reference for the specification of the manoeuvre direction.
ipa	-180 to 180 [°]	In plane angle for the direction of the manoeuvre. Angle is counted counter clockwise in the orbital plane. Zero is in the direction of flight in the reference system chosen by "ref"
oopa	-90 to 90 [°]	Out of plane angle. Positive angles produce a component of acceleration pointing into the same direction, as the normal vector of the orbital plane.
opt_time	True/ Fasle	Indicates, if the duration of this flight step is an input to the optimizer
opt_burn	True/ False	Indicates, if the burn sizing of this flight step is an input to the optimizer
opt_ipa	True/ False	Indicates, if the in plane manoeuvre angle of this flight step is an input to the optimizer
opt_oopa	True/ False	Indicates, if the out of plane manoeuvre angle of this flight step is an input to the optimizer
opt_mintime	0.001 to 600 [days]	Minimum boundary value for the duration of this flight step
opt_minburn	0 to +inf [s] / [km/s] / [kg]	Minimum boundary value for the burn sizing of this flight step
opt_minipa	-180 to < opt_maxipa > [°]	Minimum boundary value for the in plane angle of the manoeuvre
opt_minoopa	-90 to < opt_maxoopa > [°]	Minimum boundary value for the out of plane angle of the manoeuvre
opt_maxtime	< opt_mintime > to 600 [days]	Maximum boundary value for the duration of this flight step
opt_maxburn	< opt_minburn > to +inf [s] / [km/s] / [kg]	Maximum boundary value for the burn sizing of this flight step

Proclamation

Hereby I confirm that I wrote this concept paper independently and that I have not made use of any other resources or means than those indicated.

Bremen, September 2013