

Politecnico di Torino

Master's Degree in
Aerospace Engineering

Master's Degree Thesis
Targeting and Descent Trajectories for
Moon Missions



Supervisors:

Prof. Manuela Battipede
Ing. Andrea D'Ottavio

Candidate:

Marco Luigi Ottavi

OCTOBER 2023

Contents

1	Context of the Study	1
1.1	Introduction	1
1.2	Mission Overview	1
1.3	Objectives	4
2	Introduction to Computational Optimal Control	5
2.1	Introduction	5
2.2	Formulation of Optimal Control Problems	5
2.3	Nonlinear Programming and Transcription Methods	7
2.3.1	Direct Collocation Methods	9
2.3.2	Direct Single Shooting Method	12
3	Descent Trajectory Dynamics	15
3.1	Introduction	15
3.2	Reference Frames	17
3.3	The Equations of Motion	18
3.3.1	The kinematics equations	18
3.3.2	The dynamics equations	20
3.4	Summary of dynamics and kinematics	28

3.5	Transformation between the original and modified frames	30
3.5.1	From the Original to the Modified Frames	31
3.5.2	From the Modified to the Original Frames	32
3.6	Computing the Range	33
3.7	Trapezoidal Direct Collocation Method Implementation in MATLAB	34
4	Introduction to Astrodynamics	37
4.1	Newton's Law of Universal Gravitation	37
4.2	The N-Body Problem	38
4.3	The Restricted Two-Body Problem - R2BP	39
4.3.1	The two constants of orbital motion	40
4.3.2	The Trajectory Equation	43
4.3.3	Polar Equation of a Conic Section	44
4.3.4	The eccentricity vector	46
4.3.5	Relating orbit geometry with \mathbf{h} and ξ	46
4.3.6	Period of an Elliptical Orbit	47
4.4	Reference Frames	50
4.5	Classic Orbital Elements	52
5	B-Plane Targeting	54
5.1	B-Plane Parameters	54
5.2	Relating the B-Plane parameters to inclination and periapsis	59
5.3	Solving using Newton's method	60
5.4	Accuracy Problems	62

6 Lunar Targeting	64
6.1 The Earth-Moon system geometry	64
6.2 Formulation of the NLP Problem	66
6.3 The Circular Restricted Three-Body Problem	67
6.3.1 The Equations of Motion	67
6.4 SPICE-Enhanced N-Body Problem	71
6.5 The LOI Maneuver	74
6.6 SPICE-Enhanced COSMOGRAPHIA	76
7 Results	77
7.1 Descent Trajectory Optimization	77
7.1.1 Convergence Study	78
7.1.2 Effect of Maximum Thrust - T_{MAX}	78
7.1.3 Effect of Initial Lander Mass - m_0	82
7.1.4 Effect of Specific Impulse - I_{sp}	83
7.1.5 2D and 3D Model Comparison	85
7.1.6 Constraints on Maximum Pitch Rate	85
7.1.7 3D Trajectory Example	86
7.2 Transfer Trajectory Optimization	89
7.2.1 TCM and LOI using the CR3BP	89
7.2.2 TCM and LOI using SPICE	93
Appendices	96
A Spherical Trigonometry	97
B Elementary Rotation Matrices	98

C	Descent Dynamics in the Perifocal Reference Frame	100
D	Orbit Determination and Propagation	103
E	Original Descent Equations	106
F	Cosmographia Catalog Template	108

List of Figures

1.1	Moon Transfer Orbit Injection.	2
1.2	Transfer trajectory overview.	3
1.3	30 x 100 km orbit injection and PDI-Powered Descent Initiation.	4
2.1	Approximation of system state vector through a piecewise polynomial function $\tilde{x}(t)$ defined on the $t_0, \dots, t_N = t_f$ grid	10
2.2	Propagation of system trajectory from different initial conditions, to hit a certain target value at time t_f . This idea is at the base of the Direct Single Shooting Method.	13
3.1	Orientation of the \mathcal{F}_R and \mathcal{F}_T Reference Frames with respect to the celestial body of interest.	18
3.2	Orientation of the \mathcal{F}_S frame with respect to the \mathcal{F}_R frame and forces in the \mathcal{F}_S frame.	23
3.3	1-2-3 Rotation from \mathcal{F}_S to \mathcal{F}_T	25
3.4	Rotation of the $\mathcal{F}_{R'}$ frame to the \mathcal{F}_R frame	31
3.5	Descent Trajectory Range	33
4.1	Newton's Law of Gravity	38
4.2	The perifocal frame.	41
4.3	Common geometrical dimensions of all conic sections.	44
4.4	Conic sections obtained from the intersection of a plane with a circular cone.	45

4.5	Elementary angle $d\nu$, associated to dA , swept by the position vector during time dt	48
4.6	Hohmann Transfer maneuver between two circular coplanar orbits.	49
4.7	Classic Orbital Elements.	52
5.1	Sphere of Influence of a celestial body	55
5.2	The B-Plane.	56
5.3	The B-Vector in the perifocal reference frame.	56
5.4	B-Plane Spherical Trigonometry. The θ angle shown in figure is negative, but this doesn't affect the final result.	59
5.5	Example of B-Plane Parameter numerical partial derivative, evaluated for a certain $(\mathbf{r}, \dot{\mathbf{r}})$, as δV varies.	60
5.6	Moon B-Plane targeting. R2BP hypotheses satisfied.	62
5.7	Moon B-Plane targeting. R2BP hypotheses unsatisfied.	63
6.1	Earth-Moon System Geometry	65
6.2	The Synodic Reference Frame.	68
6.3	Transfer Trajectory in Synodic Frame	70
6.4	Position vectors in the J2000 Frame.	72
6.5	Lunar Targeting in the J2000 Frame	73
6.6	LOI Maneuver Geometry.	75
7.1	Convergence Study of the Direct Collocation Algorithm.	79
7.2	Convergence Study of Final Lander Mass and Final Descent Time .	80
7.3	Maximum Lander Thrust Tests.	81
7.4	Lander efficiency and descent time for different values of the thrust to mass ratio.	82
7.5	Specific Impulse Tests.	83
7.6	Lander Initial Mass Tests.	84

7.7	2D and 3D Model descent comparison	86
7.8	Effect of the $\dot{\gamma}$ constraint.	87
7.9	3D Descent Trajectory Plot	88
7.10	3D Descent Trajectory Plot. View from above the South-Pole . . .	88
7.11	Parametric Study of the TCM Delay vs ΔV cost, for one of the possible arrival dates.	90
7.12	ACM (TCM) Maneuver Cost over 2031 (Departure Dates)	90
7.13	LOI Maneuver Cost over 2031 (Departure Dates)	91
7.14	Example of trajectory in the Synodic Frame.	91
7.15	Example of trajectory in a Non-Rotating Frame. (CR3BP)	92
7.16	Transfer Trajectory in Earth-Fixed J2000.	94
7.17	Arrival Trajectory in Moon-Fixed J2000.	94
A.1	Angles of a spherical triangle.	97
B.1	Fundamental Rotations.	98
C.1	Descent trajectory in perifocal reference frame.	101
F.1	Earth departure in Cosmographia.	110
F.2	Moon arrival in Cosmographia.	110

Acknowledgments

Ringrazio innanzitutto la Prof. Manuela Battipede, per avermi dato la possibilità di svolgere la mia tesi presso TAS-I, e l'Ing. Andrea D'Ottavio, per avermi seguito e guidato durante il mio lavoro. Ringrazio inoltre i ragazzi dell'ufficio, con cui ho avuto modo di confrontarmi e apprendere durante questi mesi e il Dr. Giorgio Fasano, per gli utili spunti forniti durante la stesura del codice.

Questa tesi rappresenta però soltanto il culmine di un percorso di studi complessivamente durato cinque anni, colmi di fatiche ma anche e soprattutto di gioie, condivise con persone che hanno svolto e continueranno a svolgere un ruolo importante; per cui grazie Rocco, per avermi sopportato in questi cinque lunghi anni di convivenza nella ormai storica stanza 428, e grazie Mario, per tutti i dubbi che sollevavi due ore prima dell'esame.

Grazie a tutti i ragazzi della cucina del quinto piano e agli amici di giù e infine, soprattutto, grazie ai miei genitori, a mio fratello, nonna Pia e a tutti i miei altri familiari che mi sono sempre stati vicini durante tutto il tragitto, che non finisce certo qui.

A tutti voi, grazie di nuovo, e *ad maiora*.

Torino, Ottobre 2023,
Marco

Abstract

"It suddenly struck me that that tiny pea, pretty and blue, was the Earth. I put up my thumb and shut one eye, and my thumb blotted out the planet Earth. I didn't feel like a giant. I felt very, very small."

-Neil Armstrong

All the major Space-Powers are looking back again at the exploration of the Moon, this time “to stay”. Driven by scientific, geopolitical and economic purposes, the -still today- totally unexplored Moon South Pole is set as the preferred target region where to land. Being rich of water-ice (permafrost), this region is the most attractive for the development of stable human settlements. Through also the involvement of space private companies, the race is on!

Through the European Space Agency (ESA), Europe wants to play a key role in such a fascinating and challenging rush. Never been on the Moon, Europe is now going to put its best efforts on the so-called Argonaut program, representing the European answer to the new Moon call. The objective is to independently develop a lunar lander able to carry over the lunar surface a greater amount of mass (> 1.5 tons). Consequently, this thesis has been fit into this operative and competitive context by aiming to develop a set of tools providing autonomous mission analysis capabilities, without relying on commercial mission analysis software. The problems addressed are then two: that of computing the necessary trajectory correction maneuver and that of optimizing the descent trajectory at Moon arrival. With the purpose of fulfilling these needs, this thesis is based on the optimization methods further employed in the creation of the software developed for Thales Alenia Space Italy (TAS-I), using the MATLAB language.

The descent dynamics have been written in a Celestial Body-Fixed reference frame, continuing previous development work done at TAS-I with simpler models. The new insights into the problem allowed to improve the previous dynamical model, with the new equations given in the Appendix. The Moon targeting was instead studied using the Circular Restricted Three-Body Problem (CR3BP), and also a higher precision model, importing ephemeris data from the SPICE Toolkit, accounting for the presence of the Sun. This method also allows to export the trajectory data in the SPICE Enhanced software COSMOGRAPHIA, to realistically represent the optimized trajectories profiles. Finally, the B-Plane targeting method has been investigated too.

Chapter 1

Context of the Study

1.1 Introduction

Before delving into the mathematical aspects of optimization and orbital mechanics, it's necessary to understand the context of this study, so that we can then better grasp what the objectives of the thesis are and follow a logical thread in the discussion. As stated in the Abstract, this thesis fits into the operative context provided by the Argonaut Program, and is devoted to the analysis of one of the possible case studies. For this reason, the mission strategy that will be outlined here is not to be considered the best, but one of the possible choices, even along with more exotic ones. An example could be that of employing the weak stability boundary concept to potentially obtain a slightly lower mission cost in terms of ΔV , but at the cost of a much longer mission duration, requiring then an appropriate sizing of the on-board systems.

1.2 Mission Overview

The first step is clearly that of launching the spacecraft and putting it into a waiting orbit around Earth, which is supposed to be a circular 250×250 km orbit. At the appropriate time, the launcher provides an impulse that puts the spacecraft into an elliptical transfer orbit, with a fixed apogee height of 400.000 km (fixed launch program hypothesis) *in the Moon orbital plane*. The launch vehicle in question should be the Ariane 64 (Ariane 6 with a 4 booster configuration) Launch System, or another version derived from further development. Clearly, the Ariane 6 is a launcher that so far has never flown, so the numbers provided before, related to the types of waiting and injection orbits, are somewhat reasonable, waiting for more information from Ariane Space, but not exact. After the conclusion of a commissioning phase, lasting up to 24 hours after injection,

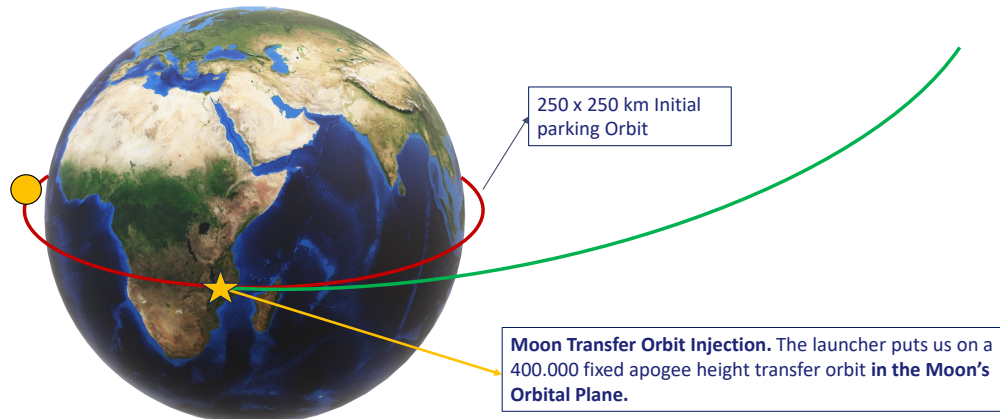


Figure 1.1: Moon Transfer Orbit Injection.

a *TCM-Trajectory Correction Maneuver* must be performed. This is due to the fact that the launcher released the spacecraft on a fixed apogee height orbit that is clearly not precise enough to correctly target the real position of the Moon. The TCM should then correct the Apogee height and target the correct orbital parameters at Moon arrival, to reduce the cost of the following maneuver, the *LOI-Lunar Orbit Injection*. Delaying the TCM further results in higher mission costs, so this 24h period was chosen as a worst case scenario. A more thorough study of the problem would also require addressing a trajectory dispersion problem, accounting for the fact that errors in performing both the injection and TCM maneuvers imply a dispersion of the trajectory and a variation of the entity of the corrective maneuvers. In fact, a single TCM might not be sufficient, and at least another one may prove to be necessary, for example when entering the Moon's sphere of influence. However, this discussion can be postponed to more advanced stages of the mission study, and for now we are going to strictly deal with the mission baseline proposed by ESA.

As mentioned before, after a certain amount of time after the execution of the TCM maneuver, an injection maneuver (LOI) must be performed to enter a circular 100×100 km circular orbit around the Moon. The LOI maneuver is the first critical maneuver of the mission: an error in providing the correct ΔV , both in magnitude and direction, is almost sure to result in a mission loss. The choice of a 100×100 km orbit is not completely arbitrary; from a certain point of view, this is the path followed by other operative programs that effectively reached the Moon (such as the Apollo missions in the past century, and the russian Luna-25 more recently), but there's a more scientific reason too. This circular orbit is in fact believed to be stable enough to allow us to remain waiting around the Moon

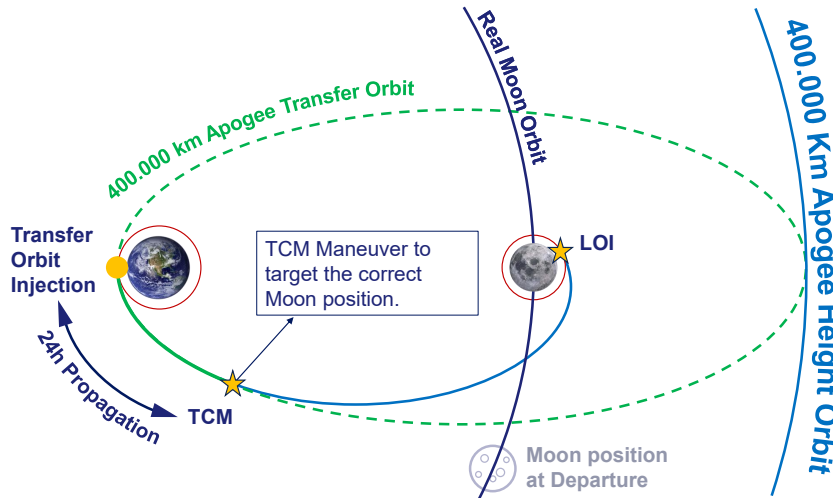


Figure 1.2: Transfer trajectory overview.

for the maximum amount of time required before initiating the subsequent phase, represented by lunar descent. This is due to the fact that a landing is not possible at an arbitrary moment, because:

- the ground track of the waiting orbit must pass over the specified landing site, to reduce the propellant burned to land;
- the landing site must be in light conditions at the time of Moon landing (requirement provided by ESA);

These two conditions may require up to one Moon's orbital period to be both satisfied, and this translates to a waiting time of up to 28 days. At this point, the problem with waiting orbits around the Moon is that, due to the irregularity of the Moon gravitational field, their perilunium height tends to lower with time. This is a huge issue, because it can cause an impact of the lander on the surface. In addition, the LOI maneuver may be performed in multiple steps instead of a single impulse, depending on the amount of thrust at disposal.

Having clarified these aspects, we can now move on to the next maneuver, that is also the *second critical maneuver*: lowering the orbit perilunium from 100 km to 30 km, performing a transfer from the 100×100 km waiting orbit to a 30×100 km one. Again, failing this maneuver could prove fatal: the Russian Lunar-25 impacted on the Moon's surface due to a magnitude error in providing the required ΔV impulse. Finally, upon reaching the perilunium of this 30×100 km orbit, the *Powered Descent* phase, considered the *third and last critical maneuver*, can begin.

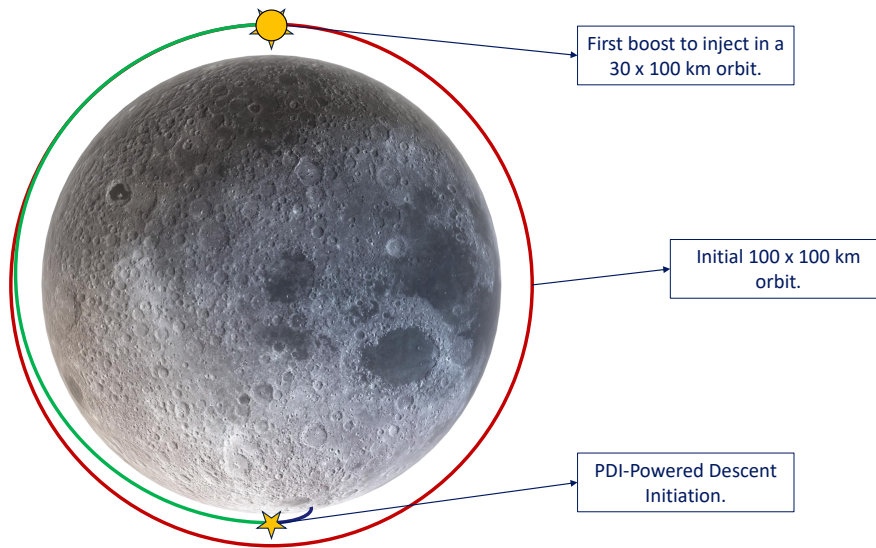


Figure 1.3: 30 x 100 km orbit injection and PDI-Powered Descent Initiation.

1.3 Objectives

Having outlined the mission profile, it is now easier to understand what the goals of this thesis are:

- to compute the magnitude of the TCM maneuver, necessary to target the correct Moon position with the required orbital parameters at arrival (inclination and perilunium), reducing the cost of the subsequent LOI maneuver preventing out of plane changes in velocity;
- to compute the optimum descent trajectory for the *Powered Descent* phase, minimizing the amount of fuel burned (or, analogously, maximizing the payload at landing) and deriving the associated thrust and attitude profiles.

To achieve these two goals, a set of different dynamic models which will be presented during the course of the thesis are necessary, along with optimization methods that are adequate for these purposes. The final output should be a set of tools capable of handling and linking these two problems, with appropriate parameters for the various possible configurations (different landing sites, different landers...). For this reason, a large part of this thesis is devoted to exposing the theoretical aspects related to the development of these tools, while the results for some possible parametric studies are presented only at the end, to validate the software.

Chapter 2

Introduction to Computational Optimal Control

2.1 Introduction

The purpose of this Chapter is to provide the reader with background information on optimal control. Some basic insights into the mathematical setting of the problem will be provided, along with implementation strategies of NLP Problems in commercial optimization software through the so called transcription methods, with particular reference to the MATLAB (Global) Optimization Toolbox. The reader is welcome to consult the references on which this brief introduction is based for more detailed information on Computational Optimal Control, although the basic information here contained should be more than sufficient to successfully implement the problems presented in the following chapters. [2] [17] [3]

2.2 Formulation of Optimal Control Problems

Optimal Control is the process of finding state and control trajectories for a dynamic system in a certain period of time, to optimize system performance in some specified sense. Different measures of system performance can be adopted, such as:

- final mass;
- total $\Delta\vec{V}$;
- TOF - time of flight

In general, the performance index represents a measure of the system control effort, of the employed amount of energy or of the trajectory tracking error with respect to a certain reference path. Clearly, the definition of the performance index isn't always straightforward, so for some problems it is possible that there is a certain grade of arbitrariness.

From a mathematical standpoint, some ingredients are usually present while formulating an optimal control problem:

- a mathematical model of the system to be controlled;
- the specification of a performance index;
- a statement of which variables are free;
- a specification of all the boundary conditions on controls and system state variables.

Starting with the mathematical model of the controlled system, it is usually represented by a set of ordinary differential equations (ODEs), which can be expressed as follows:

$$\dot{\mathbf{x}}(t) = f[\mathbf{x}(t), \mathbf{u}(t), \mathbf{p}, t], t \in [t_0, t_f] \quad (2.2.1)$$

where:

- $\mathbf{x} : [t_0, t_f] \rightarrow \mathbb{R}^{n_x}$ is the **system state vector**;
- $\mathbf{u} : [t_0, t_f] \rightarrow \mathbb{R}^{n_u}$ is the **system control vector**;
- $\mathbf{p} \in \mathbb{R}^{n_p}$ is the **static parameters vector**, i.e. elements not dependent on t ;
- $t \in [t_0, t_f] \in \mathbb{R}$ is an independent variable, usually representing time, so we will refer to it as such from now on;

On the other hand, as for the boundary conditions on system dynamics, they can be imposed on the system initial or final state (or both), or over the whole $[t_0, t_f]$ interval. They can also occur as both inequalities or equalities (the latter correspond to the first ones in case the upper boundaries are equal to the lower boundaries, so in the following writing we are not going to distinguish between them).

From a formal point of view, terminal or initial boundary conditions can be expressed as follows:

$$\mathbf{e}_L \leq \mathbf{e}[\mathbf{x}(t_0), \mathbf{u}(t_0), \mathbf{x}(t_f), \mathbf{u}(t_f), \mathbf{p}, t_0, t_f] \leq \mathbf{e}_U \quad (2.2.2)$$

As previously stated, a system is sometimes also subjected to constraints active in the whole $[t_0, t_f]$ interval. One may for example think about a maximum pitch rate, or even a maximum/minimum thrust level.

In this instance, we then have two cases: the first one, where the constraints are time dependent:

$$\mathbf{h}_L \leq \mathbf{h}[\mathbf{x}(t), \mathbf{u}(t), p, t] \leq \mathbf{h}_U, t \in [t_0, t_f] \quad (2.2.3)$$

and the second one, in which the constraints acting on static, state and control variables are not time dependent:

$$\begin{aligned} \mathbf{u}_L &\leq \mathbf{u}(t) \leq \mathbf{u}_U, t \in [t_0, t_f] \\ \mathbf{x}_L &\leq \mathbf{x}(t) \leq \mathbf{x}_U, t \in [t_0, t_f] \\ \mathbf{p}_L &\leq \mathbf{p} \leq \mathbf{p}_U \end{aligned} \quad (2.2.4)$$

In addition, the initial and final times, respectively t_0 and t_f , are not necessarily fixed:

$$\begin{aligned} t_{0L} &\leq t_0 \leq t_{0U} \\ t_{fL} &\leq t_f \leq t_{fU} \\ t_f - t_0 &\geq 0 \end{aligned} \quad (2.2.5)$$

Lastly, the performance index can be expressed in the following form:

$$J_1 = \phi_1[\mathbf{x}(\mathbf{t}_0), \mathbf{t}_0, \mathbf{x}(\mathbf{t}_f), \mathbf{t}_f, \mathbf{p}] + \int_{t_0}^{t_f} L[\mathbf{x}(\mathbf{t}), \mathbf{u}(\mathbf{t}), \mathbf{p}, \mathbf{t}] dt \quad (2.2.6)$$

where L is often referred to as the *running cost function*.

2.3 Nonlinear Programming and Transcription Methods

Nonlinear programming involves finding the solution of an optimization problem where some of the constraints or the objective function are nonlinear, where *optimizing* means maximizing or minimizing an objective function while satisfying a set of constraints. The NLP problem is often referred to as *parameter optimization*, and the solution is composed by a *finite number of variables*, which make up a so-called **decision vector** $\mathbf{z} \in \mathbb{R}^{n_z}$. In mathematical terms, we can state a NLP Problem as follows:

$$\begin{aligned} \min_{\mathbf{z}} J(\mathbf{z}) & \quad \text{subject to:} \\ & g(\mathbf{z}) \leq 0 \\ & h(\mathbf{z}) = 0 \\ & \mathbf{z}_L \leq \mathbf{z} \leq \mathbf{z}_U \end{aligned} \quad (2.3.1)$$

The main difference is that in a NLP Problem, the decision vector is finite dimensional (not defined on a continuous time interval), the decision variables are numbers and the equations are algebraic, while in an optimum control problem (trajectory optimization problem) we have differential equations and the decision variables (all the unknowns of the differentiation problem) are continuous functions defined on the $[t_0, t_f]$ interval. One could say that the NLP Problem tends to the corresponding Optimum Control Problem as the dimension of the decision vector approaches infinity. Various methods to solve NLP problems are available, such as:

- Gradient-Based Methods: These methods use information about the gradient of the objective function to guide the search for the optimal solution. Examples include the steepest descent method and the Newton-Raphson method.
- Penalty and Barrier Methods: These methods transform nonlinear constrained problems into unconstrained problems by introducing penalty or barrier functions that enforce the constraints.
- Sequential Quadratic Programming (SQP): SQP algorithms solve a sequence of quadratic approximation problems that approximate the original nonlinear programming problem.
- Evolutionary Algorithms, like genetic algorithms and particle swarm optimization, used to solve both constrained and unconstrained optimization problems, are based on a natural selection process that mimics biological evolution.

Their implementation is not part of this thesis, which instead deals only with the transformation of an optimum control problem in an NLP problem, but how can this goal be achieved? Transforming a continuous optimum control problem in a NLP problem can be achieved by *transcribing* the infinite-dimensional problem into a finite-dimensional approximation, through a so called *transcription method*, which has three fundamental steps:

- convert the dynamic system into a problem with a *finite* set of variables;
- solve the finite-dimensional problem (NLP Problem) using, for example, an SQP Algorithm;
- assess the accuracy of the obtained approximation and, if necessary, repeat the above steps;

Two different *transcription methods* will be introduced in this thesis, which will then be applied to the case studies addressed. These are the:

- *Direct Collocation Methods*, also called *Direct Transcription Methods* which will be applied to the optimization of the Descent Trajectory of a Moon Lander. They are based on "approximation", and are better suited for problems with complicated controls and/or path constraints.
- *Direct Shooting Methods*, which will be applied to the optimization of the impulsive TCM burn during Moon Transfer. They are based on "simulation", and are better suited for problems with simple controls and no path constraints.

We can now move on to the description and implementation of these algorithms in a MATLAB environment, using the `fmincon` function to solve the associated NLP problems. Of course the reader may choose to implement these methods in other optimization softwares, such as GPOPS (written in the MATLAB language) or SNOPT (written in FORTRAN, with interfaces available for MATLAB, C, C++ and Python), which is in fact available in several trajectory optimization packages, such as the Astrogator module of Systems Tool Kit - STK and the General Mission Analysis Tool - GMAT.

2.3.1 Direct Collocation Methods

Direct Collocation Methods involve the discretization of the ODEs that describe our system through a piecewise polynomial approximation, thus achieving various different direct collocation methods such as the Trapezoidal method or the Hermite-Simpson method. A set of equalities that represent the system dynamics are then obtained, and they can be implemented as equality constraints in a NLP Problem. The first step consists in dividing the $[t_0, t_f]$ interval in N sub-intervals using $N+1$ grid nodes. It's not necessary for these nodes to be equally spaced, although it may appear as the most logical choice, at least for a preliminary optimization. Adaptive algorithms in which more nodes are automatically placed where necessary, to satisfy accuracy requirements, do exist, but they are not the subject of this discussion.

We can then start by assuming that the state of the ODE system is approximated by a polynomial $\tilde{x}(t)$ of degree M over each interval $t \in [t_k, t_{k+1}]$, $k = 0, \dots, N-1$, which has the following form:

$$\tilde{\mathbf{x}}(t) = \mathbf{a}_0^{(k)} + \mathbf{a}_1^{(k)}(t - t_k) + \dots + \mathbf{a}_M^{(k)}(t - t_k)^M \quad (2.3.2)$$

The coefficients are chosen such that the approximation $\tilde{\mathbf{x}}(t)$ matches the $\mathbf{x}(t)$ function at the two extremes of the $[t_k, t_{k+1}]$ interval:

$$\begin{aligned} \tilde{\mathbf{x}}(t_k) &= \mathbf{x}(t_k) \\ \tilde{\mathbf{x}}(t_{k+1}) &= \mathbf{x}(t_{k+1}) \end{aligned} \quad (2.3.3)$$

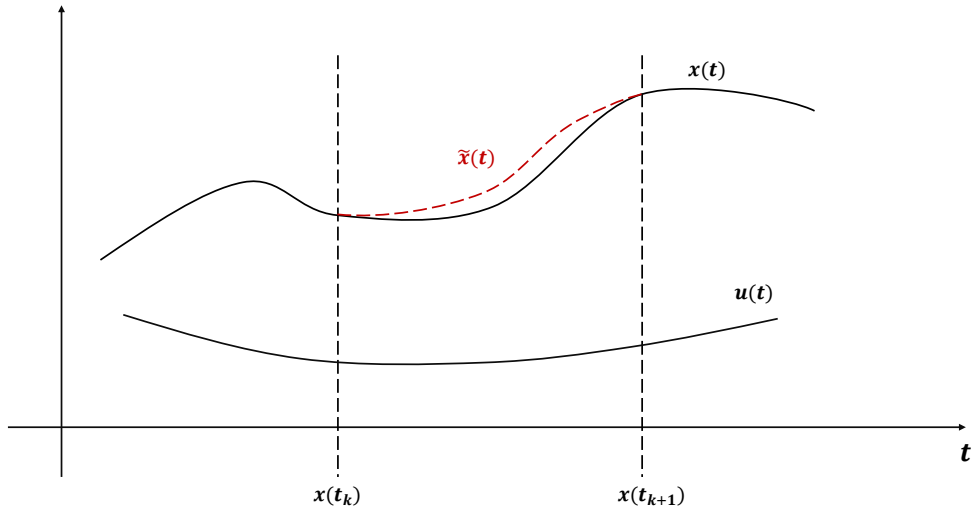


Figure 2.1: Approximation of system state vector through a piecewise polynomial function $\tilde{\mathbf{x}}(t)$ defined on the $t_0, \dots, t_N = t_f$ grid

and the first time derivative of the approximation also matches the $\mathbf{x}(t)$ function at t_k and t_{k+1} :

$$\begin{aligned}\dot{\tilde{\mathbf{x}}}(t_k) &= \dot{\mathbf{x}}(t_k) = f[\mathbf{x}(t_k), \mathbf{u}(t_k), \mathbf{p}, t_k] \\ \dot{\tilde{\mathbf{x}}}(t_{k+1}) &= \dot{\mathbf{x}}(t_{k+1}) = f[\mathbf{x}(t_{k+1}), \mathbf{u}(t_{k+1}), \mathbf{p}, t_{k+1}]\end{aligned}\tag{2.3.4}$$

Only the trapezoidal method will be addressed in this discussion, being the one that has then be implemented in the optimization of the descent trajectory. The reader is encouraged to consult the references listed at the beginning of this Chapter to delve further into the subject.

Trapezoidal Method

The trapezoidal method is based on a quadratic interpolating spline (piece-wise polynomial), written as follows:

$$\tilde{\mathbf{x}}(t) = \mathbf{a}_0^{(k)} + \mathbf{a}_1^{(k)}(t - t_k) + \mathbf{a}_2^{(k)}(t - t_k)^2, \quad t \in [t_k, t_{k+1}], \quad k = 0, \dots, N - 1 \tag{2.3.5}$$

Before proceeding further it's also appropriate to introduce the *notation* used for this section:

- $\mathbf{x}(t_k) = \mathbf{x}_k$
- $\mathbf{x}(t_{k+1}) = \mathbf{x}_{k+1}$

- $f[\mathbf{x}(t_k), \mathbf{u}(t_k), \mathbf{p}, t_k] = \mathbf{f}_k$
- $f[\mathbf{x}(t_{k+1}), \mathbf{u}(t_{k+1}), \mathbf{p}, t_{k+1}] = \mathbf{f}_{k+1}$
- $\mathbf{u}(t_k) = \mathbf{u}_k$

We now start by imposing conditions (2.3.3) and (2.3.4) on (2.3.5). Taking the first time derivative of (2.3.5), we get:

$$\dot{\tilde{\mathbf{x}}}(t) = \mathbf{a}_1^{(k)} + 2\mathbf{a}_2^{(k)}(t - t_k) \quad (2.3.6)$$

Evaluating (2.3.5) at node t_k , we get:

$$\mathbf{x}_k = \mathbf{a}_0^{(k)} = \tilde{\mathbf{x}}_k \quad (2.3.7)$$

while, evaluating Equation (2.3.6) at nodes t_k and t_{k+1} , we obtain:

$$\begin{aligned} \dot{\tilde{\mathbf{x}}}_k &= \mathbf{a}_1^{(k)} = \mathbf{f}_k = \dot{\mathbf{x}}_k \\ \dot{\tilde{\mathbf{x}}}_{k+1} &= \mathbf{a}_1^{(k)} + 2\mathbf{a}_2^{(k)}(t_{k+1} - t_k) = \mathbf{f}_{k+1} = \dot{\mathbf{x}}_{k+1} \rightarrow \\ \rightarrow \mathbf{a}_2^{(k)} &= \frac{\mathbf{f}_{k+1} - \mathbf{f}_k}{2(t_{k+1} - t_k)} \end{aligned} \quad (2.3.8)$$

Substituting (2.3.7) and (2.3.8) in Equation (2.3.5), and evaluating at t_{k+1} , we finally derive that:

$$\tilde{\mathbf{x}}_{k+1} = \tilde{\mathbf{x}}_k + \frac{1}{2}(\mathbf{f}_{k+1} + \mathbf{f}_k)(t_{k+1} - t_k) \quad (2.3.9)$$

Clearly it wouldn't be correct to say that $\tilde{\mathbf{x}}_{k+1} = \mathbf{x}_{k+1}$, thus the approximation. We have in this way obtained $N \cdot n_x$ equality constraints of the following form:

$$\mathbf{c}_k = \tilde{\mathbf{x}}_{k+1} - \tilde{\mathbf{x}}_k - \frac{1}{2}(\mathbf{f}_{k+1} + \mathbf{f}_k)(t_{k+1} - t_k), \mathbf{c}_k \in \mathbb{R}^{n_x}, k = 0, \dots, (N - 1) \quad (2.3.10)$$

which basically contain information on the dynamics of our system.

Lastly, the **decision vector** of the NLP Problem can for example be written as follows:

$$\mathbf{z} = \begin{bmatrix} \mathbf{u}_0 \\ \vdots \\ \mathbf{u}_N \\ \mathbf{x}_0 \\ \vdots \\ \mathbf{x}_N \\ \mathbf{p} \\ t_0 \\ t_f \end{bmatrix} \quad (2.3.11)$$

MATLAB implementation guidelines

A MATLAB implementation of this method shouldn't now be particularly difficult, using the `fmincon` function with the SQP algorithm or other alternatives such as the `ga` function, which implements a genetic algorithm that was however found to be much slower for our cases of interest, so its use is not recommended. The `fmincon` function has very convenient ways to input equality and inequality constraints, including the non-linear ones. Some guidelines are to:

- use the `nonlcon` input of the `fmincon` function to impose the dynamics constraints stated in Equation (2.3.10);
- use the lower and upper boundaries inputs to impose constraints on the initial and final states of the optimized trajectory, and in general avoid using the `nonlcon` input when not necessary;
- both too narrow and too loose constraints are bad for convergence, as well as an excessive number of grid nodes. It is instead better to start the optimization with a reduced number of nodes and without difficult constraints, and to reintroduce them, increasing the number of nodes, only when a basic idea of how an optimum solution should look like is acquired. This should allow us to remove the parts of the domain of the decision vector in which an optimum solution is not expected, enabling convergence even with a higher number of nodes (obtaining a more precise solution). An interpolation of previous solutions with a reduced number of nodes on a higher number of grid nodes, to be used as a *first guess* and facilitate convergence, can also be implemented.

The reader may consult the MATLAB documentation of the `fmincon` function for further details on how to properly use it. [15]

2.3.2 Direct Single Shooting Method

As stated before, the Direct Single Shooting Method is based on "simulation" rather than "approximation". This means that our optimal control problem, with certain *boundary conditions* (BVP-Boundary Value Problem), is reduced to an *initial conditions problem*. We then have to guess the initial conditions that allow us to satisfy the constraints imposed during propagation in the time interval $[t_0, t_f]$, and possibly some final state boundaries, while at the same time optimizing a certain performance index. The name of this method comes from the intuitive idea of *aiming a cannon*: the aim is adjusted until the projectile hits the target. In our case, *aiming* means *adjusting the initial conditions* and *hitting the target* means *satisfying the boundary conditions*. Obviously, there are better and worse ways to hit our target; for example one could maximize the range of the projectile

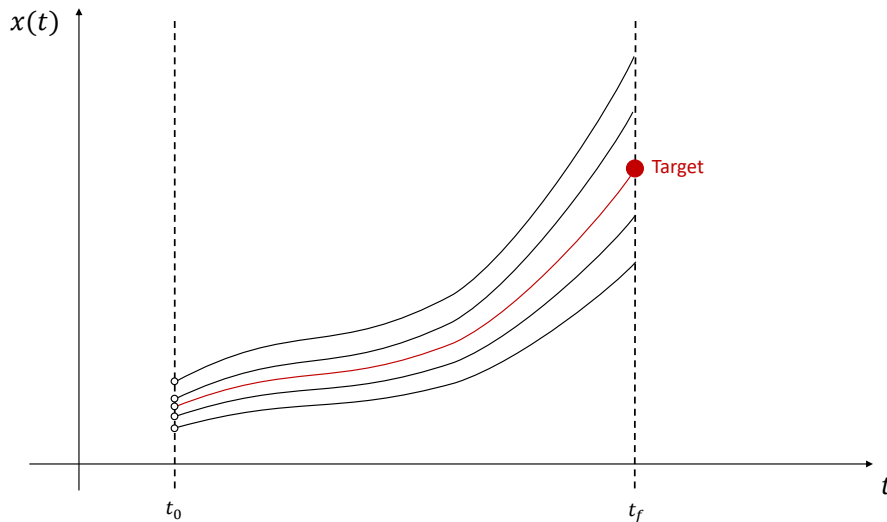


Figure 2.2: Propagation of system trajectory from different initial conditions, to hit a certain target value at time t_f . This idea is at the base of the Direct Single Shooting Method.

by tilting the cannon to a 45-degree angle, or could shoot straight at the target, requiring in this case a much higher initial speed of the projectile. Therefore, the performance-index function comes into play to account for these factors.

From an implementation standpoint, a Direct Single Shooting Algorithm should perform the following actions:

- guess the initial conditions $\mathbf{z} = \mathbf{x}(t_0)$;
- propagate the system state trajectory from t_0 to t_f (i.e shoot the cannon), using a numerical propagator (if an analytical solution is not available);
- check that the cost ($f(\mathbf{z})$) is minimized and the boundary conditions, evaluating the eventual error in meeting the boundary conditions, to adjust the initial conditions (i.e solve a NLP problem).

In reality, the decision vector \mathbf{z} does not necessarily consist only of the initial conditions of our system, but could include other parameters such as *propagation time* ($t_f - t_0$). Erring to account for all the free parameters that could improve the system performance might lead to sub-optimal solutions, or even to having no solution at all. Stating which variables should be included in the decision vector isn't possible a-priori for each problem, and could require a bit of trial and error. This method is very simple to implement and understand, but it comes with some drawbacks; one of them is that a small change in the initial conditions can cause big changes in the final ones. This is in fact really evident in space

trajectory optimization problems, where a difference of a couple m/s in the initial impulse can cause the S/C to completely miss the target, and it's also the reason for which, in fact, TCM-Trajectory Correction Maneuvers are necessary, from a practical standpoint (leaving aside the Direct Shooting Algorithms), to adjust the S/C Trajectory during transfer. This technique then requires a good initial guess of the solution (for example a two-body theory transfer could be used to initialize the problem), and it can be coupled with an uncertainty analysis to compute the worst case required burn. This matter will be briefly reprised in Chapter 6.

MATLAB implementation guidelines

Even the Direct Single Shooting Algorithm was implemented in MATLAB, using the `fmincon` function with the SQP algorithm. Some general guidelines, based on the experience gained, are to:

- propagate the trajectory in the `nonlcon` function, using the parameters (e.g initial speed, Time Of Flight...) included in the decision vector. The final system state vector can be then further elaborated to impose conditions reached at the end of the propagation (such as orbit inclination, speed, flight path angle...);
- start without difficult constraints, and reintroduce them while gradually narrowing the domain of the decision vector, on the basis of the results obtained in the previous optimization runs.

Global Optimization

One final important point is that the SQP algorithm implemented in MATLAB isn't a Global Optimization Algorithm. In fact, true Global Optimization Algorithms that allow us to always find the absolute-best solution in any case do not exist. What do exist are *globalization strategies*, and the one used in this thesis is the `multistart` function, part of the MATLAB Global Optimization Toolbox. The principle is pretty simple: the initial guess is changed for a certain number of optimization runs, and the best solution found among all of these runs is given as output. Theoretically, the greater the number of optimization runs (up to a reasonable point), the better the solution that is found; however, even in this case, nothing assures us that the found solution is the global optimum, but it will likely be better than the one found from a single run. In addition, one might be mistakenly led to think that if the algorithm always converges to the same solution, then that solution is the global optimum, but this isn't always the case. The shape of the performance function could be such that the algorithm is more likely to converge to that solution, even if it isn't the best one in the domain.

Chapter 3

Descent Trajectory Dynamics

3.1 Introduction

In this Chapter, both the derivation of the equations of motion for the descent trajectory and the implementation of the Direct Collocation Transcription Method will be introduced. These equations are based on a model widely described in Avanzini's *Entry, Descent, Landing and Ascent* SEEDS Course Notes [10], which, however, have a problem: a singularity occurs at Latitudes of $\pm 90^\circ$; but the Moon South Pole is of interest for the Argonaut Program. A procedure to "move" this singularity elsewhere, through an appropriate reference frame rotation, is then introduced. The complete equations, useful to describe even an atmospheric reentry problem in the hypothesis of a spherical body with a body-fixed atmosphere and an inertially fixed rotation axis, will be obtained, before removing the atmospheric terms to appropriately model a Moon landing case.

Nomenclature

First, it's necessary to introduce the notation used in this chapter, to avoid confusion:

\mathcal{F}_P	Celestial Body-Centered Reference Frame (pseudo-inertial, non-rotating)
\mathcal{F}_R	Celestial Body-Fixed Reference Frame (rotating).
\mathcal{F}_T	LVLH Reference Frame (Local Vertical Local Horizontal).

\mathbf{r}_P	Position Vector in the \mathcal{F}_P Reference Frame (or, alternatively, in \mathcal{F}_R or \mathcal{F}_T , depending on the subscript).
$\dot{\mathbf{r}}_P$	\mathcal{F}_P (or, alternatively, \mathcal{F}_R or \mathcal{F}_T) Reference Frame relative speed vector. In more general terms, this should be interpreted as a <i>fixed-axes</i> derivative of a vector (in this case the position one) in the reference frame specified by the subscript.
$\ddot{\mathbf{r}}_P$	\mathcal{F}_P (or, alternatively, \mathcal{F}_R or \mathcal{F}_T) Reference Frame relative acceleration vector. In more general terms, this should be interpreted as a <i>fixed-axes</i> second derivative of a vector (in this case the position one) in the reference frame specified by the subscript.
T	Thrust Magnitude.
ϵ	Angle between Thrust and Speed vectors.
σ	Bank Angle (angle between local vertical plane and vehicle plane of symmetry).
v	\mathcal{F}_R Reference Frame relative speed magnitude ($v = \dot{\mathbf{r}}_R $)
γ	Angle Between $\dot{\mathbf{r}}_R$ and local horizontal direction, measured on a vertical plane. See Figure (3.1).
ψ	Modified Heading Angle (angle between the horizontal projection of $\dot{\mathbf{r}}_R$ and the local pseudo-north direction). See Figure (3.1).
r	Vehicle Distance from Celestial Body Center.
λ	Modified Longitude. See Figure (3.1).
ϕ	Modified Latitude. See Figure (3.1).
m	Vehicle mass.
\mathbf{L}	Lift vector.
\mathbf{D}	Drag vector.
\mathbf{g}	Gravitational Acceleration vector.
I_{sp}	Specific Impulse.

$\{\hat{i}_T, \hat{j}_T, \hat{k}_T\}$	Triad of right-handed axes of the \mathcal{F}_T reference frame (or alternatively \mathcal{F}_R , depending on the subscript).
$[f_i, f_j, f_k]^T$	Sum of forces vector in the \mathcal{F}_T Reference Frame
L_{TR}	Rotation matrix from \mathcal{F}_R to \mathcal{F}_T (or between other reference frames depending on the subscript).
ω_{RT}	Angular Speed vector of \mathcal{F}_T with respect to \mathcal{F}_R (or between other frames, depending on the subscript).

3.2 Reference Frames

Three reference frames will be used in deriving the dynamics equation for the optimization of the descent trajectory:

- the **\mathcal{F}_R Reference Frame**, which is a rotating reference frame with the \hat{k}_R axis normally aligned with the celestial body angular speed, or rather this is the approach used in Reference [10]. For the sake of our analysis, to prevent singularities from appearing at the North and South poles, it's instead better to align the \hat{j}_R axis with the celestial body *angular speed vector* that, as already stated, is supposed to be *inertially fixed*;
- the **\mathcal{F}_T Reference Frame**, also called LVLH Reference Frame, in which the \hat{i}_T axis is aligned with the local Vertical direction (tangent to the local parallel), the \hat{j}_T is aligned with the local East direction and \hat{k}_T is aligned with the local North direction (tangent to the local meridian). The triad is obviously right handed;
- the **\mathcal{F}_P Reference Frame**, which is a Celestial-Body Centered Pseudo-Inertial (non-rotating) Reference Frame. This frame is used to correctly apply Newton's second principle of dynamics, but no vector will be expressed in this reference frame at any stage of the derivation procedure of our dynamics equations.

The reader is also invited to carefully observe Figure (3.1) to better understand the rotation matrices that will be introduced in the next sections. Notice the visualization of the ψ and γ angles, and in particular that of the λ and ϕ angles: they don't represent latitude and longitude as commonly intended (they are somewhat inverted; for example λ , which is the longitude, increases by moving in the North direction). Clearly, this isn't very practical for a software implementation in which longitude and latitude of a certain landing site are specified in the classical sense,

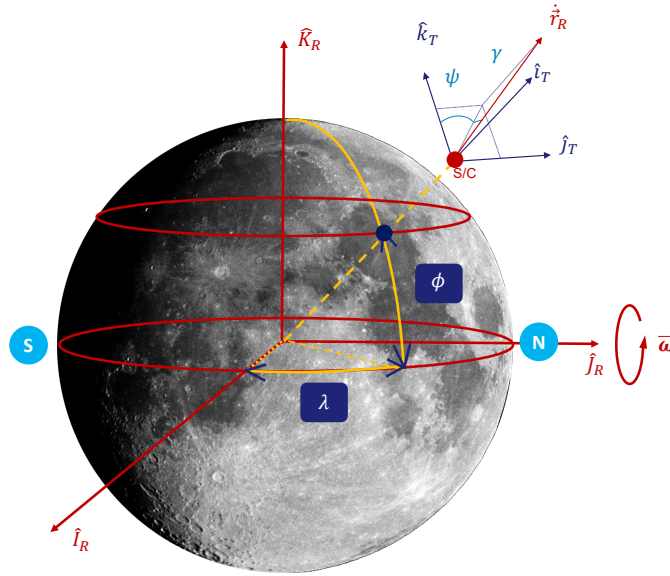


Figure 3.1: Orientation of the \mathcal{F}_R and \mathcal{F}_T Reference Frames with respect to the celestial body of interest.

so a procedure to pass from one coordinate system to the other will be presented in Section (3.5).

3.3 The Equations of Motion

3.3.1 The kinematics equations

To correctly obtain the final system of ODEs that describe the dynamics of our vehicle during descent, the first step is to introduce the necessary rotation matrices and derive the *three kinematics equations*.

To pass from the \mathcal{F}_R to the \mathcal{F}_T reference frame, the L_{TR} rotation matrix is needed, performing a $2 \rightarrow 1$ rotation; this means that the the \mathcal{F}_T reference frame is first rotated by a ϕ angle around the second axis of the right-handed triad (\hat{j}_T), and then by a $-\lambda$ angle around the third axis of the right-handed triad of the intermediate reference frame, obtained from the previous rotation. The plus or minus sign depends on the direction of rotation; a rotation is considered positive when it is counterclockwise if observed from the tip of the axis around which the rotation is happening. Since rotation matrices will be used extensively from now on, in case a review is needed the reader is invited to consult Appendix B.

The rotation matrix of interest is then expressed as:

$$\begin{aligned}
 L_{TR} &= \begin{bmatrix} \cos \phi & 0 & \sin \phi \\ 0 & 1 & 0 \\ -\sin \phi & 0 & \cos \phi \end{bmatrix} \cdot \begin{bmatrix} \cos \lambda & \sin \lambda & 0 \\ -\sin \lambda & \cos \lambda & 0 \\ 0 & 0 & 1 \end{bmatrix} = \\
 &= \begin{bmatrix} \cos \phi \cos \lambda & \cos \phi \sin \lambda & \sin \phi \\ -\sin \lambda & \cos \lambda & 0 \\ -\sin \phi \cos \lambda & -\sin \phi \sin \lambda & \cos \phi \end{bmatrix}
 \end{aligned} \tag{3.3.1}$$

In addition, the following relations are valid:

$$\begin{aligned}
 \mathbf{r}_T &= r \hat{i}_T \\
 \boldsymbol{\omega}_{RT} &= \dot{\lambda} \hat{k}_R - \dot{\phi} \hat{j}_T
 \end{aligned}$$

where the second one stems from the *theorem of composition* of angular velocities. [9]

At this point, using the Poisson Theorem, the vehicle speed relative to the \mathcal{F}_R frame can be expressed as follows:

$$\dot{\mathbf{r}}_R = \dot{\mathbf{r}}_T + \boldsymbol{\omega}_{RT} \times \mathbf{r}_T \tag{3.3.2}$$

This is an implicit vector relation. To explicit the equation in matrix form, every vector should be expressed in the same Reference Frame; hence, using the L_{TR} rotation matrix, we can write $\boldsymbol{\omega}_{RT}$ in the \mathcal{F}_T frame as:

$$\boldsymbol{\omega}_{RT} = \begin{bmatrix} 0 \\ -\dot{\phi} \\ 0 \end{bmatrix} + L_{TR} \begin{bmatrix} 0 \\ 0 \\ \dot{\lambda} \end{bmatrix} = \begin{bmatrix} \dot{\lambda} \sin \phi \\ -\dot{\phi} \\ \dot{\lambda} \cos \phi \end{bmatrix} \tag{3.3.3}$$

Expressing now everything in the \mathcal{F}_T frame, Equation (3.3.2) becomes:

$$\dot{\mathbf{r}}_R = \begin{bmatrix} \dot{r} \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & -\dot{\lambda} \cos \phi & -\dot{\phi} \\ \dot{\lambda} \cos \phi & 0 & -\dot{\lambda} \sin \phi \\ \dot{\phi} & \dot{\lambda} \sin \phi & 0 \end{bmatrix} \cdot \begin{bmatrix} r \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \dot{r} \\ r \dot{\lambda} \cos \phi \\ r \dot{\phi} \end{bmatrix} \tag{3.3.4}$$

However, by looking at Figure (3.2), we can also state, again in the same reference system \mathcal{F}_T , that:

$$\dot{\mathbf{r}}_R = \begin{bmatrix} v \sin \gamma \\ v \cos \gamma \sin \psi \\ v \cos \gamma \cos \psi \end{bmatrix} \tag{3.3.5}$$

where we remind that $v = |\dot{\mathbf{r}}_{\mathbf{R}}|$. This can be rapidly understood again by observing Figure (3.1). At this point, equating Equation (3.3.4) and Equation (3.3.5), we obtain the *three kinematics equations*

$$\begin{bmatrix} \dot{r} \\ r\dot{\lambda} \cos \phi \\ r\dot{\phi} \end{bmatrix} = \begin{bmatrix} v \sin \gamma \\ v \cos \gamma \sin \psi \\ v \cos \gamma \cos \psi \end{bmatrix} \rightarrow \begin{cases} \dot{r} = v \sin \gamma \\ \dot{\lambda} = \frac{v \cos \gamma \sin \psi}{r \cos \phi} \\ \dot{\phi} = \frac{v}{r} \cos \gamma \cos \psi \end{cases} \quad (3.3.6)$$

Nonetheless, to completely describe our problem, it's now necessary to introduce the dynamics.

3.3.2 The dynamics equations

Newton's second principle of dynamics states that, in an inertial reference frame:

$$m\ddot{\mathbf{r}} = \mathbf{F} - \dot{m}\dot{\mathbf{r}} \quad (3.3.7)$$

In our case, the (pseudo)inertial reference frame is represented by \mathcal{F}_P , so for vectors expressed in other reference frames, additional terms due to the fact that they are rotating (non-inertial) appear. The following procedure can then be applied to correctly use Equation (3.3.7); we start by expressing the inertial speed $\dot{\mathbf{r}}_P$ as follows, by applying the Poisson theorem:

$$\dot{\mathbf{r}}_P = \dot{\mathbf{r}}_{\mathbf{R}} + \boldsymbol{\omega}_{PR} \times \mathbf{r}_R \quad (3.3.8)$$

if we now take the first derivative of Equation (3.3.8), we obtain:

$$\begin{aligned} \ddot{\mathbf{r}}_P &= \ddot{\mathbf{r}}_{\mathbf{R}} + \dot{\boldsymbol{\omega}}_{PR} \times \mathbf{r}_R + \boldsymbol{\omega}_{PR} \times \dot{\mathbf{r}}_{\mathbf{R}} + \boldsymbol{\omega}_{PR} \times \dot{\mathbf{r}}_{\mathbf{R}} + \boldsymbol{\omega}_{PR} \times (\boldsymbol{\omega}_{PR} \times \mathbf{r}_R) = \\ &= \ddot{\mathbf{r}}_{\mathbf{R}} + \dot{\boldsymbol{\omega}}_{PR} \times \mathbf{r}_R + 2 \boldsymbol{\omega}_{PR} \times \dot{\mathbf{r}}_{\mathbf{R}} + \boldsymbol{\omega}_{PR} \times (\boldsymbol{\omega}_{PR} \times \mathbf{r}_R) \end{aligned} \quad (3.3.9)$$

We can then define the following vector:

$$\mathbf{w}_T = \dot{\mathbf{r}}_{\mathbf{T}} + \boldsymbol{\omega}_{RT} \times \mathbf{r}_T = \dot{\mathbf{r}}_{\mathbf{R}} \quad (3.3.10)$$

and, in accordance with what was said in Section 3.1, we can perform an \mathcal{F}_T fixed-axes time derivative of Equation (3.3.10) and state that:

$$\dot{\mathbf{w}}_T = \ddot{\mathbf{r}}_T + \dot{\boldsymbol{\omega}}_{RT} \times \mathbf{r}_T + \boldsymbol{\omega}_{RT} \times \dot{\mathbf{r}}_T \quad (3.3.11)$$

but we also know, taking the first derivative of Equation (3.3.2) and applying the Poisson Theorem, that:

$$\begin{aligned}\ddot{\mathbf{r}}_R &= \ddot{\mathbf{r}}_T + \boldsymbol{\omega}_{RT} \times \dot{\mathbf{r}}_T + \dot{\boldsymbol{\omega}}_{RT} \times \mathbf{r}_T + \boldsymbol{\omega}_{RT} \times \dot{\mathbf{r}}_T + \boldsymbol{\omega}_{RT} \times (\boldsymbol{\omega}_{RT} \times \mathbf{r}_T) \\ &= \ddot{\mathbf{r}}_T + 2 \boldsymbol{\omega}_{RT} \times \dot{\mathbf{r}}_T + \dot{\boldsymbol{\omega}}_{RT} \times \mathbf{r}_T + \boldsymbol{\omega}_{RT} \times (\boldsymbol{\omega}_{RT} \times \mathbf{r}_T) \\ &= \dot{\mathbf{w}}_T + \boldsymbol{\omega}_{RT} \times \mathbf{w}_T\end{aligned}\quad (3.3.12)$$

At this point, using Equations (3.3.8), (3.3.9),(3.3.10),(3.3.11) and (3.3.12), *assuming that $\boldsymbol{\omega}_{PR} = \text{cost}$* , we can correctly apply the second principle of dynamics and state:

$$\dot{\mathbf{w}}_T + \boldsymbol{\omega}_{RT} \times \mathbf{w}_T + 2 \boldsymbol{\omega}_{PR} \times \mathbf{w}_T + \boldsymbol{\omega}_{PR} \times (\boldsymbol{\omega}_{PR} \times \mathbf{r}_T) = \frac{\mathbf{F}}{m} - \frac{\dot{m}}{m}(\mathbf{w}_T + \boldsymbol{\omega}_{PR} \times \mathbf{r}_T)\quad (3.3.13)$$

This expression implicitly contains the three *dynamics equations* that we are looking for. The first step in correctly deriving them is to express every term in the \mathcal{F}_T frame; therefore, proceeding on this path, we can start by expressing the $\boldsymbol{\omega}_{PR}$ vector in the \mathcal{F}_T frame. We remember that, as stated in the Introduction, we want to prevent singularities at poles; to achieve this goal, we consider the \mathcal{F}_R frame to be rotating around its \hat{j}_R axis with respect to the \mathcal{F}_P frame. We can then state that, in the \mathcal{F}_T frame, the $\boldsymbol{\omega}_{PR}$ vector can be expressed as:

$$\boldsymbol{\omega}_{PR} = \begin{bmatrix} \cos \phi \cos \lambda & \cos \phi \sin \lambda & \sin \phi \\ -\sin \lambda & \cos \lambda & 0 \\ -\sin \phi \cos \lambda & -\sin \phi \sin \lambda & \cos \phi \end{bmatrix} \begin{bmatrix} 0 \\ \omega \\ 0 \end{bmatrix} = \begin{bmatrix} \omega \cos \phi \sin \lambda \\ \omega \cos \lambda \\ -\omega \sin \phi \sin \lambda \end{bmatrix}\quad (3.3.14)$$

Then, considering that:

$$\mathbf{w}_T = \begin{bmatrix} v \sin \gamma \\ v \cos \gamma \sin \psi \\ v \cos \gamma \cos \psi \end{bmatrix}\quad (3.3.15)$$

we easily obtain that:

$$\dot{\mathbf{w}}_T = \begin{bmatrix} \dot{v} \sin \gamma + v \cos \gamma \dot{\gamma} \\ \dot{v} \cos \gamma \sin \psi - v \sin \gamma \sin \psi \dot{\gamma} + v \cos \gamma \cos \psi \dot{\psi} \\ \dot{v} \cos \gamma \cos \psi - v \sin \gamma \cos \psi \dot{\gamma} - v \cos \gamma \sin \psi \dot{\psi} \end{bmatrix}\quad (3.3.16)$$

We now have all the necessary pieces, and only a few algebraic calculations remain:

$$\begin{aligned}
\boldsymbol{\omega}_{RT} \times \mathbf{w}_T &= \begin{bmatrix} 0 & -\cos \phi \dot{\lambda} & -\dot{\phi} \\ \cos \phi \dot{\lambda} & 0 & -\sin \phi \dot{\lambda} \\ \dot{\phi} & \sin \phi \dot{\lambda} & 0 \end{bmatrix} \begin{bmatrix} v \sin \gamma \\ v \cos \gamma \sin \psi \\ v \cos \gamma \cos \psi \end{bmatrix} = \\
&= \begin{bmatrix} -v \dot{\lambda} \cos \phi \cos \gamma \sin \psi - v \dot{\phi} \cos \gamma \cos \psi \\ v \dot{\lambda} \sin \gamma \cos \phi - v \dot{\lambda} \cos \gamma \cos \psi \sin \phi \\ v \dot{\phi} \sin \gamma + v \dot{\lambda} \cos \gamma \sin \psi \sin \phi \end{bmatrix} \quad (3.3.17)
\end{aligned}$$

In addition, recalling Equations (3.3.6) and plugging them in (3.3.17), we derive:

$$\boldsymbol{\omega}_{RT} \times \mathbf{w}_T = \begin{bmatrix} -\frac{v^2}{r} \cos^2 \gamma \sin^2 \psi - \frac{v^2}{r} \cos^2 \gamma \cos^2 \psi \\ \frac{v^2}{r} \sin \gamma \cos \gamma \sin \psi - \frac{v^2}{r} \cos^2 \gamma \cos \psi \tan \phi \sin \psi \\ \frac{v^2}{r} \cos \gamma \sin \gamma \cos \psi + \frac{v^2}{r} \cos^2 \gamma \sin^2 \psi \tan \phi \end{bmatrix} \quad (3.3.18)$$

Therefore, computing the remaining terms:

$$\begin{aligned}
2\boldsymbol{\omega}_{PR} \times \mathbf{w}_T &= 2 \begin{bmatrix} 0 & \omega \sin \phi \sin \lambda & \omega \cos \lambda \\ -\omega \sin \phi \sin \lambda & 0 & -\omega \cos \phi \sin \lambda \\ -\omega \cos \lambda & \omega \cos \phi \sin \lambda & 0 \end{bmatrix} \begin{bmatrix} v \sin \gamma \\ v \cos \gamma \sin \psi \\ v \cos \gamma \cos \psi \end{bmatrix} = \\
&= \begin{bmatrix} 2\omega v \sin \phi \sin \lambda \cos \gamma \sin \psi + 2\omega v \cos \gamma \cos \lambda \cos \psi \\ -2\omega v \sin \phi \sin \lambda \sin \gamma - 2\omega v \cos \phi \sin \lambda \cos \gamma \cos \psi \\ -2\omega v \sin \gamma \cos \lambda + 2\omega v \cos \phi \sin \lambda \cos \gamma \sin \psi \end{bmatrix} \quad (3.3.19)
\end{aligned}$$

$$\begin{aligned}
\boldsymbol{\omega}_{PR} \times \mathbf{r}_T &= \begin{bmatrix} 0 & \omega \sin \phi \sin \lambda & \omega \cos \lambda \\ -\omega \sin \phi \sin \lambda & 0 & -\omega \cos \phi \sin \lambda \\ -\omega \cos \lambda & \omega \cos \phi \sin \lambda & 0 \end{bmatrix} \begin{bmatrix} r \\ 0 \\ 0 \end{bmatrix} = \\
&= \begin{bmatrix} 0 \\ -\omega r \sin \phi \sin \lambda \\ -\omega r \cos \lambda \end{bmatrix} \quad (3.3.20)
\end{aligned}$$

$$\begin{aligned}
\boldsymbol{\omega}_{PR} \times (\boldsymbol{\omega}_{PR} \times \mathbf{r}_T) &= \begin{bmatrix} 0 & \omega \sin \phi \sin \lambda & \omega \cos \lambda \\ -\omega \sin \phi \sin \lambda & 0 & -\omega \cos \phi \sin \lambda \\ -\omega \cos \lambda & \omega \cos \phi \sin \lambda & 0 \end{bmatrix} \begin{bmatrix} 0 \\ -\omega r \sin \phi \sin \lambda \\ -\omega r \cos \lambda \end{bmatrix} = \\
&= \begin{bmatrix} -\omega^2 r \sin^2 \phi \sin^2 \lambda - \omega^2 r \cos^2 \lambda \\ \omega^2 r \cos \phi \sin \lambda \cos \lambda \\ -\omega^2 r \cos \phi \sin \phi \sin^2 \lambda \end{bmatrix} \quad (3.3.21)
\end{aligned}$$

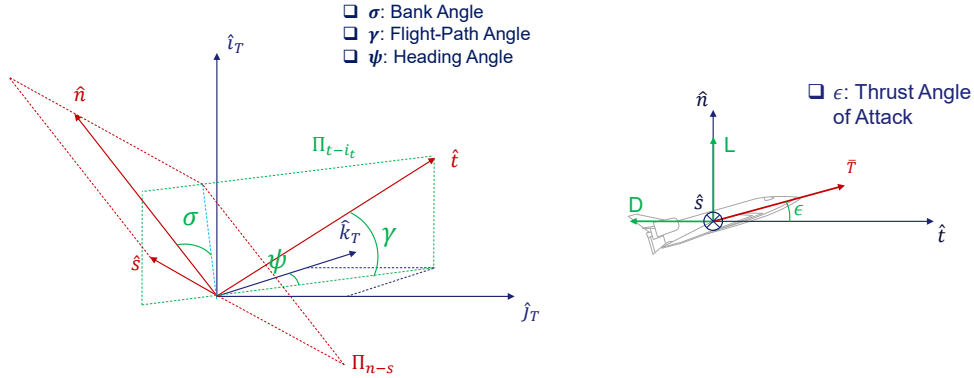


Figure 3.2: Orientation of the \mathcal{F}_S frame with respect to the \mathcal{F}_R frame and forces in the \mathcal{F}_S frame.

This addresses all of the terms in Equation (3.3.13), except for the forces vector \mathbf{F} . Assuming that we are in a symmetrical flight situation, with no slip-angle (i.e. the speed vector is contained in the vehicle plane of symmetry), we will have to take into account four forces:

- Thrust \mathbf{T} .
- Lift \mathbf{L} .
- Drag \mathbf{D} .
- Gravitational Force $m\mathbf{g}$.

The sum of these forces must be correctly written in the \mathcal{F}_T reference frame, so we can start by expressing Lift, Thrust and Drag in an auxiliary reference frame, called \mathcal{F}_S , in which:

- \hat{t} is the $\hat{\mathbf{r}}_R$ direction.
- \hat{s} is defined such that the Π_{t-n} plane is a *plane of symmetry of the vehicle*.
- $\hat{n} = \hat{t} \times \hat{s}$.

while the gravitational component is more easily added directly at the end, because in the \mathcal{F}_T frame $m\mathbf{g} = -(mg)\hat{i}_T$.

By looking at Figure (3.2), we can easily obtain the forces vector in the \mathcal{F}_S frame:

$$\begin{cases} f_t = T \cos \epsilon - D \\ f_s = 0 \\ f_n = T \sin \epsilon + L \end{cases} \quad (3.3.22)$$

while, as for the L_{TS} rotation matrix, it's possible to adopt the $1 \rightarrow 2 \rightarrow 3$ rotation sequence described in Figure (3.3):

- first, we rotate the \mathcal{F}_S frame around \hat{t} by an angle σ (bank angle), obtaining the $\mathcal{F}_{S'}$ frame;
- then we rotate the $\mathcal{F}_{S'}$ frame by a $-(\frac{\pi}{2} - \gamma)$ angle around the \hat{s}' axis obtaining the $\mathcal{F}_{S''}$ frame;
- finally, we rotate $\mathcal{F}_{S''}$ around its \hat{t}'' axis by a $-(\pi - \psi)$ angle obtaining the $\mathcal{F}_{S'''}$ frame, which is aligned with \mathcal{F}_T .

The rotation matrix is thus written as follows:

$$\begin{aligned} L_{TS} &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \sigma & -\sin \sigma \\ 0 & \sin \sigma & \cos \sigma \end{bmatrix} \cdot \begin{bmatrix} \cos(\frac{\pi}{2} - \gamma) & 0 & -\sin(\frac{\pi}{2} - \gamma) \\ 0 & 1 & 0 \\ \sin(\frac{\pi}{2} - \gamma) & 0 & \cos(\frac{\pi}{2} - \gamma) \end{bmatrix} \\ &\cdot \begin{bmatrix} \cos(\pi - \psi) & \sin(\pi - \psi) & 0 \\ -\sin(\pi - \psi) & \cos(\pi - \psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} = \\ &= \begin{bmatrix} s\gamma & -s\sigma c\gamma & c\sigma c\gamma \\ c\gamma s\psi & s\sigma s\gamma s\psi - c\sigma c\psi & -c\psi s\sigma - c\sigma s\gamma s\psi \\ c\gamma c\psi & s\sigma s\gamma c\psi + c\sigma s\psi & s\sigma s\psi - c\sigma s\gamma c\psi \end{bmatrix} \end{aligned} \quad (3.3.23)$$

We are then able to finally express \mathbf{F} in the \mathcal{F}_T frame:

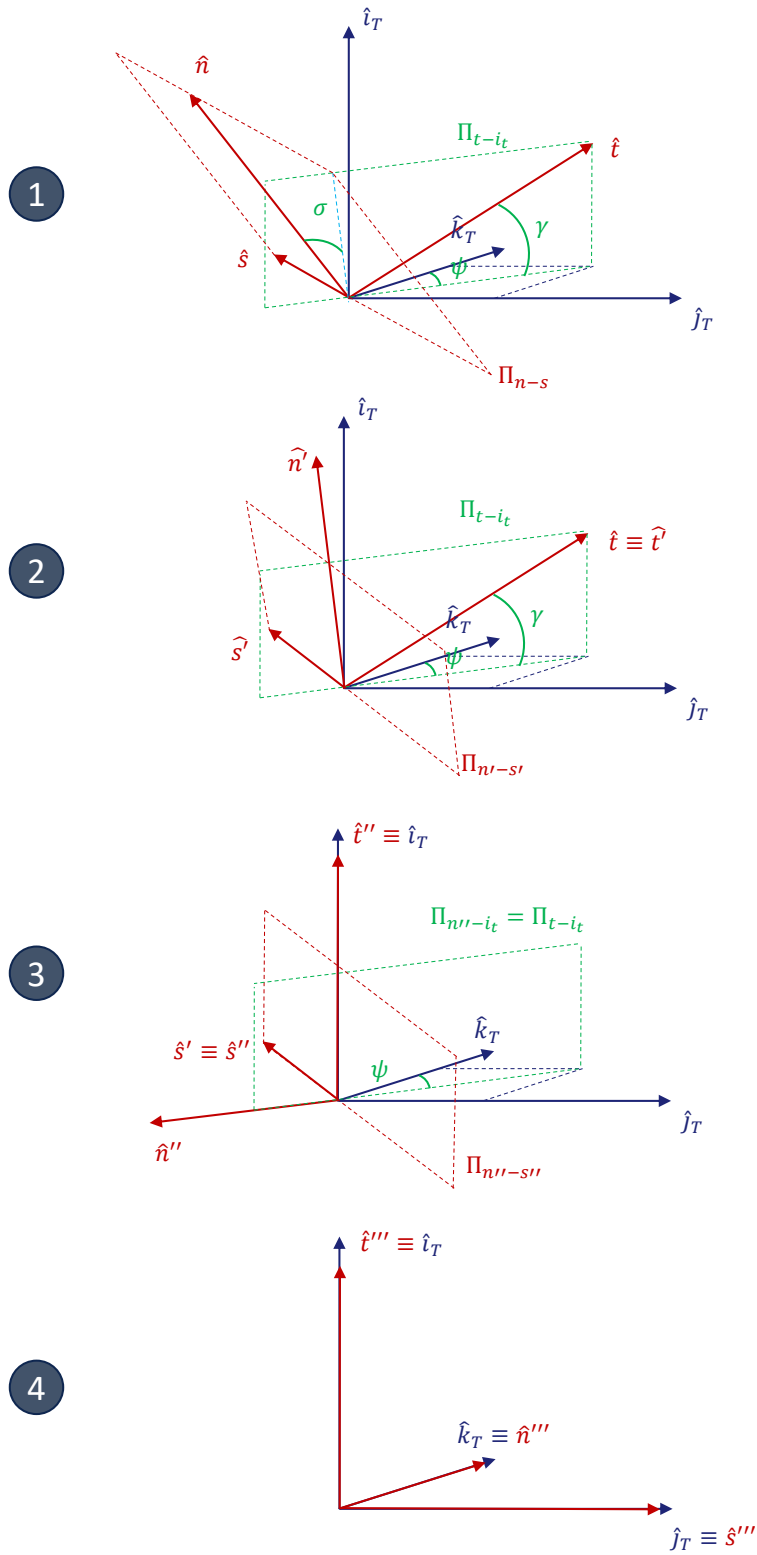


Figure 3.3: 1-2-3 Rotation from \mathcal{F}_S to \mathcal{F}_T

$$\begin{aligned}
\mathbf{F} &= \\
&= \begin{bmatrix} s\gamma & -s\sigma c\gamma & c\sigma c\gamma \\ c\gamma s\psi & s\sigma s\gamma s\psi - c\sigma c\psi & -c\psi s\sigma - c\sigma s\gamma s\psi \\ c\gamma c\psi & s\sigma s\gamma c\psi + c\sigma s\psi & s\sigma s\psi - c\sigma s\gamma c\psi \end{bmatrix} \begin{bmatrix} T \cos \epsilon - D \\ 0 \\ T \sin \epsilon + L \end{bmatrix} + \\
&+ \begin{bmatrix} -mg \\ 0 \\ 0 \end{bmatrix} = \\
&= \begin{bmatrix} (T \cos \epsilon - D) \sin \gamma + (T \sin \epsilon + L) \cos \sigma \cos \gamma - mg \\ (T \cos \epsilon - D) \cos \gamma \sin \psi - (T \sin \epsilon + L)(\cos \psi \sin \sigma + \cos \sigma \sin \gamma \sin \psi) \\ (T \cos \epsilon - D) \cos \gamma \cos \psi + (T \sin \epsilon + L)(\sin \sigma \sin \psi - \cos \sigma \sin \gamma \cos \psi) \end{bmatrix} \\
&= \begin{bmatrix} f_i \\ f_j \\ f_k \end{bmatrix}
\end{aligned} \tag{3.3.24}$$

Inserting now everything within Equation (3.3.13), we obtain:

$$\begin{aligned}
\textcircled{1} \dot{v} \sin \gamma + v \cos \dot{\gamma} - \frac{v^2}{r} \cos^2 \gamma \sin^2 \psi - \frac{v^2}{r} \cos^2 \gamma \cos^2 \psi + 2\omega v \sin \phi \sin \lambda \cos \gamma \sin \psi + \\
+ 2\omega v \cos \gamma \cos \psi \cos \lambda - \omega^2 r \sin^2 \phi \sin^2 \lambda - \omega^2 r \cos^2 \lambda = \frac{f_i}{m} - \frac{\dot{m}}{m} v \sin \gamma
\end{aligned} \tag{3.3.25}$$

$$\begin{aligned}
\textcircled{2} \dot{v} \cos \gamma \sin \psi - v \sin \gamma \sin \psi \dot{\gamma} + v \cos \gamma \cos \psi \dot{\psi} + \frac{v^2}{r} \sin \gamma \cos \gamma \sin \psi - \\
- \frac{v^2}{r} \cos^2 \gamma \cos \psi \tan \phi \sin \psi - 2\omega v \sin \phi \sin \lambda \sin \gamma - 2\omega v \cos \phi \sin \lambda \cos \gamma \cos \psi + \\
+ \omega^2 r \cos \phi \sin \lambda \cos \lambda = \frac{f_j}{m} - \frac{\dot{m}}{m} (v \cos \gamma \sin \psi - \omega r \sin \phi \sin \lambda)
\end{aligned} \tag{3.3.26}$$

$$\begin{aligned}
\textcircled{3} \dot{v} \cos \gamma \cos \psi - v \sin \gamma \cos \psi \dot{\gamma} - v \cos \gamma \sin \psi \dot{\psi} + \frac{v^2}{r} \cos \gamma \sin \gamma \cos \psi + \\
+ \frac{v^2}{r} \cos^2 \gamma \sin^2 \psi \tan \phi - 2\omega v \sin \gamma \cos \lambda + 2\omega v \cos \phi \sin \lambda \cos \gamma \sin \psi - \\
- \omega^2 r \cos \phi \sin \phi \sin^2 \lambda = \frac{f_k}{m} - \frac{\dot{m}}{m} (v \cos \gamma \cos \psi - \omega r \cos \lambda)
\end{aligned} \tag{3.3.27}$$

These three equations must be properly recombined to obtain the three *dynamics equations* in explicit form.

The operations that we have to perform are:

- ① $\sin \gamma +$ ② $\cos \gamma \sin \psi +$ ③ $\cos \gamma \cos \psi$
- ① $\cos \gamma -$ ② $\sin \gamma \sin \psi -$ ③ $\sin \gamma \cos \psi$
- ② $\cos \psi -$ ③ $\sin \psi$

In this way, the *three dynamics equations* are respectively derived:

$$\begin{aligned} \dot{v} = & -\omega^2 r (-\sin^2 \phi \sin^2 \lambda \sin \gamma - \cos^2 \lambda \sin \gamma + \cos \phi \sin \lambda \cos \lambda \cos \gamma \sin \psi - \\ & - \cos \phi \sin \phi \sin^2 \lambda \cos \gamma \cos \psi) + \frac{f_i}{m} \sin \gamma + \frac{f_j}{m} \cos \gamma \sin \psi + \frac{f_k}{m} \cos \gamma \cos \psi + \\ & + \frac{\dot{m}}{m} (-v + \omega r \sin \phi \sin \lambda \cos \gamma \sin \psi + \omega r \cos \lambda \cos \gamma \cos \psi) \end{aligned} \quad (3.3.28)$$

$$\begin{aligned} \dot{\gamma} = & \frac{1}{v} \left[\frac{v^2}{r} \cos \gamma - 2\omega v (\sin \phi \sin \lambda \sin \psi + \cos \lambda \cos \psi) - \omega^2 r (-\sin^2 \phi \sin^2 \lambda \cos \gamma - \right. \\ & - \cos^2 \lambda \cos \gamma - \cos \phi \sin \lambda \cos \lambda \sin \gamma \sin \psi + \cos \phi \sin \phi \sin^2 \lambda \sin \gamma \cos \psi) + \\ & + \frac{f_i}{m} \cos \gamma - \frac{f_j}{m} \sin \gamma \sin \psi - \frac{f_k}{m} \sin \gamma \cos \psi + \frac{\dot{m}}{m} (-\omega r \sin \phi \sin \lambda \sin \gamma \sin \psi - \\ & \left. - \omega r \cos \lambda \sin \gamma \cos \psi) \right] \end{aligned} \quad (3.3.29)$$

$$\begin{aligned} \dot{\psi} = & \frac{1}{v \cos \gamma} \left[\frac{v^2}{r} \cos^2 \gamma \sin \psi \tan \phi - 2\omega v (-\sin \phi \sin \lambda \sin \gamma \cos \psi - \cos \phi \sin \lambda \cos \gamma + \right. \\ & + \sin \gamma \cos \lambda \sin \psi) - \omega^2 r (\cos \phi \sin \lambda \cos \lambda \cos \psi + \cos \phi \sin \phi \sin^2 \lambda \sin \psi) + \\ & \left. + \frac{f_j}{m} \cos \psi - \frac{f_k}{m} \sin \psi + \frac{\dot{m}}{m} (\omega r \sin \phi \sin \lambda \cos \psi - \omega r \cos \lambda \sin \psi) \right] \end{aligned} \quad (3.3.30)$$

The mass equation

To completely describe the dynamics of our system, only the mass equation remains:

$$\dot{m} = -\frac{T}{g_{0,EARTH} I_{sp}} \quad (3.3.31)$$

where the *specific impulse* I_{sp} can be dependent on thrust.

The gravitational acceleration

Finally, to account for the variation in the value of the gravitational acceleration during descent, we can state that:

$$g = g_0 \frac{r_0^2}{r^2} \quad (3.3.32)$$

where g_0 is a reference value at a given altitude. This is in accordance with the spherical celestial body hypothesis and Newton's gravitational law, through Gauss's Theorem. Alternatively, a more accurate gravitational model of the celestial body in question can be imported.

3.4 Summary of dynamics and kinematics

Summarizing, we have:

- *three kinematics equations: \dot{r} , $\dot{\lambda}$ and $\dot{\phi}$*
- *three dynamics equations: \dot{v} , $\dot{\gamma}$ and $\dot{\psi}$*
- *one mass equation: \dot{m}*

Our model is then described by a system of *seven ODEs*, which we quote again here for clarity:

$$\dot{r} = v \sin \gamma$$

$$\dot{\lambda} = \frac{v \cos \gamma \sin \psi}{r \cos \phi}$$

$$\dot{\phi} = \frac{v}{r} \cos \gamma \cos \psi$$

$$\begin{aligned} \dot{v} = & -\omega^2 r (-\sin^2 \phi \sin^2 \lambda \sin \gamma - \cos^2 \lambda \sin \gamma + \cos \phi \sin \lambda \cos \lambda \cos \gamma \sin \psi - \\ & - \cos \phi \sin \phi \sin^2 \lambda \cos \gamma \cos \psi) + \frac{f_i}{m} \sin \gamma + \frac{f_j}{m} \cos \gamma \sin \psi + \frac{f_k}{m} \cos \gamma \cos \psi + \\ & + \frac{\dot{m}}{m} (-v + \omega r \sin \phi \sin \lambda \cos \gamma \sin \psi + \omega r \cos \lambda \cos \gamma \cos \psi) \end{aligned}$$

$$\dot{\gamma} = \frac{1}{v} \left[\frac{v^2}{r} \cos \gamma - 2\omega v (\sin \phi \sin \lambda \sin \psi + \cos \lambda \cos \psi) - \omega^2 r (-\sin^2 \phi \sin^2 \lambda \cos \gamma - \cos^2 \lambda \cos \gamma - \cos \phi \sin \lambda \cos \lambda \sin \gamma \sin \psi + \cos \phi \sin \phi \sin^2 \lambda \sin \gamma \cos \psi) + \frac{f_i}{m} \cos \gamma - \frac{f_j}{m} \sin \gamma \sin \psi - \frac{f_k}{m} \sin \gamma \cos \psi + \frac{\dot{m}}{m} (-\omega r \sin \phi \sin \lambda \sin \gamma \sin \psi - \omega r \cos \lambda \sin \gamma \cos \psi) \right]$$

$$\dot{\psi} = \frac{1}{v \cos \gamma} \left[\frac{v^2}{r} \cos^2 \gamma \sin \psi \tan \phi - 2\omega v (-\sin \phi \sin \lambda \sin \gamma \cos \psi - \cos \phi \sin \lambda \cos \gamma + \sin \gamma \cos \lambda \sin \psi) - \omega^2 r (\cos \phi \sin \lambda \cos \lambda \cos \psi + \cos \phi \sin \phi \sin^2 \lambda \sin \psi) + \frac{f_j}{m} \cos \psi - \frac{f_k}{m} \sin \psi + \frac{\dot{m}}{m} (\omega r \sin \phi \sin \lambda \cos \psi - \omega r \cos \lambda \sin \psi) \right]$$

$$\dot{m} = -\frac{T}{g_{0,EARTH} I_{sp}}$$

with:

$$\begin{bmatrix} f_i \\ f_j \\ f_k \end{bmatrix} = \begin{bmatrix} (T \cos \epsilon - D) \sin \gamma + (T \sin \epsilon + L) \cos \sigma \cos \gamma - mg \\ (T \cos \epsilon - D) \cos \gamma \sin \psi - (T \sin \epsilon + L) (\cos \psi \sin \sigma + \cos \sigma \sin \gamma \sin \psi) \\ (T \cos \epsilon - D) \cos \gamma \cos \psi + (T \sin \epsilon + L) (\sin \sigma \sin \psi - \cos \sigma \sin \gamma \cos \psi) \end{bmatrix}$$

It's now possible to understand why a singularity was present when passing above the Poles in the original equations of motion.[10] Indeed, when $\phi = \pm 90^\circ$, a zero at denominator appears in the $\dot{\lambda}$ and $\dot{\psi}$ equations, but in the new reference frame, when $\phi = \pm 90^\circ$ we are not above the poles. It's also important to note that this isn't the only singularity present in the equations; a similar problem appears when $\gamma = \pm 90^\circ$ and when $v = 0$ m/s. However, this is easily avoidable by imposing adequate boundary conditions at the end of the simulation, such as having reached an "almost vertical descent" condition ($\gamma \approx -90^\circ$) and an "almost zero" speed. This is also the approach used in other already cited references, and it can be considered a valid engineering approximation for the accuracy required for a preliminary optimization. On the contrary, removing the singularity when passing above the Poles was necessary because it would have completely prevented the optimization algorithm from converging to a solution. [4] [10] However, to completely remove all of the singularities, one should resort to using quaternions, or to express the dynamics equations directly in the perifocal reference frame (but in this case with 2D confined dynamics). For the latter case, see Appendix C.

3.5 Transformation between the original and modified frames

As stated in Section (3.2), the \mathcal{F}_R reference frame is defined differently from the Celestial Body-Fixed ones commonly used. For example the The Mean Earth-/Polar Axis (ME) reference system defines the z-axis as the mean rotational pole, with the Prime Meridian (0° Longitude) defined by the mean Earth direction. [1] In our case the mean rotational pole is represented instead by the y-axis (that we named \hat{j}_R), and, as already stated, this was necessary to avoid singularities when passing above the poles. Nevertheless, this isn't very practical for the final user of our optimization software, who almost certainly has the coordinates of the final landing site expressed in a traditional frame, so a procedure to pass from a *traditional (original) frame* to the *modified frame* introduced in this thesis is outlined in this Section, in such a way that the input and output of the software are expressed in the original frame, while the computations are done in the modified frame.

There are three quantities involved in this transformation:

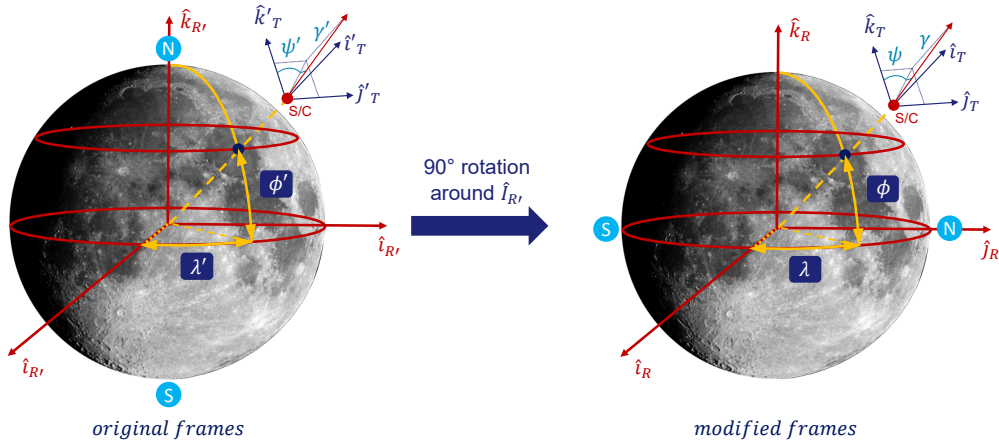
- the λ' (λ) angle, representing longitude in the original (modified) frame;
- the ϕ' (ϕ) angle, representing latitude in the original (modified) frame;
- the ψ' (ψ) angle, representing heading measured from the local north direction in the original (modified) frame;

A fourth angle, γ , will appear in our equations, but because the celestial body is modeled as *spherical*, one could verify a posteriori that in fact γ will not change.

We will also refer to:

- the modified Celestial-Body Fixed Frame as \mathcal{F}_R ;
- the modified LVLH Frame as \mathcal{F}_T ;
- the original Celestial-Body Fixed Frame as \mathcal{F}'_R ;
- the original LVLH Frame as \mathcal{F}'_T .

As already stated, all of the angles measured in the original frames will be marked by a superscript as well (γ' , ψ' , λ' and ϕ').

Figure 3.4: Rotation of the $\mathcal{F}_{R'}$ frame to the \mathcal{F}_R frame

3.5.1 From the Original to the Modified Frames

We can start by introducing the basic strategy, which is to express the position vector in cartesian components in the \mathcal{F}_R frame and the velocity vector in cartesian components in the \mathcal{F}_T frame, then to rotate these vectors in the \mathcal{F}_R and \mathcal{F}_T frames respectively and finally to calculate the ψ' , λ' and ψ' from the cartesian components of the rotated position and speed vectors. By looking at Figure (3.4), one can easily notice that the \mathcal{F}_R frame is easily obtained from the $\mathcal{F}_{R'}$ frame through a 90° rotation around the $\hat{i}_{R'}$ versor. Consequently, the $\mathcal{F}_R \rightarrow \mathcal{F}_{R'}$ rotation matrix is given by:

$$L_{R'R} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \quad (3.5.1)$$

while the L_{TR} rotation matrix has already been stated in Equation (3.3.1). Clearly there is no formal difference between the definition of the L_{TR} and $L_{T'R'}$ matrices, as long as we use the appropriate angles (the ones with the superscript in case of $L_{T'R'}$). We must also remember an important property of rotation matrices: they are orthogonal; this means that for two generic reference frames \mathcal{F}_a and \mathcal{F}_b :

$$L_{ab} = L_{ba}^T \quad (3.5.2)$$

so all of the remaining rotation matrices can be easily derived from the ones already expressed.

The conversion algorithm for λ and ϕ is then the following:

$$\textcircled{1} \begin{bmatrix} x_R \\ y_R \\ z_R \end{bmatrix} = L_{RR'} \begin{bmatrix} r \cos \phi' \cos \lambda' \\ r \cos \phi' \sin \lambda' \\ r \sin \phi' \end{bmatrix}$$

$$\textcircled{2} \phi = \text{atan2} \left(z_R, \sqrt{x_R^2 + y_R^2} \right)$$

$$\textcircled{3} \lambda = \text{atan2} (y_R, x_R)$$

$$\textcircled{4} \begin{bmatrix} w_{tx} \\ w_{ty} \\ w_{tz} \end{bmatrix} = L_{TR} L_{RR'} L_{R'T'} \begin{bmatrix} v \sin \gamma' \\ v \cos \gamma' \sin \psi' \\ v \cos \gamma' \cos \psi' \end{bmatrix}$$

$$\textcircled{5} \gamma = \text{atan2} \left(w_{tx}, \sqrt{w_{ty}^2 + w_{tz}^2} \right)$$

$$\textcircled{6} \psi = \text{atan2} (w_{ty}, w_{tz})$$

The γ equation can be used to verify that the conversion has been performed correctly ($\gamma = \gamma'$).

3.5.2 From the Modified to the Original Frames

We can simply invert the previous algorithm and obtain the following:

$$\textcircled{1} \begin{bmatrix} x_{R'} \\ y_{R'} \\ z_{R'} \end{bmatrix} = L_{R'R} \begin{bmatrix} r \cos \phi \cos \lambda \\ r \cos \phi \sin \lambda \\ r \sin \phi \end{bmatrix}$$

$$\textcircled{2} \phi' = \text{atan2} \left(z_{R'}, \sqrt{x_{R'}^2 + y_{R'}^2} \right)$$

$$\textcircled{3} \lambda' = \text{atan2} (y_{R'}, x_{R'})$$

$$\textcircled{4} \begin{bmatrix} w_{tx'} \\ w_{ty'} \\ w_{tz'} \end{bmatrix} = L_{T'R'} L_{R'R} L_{RT} \begin{bmatrix} v \sin \gamma \\ v \cos \gamma \sin \psi \\ v \cos \gamma \cos \psi \end{bmatrix}$$

$$\textcircled{5} \gamma' = \text{atan2} \left(w_{tx'}, \sqrt{w_{ty'}^2 + w_{tz'}^2} \right)$$

$$\textcircled{6} \psi' = \text{atan2} (w_{ty'}, w_{tz'})$$

3.6 Computing the Range

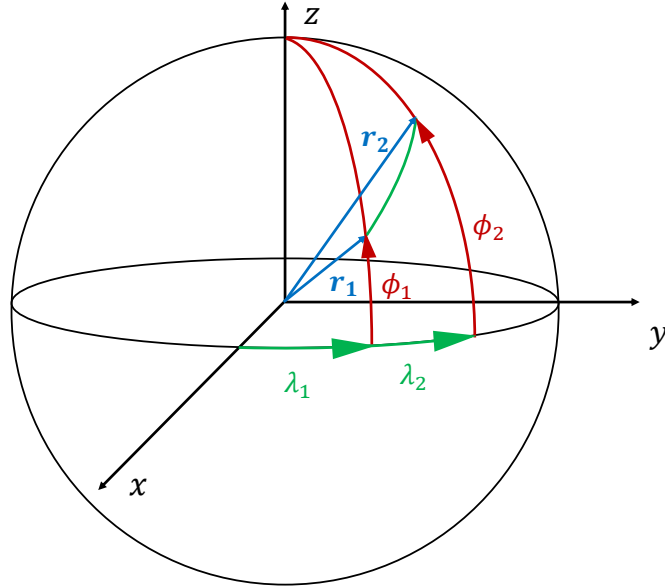


Figure 3.5: Descent Trajectory Range

Assuming a spherical body and an in-plane descent trajectory, it's easy to compute the descent trajectory range by deriving the angle θ between the two position vectors at descent initiation and landing, respectively r_1 and r_2 . This angle is not dependent on their magnitude, so we can assume that they are both unit-vectors:

$$\begin{aligned} \mathbf{r}_1 &= [\cos \phi_1 \cos \lambda_1 \quad \cos \phi_1 \sin \lambda_1 \quad \sin \phi_1]^T \\ \mathbf{r}_2 &= [\cos \phi_2 \cos \lambda_2 \quad \cos \phi_2 \sin \lambda_2 \quad \sin \phi_2]^T \end{aligned} \quad (3.6.1)$$

Furthermore, this angle doesn't depend on the orientation of the reference frame, so we can rotate it around the z-axis of an angle λ_1 , obtaining that, in the new frame:

$$\begin{aligned} \mathbf{r}_1 &= [\cos \phi_1 \quad 0 \quad \sin \phi_1]^T \\ \mathbf{r}_2 &= [\cos \phi_2 \cos(\lambda_2 - \lambda_1) \quad \cos \phi_2 \sin(\lambda_2 - \lambda_1) \quad \sin \phi_2]^T \end{aligned} \quad (3.6.2)$$

We can now compute the dot product of r_1 and r_2 :

$$\mathbf{r}_1 \cdot \mathbf{r}_2 = \cos \phi_1 \cos \phi_2 \cos(\lambda_2 - \lambda_1) + \sin \phi_1 \sin \phi_2 = \cos \theta \quad (3.6.3)$$

It's convenient to avoid computing an arccos to increase precision when the two points are close together. To achieve this, we remember that:

$$\frac{1 - \cos \theta}{2} = \sin^2 \left(\frac{\theta}{2} \right) \quad (3.6.4)$$

Replacing $\cos \theta$ in the previous expression, it reduces to:

$$\theta = 2 \arcsin \left(\sqrt{\cos \phi_1 \cos \phi_2 \sin^2 \left(\frac{\lambda_2 - \lambda_1}{2} \right) + \sin^2 \left(\frac{\phi_2 - \phi_1}{2} \right)} \right) \quad (3.6.5)$$

and consequently:

$$d = \theta r = 2r \arcsin \left(\sqrt{\cos \phi_1 \cos \phi_2 \sin^2 \left(\frac{\lambda_2 - \lambda_1}{2} \right) + \sin^2 \left(\frac{\phi_2 - \phi_1}{2} \right)} \right) \quad (3.6.6)$$

3.7 Trapezoidal Direct Collocation Method Implementation in MATLAB

Having concluded the theoretical discussion of the equations of descent dynamics, we can finally move on to their implementation in a MATLAB direct collocation algorithm. We divide the $[t_0, t_f]$ time interval into $N+1$ nodes, obtaining a $[t_0, t_1, \dots, t_{N-1}, t_N]$ grid, and we define the decision vector \mathbf{z} as follows:

$$\begin{aligned} \mathbf{z}' &= \text{vec} \begin{bmatrix} T_0 & \epsilon_0 & \sigma_0 & v_0 & \gamma_0 & \psi_0 & r_0 & \lambda_0 & \phi_0 & m_0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ T_N & \epsilon_N & \sigma_N & v_N & \gamma_N & \psi_N & r_N & \lambda_N & \phi_N & m_N \end{bmatrix} = \\ &= \text{vec} \begin{bmatrix} u_1(t_0) & u_2(t_0) & u_3(t_0) & x_1(t_0) & x_2(t_0) & x_3(t_0) & x_4(t_0) & x_5(t_0) & x_6(t_0) & x_7(t_0) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u_1(t_N) & u_2(t_N) & u_3(t_N) & x_1(t_N) & x_2(t_N) & x_3(t_N) & x_4(t_N) & x_5(t_N) & x_6(t_N) & x_7(t_N) \end{bmatrix} \\ \mathbf{z} &= \begin{bmatrix} \mathbf{z}' \\ \Delta t \end{bmatrix} \in \mathbb{R}^{10(N+1)+1} \end{aligned} \quad (3.7.1)$$

where vec is an operator that lines up the columns of a matrix, forming a column vector. Each time-dependent variable is represented by $N + 1$ values sampled on

the grid nodes. It's also necessary to include the $\Delta t = t_f - t_0$ variable in the decision vector, with $t_0 = 0$, because, obviously, we don't know how much time will be required to complete the descent maneuver. We can also identify σ , ϵ and T as the *control variables* of our problem, while v , γ , ψ , r , λ , ϕ and m are the *state variables*.

Being interested in a Moon environment, we can remove all the aerodynamic terms from the differential equations that describe our system and, using the formalism already introduced in Chapter 2, we can write the previous system of seven ODEs as follows:

$$\dot{\mathbf{x}} = f(\mathbf{x}(t), \mathbf{u}(t), \mathbf{p}, t)$$

with:

- $\mathbf{x} = [x_1, x_2, x_3, x_4, x_5, x_6, x_7]^T : [t_0, t_f] \rightarrow \mathbb{R}^7$.
- $\mathbf{u} = [u_1, u_2, u_3]^T : [t_0, t_f] \rightarrow \mathbb{R}^3$.
- $\mathbf{p} = [I_{SP}, g_0, g_{0-EARTH}]^T : [t_0, t_f] \rightarrow \mathbb{R}^3$.
- $t \in [t_0, t_f]$.

and, by applying the Trapezoidal Direct Collocation Algorithm, we obtain the following equality constraints:

$$\mathbf{c}_k = \mathbf{x}_{k+1} - \mathbf{x}_k - \frac{h}{2}(\mathbf{f}_{k+1} + \mathbf{f}_k)(t_{k+1} - t_k) = 0, \mathbf{c}_k \in \mathbb{R}^7, k = 0, \dots, (N - 1)$$

These basically are $N \cdot 7$ constraints, that should be concatenated in a single vector \mathbf{c}_{eq} when given as output by the `nonlcon` function:

$$\mathbf{c}_{eq} = \begin{bmatrix} \mathbf{c}_1 \\ \vdots \\ \mathbf{c}_N \end{bmatrix}$$

The *objective function* to be minimized is in this case represented by:

$$J(\mathbf{z}) = -z_{10(N+1)} \in \mathbb{R}$$

which corresponds to maximizing the final mass of the lander, while the upper and lower constraints:

$$\mathbf{z}_L \leq \mathbf{z} \leq \mathbf{z}_U$$

can be used for example to impose boundaries on the initial and final conditions of the various state variables, or to impose a minimum/maximum thrust magnitude. Finally, we can resort to the `c` output of the `nonlcon` function to apply constraints

on the maximum rates of change in the control variables, using for example forward finite differences of the following form:

$$c_j = \frac{u_i(t_{k+1}) - u_i(t_k)}{h} \quad (3.7.2)$$

Clearly, in applying all of these constrains, it's important to correctly extrapolate the various variables at the correct time nodes from the previously defined *decision vector*.

Chapter 4

Introduction to Astrodynamics

The reader is probably already familiar with the basics of astrodynamics, which are necessary to correctly understand the Moon Targeting techniques that will be introduced in Chapter 6. However, in an attempt to make this thesis self-contained, this chapter aims to provide all the necessary concepts to correctly understand the following material, by briefly addressing Newton's Universal Gravitation, the generic N-Body problem and the Restricted Two-Body Problem. It's clearly possible to skip this Chapter in case these notions are already known, or to consult the listed References, on which this chapter is based, for additional information.

4.1 Newton's Law of Universal Gravitation

The starting point of Newton's work can be traced back to the three Kepler's Laws:

- **First Law:** The orbit of each planet is an ellipse, with the sun at a focus;
- **Second Law:** The line joining the planet to the Sun sweeps out equal areas in equal times;
- **Third Law:** The square of the period of a planet is proportional to the cube of its mean distance from the sun;

However, these three laws were just a description of the motion of planets around the Sun, and not an explanation; but why do planets revolve in such a way? The first answer to this question was given by Isaac Newton, who conceived his law of universal gravitation during the plague epidemic between 1665 and 1666, at the sole age of 23 years old, but it was not until 1687 that his discoveries, prompted by Halley, were first published, when Newton issued *The Mathematical Principles*

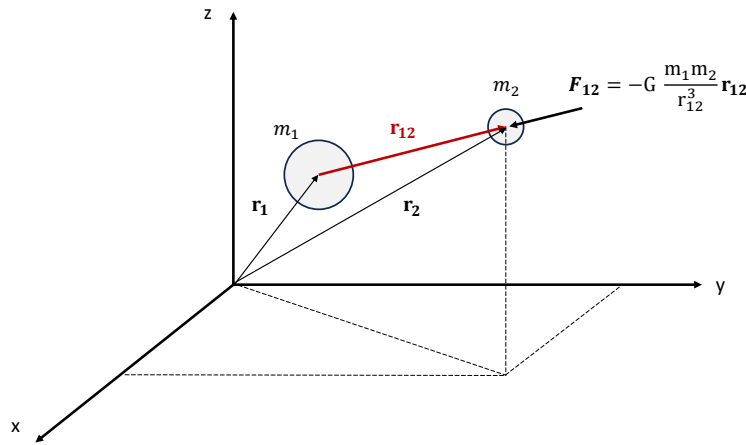


Figure 4.1: Newton's Law of Gravity

of *Natural Philosophy*, also known as *Principia*. This work is widely known for containing his three laws of motion, known as *Newton's Laws*, but, in the same book, Newton also formulated the *Universal Law of Gravity* by stating that any two bodies attract one another with a force proportional to the product of their masses and inversely proportional to the the square of the distance between them. In mathematical terms:

$$\mathbf{F}_{12} = -G \frac{m_1 m_2}{r_{12}^3} \mathbf{r}_{12} \quad (4.1.1)$$

where \mathbf{F}_{12} is the force produced by the body of mass m_1 on the body of mass m_2 , $G = 6,67 \cdot 10^{-11} \frac{N \cdot m^2}{kg^2}$ is the *universal gravity constant* and \mathbf{r}_{12} is the position vector of mass m_2 with respect to m_1 .

4.2 The N-Body Problem

Obviously, our system can be characterized by the simultaneous presence of multiple masses, all interacting with each other through gravitational forces. We also consider these masses to be spherical and uniform, so that through Gauss's theorem, one could prove that the gravitational force produced by each of them is equivalent to that of a point of equal mass located in the middle of the sphere. This is a reasonable approximation in many cases. The result is a system of differential equations of the second order that constitute the so called N-Body Problem. Let's start by remembering Newton's Second Law of Motion, which states that in

an *inertial coordinate frame* the rate of change of momentum of a body is given by the sum of the forces acting on it:

$$\frac{d}{dt}(m\dot{\mathbf{r}}) = \sum \mathbf{F} \quad (4.2.1)$$

Where \mathbf{r} is the position vector of mass m in the chosen inertial frame (see Figure 4.1). In this case the masses of the various bodies can be considered constant, so Equation (4.2.1) reduces to:

$$m\ddot{\mathbf{r}} = \sum \mathbf{F} \quad (4.2.2)$$

For each point mass, considering only the contribution of the gravitational force, we can then write the following:

$$\ddot{\mathbf{r}}_i = \sum_{j=1, j \neq i}^N -G \frac{m_j}{r_{ji}^3} \mathbf{r}_{ji}, \quad i = 1, \dots, N \quad (4.2.3)$$

This a system of $3N$ coupled differential equations of the second order. There is no analytical solution for this general case, so a numerical one is necessary. Before implementing this system in the various available *ode* routines, for example in the **MATLAB** environment, one should first transform these equations into a system of $6N$ differential equations of the first order, defining appropriate additional variables by substitution.

In addition, instead of studying the motion of the various bodies with respect to the origin of the inertial frame, one could also subtract one of the differential equations from the others to study a relative motion with respect to that mass, reducing the system to $3N - 3$ differential equations of the second order. This is also the approach that will be used later on, in Chapter 6, when targeting the Moon in the J2000 frame, with the aid of SPICE ephemeris data. There is however a particular case of great interest of the N-Body Problem, which is in many cases a valid approximation and also presents an analytical solution: the R2BP-Restricted Two-Body Problem, that we are going to discuss in the next section.

4.3 The Restricted Two-Body Problem - R2BP

Starting from Equation (4.2.3), it's possible to consider just two masses: m_1 and m_2 , with $m_2 \gg m_1$, obtaining:

$$\begin{cases} \ddot{\mathbf{r}}_1 = -G \frac{m_2}{r_{21}^3} \mathbf{r}_{21} \\ \ddot{\mathbf{r}}_2 = -G \frac{m_1}{r_{12}^3} \mathbf{r}_{12} \end{cases} \quad (4.3.1)$$

Subtracting now the second one from the first, we get:

$$\ddot{\mathbf{r}}_{21} = -G \frac{m_1 + m_2}{r_{21}^3} \mathbf{r}_{21} \quad (4.3.2)$$

because obviously $\mathbf{r}_{21} = -\mathbf{r}_{12}$. But $m_2 \gg m_1$, so:

$$\ddot{\mathbf{r}}_{21} \approx -G \frac{m_2}{r_{21}^3} \mathbf{r}_{21} \quad (4.3.3)$$

The traditional way to write the previous equation, and to which we will refer from now on, is, however, the following:

$$\ddot{\mathbf{r}} = -\frac{\mu}{r^3} \mathbf{r} \quad (4.3.4)$$

where $\mu = G(m_1 + m_2) \approx Gm_2$ is the *standard gravitational parameter* and \mathbf{r} is the position vector of the secondary (less massive body) with respect to the primary (more massive) body. It's also typical to refer to the mass of the primary body as M , and to the mass of the secondary body as m . Equation (4.3.4) basically is the starting point for all the subsequent discussion, which will lead us to prove that, in this environment, the orbits of celestial bodies as seen from a primary body (e.g. the planets and the Sun respectively, or a S/C and Earth) are *conics*, and to obtain the constants of orbital motion.

4.3.1 The two constants of orbital motion

Let us consider a cartesian coordinate system centered in the primary body of mass M , shown in Figure (4.2). We anticipate that the trajectory of the secondary body will always be contained in the same plane, on which the \hat{p} and \hat{q} versors lie, but we will prove it shortly. We know that:

$$\begin{aligned} \hat{i} &= \hat{p} \cos \nu + \hat{q} \sin \nu \\ \hat{j} &= -\hat{p} \sin \nu + \hat{q} \cos \nu \end{aligned} \quad (4.3.5)$$

Consequently, if we take the first derivative, we obtain:

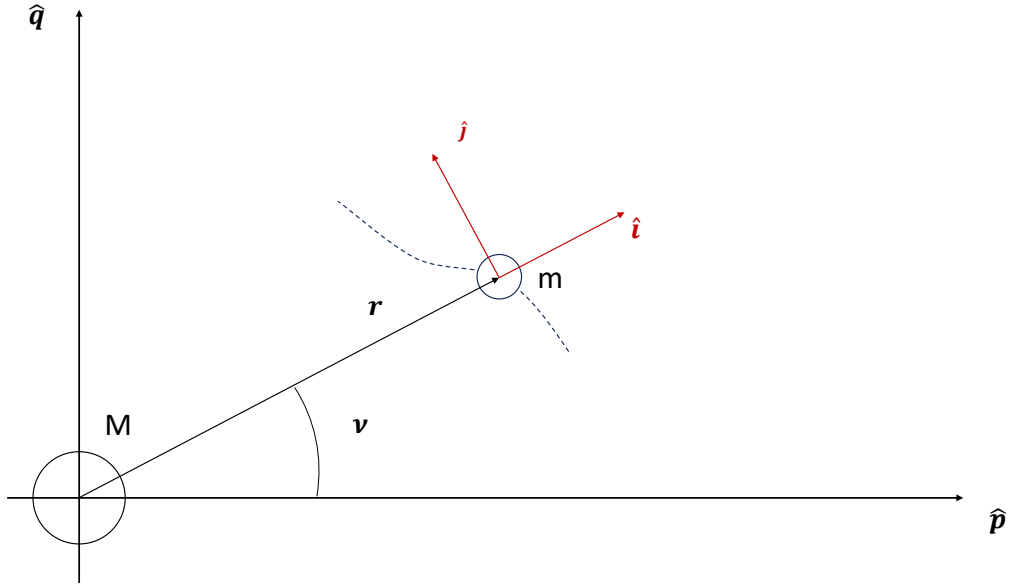


Figure 4.2: The perifocal frame.

$$\begin{aligned}\dot{\hat{i}} &= -\hat{p} \sin \nu \dot{\nu} + \hat{q} \cos \nu \dot{\nu} = \dot{\nu} \hat{j} \\ \dot{\hat{j}} &= -\hat{p} \cos \nu \dot{\nu} - \hat{q} \sin \nu \dot{\nu} = -\dot{\nu} \hat{i}\end{aligned}\quad (4.3.6)$$

Thus:

$$\begin{aligned}\mathbf{r} &= r \hat{i} \\ \dot{\mathbf{r}} &= \dot{r} \hat{i} + r \dot{\hat{i}} = \dot{r} \hat{i} + r \dot{\nu} \hat{j} = v_r \hat{i} + v_t \hat{j}\end{aligned}\quad (4.3.7)$$

Then, repeating the same process for the acceleration:

$$\begin{aligned}\ddot{\mathbf{r}} &= \ddot{r} \hat{i} + \dot{r} \dot{\hat{i}} + \dot{r} \dot{\nu} \hat{j} + r \ddot{\nu} \hat{j} + r \dot{\nu} \dot{\hat{j}} = \\ &= \ddot{r} \hat{i} + \dot{r} \dot{\nu} \hat{j} + \dot{r} \dot{\nu} \hat{j} + r \ddot{\nu} \hat{j} - r \dot{\nu}^2 \hat{i} = \\ &= (\ddot{r} - r \dot{\nu}^2) \hat{i} + (2\dot{r} \dot{\nu} + r \ddot{\nu}) \hat{j} = a_r \hat{i} + a_t \hat{j}\end{aligned}\quad (4.3.8)$$

Specific Mechanical Energy

Given Equation (4.3.4), using the expressions that we just found for a_r and a_t :

$$\begin{cases} \ddot{r} - r\dot{\nu}^2 = -\frac{\mu}{r^2} \\ 2\dot{\nu}\dot{r} + \ddot{\nu}r = 0 \end{cases} \quad (4.3.9)$$

Now, if we compute $v_r a_r + v_t a_t$, we obtain:

$$\begin{aligned} \dot{r}\ddot{r} - r\dot{\nu}^2 + 2r\dot{\nu}\dot{r} + r^2\dot{\nu}\ddot{\nu} &= -\frac{\mu}{r^2}\dot{r} \rightarrow \\ \rightarrow \dot{r}\ddot{r} + r\dot{\nu}^2 + r^2\dot{\nu}\ddot{\nu} &= -\frac{\mu}{r^2}\dot{r} \end{aligned} \quad (4.3.10)$$

The latter can also be written as follows:

$$\begin{aligned} \frac{1}{2} \frac{d}{dt}(\dot{r}^2) + \frac{1}{2} \frac{d}{dt}(r^2\dot{\nu}^2) &= \frac{d}{dt} \left(\frac{\mu}{r} \right) \rightarrow \\ \rightarrow \frac{d}{dt}(v_r^2 + v_t^2) &= \frac{d}{dt} \left(\frac{\mu}{r} \right) \end{aligned} \quad (4.3.11)$$

and integrating, we finally get:

$$\boxed{\xi = \frac{v^2}{2} - \frac{\mu}{r} = \text{const}} \quad (4.3.12)$$

Equation (4.3.12) expresses the conservation of the *specific mechanical energy*, given by the sum of *kinetic* and *potential* specific energy, which constitutes one of the two constants of orbital motion, along with *specific angular momentum*.

Specific Angular Momentum

To prove that the *specific angular momentum* is constant, we start by taking the cross product of both members of Equation (4.3.4) by \mathbf{r} , obtaining:

$$\mathbf{r} \times \ddot{\mathbf{r}} = -\frac{\mu}{r^3}(\mathbf{r} \times \mathbf{r}) = \mathbf{0} \quad (4.3.13)$$

However, $\mathbf{r} \times \ddot{\mathbf{r}} = \frac{d}{dt}(\mathbf{r} \times \dot{\mathbf{r}})$, so:

$$\boxed{\frac{d}{dt}(\mathbf{r} \times \dot{\mathbf{r}}) = 0 \rightarrow \mathbf{h} = \text{const}} \quad (4.3.14)$$

where \mathbf{h} is precisely the *specific angular momentum*. In particular, \mathbf{h} is constant in both magnitude and direction. Therefore, the orbit of the body of mass m is confined to a plane, to which we refer as *orbital plane*.

4.3.2 The Trajectory Equation

Once again starting from Equation (4.3.4), we want to derive $r(\nu)$, to prove that the trajectories traced by bodies moving under the gravitational influence of a primary body are indeed *conics*. We then take the cross product of Equation (4.3.4) by \mathbf{h} , which we just proved to be constant, resulting in:

$$\begin{aligned}
 \mathbf{h} \times \ddot{\mathbf{r}} &= \mathbf{h} \times \left(-\frac{\mu}{r^3} \mathbf{r} \right) \rightarrow \\
 \rightarrow \frac{d}{dt} (\mathbf{h} \times \dot{\mathbf{r}}) &= -\frac{\mu}{r^3} [(\mathbf{r} \times \dot{\mathbf{r}}) \times \mathbf{r}] \rightarrow \\
 \rightarrow \frac{d}{dt} (\mathbf{h} \times \dot{\mathbf{r}}) &= -\frac{\mu}{r^3} [(\mathbf{r} \cdot \mathbf{r}) \dot{\mathbf{r}} - (\mathbf{r} \cdot \dot{\mathbf{r}}) \mathbf{r}] \rightarrow \\
 \rightarrow \frac{d}{dt} (\mathbf{h} \times \dot{\mathbf{r}}) &= -\frac{\mu}{r^3} [r^2 \dot{\mathbf{r}} - r \dot{r} \mathbf{r}] \rightarrow \\
 \rightarrow \frac{d}{dt} (\mathbf{h} \times \dot{\mathbf{r}}) &= -\frac{\mu}{r} \dot{\mathbf{r}} + \frac{\mu}{r^2} \dot{r} \mathbf{r} = -\frac{d}{dt} \left(\frac{\mu}{r} \mathbf{r} \right)
 \end{aligned} \tag{4.3.15}$$

where we applied the following algebraic property: $(\mathbf{a} \times \mathbf{b}) \times \mathbf{c} = (\mathbf{a} \cdot \mathbf{c})\mathbf{b} - (\mathbf{b} \cdot \mathbf{c})\mathbf{a}$. Thus, integrating the latter expression:

$$\mathbf{h} \times \dot{\mathbf{r}} = -\frac{\mu}{r} \mathbf{r} - \mathbf{B} \tag{4.3.16}$$

where \mathbf{B} is an integration constant. Now, if we take the scalar product of both members of this latter equation by \mathbf{r} , we obtain:

$$\begin{aligned}
 \mathbf{r} \cdot [\dot{\mathbf{r}} \times \mathbf{h}] &= \mathbf{r} \cdot \left[\frac{\mu}{r} \mathbf{r} + \mathbf{B} \right] \rightarrow \\
 \rightarrow (\mathbf{r} \times \dot{\mathbf{r}}) \cdot \mathbf{h} &= \mu r + r B \cos \nu \\
 \rightarrow h^2 &= \mu r + r B \cos \nu
 \end{aligned} \tag{4.3.17}$$

because $\mathbf{a} \cdot (\mathbf{b} \times \mathbf{c}) = \mathbf{c} \cdot (\mathbf{a} \times \mathbf{b})$ and with ν being the angle between \mathbf{B} and \mathbf{r} . Rearranging this expression, we finally get the desired result:

$$r = \frac{h^2/\mu}{1 + B/\mu \cos \nu} \tag{4.3.18}$$

By observing that r is minimized when $\nu = 0$, we also understand that \mathbf{B} is a vector oriented in the direction of the orbit *periapsis*, which is the point of the

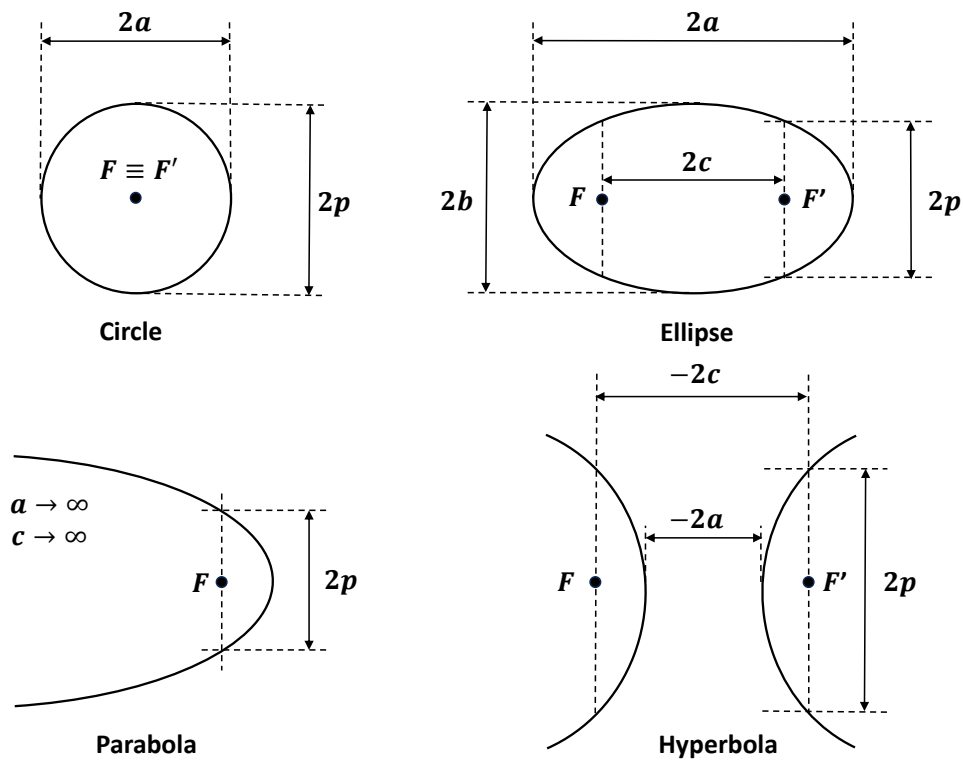


Figure 4.3: Common geometrical dimensions of all conic sections.

trajectory closest to the center of mass of the primary body, as opposed to the *apoapsis* (which exists only for elliptical orbits).

4.3.3 Polar Equation of a Conic Section

Equation (4.3.18) is in fact very similar in form to the *Polar Equation of a Conic Section*, that is the following:

$$\boxed{r = \frac{p}{1 + e \cos \nu}} \quad (4.3.19)$$

In this equation, p is the *semi-latus rectum*, while e is the *eccentricity*.

The different types of conic sections are visible in Figure (4.3), and the eccentricity allows us to distinguish between one and the other. In particular:

- $e = 0$ if the conic section is a *circle*;
- $0 < e < 1$ if the conic section is an *ellipse*;

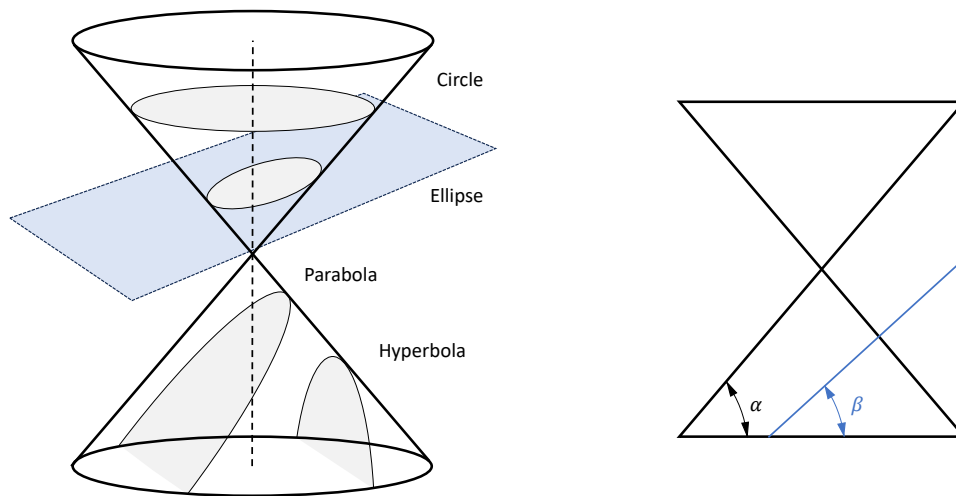


Figure 4.4: Conic sections obtained from the intersection of a plane with a circular cone.

- $e = 1$ if the conic section is a *parabola*;
- $e > 1$ if the conic section is a *hyperbola*;

These geometrical shapes are named *conic sections* because they can be obtained through the intersection of a plane and a circular cone, as shown in Figure (4.4).

Mathematically, it's possible to define the eccentricity as follows:

$$e = \frac{\sin \beta}{\sin \alpha} \quad (4.3.20)$$

and, for every conic section, except for parabolas, it follows from their definition that:

$$e = \frac{c}{a} \quad (4.3.21)$$

and

$$p = a(1 - e^2) \quad (4.3.22)$$

4.3.4 The eccentricity vector

Recalling Equation (4.3.16), we can compute the \mathbf{B} vector as follows:

$$\mathbf{B} = \dot{\mathbf{r}} \times \mathbf{h} - \frac{\mu}{r} \mathbf{r} \quad (4.3.23)$$

Then, comparing Equation (4.3.18) with Equation (4.3.19), we can immediately observe that:

$$e = \frac{B}{\mu} \rightarrow \mathbf{e} = \frac{\mathbf{B}}{\mu} \quad (4.3.24)$$

where \mathbf{e} is the *eccentricity vector*, that will come into play later on, when describing the *B-Plane Targeting Procedure* in Chapter 6, and also in the orbit determination procedure presented in Appendix D.

4.3.5 Relating orbit geometry with \mathbf{h} and ξ

It follows directly from Equations (4.3.18) and (4.3.19) that:

$$\boxed{h = \sqrt{\mu p}} \quad (4.3.25)$$

while deriving the orbit energy ξ as a function of its geometrical properties is a bit more complex. The final result will be valid for any conic section; however, we will carry out our calculations considering only ellipses and hyperbolas, and not parabolas, which constitute a particular case. We observed before that the orbit energy is constant, so we can compute it at an arbitrary point, the periapsis:

$$\xi = \frac{v_p^2}{2} - \frac{\mu}{r_P}$$

By plugging $\nu = 0$ into Equation (4.3.18), we get:

$$r_P = \frac{p}{1 + e} = a(1 - e) \quad (4.3.26)$$

This is the point where, as already stated, we leave parabolas aside, because for them the relation $p = a(1 - e^2)$ isn't valid. We also know that at periapsis and apoapsis the speed vector is tangent to the local horizontal direction, due to the fact that by definition they are the points of the trajectory which are respectively

closest and furthest from the center of mass of the main body. It follows then that:

$$h = r_P v_t = \sqrt{\mu p} = \sqrt{\mu a(1 - e^2)} \quad (4.3.27)$$

and

$$\begin{aligned} \xi &= \frac{h^2}{2r_p^2} - \frac{\mu}{r_P} = \\ &= \frac{\mu a(1 - e^2)}{2a^2(1 - e)^2} - \frac{\mu}{a(1 - e)} \end{aligned} \quad (4.3.28)$$

which, after a bit of algebra, reduces to:

$$\boxed{\xi = \frac{v^2}{2} - \frac{\mu}{r} = -\frac{\mu}{2a}} \quad (4.3.29)$$

This last equation will prove to be very useful when calculating the magnitude of the *impulsive maneuvers* to be performed to obtain, for example, a transfer trajectory to reach the Moon from Earth and to circularize the orbit at Moon arrival.

4.3.6 Period of an Elliptical Orbit

Looking at Figure (4.5), We know from elementary calculus that:

$$dA = \frac{1}{2}r^2 d\nu = \frac{1}{2}r^2 \dot{\nu} dt = \frac{h dt}{2} \quad (4.3.30)$$

The area of an ellipse is given by:

$$A = \pi(ab) \quad (4.3.31)$$

It can also be proven that for ellipses $a^2 = b^2 + c^2$; therefore, remembering that $e = \frac{c}{a}$ and that $h = \text{const}$, we integrate on the whole swept area A and obtain that the orbit period is given by:

$$T = \frac{2A}{h} = \frac{2\pi ab}{\sqrt{\mu a(1 - e^2)}} = \frac{2\pi a \sqrt{a^2(1 - e^2)}}{\sqrt{\mu a(1 - e^2)}} \quad (4.3.32)$$

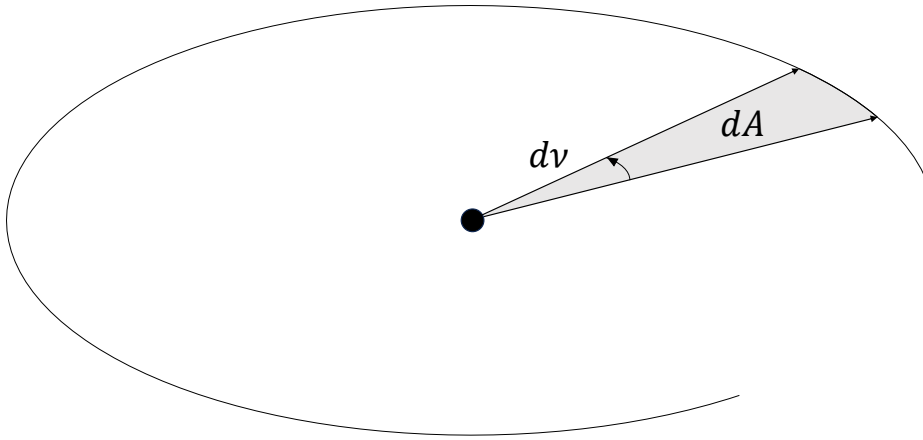


Figure 4.5: Elementary angle $d\nu$, associated to dA , swept by the position vector during time dt

which reduces to:

$$T = 2\pi\sqrt{\frac{a^3}{\mu}} \quad (4.3.33)$$

We are now going to apply these concepts to a basic maneuver to perform transfers between different orbits, the *Hohmann Transfer*.

Hohmann Transfer

It can be proven that the Hohmann Transfer is the most energy efficient two-impulse maneuver to perform a transfer between two circular coplanar orbits. These two circular orbits are connected by an elliptical orbit tangent to them at its apoapsis and periapsis, as shown in Figure (4.6). Using the *energy equation*, we can compute the speed magnitudes corresponding to the two circular orbits of radius r_1 and r_2 , respectively v_{c1} and v_{c2} , and those corresponding to the periapsis and apoapsis of the Hohmann Transfer Trajectory, respectively v_{H1} and v_{H2} . We obtain that:

- $v_{c1} = \sqrt{\frac{\mu}{r_1}}$
- $v_{c2} = \sqrt{\frac{\mu}{r_2}}$

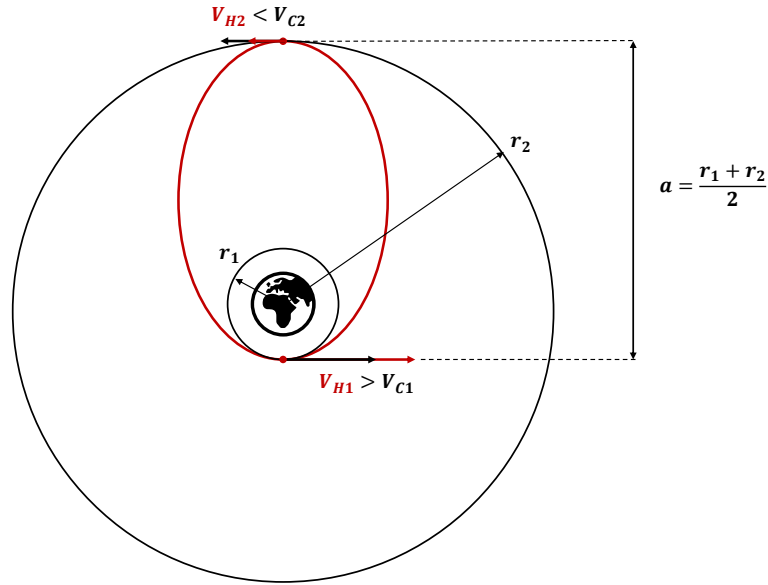


Figure 4.6: Hohmann Transfer maneuver between two circular coplanar orbits.

- $v_{H1} = \sqrt{\frac{\mu}{r_1}} \sqrt{\frac{2r_2}{r_1+r_2}}$
- $v_{H2} = \sqrt{\frac{\mu}{r_2}} \sqrt{\frac{2r_1}{r_1+r_2}}$

and this allows us to compute the absolute value of the two necessary ΔV s to perform the maneuver:

- $\Delta V_1 = \sqrt{\frac{\mu}{r_1}} \left| \sqrt{\frac{2r_2}{r_1+r_2}} - 1 \right|$
- $\Delta V_2 = \sqrt{\frac{\mu}{r_2}} \left| \sqrt{\frac{2r_1}{r_1+r_2}} - 1 \right|$

Clearly their sign depends on whether we are going from r_1 to r_2 or vice versa. In addition, it's also possible to compute the travel time Δt_H of a Hohmann Transfer maneuver by observing that we are traversing only half of the transfer orbit, so Equation (4.3.33) yields to:

$$\Delta t_H = \frac{T_H}{2} = \pi \sqrt{\frac{(r_1 + r_2)^3}{8\mu}} \quad (4.3.34)$$

This concludes the theoretical discussion of the restricted two body problem, which allowed us, among other things, to prove *Kepler's Three Laws*. [27] One could argue, however, that the assumptions on which the two body theory is based are

not always well satisfied, as in the case, for example, of an Earth-Moon transfer trajectory. In this case, before resorting to a N-Body simulation, the *Circular Restricted Three Body Problem* (CR3BP) is a good intermediate step. This topic will be addressed in Chapter 6, when discussing the various possibilities to study a Moon targeting Problem.

4.4 Reference Frames

Before proceeding further in our introduction to the fundamental concepts of astrodynamics, it's worthwhile to present the coordinate frames that will be used in our study, some of which have already appeared before.

The Perifocal Reference Frame

The *perifocal coordinate frame* is centered at the focus of the orbit and consists of the $\{\hat{p}, \hat{q}, \hat{w}\}$ triad, defined as follows:

- \hat{p} pointing from the focus of the orbit to the periapsis;
- \hat{w} pointing in the direction of the orbit angular momentum vector;
- \hat{q} completes the right-handed triad and is defined such that $\hat{w} = \hat{p} \times \hat{q}$.

In a restricted Two Body environment, this coordinate Frame is also non-rotating and non-accelerating (inertial).

The Synodic Reference Frame

The *synodic coordinate frame* is a specialized coordinate system used in the analysis of the relative motion between two celestial bodies in orbit around their center of mass. It's centered in the center of mass of the two bodies and consists of the $\{\hat{x}_S, \hat{y}_S, \hat{z}_S\}$ triad, defined as follows:

- \hat{x}_S points from the center of mass of one celestial body to that of the other;
- \hat{z}_S is oriented perpendicular to the plane defined by the two bodies' orbits and is aligned with the angular momentum vector of the two-body system;
- \hat{y}_S completes the right-handed triad and is defined such that $\hat{z}_S = \hat{x}_S \times \hat{y}_S$.

This is a non-inertial reference frame.

The J2000 Reference Frame

The definition given here is in accordance with the one provided in the NAIF SPICE documentation.[18] As every other inertial reference frame defined in SPICE, the J2000 frame is centered in the *barycenter of the solar system*, and its three axes are defined as follows:

- \hat{x}_{J2000} is defined by the intersection of the Earth Equatorial Plane with the Ecliptic Plane (defined by the orbit of the Earth around the Sun) at epoch J2000 TDB, where J2000 TDB is 2000 JAN 01 12:00:00 TDB. TDB is the Barycentric Dynamical Time, which basically is a way to count time accounting for general relativity, and is defined as a linear transformation of TCB (Barycentric Coordinate Time). TCB itself is then equivalent to the proper time experienced by a clock at rest in a coordinate frame co-moving with the barycenter (center of mass) of the Solar System . See references [13] and [12] for further reading.
- \hat{z}_{J2000} is perpendicular to the Earth mean Equator at epoch J2000, and is approximately Earth spin-axis direction at that epoch. It points towards the hemisphere containing Polaris.
- \hat{y}_{J2000} completes the right-handed triad and is defined such that $\hat{z}_{J2000} = \hat{x}_{J2000} \times \hat{y}_{J2000}$.

The Mean Earth/Polar Axis (ME) frame

As stated in the `moon_de440_220930.tf` text frame-kernel, downloadable on the NAIF SPICE website: [22]

The Lunar mean Earth/polar axis system is a lunar body-fixed reference system used in the IAU/IAG Working Group Report [14] to describe the orientation of the Moon relative to the ICRF frame (equivalent to SPICE J2000). The +Z axis of this system is aligned with the north mean lunar rotation axis, while the prime meridian contains the mean Earth direction. The x-axis is defined by the intersection of the prime Meridian with the Moon Equator, orthogonal to the mean rotation axis. This system is also sometimes called the "mean Earth/mean rotation axis" system or "mean Earth" system.

The mean directions used to define the axes of a mean Earth/polar axis reference frame realizing the lunar ME system are associated with the DE440 planetary ephemeris version in our case and, of course, this reference frame is non-inertial.

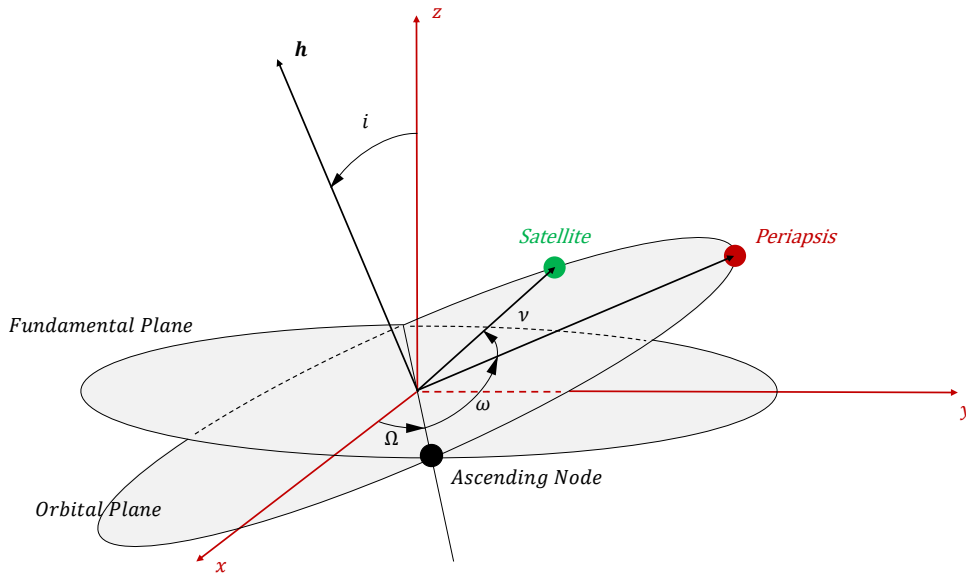


Figure 4.7: Classic Orbital Elements.

4.5 Classic Orbital Elements

We conclude our introduction to orbital mechanics by discussing orbital parameters, i.e. quantities used to specify orbits. There are many different ways to mathematically describe the same orbit, but certain schemes, each consisting of a set of six parameters, are commonly used in astronomy and orbital mechanics. A real orbit and its elements change over time due to gravitational perturbations by other objects and the effects of general relativity. A Kepler (Restricted Two-Body Theory) orbit is an idealized, mathematical approximation of the orbit at a particular time, and the classic orbital elements are also named Keplerian Elements after *Johannes Kepler*.

Considering a certain inertial frame centered in the reference body, with axes (x,y,z), there are two elements associated to the shape and dimension of the orbit:

- *eccentricity* (e) - describes the shape of the conic section representing the orbit;
- *semi-major axis* (a) - associated to the dimension of the orbit. A possible alternative is the *semilatus rectum* (p) in case the orbit is parabolic;

and other four associated with its position and orientation with respect to the main body, and the position of the secondary body along it's orbit at a certain time instant:

- *longitude of the ascending node* (Ω) - the angle measured in the fundamental (x-y) plane of a certain reference system between the \hat{i} unit vector (x-

direction) and the point where the satellite crosses through the fundamental plane in the northern direction (ascending node), measured counterclockwise when seen from the north side of the fundamental plane (from the z-axis);

- *argument of the periapsis* (ω) - the angle, in the plane of the satellite's orbit, between the ascending node and the periapsis point, measured in the direction of the satellite motion;
- *true anomaly at epoch t_0* (ν) - the angle, in the plane of the satellite's orbit, measured between the current position vector at time (t_0) and the periapsis point, measured in the direction of the satellite motion.
- *inclination* (i) - the angle between the z-direction and the angular momentum vector. If $i < 90^\circ$, the orbit is *prograde*, while if $i > 90^\circ$ the orbit is *retrograde*. Orbits with a 90-deg inclination are called *polar orbits* instead.

There are some particular cases in which some of the classical elements presented above are not defined:

- *circular orbits* - in this case ω is not defined, and ω and ν are replaced by *argument of latitude at epoch t_0* , $u = \omega + \nu$, which is the angle between the position vector at time t_0 and the ascending node, measured in the direction of satellite motion;
- *equatorial orbits* - the ascending node is not defined, so it's impossible to compute Ω . Consequently, Ω and ω are replaced by *longitude of periapsis*, $\Pi = \Omega + \omega$, which is the angle between the x-direction and periapsis point, measured counterclockwise when seen from the north side of the fundamental plane.
- *equatorial circular orbits* - neither Ω nor ω are defined, so Ω , ω and ν are replaced by *true longitude at epoch t_0* , $l = \Omega + \omega + \nu$, which is the angle between the position vector at time t_0 and x-direction, measured in the direction of satellite motion.

A procedure to compute the orbital elements given $\{\mathbf{r}, \dot{\mathbf{r}}\}$ and to then propagate the orbit is presented in Appendix D. It is useful when used together with the B-Plane algorithm presented in Chapter 5 to plot the orbit, before and after the trajectory correction maneuver. The alternative would be to simply propagate the orbit numerically, given the initial position and velocity (which one could observe to be, again, six parameters).

Chapter 5

B-Plane Targeting

In this chapter, the B-Plane Targeting problem, which was one of the steps that then led to the choice of the Moon targeting method, is addressed. It was noted that there isn't a great deal of material in literature covering this topic so, although the method will then not actually be applied to compute the TCM maneuver (but it can be useful to target a north or south injection), it was nevertheless chosen to keep this discussion within this thesis, as a useful reference for further research work that may be conducted on this method, perhaps for interplanetary missions. [5] [25] [16] [8]

5.1 B-Plane Parameters

The B-Plane is an imaginary plane defined as normal to the incoming asymptote of the hyperbolic orbit and containing the target body center of mass. The concept of an incoming asymptote of a hyperbolic orbit is associated to the so called *patched conics method*, which approximates interplanetary trajectories by *patching* two-body theory conics, each with focus on the celestial body to which the *SOI-Sphere of Influence* belongs. For example, for an Earth-Mars Hohmann transfer, according to this theory, the trajectory is composed of an hyperbola (or a parabola) with focus on Earth while inside Earth's SOI, an ellipse while inside the Sun's SOI, with focus on the Sun and another hyperbola while inside Mars' SOI, with focus on Mars.

The *SOI* is then defined as the region of space inside which the gravity of the celestial body to which the SOI belongs dominates the dynamics of a satellite. Using Laplace's perturbation approach, the radius of the SOI of a celestial body, considering two massive bodies and a much less massive spacecraft, is:

$$r_{SOI,1} = r_{12} \left(\frac{m_1}{m_2} \right)^{\frac{2}{5}} \quad (5.1.1)$$

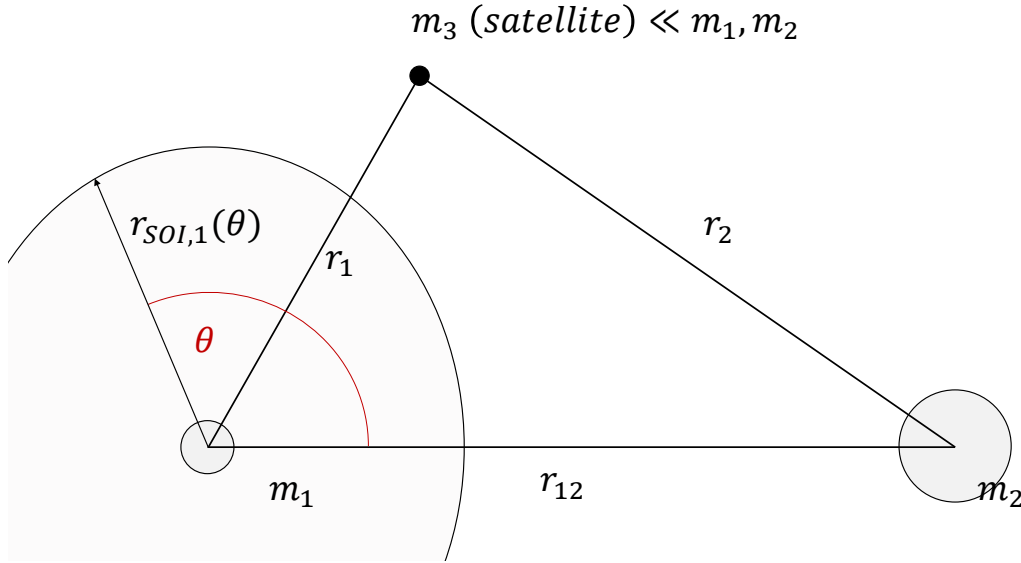


Figure 5.1: Sphere of Influence of a celestial body

where r_{12} is the distance between the two bodies, m_1 and m_2 are the masses of the two bodies and $r_{SOI,1}$ is the radius of the sphere of influence belonging to the first body. This result is only valid when $m_2 \gg m_1$. [11]

In reality the SOI is not exactly a sphere, and a more accurate formula would be the following:

$$r_{SOI,1} = r_{12} \left(\frac{m_1}{m_2} \right)^{\frac{2}{5}} \frac{1}{\sqrt[10]{1 + 3 \cos^2 \theta}} \quad (5.1.2)$$

Finally, when the masses of the two bodies are quite comparable, as in case of an Earth-Moon system, a better approximation of the SOI might be necessary:

$$r_{SOI,1} = \left[\left(\frac{m_1}{m_3} \right)^{-\frac{2}{5}} \sqrt[10]{1 + 3 \cos^2 \theta} + \frac{2}{5} \cos \theta \frac{1 + 6 \cos^2 \theta}{1 + 3 \cos^2 \theta} \right]^{-1} \quad (5.1.3)$$

The initial idea, that didn't then match the final mission strategy (see Chapter 1) is to use two-body dynamics, when inside the Moon's SOI, to compute the TCM $\Delta \vec{V}$ necessary to achieve the correct B-Plane parameters. The first step is to mathematically describe the B-Plane, and then we'll introduce the numerical procedure that allows us to compute the maneuver. Targeting the B-plane parameters instead of orbital parameters allows for a linearization of the problem, because it's possible to define partial derivatives that have a wide linear regime as we'll soon see. In addition, B-Plane parameters can be easily related to other orbital parameters of interest.

Let's start by looking at Figure (5.2). We define a $\{\hat{S}, \hat{T}, \hat{R}\}$ triad, where:

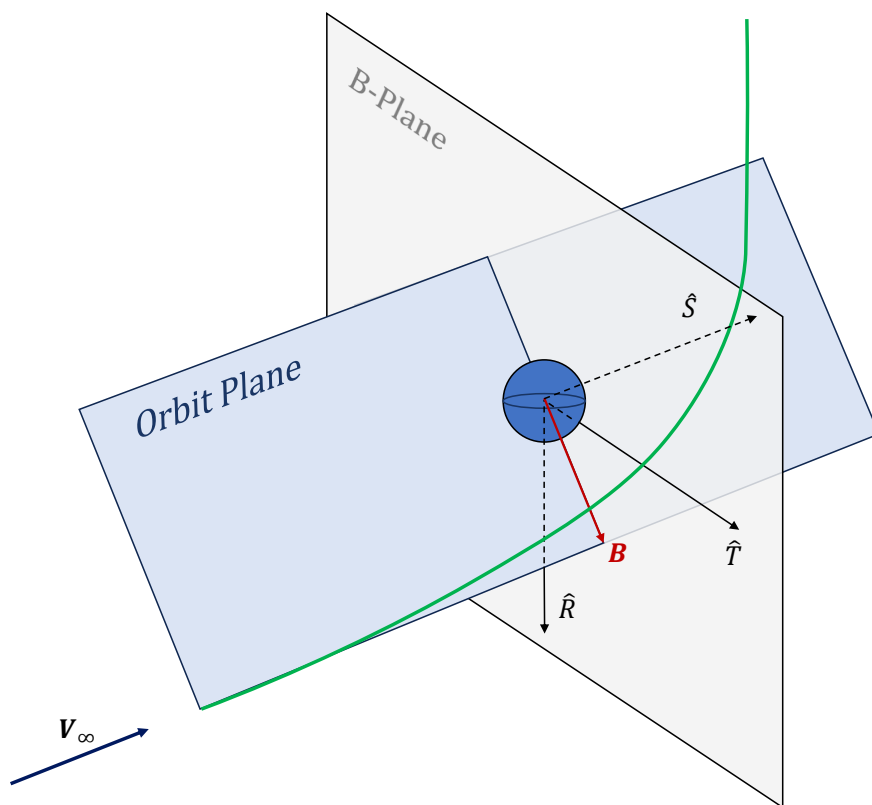


Figure 5.2: The B-Plane.

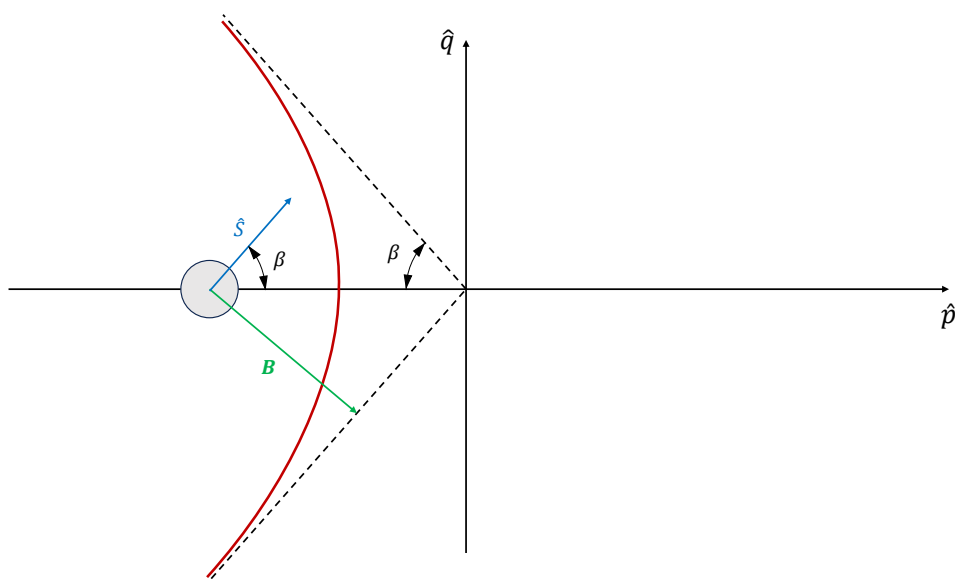


Figure 5.3: The B-Vector in the perifocal reference frame.

- \hat{S} is the incoming \vec{V}_∞ direction;
- \hat{K} is a reference unit vector that can for example be chosen to be normal to the equatorial plane and directed towards the north pole of the central body;
- $\hat{T} = \hat{S} \times \hat{K}$;
- $\hat{R} = \hat{S} \times \hat{T}$

The *B-Vector* \mathbf{B} (not to be confused with the B constant introduced in Chapter 4) is instead defined as the vector that goes from the center of mass of the central body to the point at which the hyperbolic asymptote intercepts the B-Plane.

Given $(\mathbf{r}, \dot{\mathbf{r}})$, we have to determine \hat{S} , \hat{T} , \hat{R} and \mathbf{B} . To achieve this, we resort to the restricted two-body problem.

First of all, we remember the definition of the *eccentricity vector*:

$$\mathbf{e} = \frac{1}{\mu} \left[\left(v^2 - \frac{\mu}{r} \right) \mathbf{r} - (\mathbf{r} \cdot \dot{\mathbf{r}}) \dot{\mathbf{r}} \right] \rightarrow \hat{e} = \frac{\mathbf{e}}{e}$$

The specific angular momentum can instead be computed as:

$$\mathbf{h} = \mathbf{r} \times \dot{\mathbf{r}} \rightarrow \hat{h} = \frac{\mathbf{h}}{h}$$

while the β angle in Figure (5.3) can be calculated as follows:

$$e = \frac{1}{\cos \beta} \rightarrow \beta = \arccos \left(\frac{1}{e} \right) \quad (5.1.4)$$

It's now easy to obtain the \hat{S} unit versor, again referring to Figure (5.3), as:

$$\hat{S} = \hat{e} \cos \beta + \left(\frac{\mathbf{h}}{h} \times \hat{e} \right) \sin \beta \quad (5.1.5)$$

Given the reference \hat{K} unit versor, \hat{T} and \hat{R} are then given by:

$$\begin{aligned} \hat{T} &= \hat{S} \times \hat{K} \\ \hat{R} &= \hat{S} \times \hat{T} \end{aligned} \quad (5.1.6)$$

The B-vector direction is computed as:

$$\hat{B} = \hat{S} \times \hat{h} \quad (5.1.7)$$

while the *missing distance*, which basically is the B-vector modulus, can be obtained from the conservation of angular momentum. First of all we compute the semi-major axis a and V_∞ (called hyperbolic excess velocity, which is the speed at infinite distance from the main body) using the energy equation:

$$a = -\frac{\frac{\mu}{2}}{\frac{v^2}{2} - \frac{\mu}{r}} \quad (5.1.8)$$

$$\frac{V_\infty^2}{2} = -\frac{\mu}{2a} \rightarrow V_\infty = -\frac{\mu}{a}$$

We remember that, indeed, $a < 0$ for a hyperbola. Then, using the angular momentum conservation:

$$B^2 V_\infty^2 = h^2 = \mu a (1 - e^2) \rightarrow$$

$$\rightarrow B^2 \left(-\frac{\mu}{a}\right) = \mu a (1 - e^2) \rightarrow$$

$$\rightarrow \boxed{B = -a\sqrt{e^2 - 1}}$$
(5.1.9)

where we took the minus (-) solution to get $B > 0$. It's finally possible to compute the B-Vector as:

$$\mathbf{B} = B\hat{B} = -a\sqrt{e^2 - 1}(\hat{S} \times \hat{h}) \quad (5.1.10)$$

The B-Plane parameters are now basically the cartesian or polar coordinates of intercept point of the trajectory with the B-Plane:

- **CARTESIAN FORM**

- $B_T = \mathbf{B} \cdot \hat{T}$
- $B_R = \mathbf{B} \cdot \hat{R}$

- **POLAR FORM**

- $b = |\mathbf{B}|$
- $\theta = \text{atan2}(B_R, B_T)$

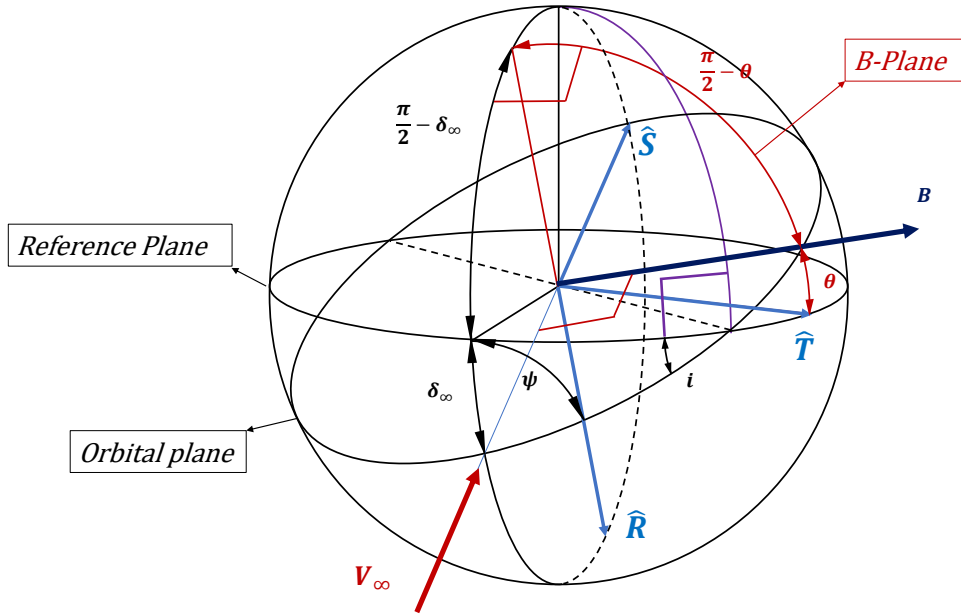


Figure 5.4: B-Plane Spherical Trigonometry. The θ angle shown in figure is negative, but this doesn't affect the final result.

5.2 Relating the B-Plane parameters to inclination and periapsis

Clearly the missing distance and the θ angle are not equal to inclination and periapsis, but we can find an analytical relation. The process is very straightforward for the periapsis; by combining the energy equation and angular momentum conservation equations, we get:

$$B = \frac{r_P}{V_\infty} \left(V_\infty^2 + \frac{2\mu}{r_P} \right)^{0.5} \quad (5.2.1)$$

while some spherical trigonometry is necessary for the inclination.

By looking at Figure (5.4) and applying the *law of sines*, we can *easily* state that:

$$\frac{\sin\left(\frac{\pi}{2} - \theta\right)}{\sin \psi} = \frac{\sin \frac{\pi}{2}}{\sin \frac{\pi}{2}} \rightarrow \boxed{\cos \theta = \sin \psi} \quad (5.2.2)$$

$$\frac{\sin \psi}{\sin \frac{\pi}{2}} = \frac{\sin\left(\frac{\pi}{2} + i\right)}{\sin\left(\frac{\pi}{2} + \delta_\infty\right)} \rightarrow \boxed{\cos \theta = \frac{\cos i}{\cos \delta_\infty}}$$

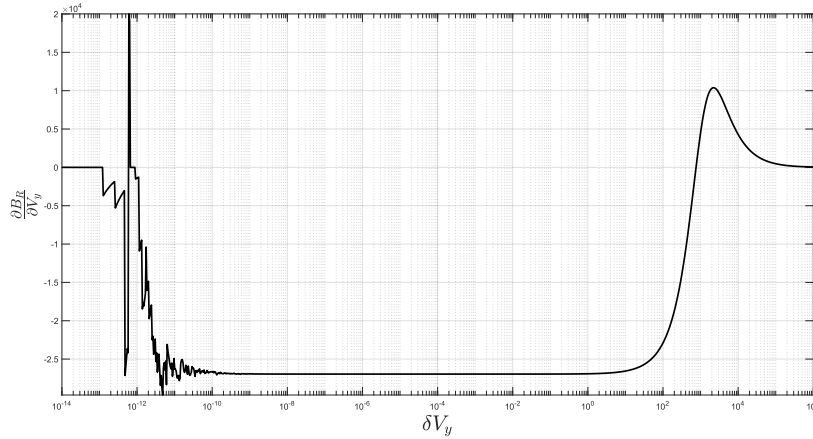


Figure 5.5: Example of B-Plane Parameter numerical partial derivative, evaluated for a certain $(\mathbf{r}, \dot{\mathbf{r}})$, as δV varies.

5.3 Solving using Newton's method

Now, given a certain $(\mathbf{r}, \dot{\mathbf{r}}) = (\mathbf{r}, \mathbf{V})$ state, how do we calculate the $\Delta \mathbf{V}$ necessary to achieve the desired B-Plane parameters? A possible method is to implement a Newton algorithm. Starting from a Taylor-series expansion, we stop at the first order:

$$\begin{bmatrix} B_T^* \\ B_R^* \end{bmatrix} = \begin{bmatrix} B_T \\ B_R \end{bmatrix} + \left[\frac{\partial \mathbf{B}}{\partial \mathbf{V}} \right] (\mathbf{V}^* - \mathbf{V}) + o(\mathbf{V}^* - \mathbf{V}) \quad (5.3.1)$$

where the "*" superscript indicates the desired final values and:

$$\left[\frac{\partial \mathbf{B}}{\partial \mathbf{V}} \right] = \begin{bmatrix} \frac{\partial B_T}{\partial V_x} & \frac{\partial B_T}{\partial V_y} & \frac{\partial B_T}{\partial V_z} \\ \frac{\partial B_R}{\partial V_x} & \frac{\partial B_R}{\partial V_y} & \frac{\partial B_R}{\partial V_z} \end{bmatrix} \in \mathbb{R}^{2 \times 3} \quad (5.3.2)$$

is the Jacobian Matrix. Neglecting the higher order terms represented by $o(\mathbf{V}^* - \mathbf{V})$, and defining $\Delta \mathbf{B} = \mathbf{B}^* - \mathbf{B}$ and $\Delta \mathbf{V} = \mathbf{V}^* - \mathbf{V}$, we get the following system:

$$\begin{bmatrix} \Delta B_T \\ \Delta B_R \end{bmatrix} = \left[\frac{\partial \mathbf{B}}{\partial \mathbf{V}} \right] \begin{bmatrix} \Delta V_x \\ \Delta V_y \\ \Delta V_z \end{bmatrix} \quad (5.3.3)$$

However the problem is undetermined, because $\left[\frac{\partial \mathbf{B}}{\partial \mathbf{V}} \right]$ is a 2x3 matrix, meaning that the number of unknowns is greater than the number of equations. To solve this problem we can :

- fix one component of the $\Delta \mathbf{V}$ vector;
- find the minimum norm $\Delta \mathbf{V}$. For a generic $A\mathbf{x} = \mathbf{b}$ system, the minimum norm solution is given by $\mathbf{x} = A^T(AA^T)^{-1}\mathbf{b}$;

- add another unknown variable. A typical choice is the TOF-Time of Flight.

The two latter options are clearly preferable. The next problem is that of computing the Jacobian Matrix. Some have attempted to provide analytical solutions, but a numerical one is also possible, although slightly less performing.[6] The procedure consists in choosing an appropriate perturbation δV , and approximating the partial derivatives using finite differences. In case a central one is used, an example formula would be:

$$\frac{\partial B_R}{\partial V_y} = \frac{B_T(V_x, V_y + \frac{\delta V}{2}, V_z) - B_T(V_x, V_y - \frac{\delta V}{2}, V_z)}{\delta V} \quad (5.3.4)$$

and similarly all the other ones can be obtained. Some common sense is required while choosing the δV perturbation, as shown in Figure (5.5). Both too small and too large values should be avoided, because of numerical cancellation problems (when the perturbation is too small) and loss of linearity (when the perturbation is too large). However, a certain range in which the partials are constant exists, so we can expect good results when implementing this method. Equation (5.3.3) has to be solved iteratively, calculating $\delta \mathbf{V}$, updating $\mathbf{V} = \mathbf{V} + \Delta \mathbf{V}$ and $\mathbf{B} = \mathbf{f}(\mathbf{r}, \mathbf{V})$ at each step, until a maximum number of iterations is reached or $|\Delta \mathbf{B}|$ is equal to less than a certain tolerance.

B-Plane Targeting Pseudocode

For a given initial $(\mathbf{r}_1, \mathbf{V}_1)$ state and some desired inclination and periapsis of the orbit i^* and r_p^* :

1. from the current position and speed $(\mathbf{r}_1, \mathbf{V}_1)$, compute the unit vectors $\{\hat{R}_1, \hat{S}_1, \hat{T}_1\}$ and the current \mathbf{B} vector, using the previous formulas;
2. compute the current B-Plane Parameters $\{B_R, B_T\}$;
3. compute $V_\infty = \sqrt{V_1^2 - \frac{2\mu}{r_1}}$;
4. compute $\delta_\infty = \text{atan2}\left(\hat{S}_{1z}, \sqrt{\hat{S}_{1x}^2 + \hat{S}_{1y}^2}\right)$;
5. compute B^* using r_p^* and V_∞ ;
6. compute θ^* using i^* and δ_∞ ; if $i^* < 0$ (north injection), $\theta^* = 2\pi - \theta^*$
7. compute B_R^* and B_T^* using B^* and θ^* ;
8. compute $\Delta \mathbf{B} = \mathbf{B}^* - \mathbf{B}$ and the Jacobian matrix (for the current position and speed $(\mathbf{r}_1, \mathbf{V}_1)$);

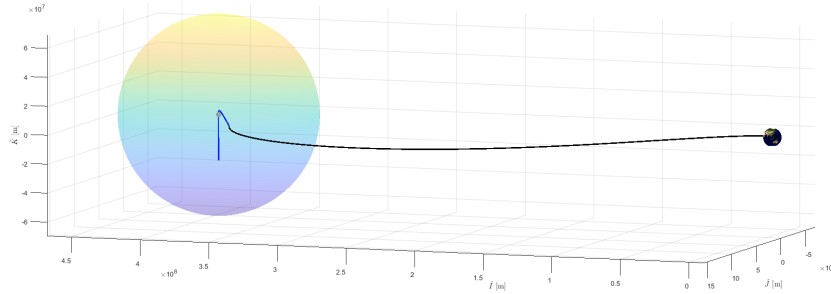


Figure 5.6: Moon B-Plane targeting. R2BP hypotheses satisfied.

9. if $|\Delta \mathbf{B}| = |\mathbf{B}^* - \mathbf{B}|$ is equal to less than a certain user-defined tolerance, or a maximum number of iterations has been reached, stop the algorithm (and print an error in the latter case);
10. compute $\Delta \mathbf{V}$ using Newton's Method (iteratively solve Equation (5.3.3));
11. update $\mathbf{V}_1 = \mathbf{V}_1 + \Delta \mathbf{V}$;
12. restart from point ①.

The final $\Delta \mathbf{V}$ is obviously the sum of all the $\Delta \mathbf{V}$ s obtained during the various iterations. The main advantage of this procedure, which goes through the B-Plane parameters rather than performing a direct targeting of inclination and periastrum, is that it's possible to remove the ambiguity on the inclination of the orbit by choosing between north and south injection, taking advantage of the θ angle, as specified in the pseudo-code. This wouldn't have been possible without it, because both a north and south injection are possible with the same inclination, and $\cos(i) = \cos(-i)$, so there would have been an ambiguity in the algorithm.

5.4 Accuracy Problems

The previous discussion is based on the assumption that the actual orbital mechanics can be approximated using the R2BP (Restricted Two-Body Problem), but for a Moon mission this may not be the case because of:

- the non-inertiality of a Moon-Centered reference frame;

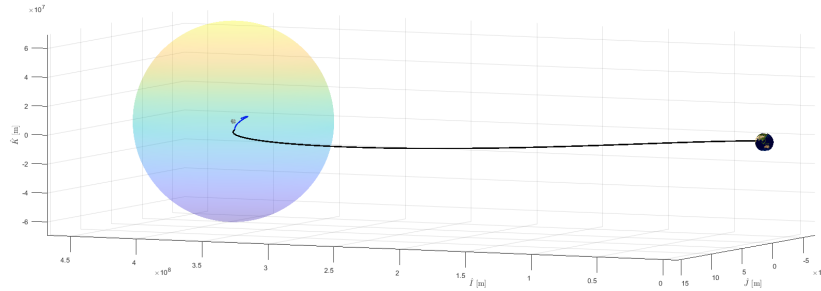


Figure 5.7: Moon B-Plane targeting. R2BP hypotheses unsatisfied.

- the fact that the gravitational pull of the Earth, and to a lesser extent the Sun, is not negligible even when inside the Moon's SOI (defined according to Equation (5.1.1));

In fact, to make the maneuver accurate, it would have to be performed so close to the Moon that it becomes completely unpractical from a ΔV cost standpoint. This problem is highlighted in Figure (5.6) and Figure (5.7), where an attempt to target a 90° inclination was made. However, the maneuver was accurate only in Figure (5.7), where the rotation of the Moon was "forcefully" stopped and Earth's gravitational pull was "turned off" after computing the TCM at the SOI.

In conclusion, this method can't be properly applied to our case of interest, but it can be interesting to compute TCMs for interplanetary missions, when entering the SOI of the arrival planet, because in that case a restricted two-body approximation is surely more appropriate. In addition, the B-Plane B_R parameter can be used in the next targeting methods to select a north or south injection, by imposing adequate constraints on its value (this idea was successfully implemented in the CR3BP targeting model).

Chapter 6

Lunar Targeting

As seen in Chapter 5, the R2BP isn't accurate enough to handle our case of interest. Therefore, two different dynamic models will be introduced here: the CR3BP and the SPICE-Enhanced N-Body Problem. We will also briefly look at how to analytically compute the LOI-Lunar Orbit Injection at Moon arrival and how to export the trajectory data to COSMOGRAPHIA, to obtain a 3D visualization of the optimized trajectory.

6.1 The Earth-Moon system geometry

Before proceeding further, it's necessary to gain awareness of the real Earth-Moon system geometry. This will enable us to understand why the CR3BP can be considered a good first approximation, but also why more accurate methodologies may then be needed. Let's start by looking at Figure (6.1), where the relative orientation of the Ecliptic plane, Equatorial plane and Moon's Orbital plane is highlighted. When viewed from the center of the Earth, Moon's orbit can also be described through six orbital parameters, such as:

- a -semimajor axis;
- e -eccentricity;
- i -inclination;
- Ω -longitude of the ascending node;
- ω -argument of perigee;
- α -right ascension at epoch

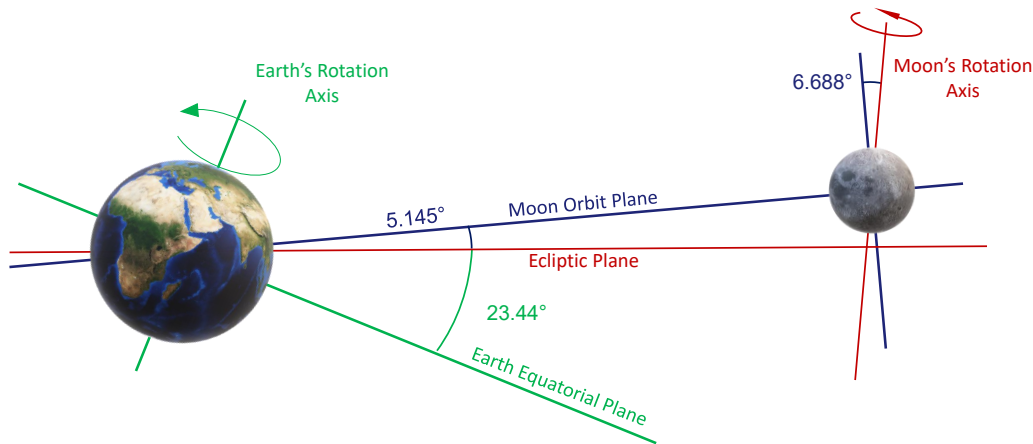


Figure 6.1: Earth-Moon System Geometry

where the first five were already described in Chapter 4, while α is the angle measured east-ward from the direction of the ascending node of the Moon to the projection of the Moon's position vector on the equatorial plane. In a classical R2BP the orbital parameters are constant but, in case of the Moon's Orbit, mainly due to the perturbative effect of the Sun, the orbital parameters are constantly changing with time, and their precise value at any particular time has to be obtained from lunar ephemeris. We can list some of the main perturbations:

- The mean value of the *semimajor axis* is 384400 km, and the average time for the Moon to complete one revolution around Earth relative to the stars is around 27.3h. However, due to solar perturbations, the sidereal period can vary as much as 7h.
- The mean eccentricity of the Moon's orbit is 0.0549, but the *evection* effect causes small changes in the orbital eccentricity, with a period of 31.8 days.
- The line of nodes, which is the intersection of the Moon's orbit with the ecliptic plane, rotates westward, making one complete revolution in 18.6 years.
- The inclination of the Moon's Orbit with respect to the ecliptic has a mean value of 5.15° , but it actually varies between 4.98° and 5.30° .
- The line of apsides (joining perigee and apogee) rotates in the direction of the Moon's orbital Motion causing ω to change by 360° in 8.9 years.

In reality, it wouldn't be completely correct to say that the Moon revolves around the center of mass of the Earth, because both the Earth and the Moon are orbiting around the center of mass of the Earth-Moon system. The CR3BP assumes that this orbit is circular for both, while the ER3BP-Elliptical Restricted 3-Body

Problem assumes that it's elliptical. Neither of these two interpretations are completely correct; however, the CR3BP can be used in our case to provide a first reasonable approximation to study a transfer trajectory, before resorting to a SPICE-augmented N-Body problem.

6.2 Formulation of the NLP Problem

The mission baseline was addressed in Chapter 1; recalling it briefly, the trajectory has to be propagated for 24h after the initial burn (commissioning phase) and then a TCM must be executed to target the correct Moon position and reach it with the correct orbital parameters, to perform the subsequent LOI maneuver. The landing sites of interest are located near the South Pole, so a 90° inclination injection orbit is the obvious choice.

The NLP Problem, applying a *Direct Single Shooting Algorithm*, can then be stated in a similar form for the two approaches presented in the previous section (CR3BP and SPICE N-Body):

$$\begin{aligned} \min_{\mathbf{z}} J(\mathbf{z}) &= \sqrt{\Delta V_x^2 + \Delta V_y^2 + \Delta V_z^2} && \text{subject to:} \\ i(t_f) &= i_{des} \\ r(t_f) &= r_{P,des} \\ \phi(t_f) &= 0 \end{aligned} \quad (6.2.1)$$

$$\text{with } \mathbf{z} = [\Delta V_x \quad \Delta V_y \quad \Delta V_z \quad RTOF]^T$$

where ΔV_x , ΔV_y and ΔV_z are the three TCM- ΔV components and *RTOF* is the *Remaining Time of Flight*, i.e. the propagation time required to arrive at the Moon or, more precisely, at the instant of execution of the LOI maneuver. As already stated, we also want to inject on a Polar Orbit and with the required periselenium for the subsequent LOI, so the three constraints are necessary to account for that. In addition, we could compute the B-Plane B_R parameter and impose an additional constraint to force a north ($B_R > 0$) or south ($B_R < 0$) injection. One could also argue that defining the final inclination is somewhat ambiguous, and that is true in the CR3BP because the components of the unit-vector orthogonal to the Moon's Equatorial Plane isn't exactly known. The problem can be easily solved in the SPICE approach using the included reference frame rotation routines, while for the CR3BP a valid engineering approximation is to compute the inclination using the synodic frame unit-vector $[0 \ 0 \ 1]$, although remembering that the Moon's Mean Rotation Axis isn't really normal to its orbital plane, as shown in Figure (6.1).

Optimization Strategy

- Compute the initial conditions (position and velocity in the appropriate reference frame) associated to the spacecraft at lunar transfer orbit injection.
- Propagate the trajectory until the desired time of execution of the TCM.
- Using the current (at TCM) position and speed state of the spacecraft, compute the optimum ΔV to reach the Moon with the desired boundary values solving the associated NLP Problem.
- Analytically compute the LOI Maneuver.
- Numerically propagate the whole trajectory to visualize it and verify its correctness.

With this matter cleared, we can now move on to the two dynamics models previously mentioned, starting with the CR3BP.

6.3 The Circular Restricted Three-Body Problem

The *Circular Restricted Three-Body Problem* (CR3BP) is a widely known model in celestial mechanics that deals with the motion of a smaller, negligible mass (e.g. a spacecraft) in the gravitational field produced by two larger masses, moving around their center of mass, which lies between the two objects, on circular orbits. One could immediately notice that this is a good approximation of the Earth-Moon system environment, reason why it is definitely worthwhile to at least derive the equations of motion and apply them to lunar targeting. However, the matter is certainly worthy of a more extensive discussion, which is beyond the scope of this thesis. [26] We will then see in Section 6.4 how to possibly improve the precision of our model by importing Moon ephemeris data from the SPICE Toolkit.

6.3.1 The Equations of Motion

It's convenient to study the dynamics of this system in the synodic frame, in which the position of the two main bodies is apparently fixed, as shown in Figure (6.2).

Obviously, it's necessary to account for the apparent acceleration terms, due to the rotation of the synodic frame. In this case the second principle of dynamics, considering the origin non-accelerating, constant angular velocity, constant S/C mass and accounting only for gravitational forces, becomes:

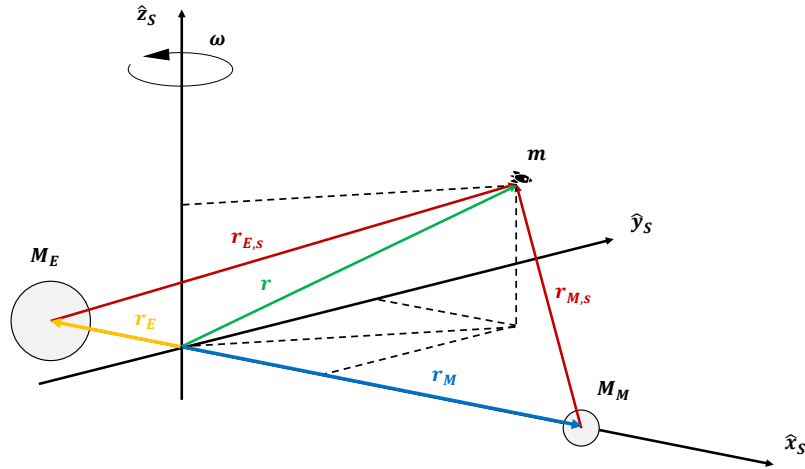


Figure 6.2: The Synodic Reference Frame.

$$\ddot{\mathbf{r}}_s + 2\boldsymbol{\omega} \times \dot{\mathbf{r}}_s + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{r}_s) = -G \frac{M_E}{r_{E,s}^3} \mathbf{r}_{E,s} - G \frac{M_M}{r_{M,s}^3} \mathbf{r}_{M,s} \quad (6.3.1)$$

Let us now introduce the *mass parameter*:

$$\mu = \frac{M_M}{M_E + M_M} \quad (6.3.2)$$

the *total mass*:

$$M = M_E + M_M \quad (6.3.3)$$

and the *distance between the two main bodies*:

$$R = r_E + r_M \quad (6.3.4)$$

The distance of the system barycenter from the center of mass of the first body is given by:

$$d = \frac{M_M R}{M_E + M_M} = \mu R \quad (6.3.5)$$

and the coordinates of the two main bodies in the synodic frame are then:

$$\mathbf{r}_E = \begin{bmatrix} -\mu R \\ 0 \\ 0 \end{bmatrix} \quad \mathbf{r}_M = \begin{bmatrix} (1-\mu)R \\ 0 \\ 0 \end{bmatrix} \quad (6.3.6)$$

From a simple force balance equation we can also compute the angular speed ω , equating the centripetal force with the gravitational force acting on one of the two bodies:

$$M_E \omega^2 \mu R = G \frac{M_E M_M}{R^2} \rightarrow \omega = \sqrt{\frac{GM}{R^3}} \quad (6.3.7)$$

At this point, indicating with $\mathbf{r}_s = [x, y, z]^T$ the coordinates of the spacecraft in the synodic frame, we can compute the $\mathbf{r}_{E,s}$ and $\mathbf{r}_{M,s}$ vectors:

$$\mathbf{r}_{E,s} = \begin{bmatrix} x + \mu R \\ y \\ z \end{bmatrix} \quad \mathbf{r}_{M,s} = \begin{bmatrix} x - (1-\mu)R \\ y \\ z \end{bmatrix} \quad (6.3.8)$$

the *centrifugal force*:

$$\boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{r}_s) = \begin{bmatrix} 0 & -\omega & 0 \\ \omega & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & -\omega & 0 \\ \omega & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} -\omega^2 x \\ -\omega^2 y \\ 0 \end{bmatrix} \quad (6.3.9)$$

and the *Coriolis acceleration* (or, better, its opposite):

$$2\boldsymbol{\omega} \times \dot{\mathbf{r}}_s = 2 \begin{bmatrix} 0 & -\omega & 0 \\ \omega & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} -2\omega \dot{y} \\ 2\omega \dot{x} \\ 0 \end{bmatrix} \quad (6.3.10)$$

By plugging everything into Equation (6.3.1), we finally obtain the desired result:

$$\begin{cases} \ddot{x} - \omega^2 x - 2\omega \dot{y} = -GM \frac{1-\mu}{r_E^3} (x + \mu R) - GM \frac{\mu}{r_M^3} (x - (1-\mu)R) \\ \ddot{y} - \omega^2 y + 2\omega \dot{x} = -GM \frac{1-\mu}{r_E^3} y - GM \frac{\mu}{r_M^3} y \\ \ddot{z} = -GM \frac{1-\mu}{r_E^3} z - GM \frac{\mu}{r_M^3} z \end{cases} \quad (6.3.11)$$

There is no analytical solution for these equations, as opposed to what was seen for the Two-Body restricted Problem, so a numerical integration is necessary.

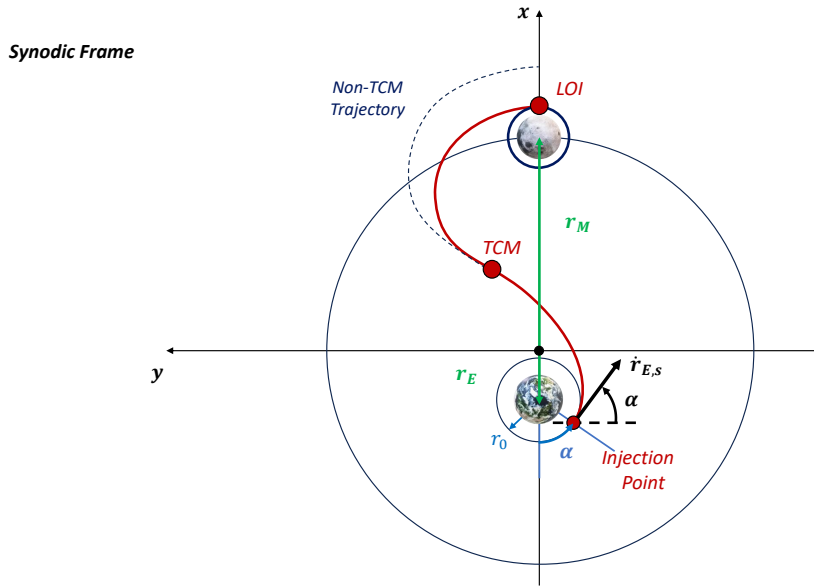


Figure 6.3: Transfer Trajectory in Synodic Frame

State Vector at Injection Point in Synodic Frame

Using Two-Body theory, it's easy to analytically compute a plausible initial state vector to propagate the trajectory. Assuming a Hohmann-like transfer trajectory in the Moon's orbital plane and an initial circular parking orbit around Earth, with the help of Figure (6.3), we obtain that *in the synodic frame*:

$$\mathbf{r}(t_0) = \begin{bmatrix} -r_E - r_0 \cos \alpha \\ -r_0 \sin \alpha \\ 0 \end{bmatrix} \quad (6.3.12)$$

where $\alpha = \omega \cdot T_H$ and $T_H = \pi \sqrt{\frac{(r_E + r_M + r_0)^3}{8GM_E}}$. The *Earth-Relative Speed* is instead given by:

$$\dot{\mathbf{r}}_{\mathbf{E},s}(t_0) = \begin{bmatrix} V_H \sin \alpha \\ -V_H \cos \alpha \\ 0 \end{bmatrix} \quad (6.3.13)$$

where:

$$V_H = \sqrt{\frac{GM_E}{r_0}} \sqrt{\frac{2(r_E + r_M)}{r_E + r_M + r_0}} \quad (6.3.14)$$

To simulate a *fixed launch program*, the value of $r_E + r_M$ in the V_H formula can be replaced with the *fixed apogee distance from Earth*. In addition, being in a rotating frame, this speed has to be corrected as follows:

$$\dot{\mathbf{r}}_s(t_0) = \dot{\mathbf{r}}_{\mathbf{E},s}(t_0) - \boldsymbol{\omega} \times (\mathbf{r}(t_0) - \mathbf{r}_{\mathbf{E}}) \quad (6.3.15)$$

obtaining the *initial speed expressed in the synodic frame*.

Visualizing the Trajectory in a Non-Rotating Frame

It's possible to use a time-dependant rotation matrix to plot the trajectory in a non-rotating pseudo-inertial frame. This non-rotating frame could simply be the synodic frame "frozen" at the time of departure from Earth. Indicating with t the time elapsed since Earth departure, the rotation matrix is then given by:

$$R(t) = \begin{bmatrix} \cos(\omega t) & -\sin(\omega t) & 0 \\ \sin(\omega t) & \cos(\omega t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (6.3.16)$$

This non rotating frame can then itself be rotated on more conventional inertial frames, such as the J2000 frame.

6.4 SPICE-Enhanced N-Body Problem

Using SPICE, we now have the ability to increase the accuracy of the dynamic model, accounting for the real positions of the various celestial bodies involved. In particular, it's not hard to include the gravitational pull of the Sun too, so we will include it in the dynamics equations. If its effect is deemed negligible, it is then legitimate to remove the associated terms.

We remember that the J2000 frame is an *inertial reference frame* (it's *non-rotating* and it's centered in the center of mass of the solar system, so it's *non-accelerating* too), so it's possible to apply the second principle of dynamics without any additional term. By doing so, we get:

$$\ddot{\mathbf{r}}_s = -G \frac{M_S}{r_{S,s}^3} \mathbf{r}_{S,s} - G \frac{M_E}{r_{E,s}^3} \mathbf{r}_{E,s} - G \frac{M_M}{r_{M,s}^3} \mathbf{r}_{M,s} \quad (6.4.1)$$

However, it's more practical to describe the motion of the spacecraft as seen from Earth so, *neglecting the mass of the spacecraft*, we obtain that for the Earth:

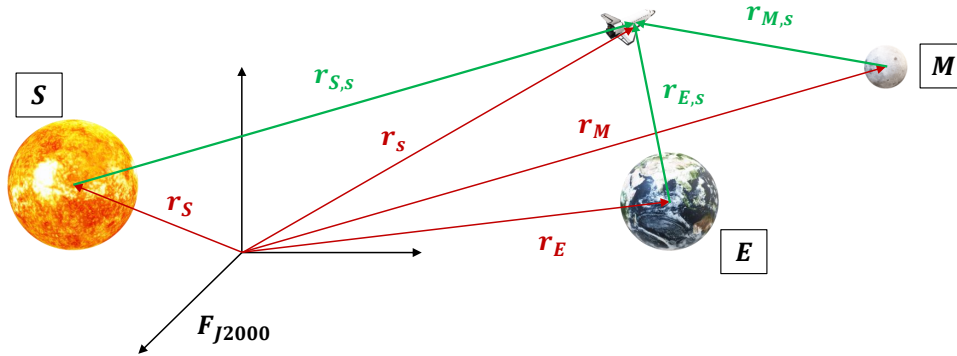


Figure 6.4: Position vectors in the J2000 Frame.

$$\ddot{\mathbf{r}}_{\mathbf{E}} = -G \frac{M_S}{|\mathbf{r}_{\mathbf{E}} - \mathbf{r}_{\mathbf{S}}|^3} (\mathbf{r}_{\mathbf{E}} - \mathbf{r}_{\mathbf{S}}) - G \frac{M_M}{|\mathbf{r}_{\mathbf{E}} - \mathbf{r}_{\mathbf{M}}|^3} (\mathbf{r}_{\mathbf{E}} - \mathbf{r}_{\mathbf{M}}) \quad (6.4.2)$$

Subtracting Equation (6.4.2) from Equation (6.4.1), we finally get:

$$\begin{aligned} \ddot{\mathbf{r}}_{\mathbf{E},s} = & -G \frac{M_S}{r_{S,s}^3} \mathbf{r}_{\mathbf{S},s} - G \frac{M_E}{r_{E,s}^3} \mathbf{r}_{\mathbf{E},s} - G \frac{M_M}{r_{M,s}^3} \mathbf{r}_{\mathbf{M},s} + \\ & + G \frac{M_S}{|\mathbf{r}_{\mathbf{E}} - \mathbf{r}_{\mathbf{S}}|^3} (\mathbf{r}_{\mathbf{E}} - \mathbf{r}_{\mathbf{S}}) + G \frac{M_M}{|\mathbf{r}_{\mathbf{E}} - \mathbf{r}_{\mathbf{M}}|^3} (\mathbf{r}_{\mathbf{E}} - \mathbf{r}_{\mathbf{M}}) \end{aligned} \quad (6.4.3)$$

The last two terms of Equation (6.4.3) are derived from the fact that an Earth-Centered frame isn't inertial, due to the gravitational pull of the Sun, the Moon and all the other celestial bodies in the Solar system. Clearly, for our purposes, it's sufficient to account just for the Sun and Moon effect.

In addition, when solving an N-Body Problem it would normally be necessary to derive the position vectors of every celestial body involved, but this can be avoided if their position is instead extracted from an external ephemerides data sheet (provided by SPICE in our case). We then only have to worry about propagating the spacecraft position.

State Vector at Injection Point in Earth-Centered J2000

The first thing we have to do is positioning the injection point. Always under the assumption that we are initially on a circular parking orbit of radius r_0 and that the transfer trajectory is a pseudo-Hohmann Transfer:

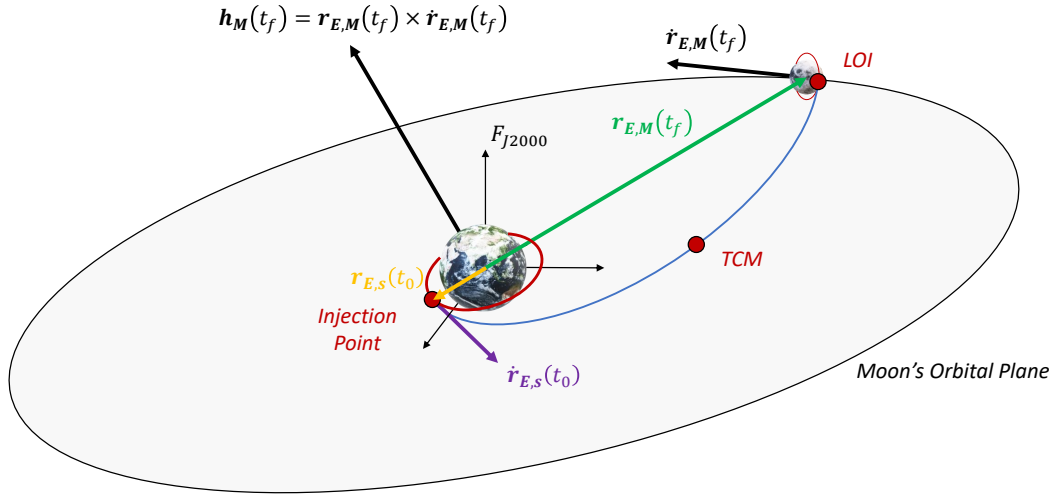


Figure 6.5: Lunar Targeting in the J2000 Frame

$$\mathbf{r}_{\mathbf{E},s}(t_0) = -\frac{\mathbf{r}_{\mathbf{E},M}(t_f)}{r_{\mathbf{E},M}(t_f)} r_0 \quad (6.4.4)$$

That is, the starting point is diametrically opposite to the Moon at the time of arrival t_f . It remains to compute the initial speed vector $\dot{\mathbf{r}}_{\mathbf{E},s}$ of the spacecraft, but the process is pretty straightforward even in this case:

$$\mathbf{h}_M(t_f) = \mathbf{r}_{\mathbf{E},M}(t_f) \times \dot{\mathbf{r}}_{\mathbf{E},M}(t_f) \quad (6.4.5)$$

$$\dot{\mathbf{r}}_{\mathbf{E},s}(t_0) = \frac{\mathbf{h}_M(t_f) \times \mathbf{r}_{\mathbf{E},s}(t_0)}{|\mathbf{h}_M(t_f) \times \mathbf{r}_{\mathbf{E},s}(t_0)|} V_H$$

where:

$$V_H = \sqrt{\frac{GM_E}{r_0}} \sqrt{\frac{2r_{\mathbf{E},M}(t_f)}{2(r_0 + r_{\mathbf{E},M}(t_f))}} \quad (6.4.6)$$

Useful SPICE functions

The SPICE Toolkit provides users with a series of routines that can be easily used to derive all the variables that we are interested in. In particular, for our case, we employ:

- the `cspice_spkezr` function to obtain the position and speed vectors of the various celestial bodies. It's possible to specify the *reference frame* in which these vectors will be expressed (in our case the J2000 frame) and the *observing body* relative to which they are computed (Earth in our case);
- the `cspice_pxform` routine to rotate the Moon's Equatorial plane orthogonal unit vector (expressed as $\hat{k} = [0 \ 0 \ 1]^T$ in the The Mean Earth/Polar Axis (ME) frame) in the J2000 frame, to impose the inclination constraint at arrival;

Their documentation is available on the NAIF SPICE website. [21] [20]

SPICE Kernels

Using SPICE commands is very straightforward after reading their related documentation. In fact, the main difficulty is more about finding all the required kernel files for a certain problem, where a kernel is essentially a text or binary file containing some type of data (ephemeris data, reference frames orientations...), depending on the type of kernel in question. Again, to better understand how to correctly use this tool, it's indeed better to refer directly to the SPICE documentation. In any case, the list of kernels employed is the following:

- `naif0012.tls` → Generic Leapseconds kernel.
- `de440.bsp` → Planetary Ephemeris kernel (includes Moon ephemeris).
- `moon_pa_de440_200625.cmt.bpc` → Moon Orientation Data kernel.
- `moon_de440_220930.tf` → Moon Reference Frames kernel.

6.5 The LOI Maneuver

The execution of the TCM Maneuver involves solving a NLP problem as explained in the previous section while the computation of the LOI maneuver, in a very similar fashion to the computation of the initial state vector, can be performed analytically, with simple geometrical considerations and vector operations, which can be easily understood by looking at Figure (6.6). The tangential unit-vector $\hat{\mu}_t$ is given by:

$$\hat{\mu}_t = \frac{(\mathbf{r}_{M,s}(t_f) \times \dot{\mathbf{r}}_{M,s}(t_f)) \times \mathbf{r}_{M,s}(t_f)}{|(\mathbf{r}_{M,s}(t_f) \times \dot{\mathbf{r}}_{M,s}(t_f)) \times \mathbf{r}_{M,s}(t_f)|} \quad (6.5.1)$$

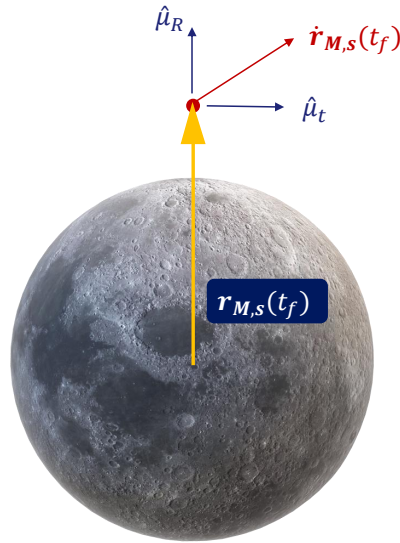


Figure 6.6: LOI Maneuver Geometry.

while the speed of a circular orbit of radius $\mathbf{r}_{M,s}(t_f)$ around the Moon, according to the R2BP, can be computed as:

$$v_c = \sqrt{\frac{GM_M}{r_{M,s}(t_f)}} \quad (6.5.2)$$

so the speed vector after the LOI maneuver ($t_f = t_{LOI}$) is:

$$\dot{\mathbf{r}}_{M,s}(t_{LOI}) = v_c \hat{\mu}_t \quad (6.5.3)$$

Notice that this is a *Moon Relative Speed*, which has to be properly converted to obtain a well propagated circular trajectory. For example, the *synodic frame relative speed* would be given by:

$$\dot{\mathbf{r}}_s = \dot{\mathbf{r}}_{M,s} - \boldsymbol{\omega} \times \mathbf{r}_{M,s} \quad (6.5.4)$$

while in the *Earth centered J2000 frame* the relative speed is:

$$\dot{\mathbf{r}}_{E,s} = \dot{\mathbf{r}}_{M,s} + \dot{\mathbf{r}}_{E,M} \quad (6.5.5)$$

6.6 SPICE-Enhanced COSMOGRAPHIA

COSMOGRAPHIA is an open source visualization tool used to produce 3D visualizations of planet ephemerides, sizes and shapes, spacecraft trajectories and orientations and instrument fields-of-view and footprints. NAIF offers a SPICE-Enhanced version of COSMOGRAPHIA, employing SPICE Kernel files to realistically propagate the trajectories of the various celestial bodies in the solar system. The documentation and templates available on the COSMOGRAPHIA and NAIF website allow us to visualize our trajectories with moderate effort. [23] [24] There are various ways to export a trajectory in COSMOGRAPHIA, such as transforming it in a SPICE kernel file, but the method we are going to use is that of recurring to a ".xyzv" file; it's essentially a text file made up of seven columns of data, in which:

- the first column contains a time value expressed in TDB Julian Days (JD), i.e. days since I Jan 4713 a.C 12:00:00 (TDB);
- the subsequent three columns contain the x,y e z coordinates of a point of the trajectory associated to the JD in the first column, expressed in a certain frame and relatively to an observing body, both specified in a catalog .json file.
- the latest three columns contain the v_x , v_y and v_z speed components expressed in a certain frame and relatively to an observing body, both specified in the same previous .json catalog file.

The `cspice_timeout` function can be used to convert ephemeris time ("et", number of seconds since J2000 i.e. 1 Jan 2000 12:00:00 TDB), used in the `cspice_spkpos` and `cspice_pxform` functions too. It's then convenient to propagate the trajectory using ephemeris time in MATLAB to easily visualize it in COSMOGRAPHIA. The documentation of the `cspice_timeout` function is again available on the NAIF website, to correctly understand its use. However, it's important to remember that the output time must be specified as Julian Day (JD) in TDB, as follows:

- `cspice_timeout(et, 'JULIAND.#####:TDB')`

In addition, the output is in this case a string, so the `str2double` MATLAB function can be employed to convert it in a double precision number.

A template of the .json catalogue file is available in Appendix F

Chapter 7

Results

Having concluded the theoretical discussion of all the different notions required to study the transfer and descent trajectories, it's finally possible to present some of the results produced with the tools developed as part of this thesis work at TAS-I. For the optimization of descent trajectories it's not possible to present real data due to a non-disclosure agreement. Some fictitious configurations, that are anyway representative of the real problem and of the type of results we can expect, were therefore analyzed instead.

7.1 Descent Trajectory Optimization

The objective of this Section isn't that of performing trade-off analyses, but to validate the descent tool. For this reason, tests will be carried out as certain parameters of interest change, such as maximum thrust and initial mass of the lander (thus acting on the thrust to mass ratio) , checking how this affects the final shape of the trajectory and verifying that the results are compatible with the ones already known in literature. Then, to further validate the descent model and increase the robustness of the analyses, a comparison with the results obtained from a slightly modified version of a 2D descent tool, previously developed at TAS-I, is provided too. This 2D tool makes use of the dynamics equations given in Appendix C to study the descent trajectory in a perifocal frame.

It is worth specifying that, since we are not interested in diverting maneuvers, the bank angle is set equal to zero. It was also noted that the solutions quality improves if throttle is fixed at its maximum value, and computation time is reduced too. Finally, all of the analyses presented herein were performed with constraints on the initial and final conditions compatible with the mission strategy outlined in Chapter 1, such as:

- initial height and speed compatible with the perilunium of a 30 x 100 km orbit;
- vertical lander configuration at arrival ($\epsilon = \pi$ and $\gamma \approx -90$ deg);
- reduced speed at arrival ($< 2\text{m/s}$).

7.1.1 Convergence Study

The number of nodes used in the direct collocation algorithm is gradually increased to verify convergence of the final optimized mass and of the control and state variables. This allows us to check the numerical stability of the solution and get an idea of its accuracy. The results are shown in Figures (7.1) and (7.2), for the following lander configuration:

- $T = 5 \times 4 \text{ kN} = 20 \text{ kN}$ (five engines configuration).
- Initial Mass $m_0 = 6000 \text{ kg}$.
- $I_{sp} = 400 \text{ s}$

It can be noted that the final values of mass and descent time are somewhat accurate even with an extremely reduced number of nodes (10), and it is clearly possible to see a convergence of all variables of interest as the number of nodes increases. In particular, the final value of the optimized mass increases slightly as the number of nodes increases, thanks to a refinement in system control, but the variations are extremely marginal. For this reason, the following analyses will be performed with an intermediate number of nodes, 50, to reduce computational costs without excessively compromising accuracy.

7.1.2 Effect of Maximum Thrust - T_{MAX}

A total of four configurations, with a different number of engines, and different maximum levels of thrust, have been tested. The parameters describing these configurations are given in Table (7.1), and the number of nodes used for the optimization is 50.

The main features of the optimized trajectories, as shown in Figures (7.3) and (7.4), are:

- an increase in time required to land as available thrust reduces, as indeed was to be expected;

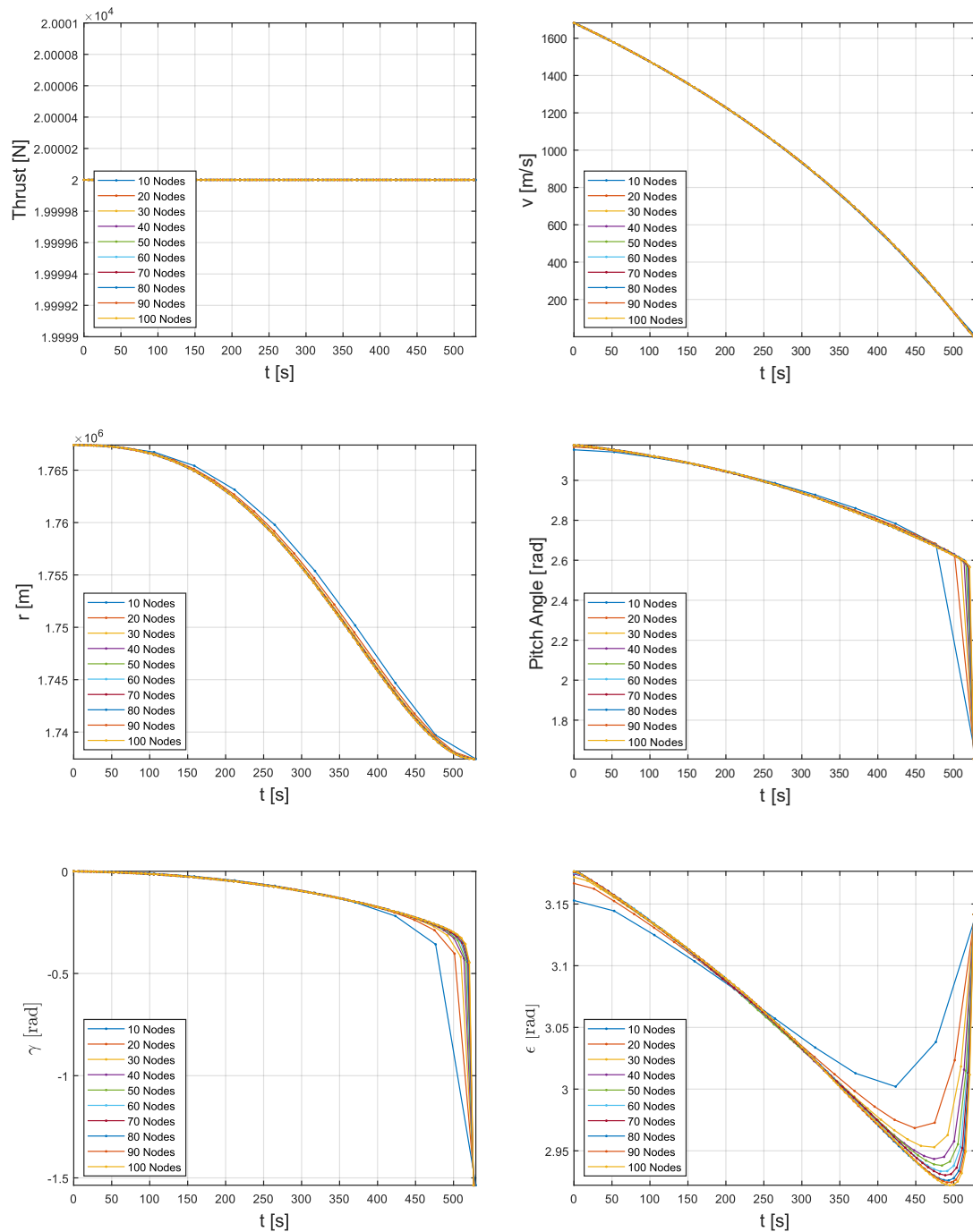


Figure 7.1: Convergence Study of the Direct Collocation Algorithm.

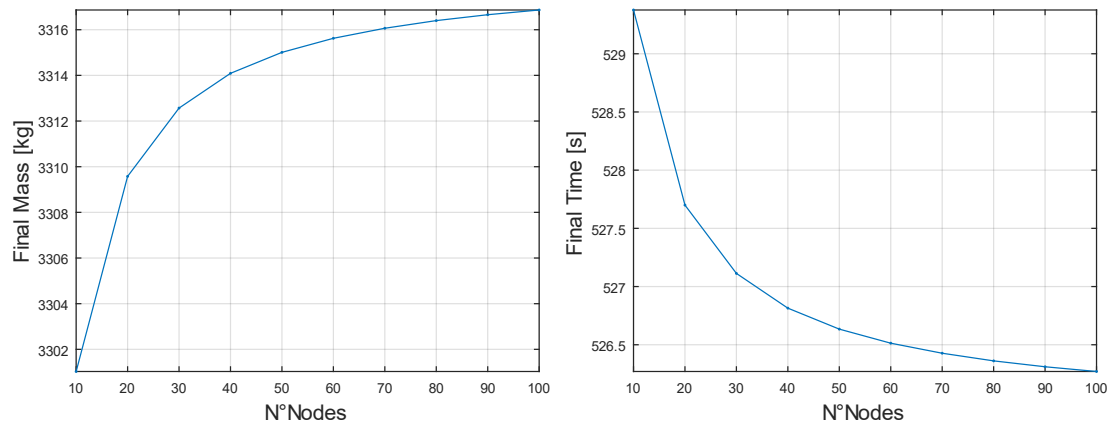


Figure 7.2: Convergence Study of Final Lander Mass and Final Descent Time

Table 7.1: Lander configurations and results for the maximum thrust test.

Configuration	1	2	3	4
Initial Mass [kg]	6000	6000	6000	6000
Thrust per Engine [kN]	4	4	4	4
N° Engines	5	4	3	2
I_{sp} [s]	400	400	400	400
Thrust to Mass Ratio [N/Kg]	10/3	8/3	2	4/3
Landing Time [s]	526.6	665.9	912.4	1488.8
Final Mass [kg]	3315	3284	3209	2964
Propellant Burned [kg]	2685	2716	2791	3036
Lander Efficiency $\frac{m_f}{m_0}$	0.553	0.547	0.535	0.494

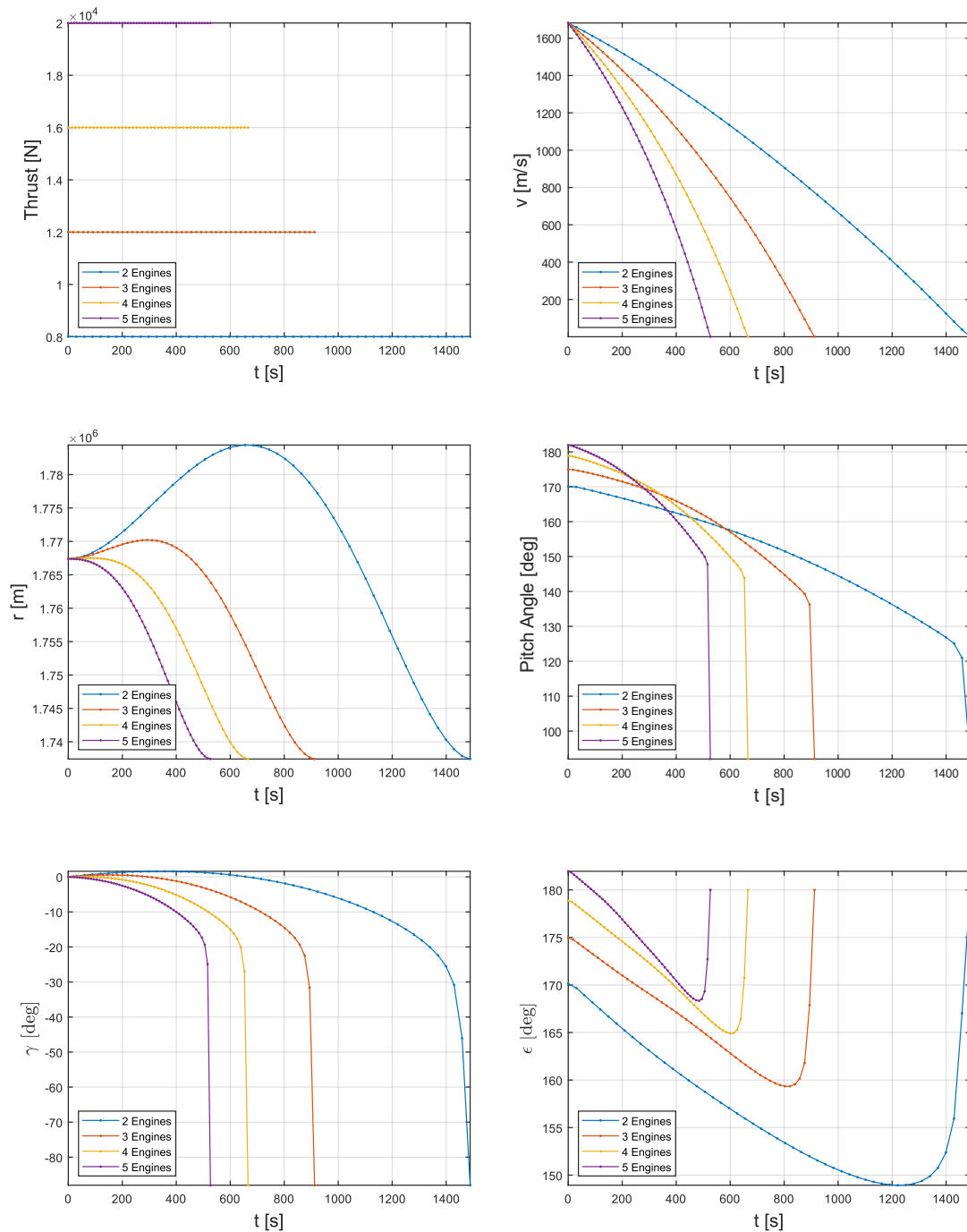


Figure 7.3: Maximum Lander Thrust Tests.

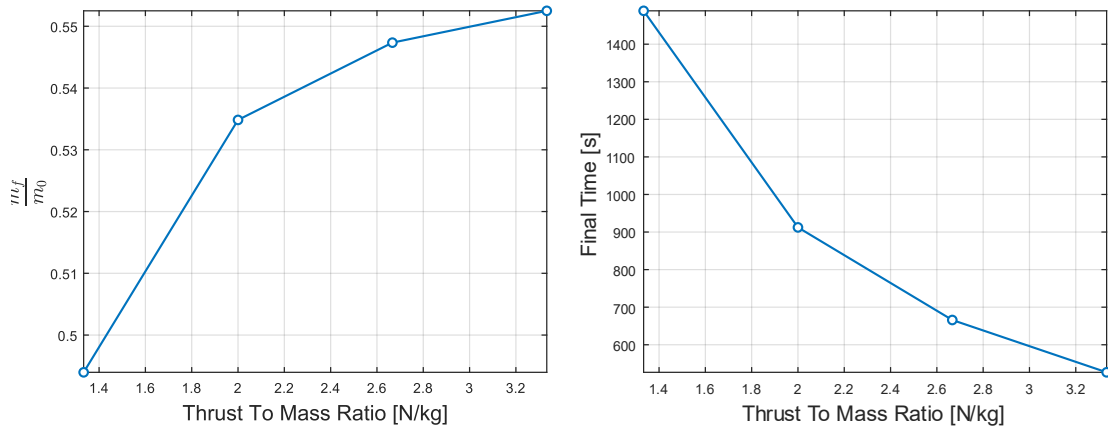


Figure 7.4: Lander efficiency and descent time for different values of the thrust to mass ratio.

- an increase in final mass at landing as available thrust increases, due to a reduction in gravitational losses;
- an initial increase in trajectory height when the available thrust falls under a certain threshold. This also reflected in values of γ greater than 0 at the beginning.

These results are in fact compatible with those of other studies available in literature.[7]. Obviously, to save up on fuel, it's more convenient to have a higher amount of trust at disposal or, more precisely, a higher *thrust to mass ratio*, as shown in the results provided in the next section.

7.1.3 Effect of Initial Lander Mass - m_0

The same behavior shown in Section 7.1.2 can be replicated by changing the initial value of the lander mass, with a fixed available thrust, as can be seen in Figure (7.6). In fact, not only have the trajectories of the various landers got the same shape when their thrust to mass ratio is equal, but their landing time and efficiency are the same too, as seen in Tables (7.1) and (7.2). It is clear from these analyses that, all other performance parameters being equal, a higher value of the thrust to mass ratio is advantageous, at least within certain limits; in fact, as can be seen in Figure (7.4), the efficiency value tends to grow less and less rapidly as the thrust to mass ratio increases. In any case, this parameter is therefore of utmost importance in determining the final performance of the lander.

The fact that the results obtained are identical can be explained analytically in a very simple way. By looking at the kinematics and dynamics equations in Section 3.4, it's easy to notice that, all other parameters being equal, if the thrust to mass ratio is identical then the ODEs are equal too.

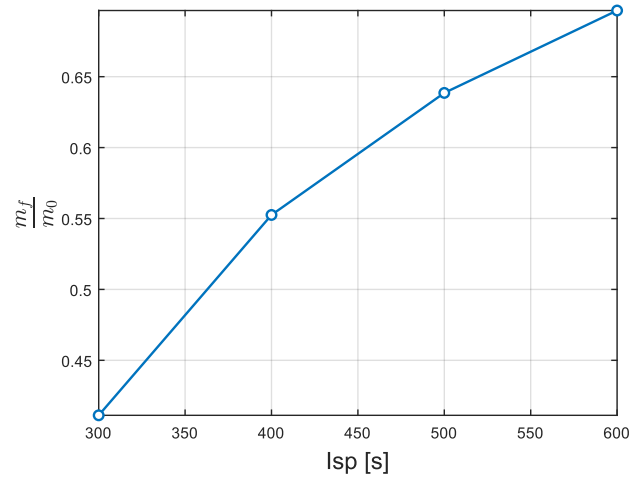


Figure 7.5: Specific Impulse Tests.

7.1.4 Effect of Specific Impulse - I_{sp}

An increase in specific impulse, on the other hand, would undoubtedly correlate with a reduction in propellant consumed, as shown in Figure (7.5). This is for two reasons:

- a higher I_{sp} value means less propellant burnt with equal thrust;
- since higher I_{sp} values imply a smaller (in modulus) \dot{m} , and due to the fact that $m\ddot{\mathbf{r}} = \mathbf{F} - \dot{m}\dot{\mathbf{r}}$, the effect of the $\dot{m}\dot{\mathbf{r}}$ term, especially relevant at high speeds and opposing the decelerating action of \mathbf{F} , becomes less.

This second effect is very important; it's common to forget about the \dot{m} term when deriving the dynamics equations, but this can, in some cases, lead to a severe overestimation of the final optimized lander mass.

However, since there is often little flexibility in the choice of engines, perhaps supplied by outside companies with a certain maximum achievable performance, it is difficult to be able to go beyond certain values of I_{sp} .

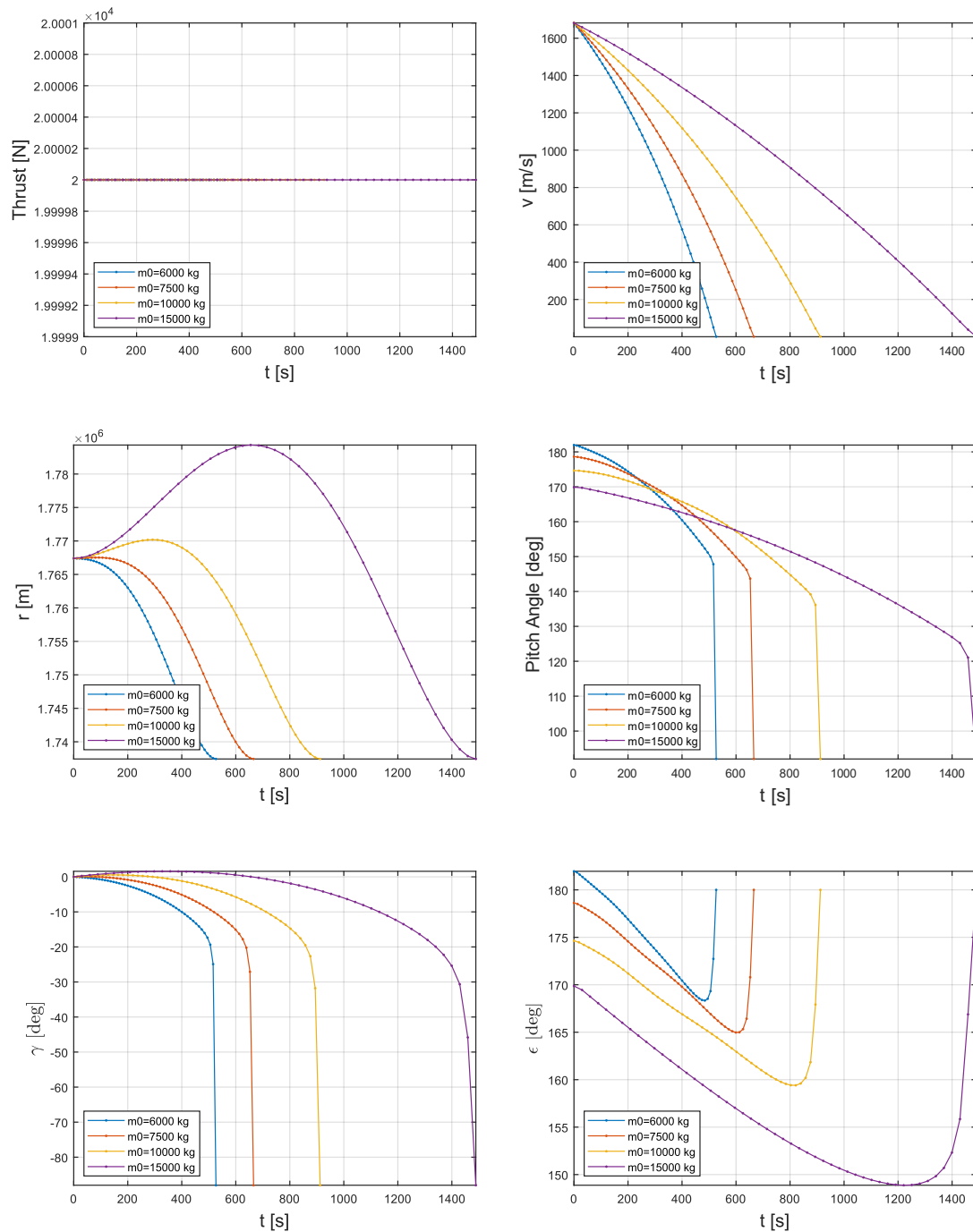


Figure 7.6: Lander Initial Mass Tests.

Table 7.2: Lander configurations and results for the initial mass test.

Configuration	1	2	3	4
Initial Mass [kg]	6000	7500	10000	15000
Thrust per Engine [kN]	4	4	4	4
N° Engines	5	5	5	5
I _{sp} [s]	400	400	400	400
Thrust to Mass Ratio [N/Kg]	10/3	8/3	2	4/3
Landing Time [s]	526.6	665.9	912.4	1488.8
Final Mass [kg]	3315	4105	5348	7410
Propellant Burned [kg]	2685	3395	4652	7590
Lander Efficiency $\frac{m_f}{m_0}$	0.553	0.547	0.535	0.494

7.1.5 2D and 3D Model Comparison

To further increase the robustness of the analyses, a comparison between the 2D descent model previously developed at TAS-I and the 3D model developed as part of this thesis has been performed. The lander configuration and results are presented in Table (7.3) and Figure (7.7). The final results obtained with both tools show great accordance between them.

Table 7.3: 2D and 3D descent models comparison.

	2D	3D
Initial Mass [kg]	6000	
Thrust Per Engine [kN]	4	
N°Engines	5	
I _{sp} [s]	400	
Thrust to Mass Ratio [N/kg]	10/3	
Landing Time [s]	528.7	526.6
Final Mass [kg]	3304	3315
Propellant Burned [kg]	2696	2685
Range [km]	506.3	507.4

7.1.6 Constraints on Maximum Pitch Rate

It's possible to adopt two approaches:

- given $Pitch = \gamma + \epsilon$ (this relation is valid because $\sigma = 0$), approximate its first derivative using finite differences and impose a constraint on this quantity;

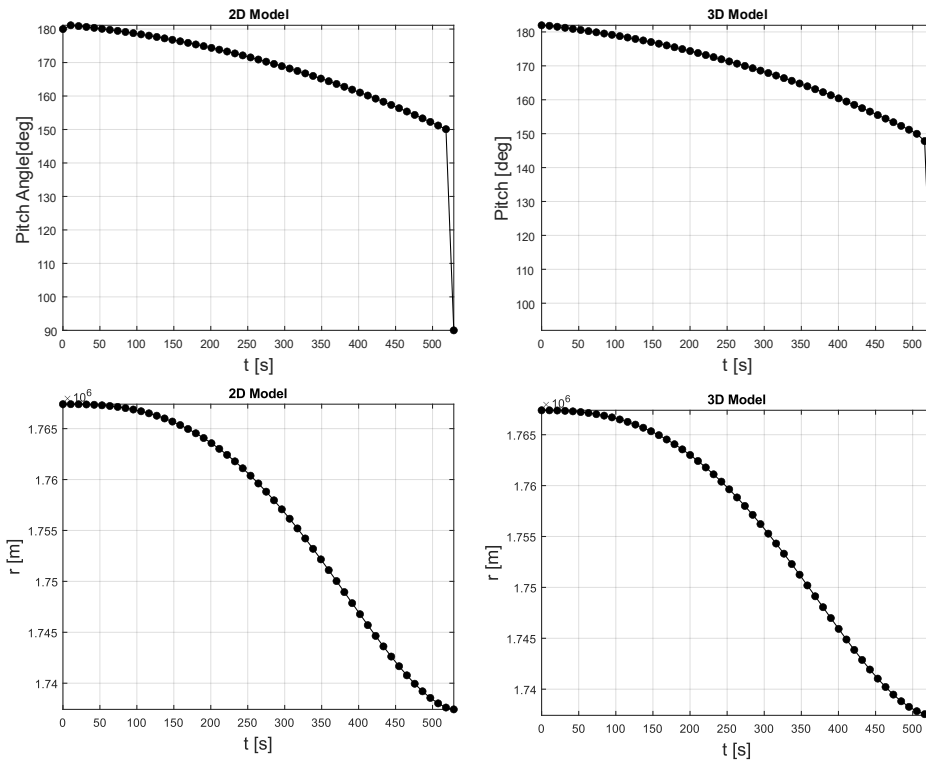


Figure 7.7: 2D and 3D Model descent comparison

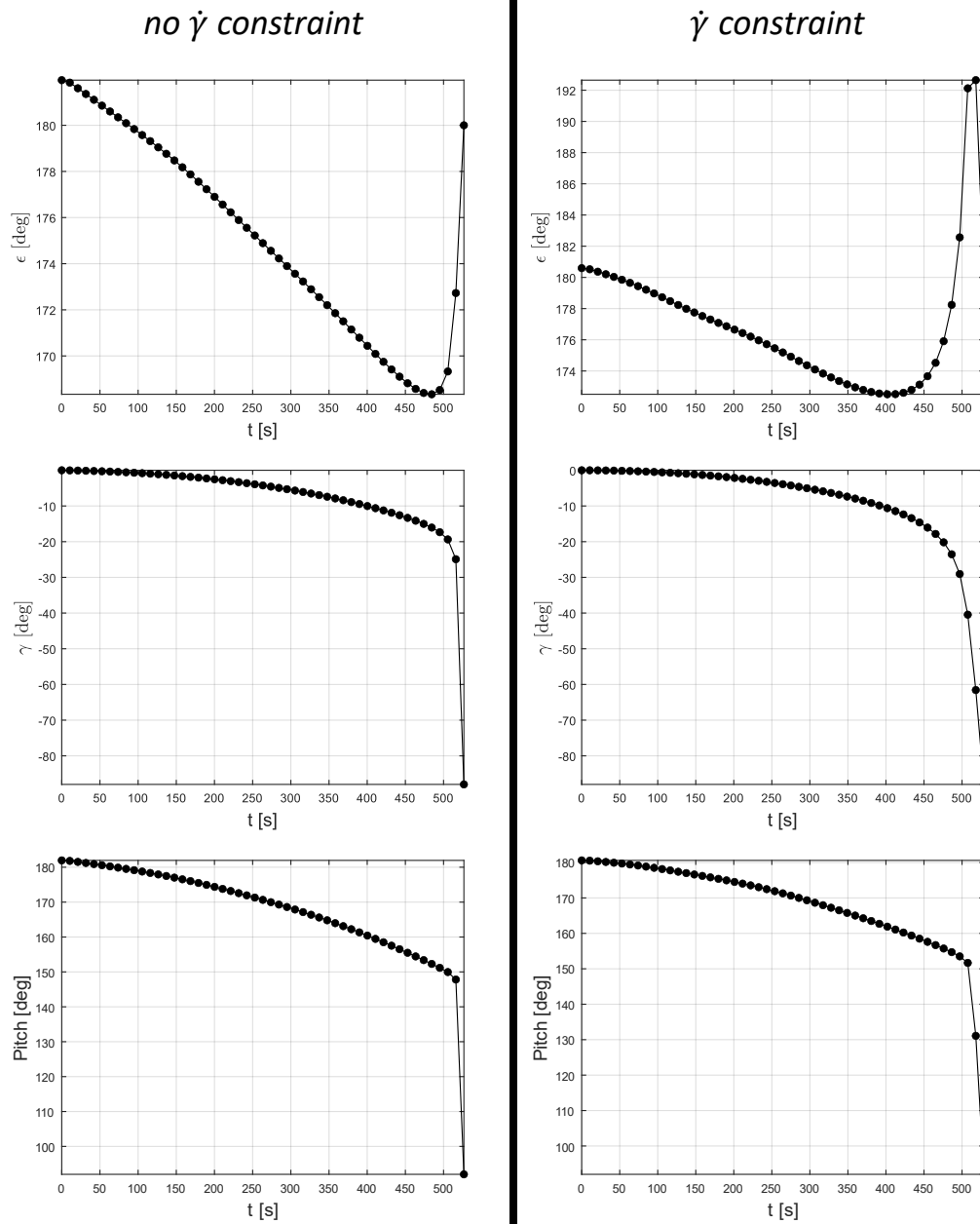
- impose a constraint on $\dot{\gamma}$, which is one of the outputs of the $\mathbf{f}(\mathbf{x}, \mathbf{u}, t)$ function, and verify a posteriori that the Pitch rate constraint is satisfied.

The second approach was the one adopted while performing analyses in presence of a maximum Pitch Rate constraint, because the quality and regularity of the solutions was observed to be superior. A comparison of the results obtained with and without a $|\dot{\gamma}| < 2.5$ deg/s constraint is shown in Figure (7.8) and Table (7.4). The objective was to keep the maximum pitch rate below 5 deg/s. Observe that the results shown in Table (7.4) for the $\dot{\gamma}$ constraint case are very similar to those of the 2D Model (where no $\dot{\gamma}$ constraints were imposed), but this is completely accidental.

7.1.7 3D Trajectory Example

Finally, a 3D plot of the optimized trajectory obtained for the configuration used in Section 7.1.5 is provided in Figures (7.9) and (7.10).

The descent trajectory is compatible with a polar waiting orbit, in accordance with the mission overview given in Chapter 1. The tool is also capable of handling passages above the poles of the celestial body of interest without incurring in singularities, thanks to the frame rotation procedure introduced in Chapter 3.

Figure 7.8: Effect of the $\dot{\gamma}$ constraint.

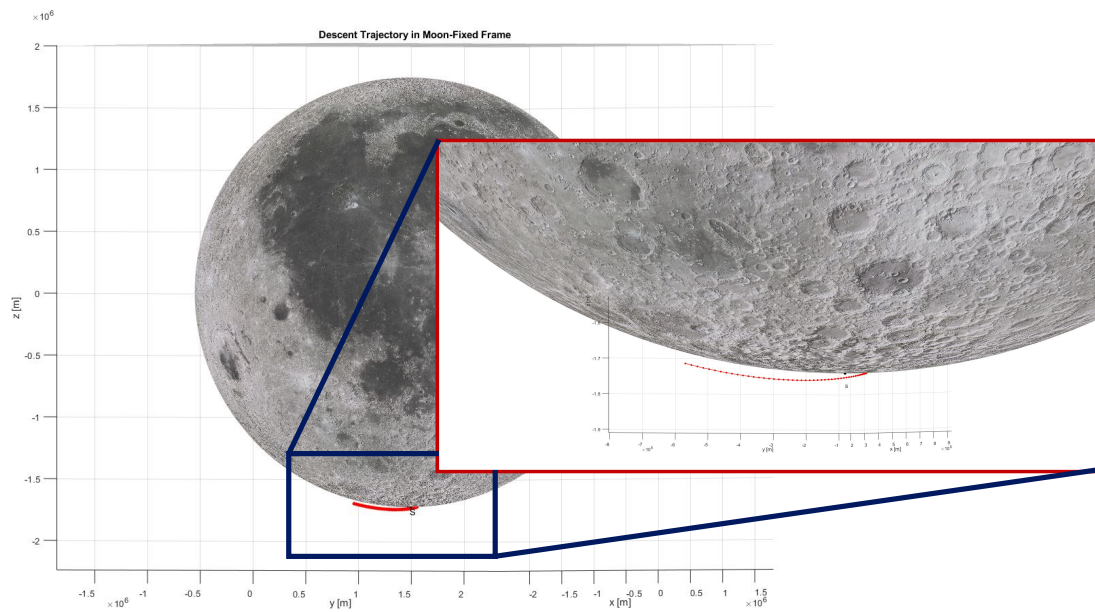


Figure 7.9: 3D Descent Trajectory Plot

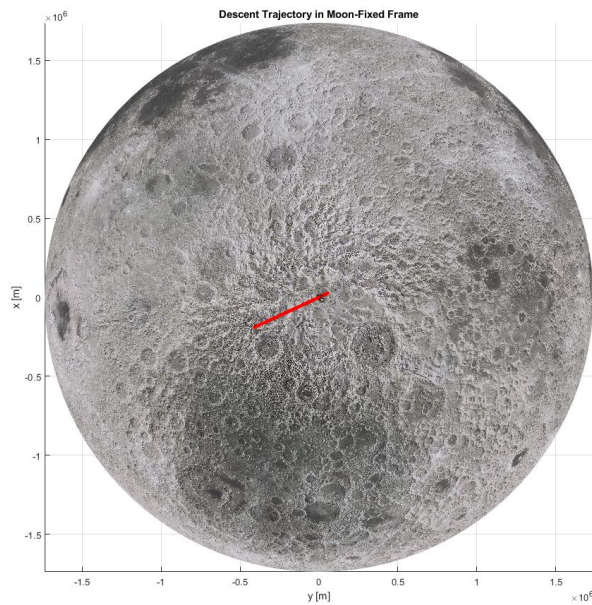


Figure 7.10: 3D Descent Trajectory Plot. View from above the South-Pole

Table 7.4: Final Results with and without the $\dot{\gamma}$ constraint.

	$\dot{\gamma}$	No $\dot{\gamma}$
Initial Mass [kg]	6000	
Thrust Per Engine [kN]	4	
N°Engines	5	
I_{sp} [s]	400	
Thrust to Mass Ratio [N/kg]	10/3	
Landing Time [s]	528.7	526.6
Final Mass [kg]	3304	3315
Propellant Burned [kg]	2696	2685
Range [km]	505.3	507.4
Max Pitch Rate reached [deg/s]	3.7	5.3

7.2 Transfer Trajectory Optimization

As already stated in the previous chapters, two different dynamic models, the Spice Enhanced N-Body Problem and the CR3BP, have been used to compute the optimal TCM, employing in both cases a direct single shooting algorithm. The results obtained are presented in this section, highlighting and justifying any differences.

7.2.1 TCM and LOI using the CR3BP

This approach is characterized by a low computational cost, allowing to compute almost 2000 TCMs in just a couple of hours or less, depending on the processing power at disposal. The results obtained for all of the arrival dates provided by TAS-I are given in Figures (7.12) and (7.13). According to the CR3BP, one should expect a cost of about 815 m/s for the LOI Maneuver and from 20 to 45 m/s, if performed at the 24h mark. One could notice, by observing Figure (7.11), that performing the TCM maneuver both too early and too late leads to an increase in its cost. In addition, there is a certain time interval before the 24 hours mark when the cost of the maneuver is lower, albeit slightly. These values are in line with those provided by the European Space Agency so, for a very preliminary analysis, they could be considered sufficient. It is unlikely, however, that, after introducing the perturbative effect of the Sun and the real orbit of the Moon, the numbers will be the same. In addition, the CR3BP doesn't allow for a correct targeting of the LLO inclination, because the coordinates of the unit-vector normal to the equatorial plane aren't known as already discussed. For all of these reasons, to increase precision, it's necessary to recur to a more accurate model, using data provided by SPICE.

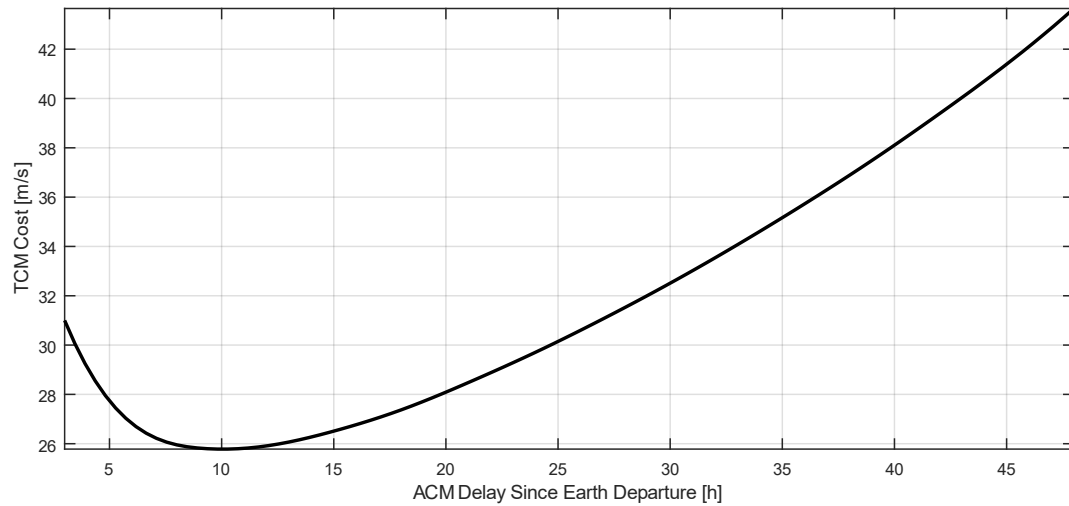


Figure 7.11: Parametric Study of the TCM Delay vs ΔV cost, for one of the possible arrival dates.

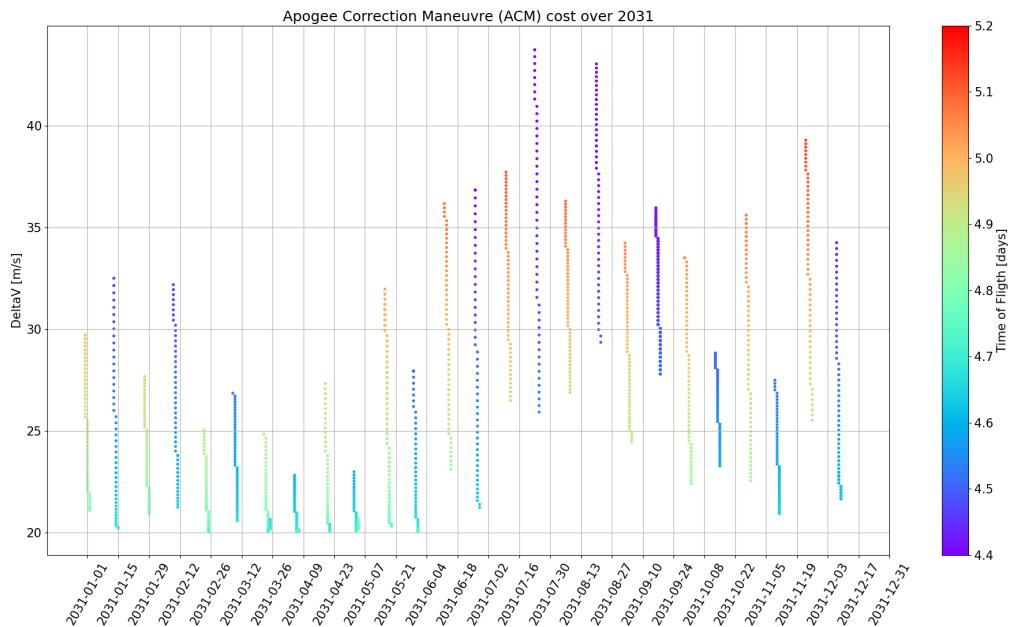


Figure 7.12: ACM (TCM) Maneuver Cost over 2031 (Departure Dates)

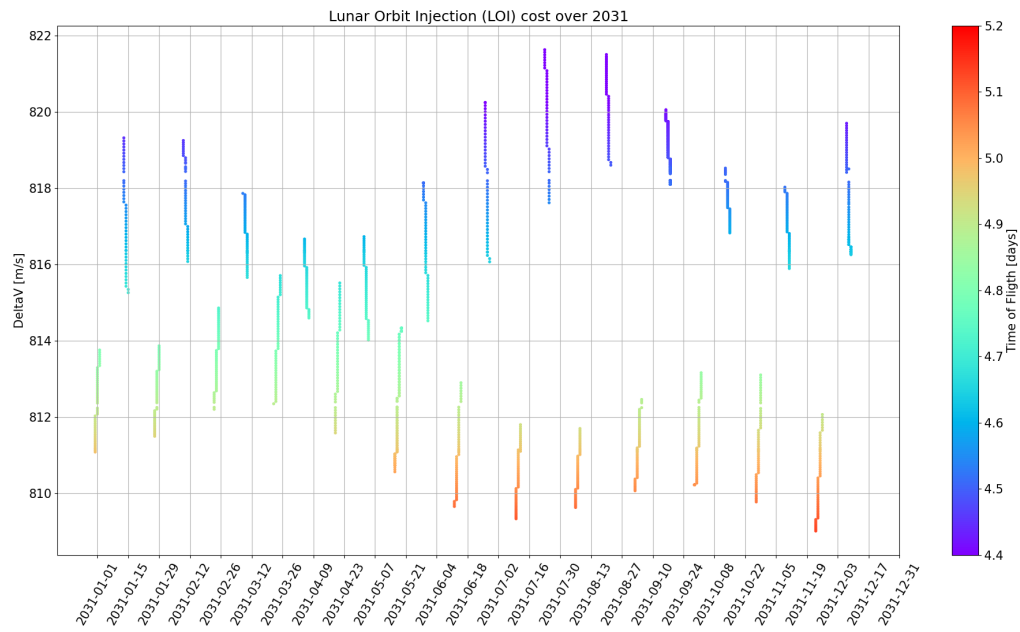


Figure 7.13: LOI Maneuver Cost over 2031 (Departure Dates)

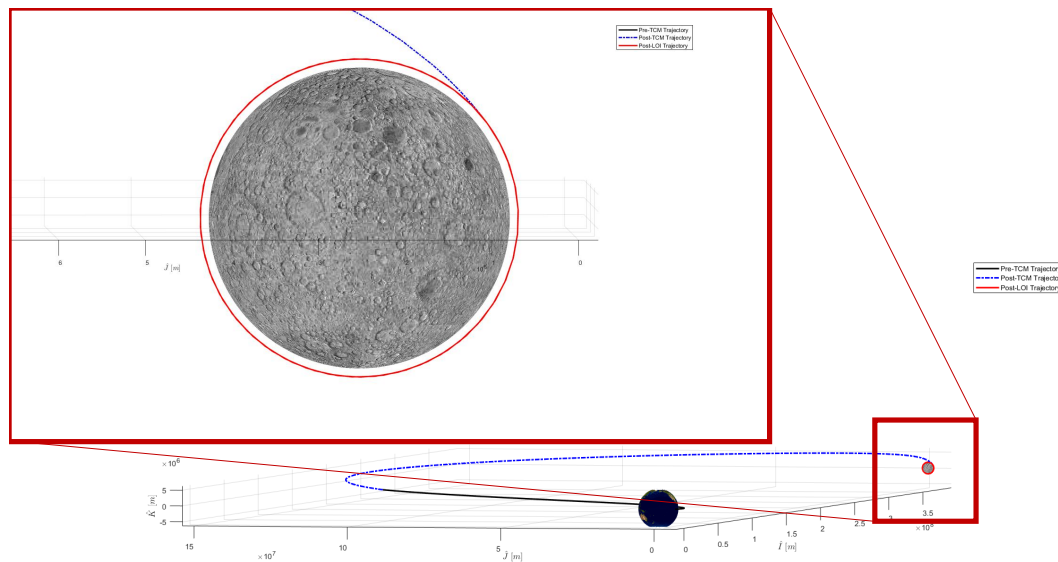


Figure 7.14: Example of trajectory in the Synodic Frame.

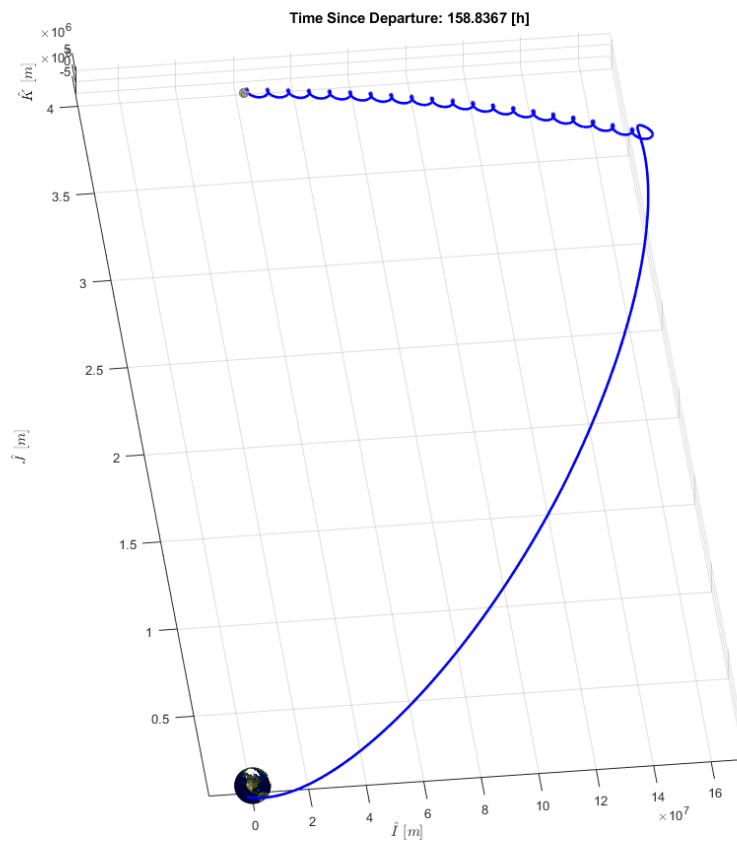


Figure 7.15: Example of trajectory in a Non-Rotating Frame. (CR3BP)

Table 7.5: Cost of LOI and TCM Maneuvers for various Departure Dates, using SPICE.

Departure (UTC)	Arrival (UTC)	ΔV_{TCM} [km/s]	ΔV_{LOI} [km/s]
01 Jan 2031 23:56	07 Jan 2031 03:27	0.019	0.800
17 Jan 2031 00:11	22 Jan 2031 07:17	0.074	0.846
31 Jan 2031 00:11	05 Feb 2031 03:42	0.021	0.801
11 Mar 2031 22:25	17 Mar 2031 04:34	0.056	0.831
25 Mar 2031 16:15	30 Mar 2031 19:23	0.054	0.831
23 Apr 2031 18:26	28 Apr 2031 21:33	0.019	0.799
19 May 2031 14:14	24 May 2031 17:18	0.054	0.829
20 May 2031 16:50	25 May 2031 19:36	0.019	0.800
16 Jun 2031 13:57	21 Jun 2031 17:09	0.019	0.802
29 Jun 2031 07:29	04 Jul 2031 14:43	0.019	0.801
09 Aug 2031 02:35	14 Aug 2031 06:07	0.078	0.849
22 Aug 2031 08:52	27 Aug 2031 15:12	0.019	0.801
07 Sep 2031 01:54	12 Sep 2031 05:16	0.019	0.797
18 Sep 2031 18:11	23 Sep 2031 23:51	0.060	0.832
18 Sep 2031 22:03	24 Sep 2031 03:46	0.019	0.798
19 Sep 2031 17:11	24 Sep 2031 23:08	0.019	0.798
19 Sep 2031 20:58	25 Sep 2031 03:25	0.056	0.831
03 Oct 2031 21:55	09 Oct 2031 01:10	0.064	0.840
12 Nov 2031 17:43	18 Nov 2031 00:01	0.020	0.804
26 Nov 2031 10:56	01 Dec 2031 14:01	0.055	0.830

7.2.2 TCM and LOI using SPICE

The computational cost is in this case much higher, and a workstation is needed to perform all of the optimizations in a reasonable amount of time. Because of this, only some reference results will be provided to understand, indicatively, how far we deviate from the previous model.

Some of the values obtained for the cost of the TCM and LOI Maneuver are given in Table (7.5), and they can differ by as much as several tens of meters per second from the ones obtained through the CR3BP, but that was to be expected since the two models are quite different dynamically. Further investigation, perhaps comparing these results with the ones obtained through commercial mission analysis software (e.g. STK), is needed to ensure their correctness, despite being, however, still in line with the reference values provided by ESA. Finally, an example of a trajectory obtained in this environment is visualized in Figures (7.16) and (7.17).

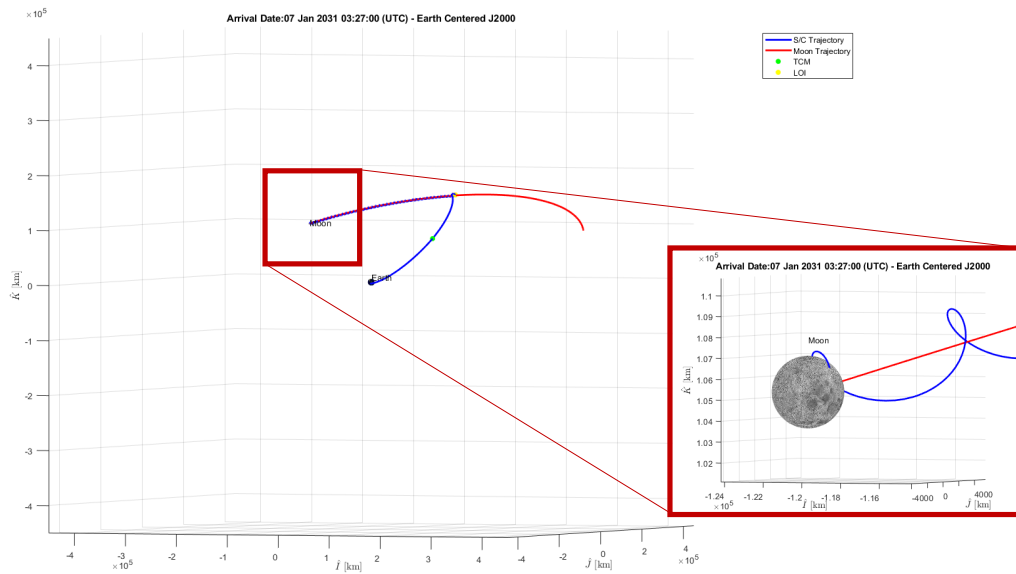


Figure 7.16: Transfer Trajectory in Earth-Fixed J2000.

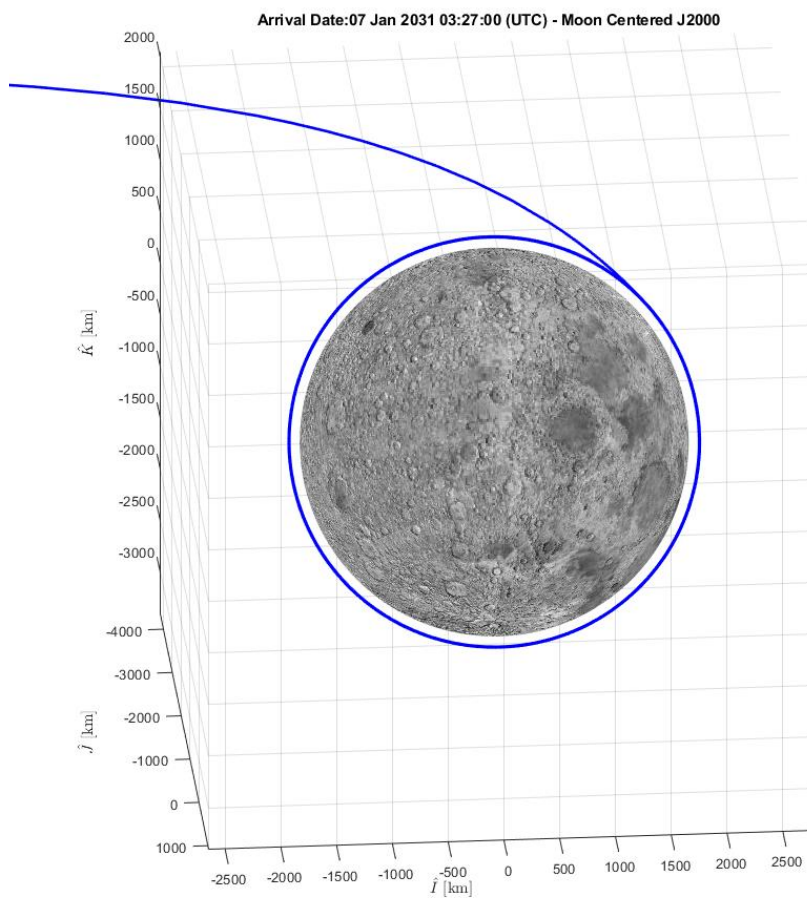


Figure 7.17: Arrival Trajectory in Moon-Fixed J2000.

Conclusions

In conclusion, this thesis has successfully tackled the challenging issues related to Moon Targeting and descent trajectory optimization by employing various dynamic models. The primary aim was to develop a comprehensive toolkit capable of efficiently handling these two critical aspects while establishing a connection between them, ensuring the compatibility of transfer trajectories with subsequent descent phases.

The outcomes of this research are aligned with the findings presented by the European Space Agency and existing literature. However, the effort does not end here. Future developments should encompass a more comprehensive comparison with commercial mission analysis software, with particular reference to the calculation of the TCM maneuver, allowing for a more robust validation of the developed tools.

Moreover, an essential direction for further research lies in strengthening the integration of Moon targeting and descent trajectory optimization, moving away from treating them as distinct phases. This future work should involve exploring methods to determine the optimal initiation time for descent after a specific transfer trajectory, ensuring that all constraints outlined by ESA and other relevant stakeholders are met seamlessly. Finally, a more accurate model of the Moon gravitational field could be implemented too, to better study the descent trajectory of the lander, and adaptive direct collocation methods could be explored too, to increase accuracy.

Appendices

Appendix A

Spherical Trigonometry

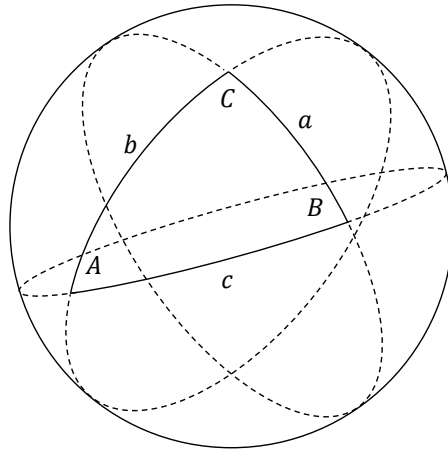


Figure A.1: Angles of a spherical triangle.

Law of Cosines

$$\begin{aligned}\cos a &= \cos b \cos c + \cos A \sin b \sin c \\ \cos b &= \cos a \cos c + \cos B \sin a \sin c \\ \cos c &= \cos a \cos b + \cos C \sin a \sin b\end{aligned}\tag{A.0.1}$$

Law of Sines

$$\frac{\sin a}{\sin A} = \frac{\sin b}{\sin B} = \frac{\sin c}{\sin C}\tag{A.0.2}$$

Appendix B

Elementary Rotation Matrices

Rotation matrices are widely used in this thesis, so it may be appropriate to briefly review the topic to better understand the mathematical procedures presented.

A rotation matrix is a matrix that allows to transform the coordinates expressed in a reference frame in those of another (in case they have the same origin and they only differ in their orientation, otherwise a translation would be necessary too). The basic concept is then to project the axes of a reference frame on those of the other one, using direction cosines. Considering a transformation of coordinates expressed in a generic $\{\hat{I}_1, \hat{J}_1, \hat{K}_1\}$ frame in a $\{\hat{I}_2, \hat{J}_2, \hat{K}_2\}$ frame, the rotation matrix would be:

$$\begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = \begin{bmatrix} \hat{I}_1 \cdot \hat{I}_2 & \hat{J}_1 \cdot \hat{I}_2 & \hat{K}_1 \cdot \hat{I}_2 \\ \hat{I}_1 \cdot \hat{J}_2 & \hat{J}_1 \cdot \hat{J}_2 & \hat{K}_1 \cdot \hat{J}_2 \\ \hat{I}_1 \cdot \hat{K}_2 & \hat{J}_1 \cdot \hat{K}_2 & \hat{K}_1 \cdot \hat{K}_2 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} \quad (\text{B.0.1})$$

However computing these dot products isn't trivial at all in many cases, and it's more practical to perform the rotations using a sequence of intermediate reference frames, through the so-called Euler angles. There are three fundamental rotations, that can be combined to perform any generic frame rotation, with three corresponding matrices, as shown in Figure (B.1).

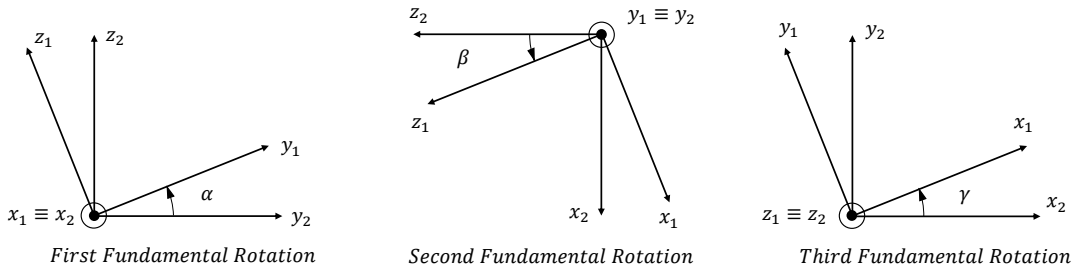


Figure B.1: Fundamental Rotations.

These matrices are:

$$R_I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha \\ 0 & \sin \alpha & \cos \alpha \end{bmatrix}$$
$$R_J = \begin{bmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{bmatrix} \tag{B.0.2}$$
$$R_K = \begin{bmatrix} \cos \gamma & -\sin \gamma & 0 \\ \sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Appendix C

Descent Dynamics in the Perifocal Reference Frame

In this Appendix we briefly derive the differential equations that describe the descent dynamics in the *perifocal reference frame* $\{\hat{p}, \hat{q}, \hat{w}\}$. The reader could then choose to implement these in a *Direct Collocation Algorithm*, and compare the results with those obtained through the more complex ones derived in Chapter 3.

The *perifocal reference frame* has already been described in Section 4.4, and for the present case is shown in Figure (C.1).

Let us start by expressing the position of the satellite in cartesian components:

$$\begin{aligned}x &= r \cos \nu \\y &= r \sin \nu \\z &= 0\end{aligned}\tag{C.0.1}$$

The motion is contained in the $\hat{p} - \hat{q}$ plane, so from now on we will discard the z component. By taking the first derivative we then get:

$$\begin{aligned}\dot{x} &= \dot{r} \cos \nu - r \sin \nu \dot{\nu} = v_r \cos \nu - v_t \sin \nu \\ \dot{y} &= \dot{r} \sin \nu + r \cos \nu \dot{\nu} = v_r \sin \nu + v_t \cos \nu\end{aligned}\tag{C.0.2}$$

We then study the second derivative:

$$\begin{aligned}\ddot{x} &= \ddot{r} \cos \nu - \dot{r} \sin \nu \dot{\nu} - \dot{r} \sin \nu \dot{\nu} - r \cos \nu \dot{\nu}^2 - r \sin \nu \ddot{\nu} \\ \ddot{y} &= \ddot{r} \sin \nu + \dot{r} \cos \nu \dot{\nu} + \dot{r} \cos \nu \dot{\nu} - r \sin \nu \dot{\nu}^2 + r \cos \nu \ddot{\nu}\end{aligned}\tag{C.0.3}$$

We now apply the second principle of dynamics, in a case where the mass is variable:

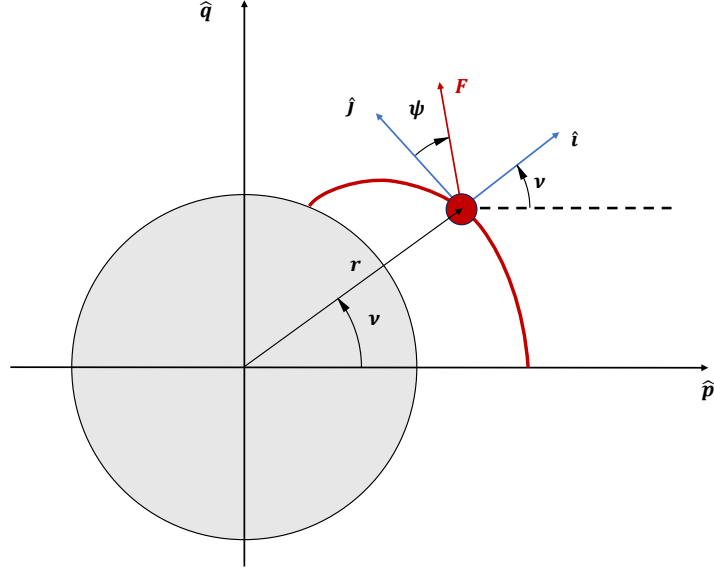


Figure C.1: Descent trajectory in perifocal reference frame.

$$\begin{aligned} m\ddot{x} + \dot{m}\dot{x} &= F_r \cos \nu - F_t \sin \nu \\ m\ddot{y} + \dot{m}\dot{y} &= F_r \sin \nu + F_t \cos \nu \end{aligned} \quad (\text{C.0.4})$$

We replace the first and second derivatives that we previously found, obtaining:

$$\begin{aligned} \ddot{r} \cos \nu - \dot{r} \sin \nu \dot{\nu} - \dot{r} \sin \nu \dot{\nu} - r \cos \nu \dot{\nu}^2 - r \sin \nu \ddot{\nu} + \frac{\dot{m}}{m}(\dot{r} \cos \nu - r \sin \nu \dot{\nu}) &= \\ = \frac{F_r}{m} \sin \nu - \frac{F_t}{m} \sin \nu & \\ \ddot{r} \sin \nu + \dot{r} \cos \nu \dot{\nu} + \dot{r} \cos \nu \dot{\nu} - r \sin \nu \dot{\nu}^2 + r \cos \nu \ddot{\nu} + \frac{\dot{m}}{m}(\dot{r} \sin \nu + r \cos \nu \dot{\nu}) &= \\ = \frac{F_r}{m} \sin \nu + \frac{F_t}{m} \cos \nu & \end{aligned} \quad (\text{C.0.5})$$

which, by remembering the definitions of $v_r = \dot{r}$ and $v_t = r\dot{\nu}$, reduce to:

$$\begin{aligned} \textcircled{1} \quad \dot{v}_r \cos \nu - \dot{v}_t \sin \nu - \dot{r} \sin \nu \dot{\nu} - \frac{v_t^2}{r} \cos \nu + \frac{\dot{m}}{m}(v_r \cos \nu - v_t \sin \nu) &= \\ = \frac{F_r}{m} \cos \nu - \frac{F_t}{m} \sin \nu & \\ \textcircled{2} \quad \dot{v}_r \sin \nu + \dot{v}_t \cos \nu - \frac{v_t^2}{r} \sin \nu + \dot{r} \cos \nu \dot{\nu} + \frac{\dot{m}}{m}(v_r \sin \nu + v_t \cos \nu) &= \\ = \frac{F_r}{m} \sin \nu + \frac{F_t}{m} \cos \nu & \end{aligned} \quad (\text{C.0.6})$$

At this point it's possible to rearrange these equations to explicit \dot{v}_r and \dot{v}_t :

$$\begin{aligned} \textcircled{1} \cos \nu + \textcircled{2} \sin \nu &\rightarrow \dot{v}_r = \frac{v_t^2}{r} - \frac{\dot{m}}{m} v_r + \frac{F_r}{m} \\ -\textcircled{1} \sin \nu + \textcircled{2} \cos \nu &\rightarrow \dot{v}_t = -\frac{v_r v_t}{r} - \frac{\dot{m}}{m} v_t + \frac{F_t}{m} \end{aligned} \quad (\text{C.0.7})$$

The final system of ODEs to be implemented is then:

$$\left\{ \begin{aligned} \dot{r} &= v_r \\ \dot{\nu} &= \frac{v_t}{r} \\ \dot{v}_r &= \frac{v_t^2}{r} - \frac{\dot{m}}{m} v_r + \frac{F_r}{m} \\ \dot{v}_t &= -\frac{v_r v_t}{r} - \frac{\dot{m}}{m} v_t + \frac{F_t}{m} \\ \dot{m} &= -\frac{T}{gI_{sp}} \end{aligned} \right. \quad (\text{C.0.8})$$

with $F_r = T \sin \psi - mg$ and $F_t = T \cos \psi$, considering only the thrust and gravitational forces in our model.

Appendix D

Orbit Determination and Propagation

We start with the code that allows us to determine the orbit given the position and velocity of a satellite (\mathbf{r}, \mathbf{v}). Then, by computing the rotation matrix that allows for a coordinate transformation from the perifocal reference frame to the reference $\{\hat{I}, \hat{j}, \hat{K}\}$ frame, used to calculate the orbital parameters, we can propagate the orbit. This procedure can be coupled with a B-Plane Targeting Algorithm (see Chapter 5) to visualize the final corrected trajectory, according to a restricted two-body theory.

Orbit Determination Pseudo-code

The following pseudo-code accounts for all possible singularities by introducing the *longitude of periapsis*, *argument of latitude at epoch t_0* and *true longitude at epoch t_0* , already discussed in Section 4.5, in case the orbit is equatorial, circular or both. However, they are not defined using new names, to make the code more practical to implement. For example, in case the orbit is equatorial (but not circular), then the orbital parameter ω corresponds in reality to the *longitude of periapsis* $\omega + \nu$.

- $\hat{I} = [1, 0, 0]^T$
- $\hat{K} = [0, 0, 1]^T$
- $\mathbf{h} = \mathbf{r} \times \mathbf{v} \rightarrow \hat{h} = \frac{\mathbf{h}}{h}$
- $i = \text{atan2}(|\hat{h} \times \hat{I}|, \hat{h} \cdot \hat{I})$
- $\mathbf{e} = \frac{1}{\mu} [(v^2 - \frac{\mu}{r})\mathbf{r} - (\mathbf{r} \cdot \mathbf{v})\mathbf{v}]$
 - If $e \neq 0 \rightarrow \hat{e} = \frac{\mathbf{e}}{e}$

$$- \text{ If } e = 1 \rightarrow a = \frac{h^2}{\mu} (\text{semilatus rectum}), \text{ else } a = -\frac{\mu/2}{v^2/2-\mu/r} (\text{semimajor axis})$$

If $i \neq 0$ and $i \neq \pi$:

- $\hat{n} = \hat{h} \times \hat{k}$
- $\Omega = \text{atan2}(|\hat{h} \times \hat{I}|, \hat{h} \cdot \hat{I}) \rightarrow \text{ If } n(2) < 0 : \Omega = 2\pi - \Omega$
- If $e \neq 0$:
 - $\omega = \text{atan2}(|\hat{e} \times \hat{n}|, \hat{e} \cdot \hat{n}) \rightarrow \text{ If } e(3) < 0 : \omega = 2\pi - \omega$
 - $\nu = \text{atan2}(|\mathbf{r} \times \hat{e}|, \mathbf{r} \cdot \hat{e}) \rightarrow \text{ If } \mathbf{r} \cdot \mathbf{v} : \nu = 2\pi - \nu$
- If $e = 0$:
 - $\omega = 0$
 - $\nu = \text{atan2}(|\mathbf{r} \times \hat{n}|, \mathbf{r} \cdot \hat{n}) \rightarrow \text{ If } \mathbf{r}(3) < 0 : \nu = 2\pi - \nu$

If $i = 0$ or $i = \pi$:

- $\Omega = 0$
- If $e \neq 0$
 - $\omega = \text{atan2}(|\hat{e} \times \hat{I}|, \hat{e} \cdot \hat{I}) \rightarrow \text{ If } \begin{cases} i = 0 \text{ and } e(2) < 0 \rightarrow \omega = 2\pi - \omega \\ i = \pi \text{ and } e(2) > 0 \rightarrow \omega = 2\pi - \omega \end{cases}$
 - $\nu = \text{atan2}(|\mathbf{r} \times \hat{e}|, \mathbf{r} \cdot \hat{e}) \rightarrow \text{ If } \mathbf{r} \cdot \mathbf{v} < 0 : \nu = 2\pi - \nu$
- If $e = 0$
 - $\omega = 0$
 - $\nu = \text{atan2}(|\mathbf{r} \times \hat{I}|, \mathbf{r} \cdot \hat{I}) \rightarrow \text{ If } \begin{cases} i = 0 \text{ and } r(2) < 0 \rightarrow \nu = 2\pi - \nu \\ i = \pi \text{ and } r(2) > 0 \rightarrow \nu = 2\pi - \nu \end{cases}$

Orbit Propagation

We rotate transform the cartesian coordinates in the *perifocal reference frame* into cartesian coordinates in the $(\hat{I}, \hat{J}, \hat{K})$ frame using the following matrix:

$$L_{IJK,pqw} = \begin{bmatrix} \cos \Omega & -\sin \Omega & 0 \\ \sin \Omega & \cos \Omega & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos i & -\sin i \\ 0 & \sin i & \cos i \end{bmatrix} \begin{bmatrix} \cos \omega & -\sin \omega & 0 \\ \sin \omega & \cos \omega & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (\text{D.0.1})$$

The coordinates vector in the perifocal reference frame is then simply given by:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} r \cos \nu \\ r \sin \nu \\ 0 \end{bmatrix} \quad (\text{D.0.2})$$

One can at this point choose an appropriate interval $[\nu_1, \nu_2] \in [0, 2\pi]$, calculate the corresponding cartesian coordinates and rotate them into the IJK frame, obtaining a visualization of the satellite trajectory.

Appendix E

Original Descent Equations

The original equations presented in Bett's book and in the SEEDS Master *Entry, Descent, Landing and Ascent* course are reported here for future reference, in case it's not necessary to optimize a polar landing trajectory, to be implemented in a direct collocation algorithm. [10] [4]

$$\dot{r} = v \sin \gamma$$

$$\dot{\lambda} = \frac{v \cos \gamma \sin \psi}{r \cos \phi}$$

$$\dot{\phi} = \frac{v}{r} \cos \gamma \cos \psi$$

$$\begin{aligned} \dot{v} = & \omega^2 r \cos \phi (\cos \phi \sin \gamma - \cos \gamma \cos \psi \sin \phi) + \frac{f_i}{m} \sin \gamma + \frac{f_j}{m} \cos \gamma \sin \psi + \frac{f_k}{m} \cos \gamma \cos \psi - \\ & - \frac{\dot{m}}{m} (v + \omega r \cos \gamma \cos \phi \sin \psi) \end{aligned}$$

$$\begin{aligned} \dot{\gamma} = & \frac{1}{v} \left[\frac{v^2}{r} \cos \gamma + 2\omega v \cos \phi \sin \psi + \omega^2 r \cos \phi (\cos \phi \cos \gamma + \cos \psi \sin \gamma \sin \phi) + \right. \\ & \left. + \frac{f_i}{m} \cos \gamma - \frac{f_j}{m} \sin \gamma \sin \psi - \frac{f_k}{m} \sin \gamma \cos \psi + \frac{\dot{m}}{m} \omega r \cos \phi \sin \gamma \sin \psi \right] \end{aligned}$$

$$\begin{aligned} \dot{\psi} = & \frac{1}{v \cos \gamma} \left[\frac{v^2}{r} \cos^2 \gamma \sin \psi \tan \phi + 2\omega v (\sin \phi \cos \gamma - \cos \phi \cos \psi \sin \gamma) \right. \\ & \left. + \omega^2 r (\cos \phi \sin \phi \sin \psi) + \frac{f_j}{m} \cos \psi - \frac{f_k}{m} \sin \psi - \frac{\dot{m}}{m} \omega r \cos \phi \cos \psi \right] \end{aligned}$$

$$\dot{m} = -\frac{T}{gI_{sp}}$$

with:

$$\begin{bmatrix} f_i \\ f_j \\ f_k \end{bmatrix} = \begin{bmatrix} (T \cos \epsilon - D) \sin \gamma + (T \sin \epsilon + L) \cos \sigma \cos \gamma \\ (T \cos \epsilon - D) \cos \gamma \sin \psi - (T \sin \epsilon + L)(\cos \psi \sin \sigma + \cos \sigma \sin \gamma \sin \psi) \\ (T \cos \epsilon - D) \cos \gamma \cos \psi + (T \sin \epsilon + L)(\sin \sigma \sin \psi - \cos \sigma \sin \gamma \cos \psi) \end{bmatrix}$$

Appendix F

Cosmographia Catalog Template

```
{
  "version": "1.0",
  "name": "Moon Targeting",
  "items": [
    {
      "class": "spacecraft",
      "name": "Spacecraft Name",
      "startTime": "YYYY-MM-DD HR:MN:SS UTC",
      "endTime": "YYYY-MM-DD HR:MN:SC UTC",
      "center": "Earth",
      "trajectoryFrame": "ICRF",
      "trajectory": {
        "type": "InterpolatedStates",
        "source": "trajectory.xyzv"
      },
      "bodyFrame": {
        "type": "ICRF"
      },
      "geometry": {
        "type": "Mesh",
        "meshRotation": [
          1,
          0,
          0,
          0
        ],
        "size": 0.5,
        "source": "CAD Model directory, relative to the folder
in which this json file is located. A supported CAD
format is .cmod"
      }
    }
  ]
}
```

```

    },
    "label": {
      "color": [
        1,
        0,
        0
      ]
    },
    "trajectoryPlot": {
      "color": [
        1,
        0,
        0
      ],
      "duration": "10 d",
      "lead": "0 d",
      "fade": 1,
      "sampleCount": 100
    }
  ]
}

```

Replace:

- the `startTime` argument with the desire start time, expressed as in template, for example 2031-01-25 12:00:00.00 UTC.
- the `endTime` argument with the desire stop time, expressed as stated for `startTime`.
- the `trajectoryFrame` argument with the frame in which the trajectory is expressed. A list of supported frames is available in the documentation of COSMOGRAPHIA and the ICRF frame is basically the J2000 frame.
- the `center` argument with the name of the observing body in which the `trajectoryFrame` is centered (consult the documentation for a list of supported bodies).
- the `source` argument under `trajectory` with the directory, relative to the folder where this catalogue file is stored, of the `.xyzv` file containing the trajectory data.
- the `source` argument under `geometry` with the directory, relative to the folder where this catalogue file is stored, of the CAD file of the spacecraft. A supported format is `.cmod`

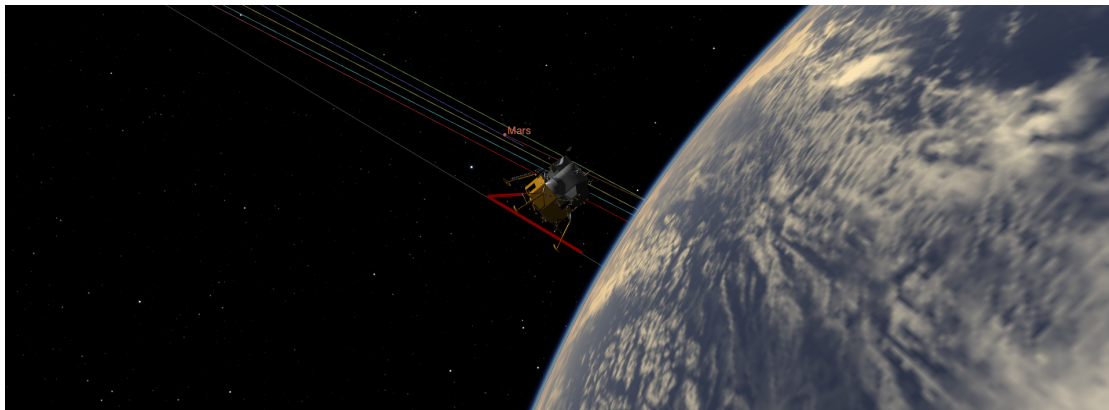


Figure F.1: Earth departure in Cosmographia.

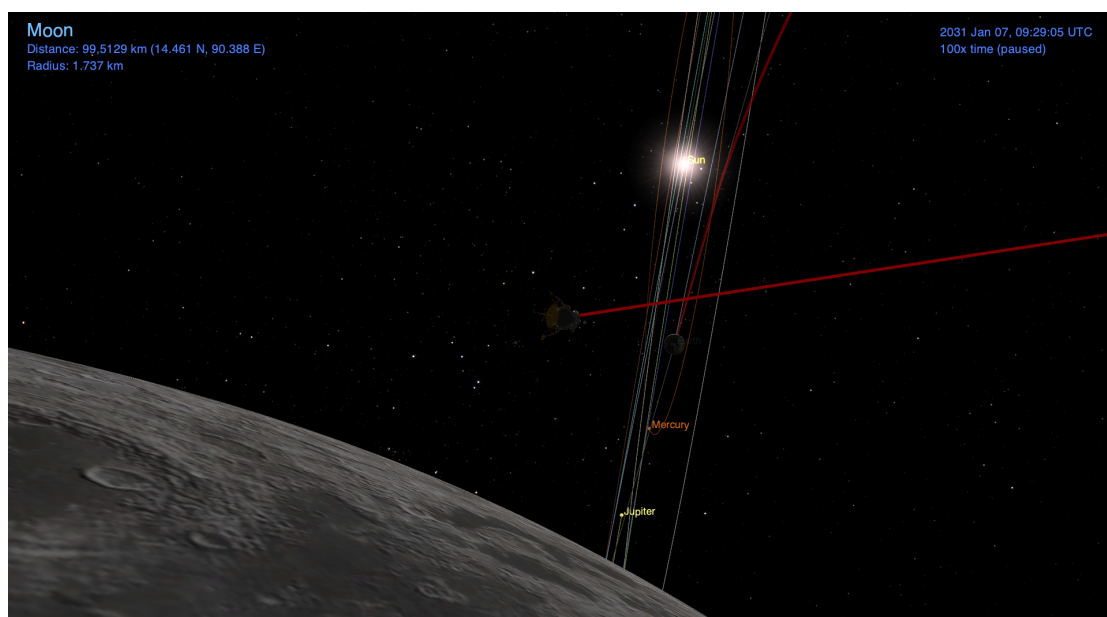


Figure F.2: Moon arrival in Cosmographia.

It's possible to further customize this file, referring to the COSMOGRAPHIA documentation. [19]

Bibliography

- [1] *A Standardized Lunar Coordinate System for the Lunar Reconnaissance Orbiter and Lunar datasets - LRO Project and LGCWG White Paper Version 5*. Tech. rep. NASA Goddard Space Flight Center, 2008.
- [2] Victor M. Becerra. “Practical Direct Collocation Methods for Computational Optimal Control”. In: *Modeling and Optimization in Space Engineering*. Ed. by Giorgio Fasano and János D. Pintér. New York, NY: Springer New York, 2013, pp. 33–60. ISBN: 978-1-4614-4469-5. DOI: 10.1007/978-1-4614-4469-5_2. URL: https://doi.org/10.1007/978-1-4614-4469-5_2.
- [3] John T. Betts. *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming, Second Edition*. Second. Society for Industrial and Applied Mathematics, 2010, pp. 1, 91–95. DOI: 10.1137/1.9780898718577. eprint: <https://epubs.siam.org/doi/pdf/10.1137/1.9780898718577>. URL: <https://epubs.siam.org/doi/abs/10.1137/1.9780898718577>.
- [4] John T. Betts. *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming, Second Edition*. Second. Society for Industrial and Applied Mathematics, 2010, pp. 247–248. DOI: 10.1137/1.9780898718577. eprint: <https://epubs.siam.org/doi/pdf/10.1137/1.9780898718577>. URL: <https://epubs.siam.org/doi/abs/10.1137/1.9780898718577>.
- [5] Collin Bezrouk. *The B-Plane Interplanetary Mission Design*. 2015. URL: <https://slideplayer.com/slide/14571565/> (visited on 09/18/2023).
- [6] Dong-Hyun Cho, Youngsuk Chung, and Hyochoong Bang. “Trajectory correction maneuver design using an improved B-plane targeting method”. In: *Acta Astronautica* 72 (2012), pp. 47–61. ISSN: 0094-5765. DOI: <https://doi.org/10.1016/j.actaastro.2011.11.009>. URL: <https://www.sciencedirect.com/science/article/pii/S0094576511003444>.
- [7] Dong-Hyun Cho, Donghoon Kim, and Henzeh Leeghim. “Optimal Lunar Landing Trajectory Design for Hybrid Engine”. In: *Mathematical Problems in Engineering* 2015 (June 2015), p. 8. DOI: 10.1155/2015/462072.
- [8] Andrey B. Sergeevsky; Gerald C. Snyder; Ross A. Cunniff. “Interplanetary Mission Design Handbook, Volume I, Part 2 - Earth to Mars Ballistic Mission Opportunities, 1990-2005”. In: (). URL: <https://ntrs.nasa.gov/api/citations/19840010158/downloads/19840010158.pdf>.

- [9] Jason Forshaw. “Spacecraft Dynamics and Control: an Introduction A. H. J. de Ruiter et al John Wiley and Sons, The Atrium, Southern Gate, Chichester, West Sussex, PO19 8SQ, UK. 2013. 569pp. Illustrated. ISBN 978-1-118-34236-7.” In: *The Aeronautical Journal* 118.1209 (2014), pp. 37–38. DOI: 10.1017/S0001924000010058.
- [10] Avanzini Giulio. “Entry, Descent, Landing and Ascent”. PoliTO SEEDS Master. Lecture Notes. 2009.
- [11] Richard H. Battin. *An Introduction to the Mathematics and Methods of Astrodynamics, Revised Edition*. AIAA, Inc., 1999, pp. 395–398.
- [12] IAU. *IAU 2006 Resolution B3*). URL: https://www.iau.org/static/resolutions/IAU2006_Resol3.pdf (visited on 09/10/2023).
- [13] IAU. *IAU(1991) RECOMMENDATION III*. URL: <https://web.archive.org/web/20070927234636/http://www.iers.org/MainDisp.csl?pid=98-133> (visited on 09/10/2023).
- [14] IAU. *Report of the IAU Working Group on Cartographic Coordinates and Rotational Elements: 2009*. URL: <https://aa.usno.navy.mil/downloads/reports/Archinaletal2011a.pdf> (visited on 09/10/2023).
- [15] The MathWorks Inc. *fmincon user manual*. URL: <https://it.mathworks.com/help/optim/ug/fmincon.html> (visited on 08/29/2023).
- [16] Moriba K. Jah. “DERIVATION OF THE B-PLANE (BODY PLANE) AND ITS ASSOCIATED PARAMETERS”. In: 2002. URL: <https://api.semanticscholar.org/CorpusID:220880539>.
- [17] Matthew Kelly. “An Introduction to Trajectory Optimization: How to Do Your Own Direct Collocation”. In: *SIAM Review* 59.4 (2017), pp. 849–904. DOI: 10.1137/16M1062569. eprint: <https://doi.org/10.1137/16M1062569>. URL: <https://doi.org/10.1137/16M1062569>.
- [18] NASA’s Navigation and Ancillary Information Facility (NAIF). *An Overview of Reference Frames and Coordinate Systems in the SPICE Context (April 2023)*. URL: https://naif.jpl.nasa.gov/pub/naif/toolkit_docs/Tutorials/pdf/individual_docs/17_frames_and_coordinate_systems.pdf (visited on 09/10/2023).
- [19] NASA’s Navigation and Ancillary Information Facility (NAIF). *Cosmographia spacecraft type catalog*. URL: <https://cosmoguide.org/catalog-file-defining-a-spacecraft/> (visited on 09/23/2023).
- [20] NASA’s Navigation and Ancillary Information Facility (NAIF). *cspice_pxform documentation*. URL: https://naif.jpl.nasa.gov/pub/naif/toolkit_docs/IDL/icy/cspice_pxform.html (visited on 09/23/2023).
- [21] NASA’s Navigation and Ancillary Information Facility (NAIF). *cspice_spekezzr documentation*. URL: https://naif.jpl.nasa.gov/pub/naif/toolkit_docs/IDL/icy/cspice_spekezzr.html (visited on 09/23/2023).

-
- [22] NASA's Navigation and Ancillary Information Facility (NAIF). *Generic Kernels*. URL: https://naif.jpl.nasa.gov/pub/naif/generic_kernels/ (visited on 09/10/2023).
- [23] NASA's Navigation and Ancillary Information Facility (NAIF). *Spice Enhanced Cosmographia catalog files*. URL: https://naif.jpl.nasa.gov/naif/cosmographia_components.html (visited on 09/23/2023).
- [24] NASA's Navigation and Ancillary Information Facility (NAIF). *Spice Enhanced Cosmographia catalog files*. URL: <https://cosmoguide.org/catalog-file-overview/> (visited on 09/23/2023).
- [25] D. Vallado. *Fundamentals of Astrodynamics and Applications*. Fourth. Microcosm Press, 2013, pp. 961–964.
- [26] D. Vallado. *Fundamentals of Astrodynamics and Applications*. Fourth. Microcosm Press, 2013, pp. 967–986.
- [27] Roger R. Bate; Donald D. Mueller; Jerry White. *Fundamentals of Astrodynamics*. First. Dover Publications, 1971, pp. 1–49.