

POLITECNICO DI TORINO

Master Degree Course in Aerospace Engineering

Master Degree Thesis

**Analysis of phasing  
maneuver performed by a  
solar electric platform**



**Advisor**  
Prof.ssa Nicole Viola

**Candidate**  
Giuseppe Narducci

April 2020

This work is subject to the Creative Commons Licence

# Summary

This thesis focuses on the problem of the phasing maneuver for a solar electric platform.

By giving as input to the program the orbital elements, in order to define the initial and final orbit, the position we want the satellite to reach, the phasing altitude, and the propulsive system, system mass and dimensions that characterize the platform, the program performs the phasing maneuver.

The phasing maneuver controls the position of the solar electric platform in terms of longitude (for geostationary orbit), true longitude (equatorial orbit) or argument of latitude (inclined orbit).

For the analysis of the maneuver to be performed, we start from chemical propulsion's maneuver logic. The satellite performs the phasing maneuver with a first EOR (electric orbit raising) raising its semi-major axis by an input values (phasing altitude). Then, the platform waits on a waiting orbit and, at the end of the maneuver, it returns to the starting orbit with a second EOR.

The main goal of this first part of the thesis is to perform a change of reference system from ECI (Earth - Centered Inertial) to ECEF (Earth - Centered, Earth - Fixed) so that the program can perform the maneuver controlling the longitude.

In the second part of the thesis, in order to optimize the maneuver, the waiting phase has been avoided. In this way, the solar electric platform is able to reach the desired position by performing two consecutive EOR maneuvers. In this case the phasing altitude is no longer an input, but an output. To determine the phasing altitude needed to reach the desired position, the program interpolates data. Furthermore, the program saves the various interpolation data (position reached and phasing altitude) in an EXCEL file, in order to be able to reduce the computational cost when the orbital parameters of a subsequent simulation are equal to a simulation already carried out.

To validate the results obtained, STK (Satellite Tool Kit) was used to perform various simulations.



*This thesis is dedicated  
to my family*

*To my father Cosmo, my mother Tiziana  
and my sisters Karine and Francesca.*

# Contents

<b>List of Tables</b>	IX
<b>List of Figures</b>	X
<b>1 MAGNETO</b>	1
1.1 Introduction . . . . .	1
1.2 On-orbit servicing mission scenarios . . . . .	2
1.2.1 Space Tug . . . . .	2
1.2.2 All electric platform . . . . .	3
1.3 MAGNETO . . . . .	4
<b>2 Fundamental of Astrodynamics</b>	7
2.1 Kepler's Laws of planetary motion . . . . .	7
2.2 Newton's Laws of motion . . . . .	7
2.2.1 Two - body problem . . . . .	8
2.2.2 Equation of motion . . . . .	9
2.2.3 Conservation of Mechanical Energy . . . . .	10
2.2.4 Conservation of angular momentum . . . . .	11
2.2.5 Trajectory Equation . . . . .	11
2.3 Geometrical properties common to all conic section . . . . .	11
2.3.1 Elliptical orbit . . . . .	12
2.3.2 Circular orbit . . . . .	12
2.4 Orbital parameters . . . . .	12
2.5 Satellite Speed . . . . .	14
2.5.1 Circular satellite speed . . . . .	14
2.5.2 Escape speed . . . . .	14
2.6 In - plane Orbit Changes . . . . .	14
2.6.1 Adjustment of Perigee and Apogee . . . . .	14
2.6.2 The Hohmann Transfer . . . . .	15
2.6.3 General coplanar Transfer between Circular Orbit . . . . .	17
2.6.4 Phasing maneuver . . . . .	18

<b>3</b>	<b>Reference Systems</b>	21
3.1	Central Body Coordinate Systems . . . . .	21
3.1.1	Earth Centered Inertial . . . . .	22
3.1.2	Earth-Centered, Earth-Fixed . . . . .	23
3.2	ECI to ECEF . . . . .	24
3.2.1	Rotation angle . . . . .	25
3.2.2	Nutation . . . . .	26
3.2.3	Precession . . . . .	26
3.3	Latitude and Longitude . . . . .	27
3.3.1	Cartesian and Polar coordinates . . . . .	28
3.3.2	Parametric Latitude . . . . .	28
3.3.3	Geodetic Latitude, Longitude and Altitude . . . . .	29
<b>4</b>	<b>Electric Orbit - Raising</b>	33
4.1	Generalities of Electric propulsion . . . . .	33
4.1.1	Rocket Equation . . . . .	35
4.1.2	Velocity Losses . . . . .	35
4.1.3	Electric propulsion maneuver . . . . .	36
4.1.4	Variation of a . . . . .	38
4.2	Electromagnetic Theory . . . . .	39
4.2.1	Electric Field . . . . .	39
4.2.2	Currents . . . . .	39
4.2.3	Magnetic Field . . . . .	40
4.2.4	Maxwell Equations . . . . .	40
4.3	Particle Motion . . . . .	41
4.3.1	$B = 0$ . . . . .	41
4.3.2	$E = 0$ . . . . .	42
4.3.3	$E, B = \text{constant}$ . . . . .	42
4.4	HT Thruster . . . . .	43
4.4.1	SITAEL HT20k Thruster . . . . .	44
<b>5</b>	<b>EOR - MAGNETO</b>	47
5.0.1	JPL Spice . . . . .	47
5.1	EOR main . . . . .	48
5.1.1	Eclipse calculator . . . . .	50
5.2	Integration . . . . .	53
5.2.1	Propulsive thrust . . . . .	54
5.2.2	Non-spherical gravity Earth . . . . .	54
5.2.3	Aerodynamic Drag . . . . .	55
5.2.4	Secondary Body perturbation . . . . .	55
5.2.5	Solar - Radiation Pressure . . . . .	56
5.3	Modified equinoctial form of the orbital equations of motion . . . . .	56

<b>6</b>	<b>EOR simulation</b>	<b>59</b>
6.1	LEO to GEO . . . . .	59
6.2	EOR - $\Delta$ altitude . . . . .	66
<b>7</b>	<b>Phasing maneuver with waiting phase</b>	<b>73</b>
7.1	Maneuver logic . . . . .	73
7.2	Simulation - Geostationary orbit . . . . .	75
7.2.1	Results - Geostationary orbit . . . . .	76
7.3	Generalization . . . . .	83
7.3.1	Controls . . . . .	83
7.4	Simulations - Generalization . . . . .	84
7.4.1	Results - Generalization . . . . .	85
<b>8</b>	<b>No waiting</b>	<b>95</b>
8.1	Simulation - Geostationary orbit . . . . .	96
8.2	Simulations . . . . .	102
<b>9</b>	<b>Interpolation</b>	<b>113</b>
9.1	Maneuver logic . . . . .	113
9.1.1	Initial case . . . . .	115
9.1.2	Results - Geostationary case . . . . .	119
9.1.3	Following case . . . . .	124
9.2	Simulation . . . . .	125
9.3	Eclipsed positions . . . . .	134
9.3.1	Eclipsed true longitude . . . . .	135
9.3.2	Circular and equatorial orbit - 13 days after . . . . .	137
<b>10</b>	<b>Validation</b>	<b>141</b>
10.0.1	STK . . . . .	141
10.1	Simulation 1 . . . . .	142
10.2	Simulation 2 . . . . .	150
<b>11</b>	<b>Conclusions</b>	<b>159</b>
	<b>Bibliography</b>	<b>161</b>

# List of Tables

4.1	HT thruster performance, [7]. . . . .	44
4.2	Working Point characteristics (credit: SITAEL, [8]). . . . .	45
6.1	Input (EOR - LEO to GEO). . . . .	60
6.2	Results (EOR - LEO to GEO). . . . .	65
6.3	Input (EOR). . . . .	66
6.4	Results (EOR). . . . .	72
7.1	Input Phasing maneuver. . . . .	75
7.2	Results. . . . .	83
7.3	Input Phasing maneuver (Generalization). . . . .	85
7.4	Results (Generalization). . . . .	93
8.1	Input Phasing maneuver (Geostationary orbit - No Waiting). . . . .	97
8.2	Results (Geostationary orbit - No Waiting) . . . . .	101
8.3	Input Phasing maneuver (No waiting). . . . .	102
8.4	Results (No waiting). . . . .	111
9.1	Input Phasing maneuver (Geostationary orbit - Interpolation). . . . .	114
9.2	Results (Geostationary orbit - Interpolation). . . . .	123
9.3	Input Phasing maneuver (Interpolation). . . . .	125
9.4	Results (Interpolation). . . . .	134
9.5	Results (Circular and equatorial orbit - 13 days after - Interpolation). . . . .	139
10.1	Input first simulation for Validation. . . . .	142
10.2	Results - simulation 1. . . . .	149
10.3	Percentage difference - Simulation 1. . . . .	150
10.4	Input simulation for Validation- 2. . . . .	151
10.5	Results - simulation 2. . . . .	158
10.6	Percentage difference - Simulation 2. . . . .	158

# List of Figures

1.1	Design Reference Mission Space Tug, [1]. . . . .	3
1.2	Design reference mission all-electric platform, [1]. . . . .	4
1.3	MAGNETO software structure, [1]. . . . .	5
2.1	Two-body problem. . . . .	8
2.2	Two-body problem in Cartesian reference frame. . . . .	9
2.3	Velocity components. . . . .	10
2.4	Elliptical orbit. . . . .	12
2.5	Orbital Parameters [Wikipedia]. . . . .	13
2.6	Adjustment of Perigee and Apogee. . . . .	15
2.7	The Hohmann Transfer. . . . .	16
2.8	General coplanar transfer between circular orbit. . . . .	17
2.9	Phasing maneuver. . . . .	18
3.1	Fixed reference system (credit: Analytical Graphics, Inc (AGI), [6]).	22
3.2	Inertial reference system (credit: AGI, [6]). . . . .	22
3.3	ECI frame. . . . .	23
3.4	ECEF frame, [13]. . . . .	24
3.5	ECI - ECEF frames, [5]. . . . .	25
3.6	Precession and Nutation . . . . .	27
3.7	Cartesian and polar coordinates, [5]. . . . .	28
3.8	Geocentric, parametric and geodetic latitudes in meridional plane, [5].	29
4.1	Internal and external forces acting on a spacecraft, [7]. . . . .	35
4.2	Thrust. . . . .	37
4.3	In-plane accelerations. . . . .	38
4.4	Electric Field, [7]. . . . .	39
4.5	Magnetic Field, [7]. . . . .	40
4.6	Particle motion - $E = 0$ , [7]. . . . .	42
4.7	Particle motion - $E, B = constant$ , [7]. . . . .	43
4.8	SITAEL HT20k Thruster (credit: SITAEL, [8]). . . . .	45
5.1	<i>EOR_main.m</i> . . . . .	48
5.2	Umbral cone geometry [12]. . . . .	51
5.3	Penumbral cone geometry [12]. . . . .	51
5.4	$\vec{r}_s$ and $\delta$ [12]. . . . .	52

5.5	Umbra [12]. . . . .	53
5.6	Penumbra [12]. . . . .	53
6.1	EOR maneuver in ECI (EOR - LEO to GEO). . . . .	60
6.2	EOR maneuver in ECEF (EOR - LEO to GEO). . . . .	61
6.3	EOR maneuver (EOR - LEO to GEO). . . . .	61
6.4	Time - semi-major axis (EOR - LEO to GEO). . . . .	62
6.5	Time - Orbital eccentricity (EOR - LEO to GEO). . . . .	62
6.6	Time - Orbital inclination (EOR - LEO to GEO). . . . .	62
6.7	Time - Delta V (EOR - LEO to GEO). . . . .	63
6.8	Time - $m_{tot}$ (EOR - LEO to GEO). . . . .	63
6.9	Time - Eclipse time (EOR - LEO to GEO). . . . .	63
6.10	Time - Thrust angles (EOR - LEO to GEO). . . . .	64
6.11	Time - Thrust weights (EOR - LEO to GEO). . . . .	65
6.12	Time - Velocity (EOR - LEO to GEO). . . . .	65
6.13	EOR maneuver . . . . .	67
6.14	Time - semi-major axis. . . . .	68
6.15	Time - Delta V. . . . .	68
6.16	Time - $m_{tot}$ . . . . .	69
6.17	Time - Eclipse time. . . . .	69
6.18	Time - Thrust angles. . . . .	70
6.19	Time - Thrust weights. . . . .	71
6.20	Time - Velocity. . . . .	71
7.1	Phasing maneuver Logic. . . . .	74
7.2	Phasing maneuver . . . . .	76
7.3	Time - semi-major axis. . . . .	77
7.4	Time - True longitude. . . . .	78
7.5	Time - Longitude. . . . .	78
7.6	Time - Delta V. . . . .	79
7.7	Time - $m_{tot}$ . . . . .	80
7.8	Time - Eclipse time. . . . .	80
7.9	Time - Thrust angles. . . . .	81
7.10	Time - Thrust weights. . . . .	82
7.11	Time - Velocity. . . . .	82
7.12	Phasing maneuver (Generalization). . . . .	86
7.13	Time - semi-major axis (Generalization). . . . .	87
7.14	Time - True Longitude (Generalization). . . . .	88
7.15	Time - Argument of Latitude (Generalization). . . . .	88
7.16	Time - Longitude (Generalization). . . . .	89
7.17	Time - Delta V (Generalization). . . . .	90
7.18	Time - $m_{tot}$ (Generalization). . . . .	90
7.19	Time - Eclipse time (Generalization). . . . .	91
7.20	Time - Thrust angles (Generalization). . . . .	92

7.21	Time - Thrust weights (Generalization).	92
7.22	Time - Velocity (Generalization).	93
8.1	No Waiting - Logic.	96
8.2	Phasing maneuver.	98
8.3	Time - semi-major axis.	98
8.4	Time -Longitude.	99
8.5	Time - $\Delta V$ .	99
8.6	Time - $m_{tot}$ .	99
8.7	Time - $t_{eclipse}$ .	100
8.8	Time - $\alpha$ and $\beta$ .	100
8.9	Time - <i>Thrust weights</i> .	101
8.10	Time - <i>Velocity vector</i> .	101
8.11	Results for the phasing maneuver at 20km, [1].	102
8.12	Phasing maneuver (No waiting).	104
8.13	Time - semi-major axis (No waiting).	105
8.14	Time - True Longitude (No waiting).	106
8.15	Time - Argument of Latitude (No waiting).	106
8.16	Time - Delta V (No waiting).	107
8.17	Time - $m_{tot}$ (No waiting).	107
8.18	Time - Eclipse time (No waiting).	108
8.19	Time - Thrust angles (No waiting).	109
8.20	Time - Thrust weights (No waiting).	110
8.21	Time - Velocity (No waiting).	111
9.1	Initial maneuver logic.	115
9.2	NO WAITING INTERPOLATION logic.	116
9.3	EXCEL file examples.	117
9.4	Interpolation $phasing_{altitude}/\Delta position$ .	118
9.5	AFTER INTERPOLATION logic.	119
9.6	Phasing maneuver.	120
9.7	Time - semi-major axis.	120
9.8	Time - Longitude.	121
9.9	Time - $\Delta V$ .	121
9.10	Time - $m_{tot}$ .	121
9.11	Time - $t_{eclipse}$ .	122
9.12	Time - $\alpha$ and $\beta$ .	122
9.13	Time - <i>Thrust weights</i> .	123
9.14	Time - <i>Velocity vector</i> .	123
9.15	Maneuver logic.	124
9.16	Interpolation $phasing_{altitude}/\Delta position$ .	126
9.17	Phasing maneuver (Interpolation).	127
9.18	Time - semi-major axis (Interpolation).	128
9.19	Time - True Longitude (Interpolation).	129

9.20	Time - Argument of Latitude (Interpolation).	129
9.21	Time - Delta V (Interpolation).	130
9.22	Time - $m_{tot}$ (Interpolation).	130
9.23	Time - Eclipse time (Interpolation).	131
9.24	Time - Thrust angles (Interpolation).	132
9.25	Time - Thrust weights (Interpolation).	133
9.26	Time - Velocity (Interpolation).	134
9.27	Phasing maneuver (Circular and equatorial orbit $\rightarrow$ <i>true longitude</i> = 180°).	135
9.28	Eclipse time - Eclipsed true longitude.	135
9.29	Eclipsed position - day.	136
9.30	Penumbra cone angle - day.	137
9.31	Interpolation + Eclipsed true longitude (Circular and equatorial orbit - 13 days after).	138
9.32	Phasing maneuver / Time - True Longitude (Circular and equatorial orbit - 13 days after).	138
10.1	Input - STK.	143
10.2	Phasing maneuver (Simulation 1).	144
10.3	Time - ECI Position (Simulation 1).	144
10.4	Time - ECEF Position (Simulation 1).	145
10.5	Time - semi-major axis (Simulation 1).	146
10.6	Time - Longitude (Simulation 1).	147
10.7	Time - True Longitude (Simulation 1).	147
10.8	Time - $\Delta V$ (Simulation 1).	148
10.9	Time - $m_{tot}$ (Simulation 1).	149
10.10	Input - STK.	152
10.11	Phasing maneuver (Simulation 2).	153
10.12	Time - ECI Position (Simulation 2).	154
10.13	Time - ECEF Position (Simulation 2).	154
10.14	Time - semi-major axis (Simulation 2).	155
10.15	Time - Longitude (Simulation 2).	155
10.16	Time - True Longitude (Simulation 2).	156
10.17	Time - $\Delta V$ (Simulation 2).	157
10.18	Time - $m_{tot}$ (Simulation 2).	157

# Chapter 1

## MAGNETO

Nowadays the trend is to develop large constellations of satellite in low orbit instead of placing large GEO satellites [1]. In fact, the telecommunication satellite market in geostationary orbit brings some uncertainties for what concern the payload size, the satellite operational lifetime and performance.

It's necessary to bring new technologies in order to make competitive again GEO satellites, and one of the solutions could be the high-power electric propulsion (HP - EP). Coupling HP-EP with reusable platforms can bring us two mission concepts:

- Reusable space tug;
- Integrated reusable platform with high-power electric propulsion.

### 1.1 Introduction

As reported in [1], during last years, the trend was to launch constellations of small satellites in low-orbit instead of launching satellites in geostationary orbit.

SpaceX's "Starlink" is one of the most recent example of small satellite constellation in LEO (Low Earth Orbit). This led the GEO market to suffer a collapse in terms of orders (about ten heavy-class satellites in 2019 and five in 2018, against the average rate of 20-25 GEO satellites per years recorded during the previous decade). But the GEO market's future is still not defined. To revitalize it, new design solutions and mission scenarios must be analyzed such the possibilities of in-orbit assembling and new technologies. The possibilities of reducing the satellites' size and so the launchers led to new on-orbit servicing (OOS). The chance to develop an innovative propulsion strategy with a brand new propellant management could satisfies easily the stakeholders requirements.

One of the factors that may bring new life to the GEO market is the development of electric in-space propulsion technologies, in particularly of high-power Hall

thrusters.

## 1.2 On-orbit servicing mission scenarios

The OOS (On-Orbit Servicing) functionalities [1] are:

- Relocate: to place a satellite in its target position;
- Restore: the operations that bring the satellite to its initial condition;
- Upgrade: the operations that allow the satellite to increase its capabilities;
- Inspect: all the activities relevant for a long permanence satellite;
- Assemble: all the activities that regards two large structure that cannot be assembled before the launch, but only in orbit.

All these functionalities must be implemented, but, unfortunately, that's not so easy. The introduction of new technologies is not so immediate and the benefits are not easy to evaluate in short time. In fact, on the one hand, there are features that can be implemented directly in an old mission scenario, but on the other hand, there may be changes that cause serious problems for what concern the design.

### 1.2.1 Space Tug

The Space Tug is an electric platform that has been created in order to rendezvous with satellites and perform a manuever in a way to bring it to the desired orbit.

The Space Tug's On-orbit scenario are:

- Transportation and deployment of satellite;
- Relocation services;
- Disposal servicing;
- In orbit assembly.

The Space Tug must be able to perform guidance, navigation and control during the transfer in order to make simpler the AOCS (Attitude and Orbit Control System) on board the payload and so reduce its complexity and size.

It must be flexible, reusable and be able to perform many operations and transfer during all their operational lifetime thanks to a refueling system.

The Design Reference Mission (Fig. 1.1) may be divided in:

- A: The Space Tug is launched in the parking orbit;
- B: It performs the rendezvous with the payload;

- C: It performs the transfer to the desired orbit;
- D: It releases the payload;
- E: It performs a phasing maneuver in order to outdistance the payload;
- F: It moves to a parking orbit and waits for the next operation.

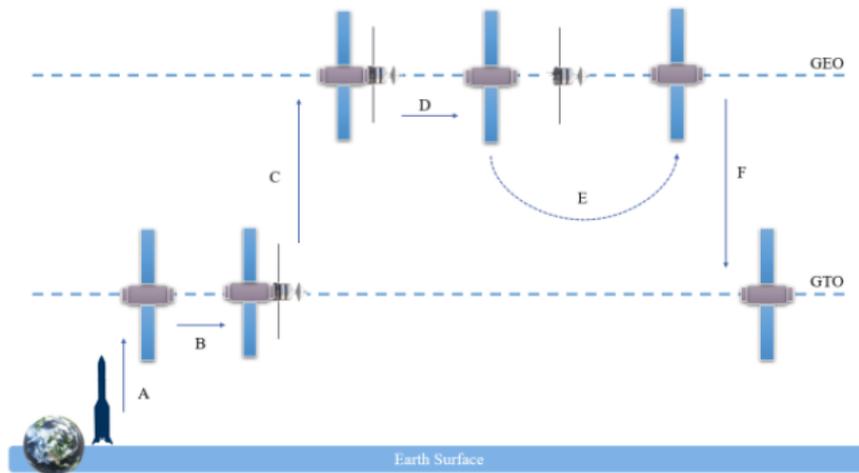


Figure 1.1: Design Reference Mission Space Tug, [1].

### 1.2.2 All electric platform

These platforms use an electric propulsion subsystem in order to make transfer to GEO orbit and station-keeping maneuver during all their operational life. By eliminating the chemical propulsion subsystem, we have a great mass saving which allow to augment the payload mass and so its capabilities and reduce the launch vehicle cost.

The Design Reference Mission of the all-electric platform (Fig. 1.2) is much simpler than the Space Tug one.

The mission phases the all-electric platform is going through to reach its desired orbit are:

- A: the launcher delivers the platform;
- B: the electric platform performs an EOR (electric orbit raising) from its initial orbit to the desired orbit;
- C: It performs a phasing maneuver in order to reach its desired position.

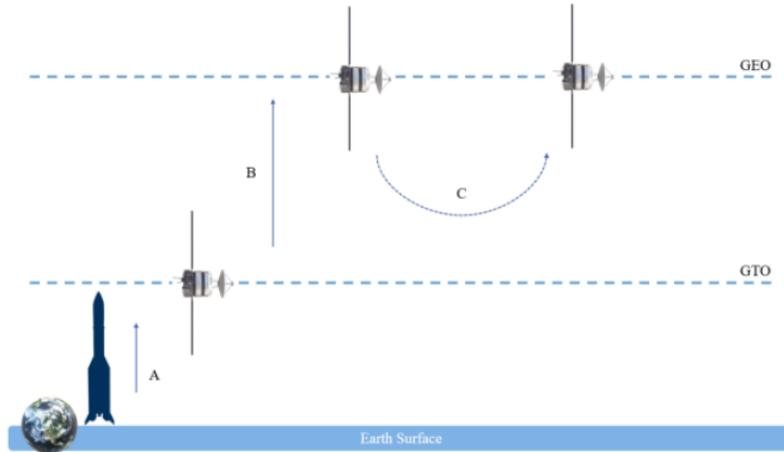


Figure 1.2: Design reference mission all-electric platform, [1].

## 1.3 MAGNETO

"The MultidisciplinAry desiGN Electric Tug tOol (MAGNETO) is a multi-input and multi-output tool designed to enable mission analysis and design of electrically-propelled space platform" [1].

This tool has been developed by Politecnico di Torino, in collaboration with ESTEC, and it's able to perform a preliminary design of a Space Tug [2].

MAGNETO represents an improvement of MISS (Mission and Space System, [3]), a design tool developed in order to perform a preliminary size of a spacecraft with high-power electric propulsion.

The tool is composed of three macro-modules (Fig. 1.3):

- Scenario definition: the scenario and architecture must be selected and so the thruster data;
- Scenario analysis: in this module there is a preliminary trajectory analysis, a preliminary subsystem sizing and a system budget propellant mass estimation;
- Scenario optimization: where there is a propagation of the trajectory, a new subsystem sizing, new spacecraft budget and in the end the result post-processing.

In the first module the mission is analyzed from the user which define his needs and some mission features. The second module define the preliminary sizing of the electrical platform and also analyze the trajectory. In the third macro-module the scenario is optimized, and new values are provided for what concern the mass and trajectory.

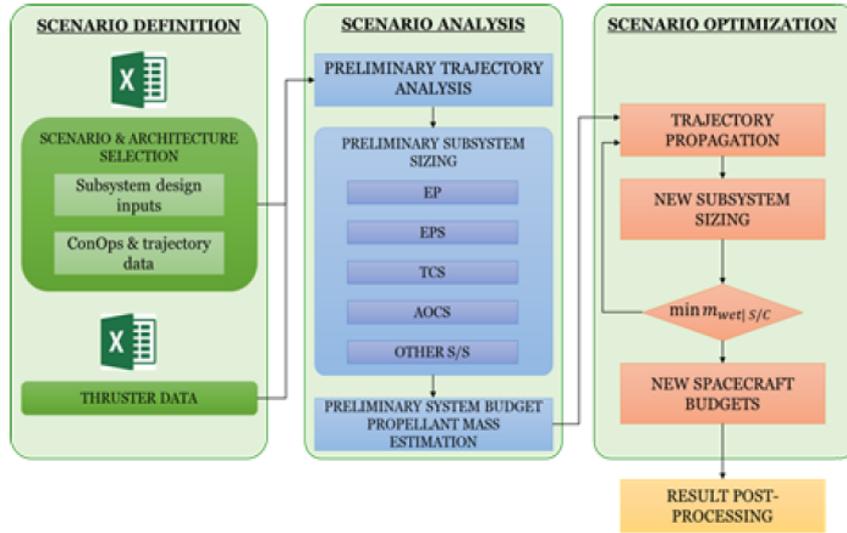


Figure 1.3: MAGNETO software structure, [1].

We are going to work on the trajectory generation module which is a low thrust propagator able to search for the optimal maneuver in order to reach the desired orbit in terms of orbital parameters such as semi-major axis, eccentricity and inclination.

In particular, the subject of this thesis is the phasing maneuver. While the orbital parameters to be checked in EOR are semi-major axis, eccentricity and inclination, for what concern the phasing maneuver, the parameter to be checked is the position of the satellite in terms of longitude, true longitude or argument of periapsis, depending on the orbit we are dealing with.



## Chapter 2

# Fundamental of Astrodynamics

In this chapter will be analyzed some generalities of Astrodynamics [4].

The equations of motion that are obtained starting from the Newton's laws and so the orbit features from the Kepler's laws will be analyzed in order to define the classic orbital elements and the characteristic velocities.

### 2.1 Kepler's Laws of planetary motion

Johannes Kepler (27 December 1571, Weil der Stadt, Württemberg - 15 November 1630, Regensburg) published between 1609 and 1619 the Kepler's laws of planetary motion [18]. These laws, as the name implies, describe the motion of planets around the Sun.

- **FIRST LAW:** "The orbit of each planet is an ellipse with the Sun at one focus" [17].
- **SECOND LAW:** "The line joining the planet to the Sun sweeps out equal areas in equal times" [17].
- **THIRD LAW:** "The square of the period of a planet is proportional to the cube of its mean distance to the Sun" [17].

### 2.2 Newton's Laws of motion

Sir Isaac Newton (25 December 1642, Woolsthorpe, Lincolnshire, England - 20 March 1726/27, London) is recognized as one of the most important all-time scientists [19]. In 1687 he published *Philosophiæ Naturalis Principia Mathematica*, where he introduced the Laws of motion:

- FIRST LAW: "Every body continues in its state of rest, or of uniform motion in a right [straight] line, unless it is compelled to change that state by forces impressed upon it" [17];
- SECOND LAW: "The change of motion is proportional to the motive force impressed and is made in the direction of the right line in which that force is impressed" [17];
- THIRD LAW: "To every action there is always opposed an equal reaction: or, the mutual action of two bodies upon each other are always equal and directed to contrary parts" [17].

### 2.2.1 Two - body problem

The second law leads to the Newton's Law of Universal Gravitation (Fig. 2.1):

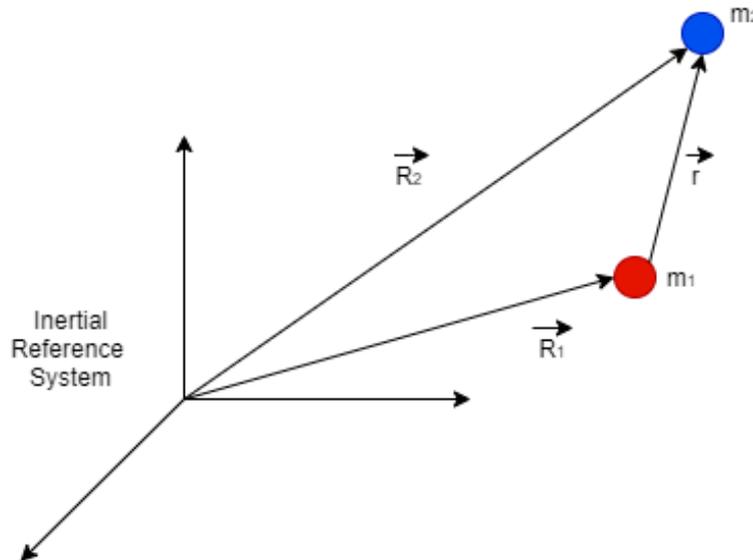


Figure 2.1: Two-body problem.

$$F_g = -\frac{Gm_1m_2}{r^2} \frac{\vec{r}}{r}$$

Where:

- $F_g$  is the force on mass  $m_2$  due to  $m_1$ ;
- $\vec{r}$  is the vector from  $m_1$  to  $m_2$ ;
- $G$  is the universal gravitational constant.

### 2.2.2 Equation of motion

Considering the two-body problem, the equation of motion results:

$$\ddot{\vec{r}} = -G \frac{M}{r^2} \frac{\vec{r}}{r} = \frac{\mu}{r^2} \frac{\vec{r}}{r}$$

Where:

- $M$  = principal body mass;
- $\mu$  = gravitational parameter of the principal body.

Let's consider a Cartesian reference system centered on the principal body (see Fig. 2.2) where  $\nu$  (nu) is the true anomaly.

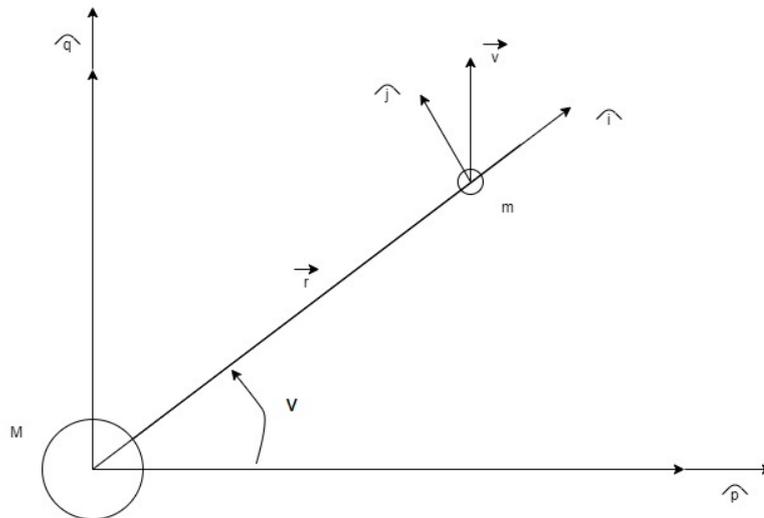


Figure 2.2: Two-body problem in Cartesian reference frame.

In Fig. 2.3 there are the components of the velocity:

- $v_r$  = radial component of velocity;
- $v_t$  = tangent component of velocity.

And  $\phi$  (phi), known as flight-path angle.

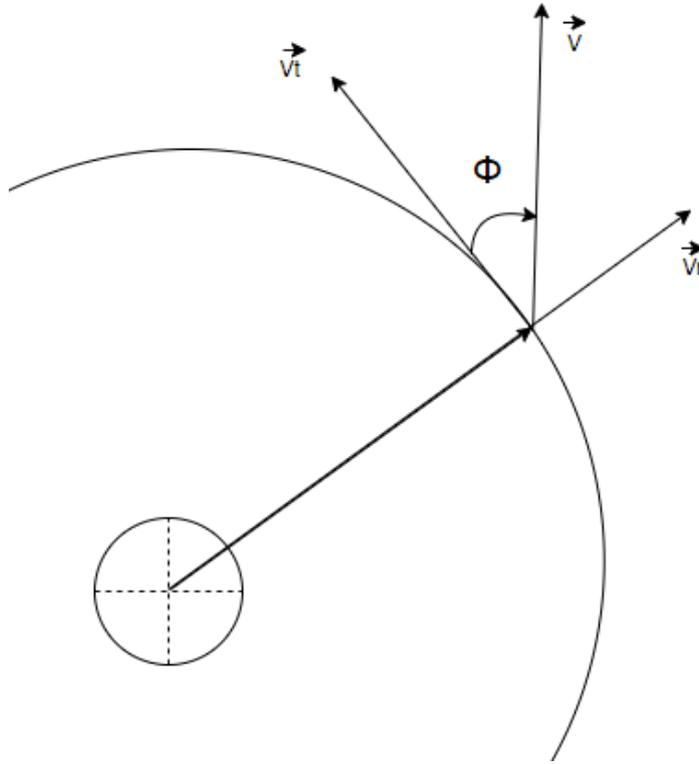


Figure 2.3: Velocity components.

The velocity can be divided in:

$$\begin{cases} v_r = \dot{r} \\ v_t = r\dot{\nu} \end{cases} \quad (2.1)$$

In the same way the acceleration is:

$$\begin{cases} a_r = \ddot{r} - r\dot{\nu}^2 \\ a_t = 2\dot{r}\dot{\nu} + r\ddot{\nu} \end{cases} \quad (2.2)$$

### 2.2.3 Conservation of Mechanical Energy

The mechanical energy of a satellite is defined as:

$$E = \frac{v^2}{2} - \frac{\mu}{r}$$

### 2.2.4 Conservation of angular momentum

A satellite must keep moving on its orbital plane as described by the specific angular momentum.

$$h = r \times v$$

It can also be expressed in terms of flight-path angle:

$$h = rv \cos \phi$$

### 2.2.5 Trajectory Equation

The trajectory equation is obtained by the integration of the equation of motion:

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

Where:

- $B$  is the vector constant of integration;
- $\nu$  is the angle between the constant vector  $B$  and the vector  $r$ .

In polar coordinates:

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

## 2.3 Geometrical properties common to all conic section

Here some geometrical properties in common to all the kind of orbit:

- $e = c/a =$  eccentricity;
- $p = a(1 - e^2) =$  periapsis;
- $r_{min} = r_{periapsis} = p/(1 + e) = a(1 - e) =$  radius of periapsis;
- $r_{max} = r_{apoapsis} = p/(1 - e) = a(1 + e) =$  radius of apoapsis;
- $E = -\frac{\mu}{2a} =$  specific mechanical energy;

### 2.3.1 Elliptical orbit

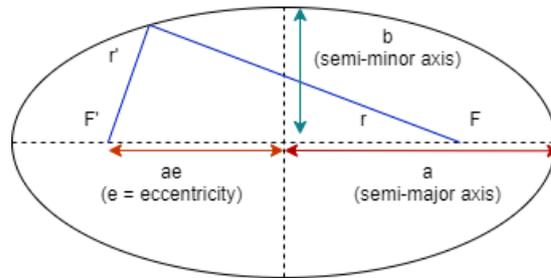


Figure 2.4: Elliptical orbit.

In Fig. 2.4 we can see an elliptical orbit, which is characterized by:

- $r + r' = 2a$ ;
- $r_p + r_a = 2a$ ;
- $e = \frac{r_a - r_p}{r_a + r_p}$  ( $0 < e < 1$ );
- Period of an elliptical orbit =  $T_P = \frac{2\pi}{\sqrt{\mu}} a^{3/2}$

### 2.3.2 Circular orbit

The circular orbit is just a particular case of the elliptical orbit, where:

- $e = 0$ ;
- $r_p = r_a$ .

## 2.4 Orbital parameters

The orbital elements are those parameters necessary to identify the typology of an orbit.

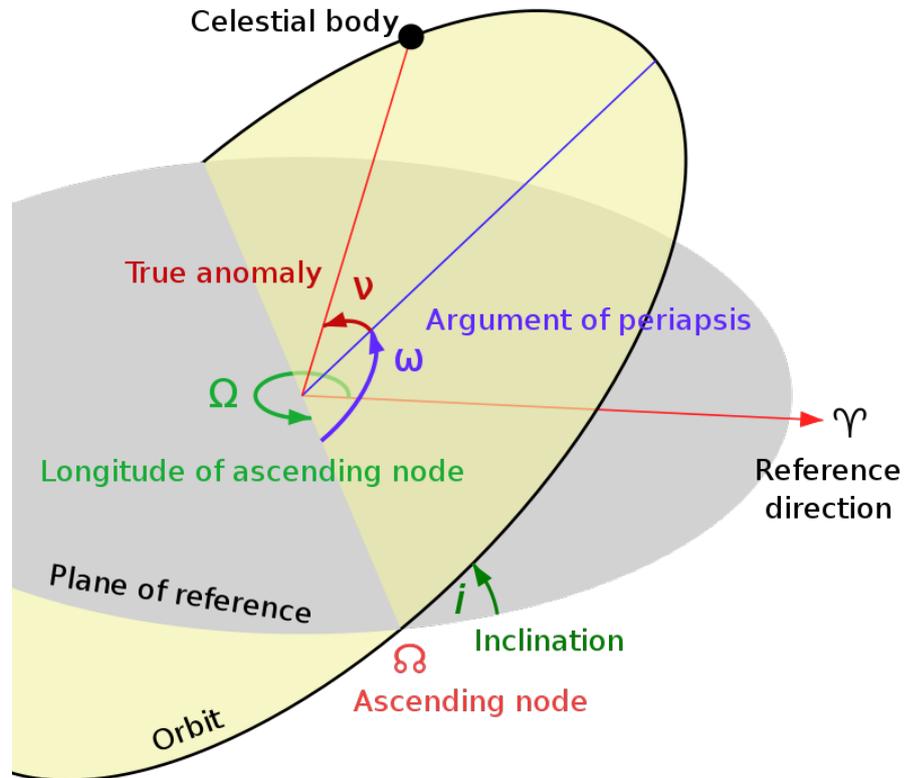


Figure 2.5: Orbital Parameters [Wikipedia].

As we can see from 2.5, there are several orbital parameters:

- Eccentricity ( $e$ );
- Semi-major axis ( $a$ );
- Inclination ( $i$ );
- Longitude of the ascending node ( $\Omega$  or *RAAN*);
- Argument of periapsis ( $\omega$ );
- True anomaly ( $\nu$ ) at epoch ( $t_0$ ).

The first two orbital elements are useful to describe the shape of the ellipse,  $i$  and  $\Omega$  define the orbital plane,  $\omega$  defines the orientation of the orbit on its plane and  $\nu$  defines the position of the orbiting object we are considering. We may also consider as orbital parameter the mean anomaly ( $M$ ) which is not a real geometric angle.

## 2.5 Satellite Speed

### 2.5.1 Circular satellite speed

In order to reach a circular orbit, we must have  $v_r = 0$  and  $v_t = r\dot{\nu} = \text{constant}$ :

$$v_c = \sqrt{\frac{\mu}{r}}$$

### 2.5.2 Escape speed

The gravitational effect of a celestial body decreases rapidly with the distance, even though it still affects the satellite with a small amount of kinetic energy. The escape speed allows the satellite to escape the effects of gravity:

$$E = \frac{v_{esc}^2}{2} - \frac{\mu}{r} = 0$$

$$v_{esc} = \sqrt{\frac{2\mu}{r}} = 2v_c$$

## 2.6 In - plane Orbit Changes

The orbit we want the satellite to reach, may not be reached because of small error that may occur in burnout altitude, speed or flight-path angle [4]. In order to reach a precise orbit, we have to make small correction in the orbit. This can be done by making some small speed changes ( $\Delta V$ ).

The In-Plane orbit maneuver is a coplanar maneuver. As the name implies, the initial and the final orbit plane lies in the same plane. Using these maneuvers, it's possible to change the semi-major axis, eccentricity (orbit's size and shape) and the argument of perigee (line of apsides). For what concerns the chemical propulsion, this maneuver can be tangential or non-tangential.

### 2.6.1 Adjustment of Perigee and Apogee

As shown in Fig. 2.6, this is an uno-tangent burn maneuver: it consists of one tangential burn in order to change semi-major axis or the location of the line of apsides [4]. The requirement for tangential burns is that the flight-path angle must be  $\phi = 0^\circ$ .

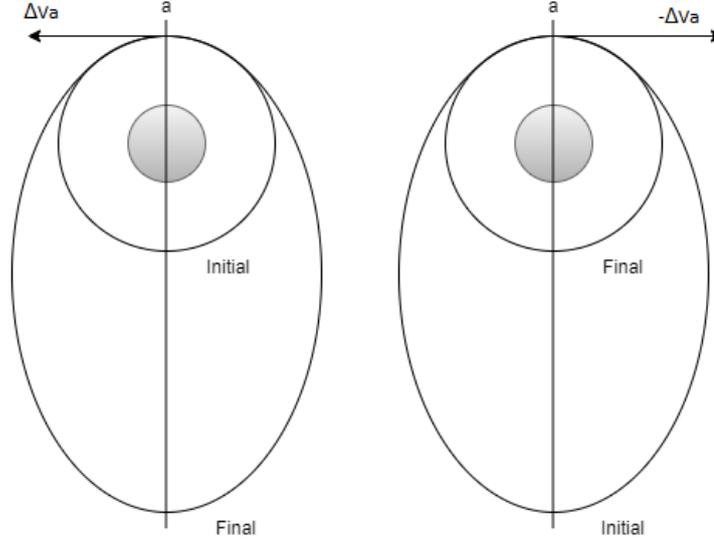


Figure 2.6: Adjustment of Perigee and Apogee.

Considering:

$$E = \frac{v^2}{2} - \frac{\mu}{r} = -\frac{\mu}{2a}$$

Solving for  $v^2$ , we obtain:

$$v^2 = \mu \left( \frac{2}{r} - \frac{1}{a} \right)$$

If we change  $v$ , leaving  $r$  unchanged:

$$2v dv = \frac{\mu}{a^2} da \text{ or } da = \frac{2a^2}{\mu} v dv$$

If we change infinitesimally the speed ( $dv$ ), we obtain a change in semi-major axis ( $da$ ).

If we apply  $\Delta V$  at the apogee we get a change in perigee height, while if the  $\Delta V$  is applied at the perigee, the change will be in apogee height.

$$\begin{cases} \Delta h_p \approx \frac{4a^2}{\mu} v_a \Delta v_a \\ \Delta h_a \approx \frac{4a^2}{\mu} v_p \Delta v_p \end{cases} \quad (2.3)$$

It is convenient to maneuver at the perigee where the speed is higher, although in terms of energy it is the same thing.

## 2.6.2 The Hohmann Transfer

The Hohmann transfer (Fig. 2.7) is a maneuver between two circular and coplanar orbit [4]. It is a two tangential burns maneuver and it's achieved by using a doubly-tangent transfer ellipse.

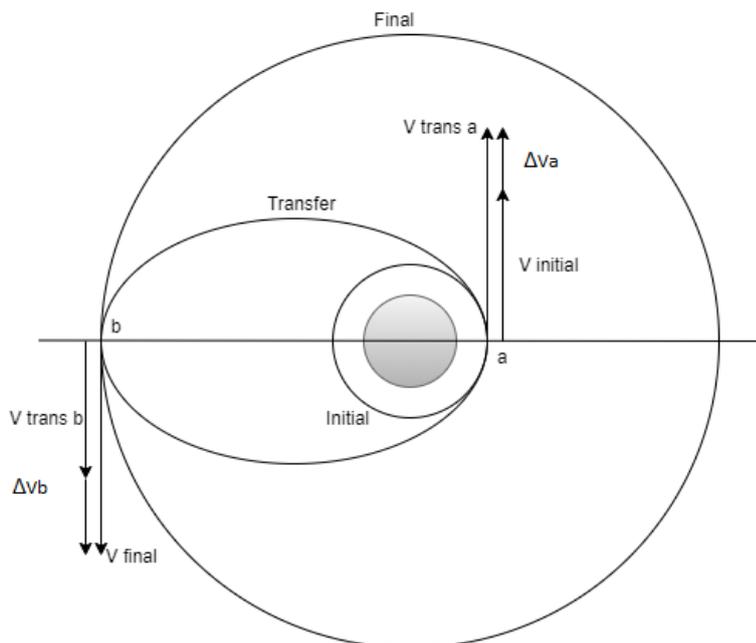


Figure 2.7: The Hohmann Transfer.

If we want to travel from the initial circular orbit ( $r_1$ ) to the final circular orbit ( $r_2$ ), we must consider:

$$2a_{transfer} = r_1 + r_2$$

$$E_{transfer} = -\frac{\mu}{2a_{transfer}}$$

Solving the energy equation at point  $a$  (Fig. 2.7), we get:

$$v_1 = \sqrt{2\left[\frac{\mu}{r_1} + E_{transfer}\right]}$$

Since the satellite was in a circular orbit, its speed was  $v_{c1} = \sqrt{\frac{\mu}{r_1}}$ . So, in order to increase the satellite speed, we must apply:

$$\Delta v_1 = v_1 - v_{c1}$$

In the same way, we may calculate the  $\Delta v_2$  necessary to circularize the orbit once the satellite reach the point  $b$ . This maneuver can be also done in the opposite direction.

The time-of-flight is half the period of the transfer ellipse orbit:

$$T = \pi \sqrt{\frac{a_{transfer}^3}{\mu}}$$

This is the most economical maneuver, but also it is the slower.

### 2.6.3 General coplanar Transfer between Circular Orbit

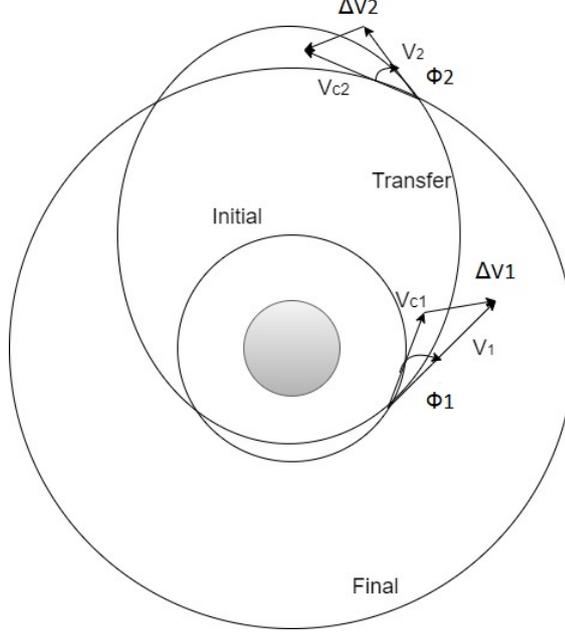


Figure 2.8: General coplanar transfer between circular orbit.

The condition to make a General Coplanar transfer between circular orbit (Fig. 2.8) are, according to [4]:

$$\begin{cases} r_p = \frac{p}{1+e} \leq r_1 \\ r_a = \frac{p}{1-e} \geq r_2 \end{cases} \quad (2.4)$$

In this way the initial and final orbits will intersect or at least be tangent to both circular orbits.

For what concern the transfer:

$$\begin{cases} E_{transfer} = \mu(1 - e^2)/2p \\ h_{transfer} = \sqrt{\mu p} \end{cases} \quad (2.5)$$

Solving the energy equation as the same way as the Hohmann transfer, we obtain:

$$v_1 = \sqrt{2\left(\frac{\mu}{r_1} + E_{transfer}\right)}$$

The angle between  $v_1$  and  $v_{c1}$  is the flight-path angle ( $\phi_1$ ). Since  $h = rv \cos \phi$ :

$$\cos \phi_1 = \frac{h_{transfer}}{r_1 v_1}$$

In this way we get the first  $\Delta V$ :

$$\Delta v_1^2 = v_1^2 + v_{c1}^2 - 2v_1v_{c1} \cos \phi_1$$

The  $\Delta v_2$  may be computed in a similar way.

The Hohmann transfer is a special case of a General coplanar transfer between circular orbit, where  $\phi_1 = 0^\circ$ ,  $r_p = r_1$  and  $r_a = r_2$ .

### 2.6.4 Phasing maneuver

The phasing maneuver is a two-impulse maneuver (Fig. 2.9). It's an important maneuver for what concern the station-keeping.

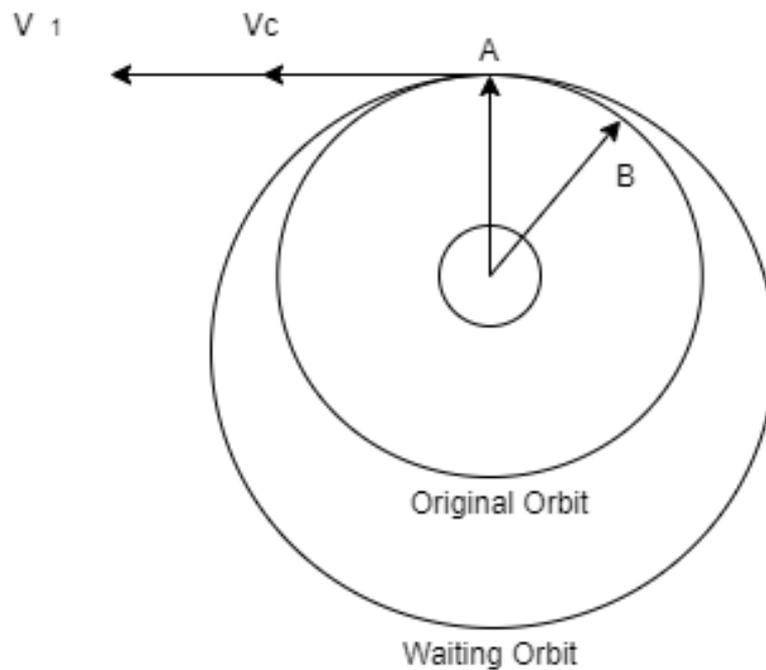


Figure 2.9: Phasing maneuver.

If the satellite is in "A", but we wanted it to be in "B", we must change its longitude:

$$|\Delta Longitude| = |Longitude_B - Longitude_A|$$

In order to make this change, the satellite should stop and wait for the Earth rotation for about:

$$\Delta T = \frac{\Delta Longitude}{\omega_{Earth}}$$

That's not possible, so we have to bring the satellite in a waiting orbit with:

$$T_{WaitingOrbit} = T_{OriginalOrbit} + \Delta T$$

Where:

- $\Delta T = n(T_{WaitingOrbit} - T_{OriginalOrbit});$
- $T_{WaitingOrbit} = 2\pi\sqrt{\frac{a_{WaitingOrbit}^3}{\mu}};$

Where  $n$  is the number of complete orbit that the satellite performs in waiting orbit.

We get:  $E_{g_{WaitingOrbit}} = \frac{\mu}{2a_{WaitingOrbit}}$  and considering the initial orbit as a circular one, we obtain:

$$\Delta V_1 = v_1 - v_c = v_1 - \sqrt{\frac{\mu}{r}}$$

The more  $n$  is high, the less is  $\Delta V$  value.

This is because if we go to a nearby orbit, we spend less than a more distant orbit, but it will be necessary to wait more.

To return to the original orbit, the satellite will have to make an equal impulse in module but opposite to the first:

$$\Delta v = 2\Delta v_1$$



## Chapter 3

# Reference Systems

One of the goals of this thesis is to carry out a change of reference system passing from the ECI reference system to ECEF. This change of reference system is necessary to perform the phasing maneuver. In fact, if the satellite is in a geostationary orbit, to bring the satellite to the desired position, we must know ECEF satellite coordinates in order to obtain the longitude.

To define a right-handed orthogonal reference system (see [6]), we need 3 elements:

- Origin of the reference system (center of gravity of the main body);
- Fundamental plane (where the x and y axes lie);
- Positive direction of the z axis.

### 3.1 Central Body Coordinate Systems

As we already saw, the origin of a reference system is in the center of gravity of the main body.

The differences between a reference system and another one are the axes we are using [6].

We can divide the typology of reference system in:

- Fixed coordinate system: the axes rotate with the central body, Fig. 3.1;
- Inertial coordinate system: the axes do not rotate with the central body, Fig. 3.2.

Both these reference frames are supported by all central bodies.

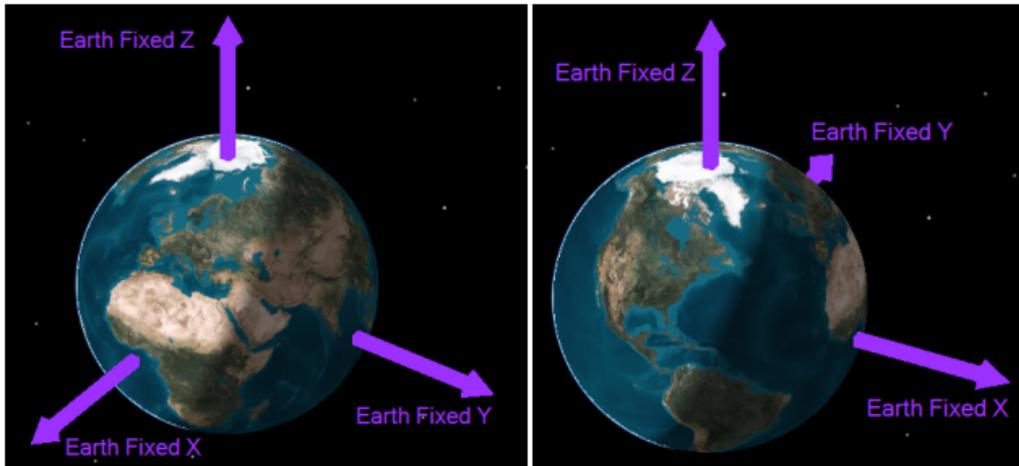


Figure 3.1: Fixed reference system (credit: Analytical Graphics, Inc (AGI), [6]).

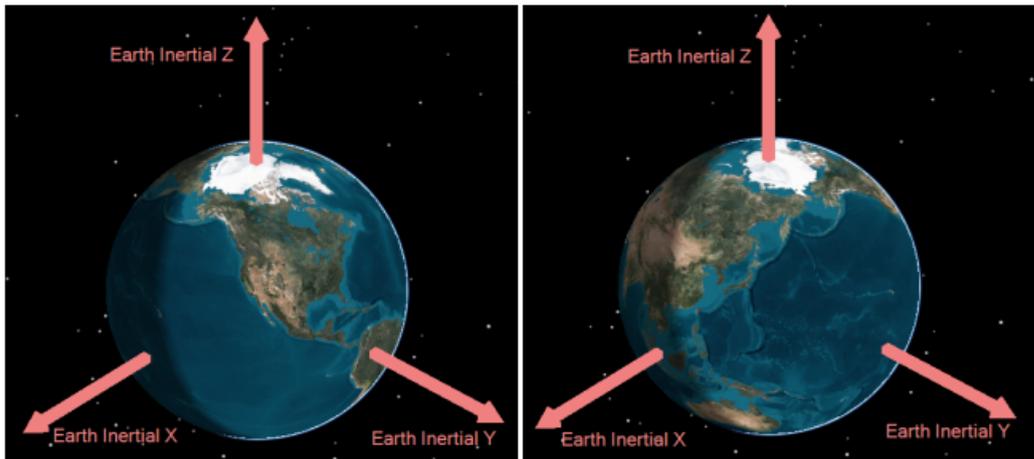


Figure 3.2: Inertial reference system (credit: AGI, [6]).

### 3.1.1 Earth Centered Inertial

The Earth-centered inertial (ECI) coordinate frames have its origin at the center of mass of the central body (Earth) and, as said before, it doesn't rotate with respect to the stars (Fig. 3.3).

This reference frame is used because the equations of motion are simpler in a non-rotating frame. In fact, this frame is used to specify the direction toward celestial objects.

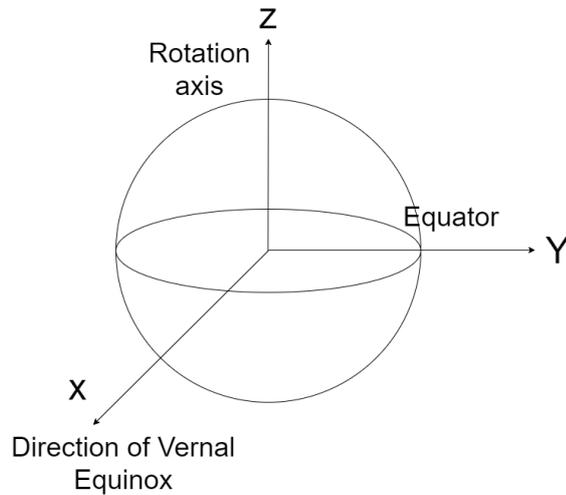


Figure 3.3: ECI frame.

The Earth's orbit plane, also known as ecliptic, doesn't coincide with the Earth's equatorial plane. In fact, the angle between these two planes,  $\epsilon$  (or obliquity of the ecliptic) measures  $\approx 23.4^\circ$ .

The x - axis is defined as the intersection of the ecliptic plane and the equator. This axis is also directed to Aries constellation, which is the direction where the Sun lies during Spring (or Vernal) Equinox. The z - axis coincides with the rotational axis of the Earth, as the same way as ECEF frame.

### 3.1.2 Earth-Centered, Earth-Fixed

ECEF (Earth-Centered, Earth-Fixed) is a Cartesian coordinate system (Fig. 3.4).

The origin of this reference system is the Earth's center of mass.

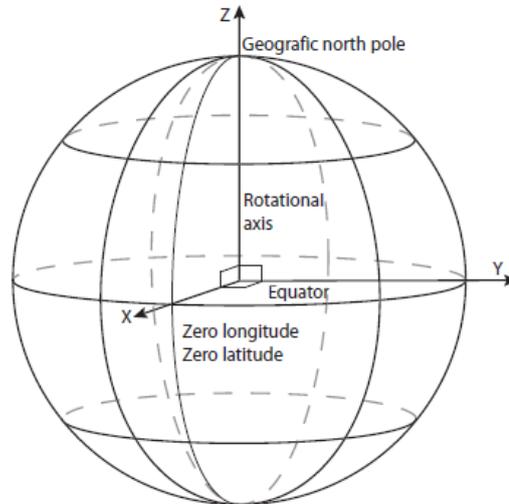


Figure 3.4: ECEF frame, [13].

In order to define the reference axes, we must define:

- International reference Prime Meridian (or prime meridian in Greenwich): where longitude =  $0^\circ$  and it identifies the x - axis;
- Equator : latitude =  $0^\circ$ ;

The ECEF reference frame rotates with the Earth and so the coordinates of a fixed point on the surface doesn't change in function of time. That's the reason why it's easier to represent positions and velocities in ECEF frame in order to obtain latitude, longitude and altitude in respect of the ECI frame.

Longitude in ECEF coordinates is considered positive in the EAST direction in respect of the prime meridian (Greenwich, UK) and vice versa ( $longitude < 0 \rightarrow WEST$ ), while latitude is considered positive in the Boreal hemisphere and negative in the Austral hemisphere.

## 3.2 ECI to ECEF

As seen previously, both ECI and ECEF frames have same origin (the center of mass of the Earth) and they share the same z axis, but the ECI frame doesn't rotate around the z axis differently from ECEF frame. So, the two frames differ in a linear function of time [5].

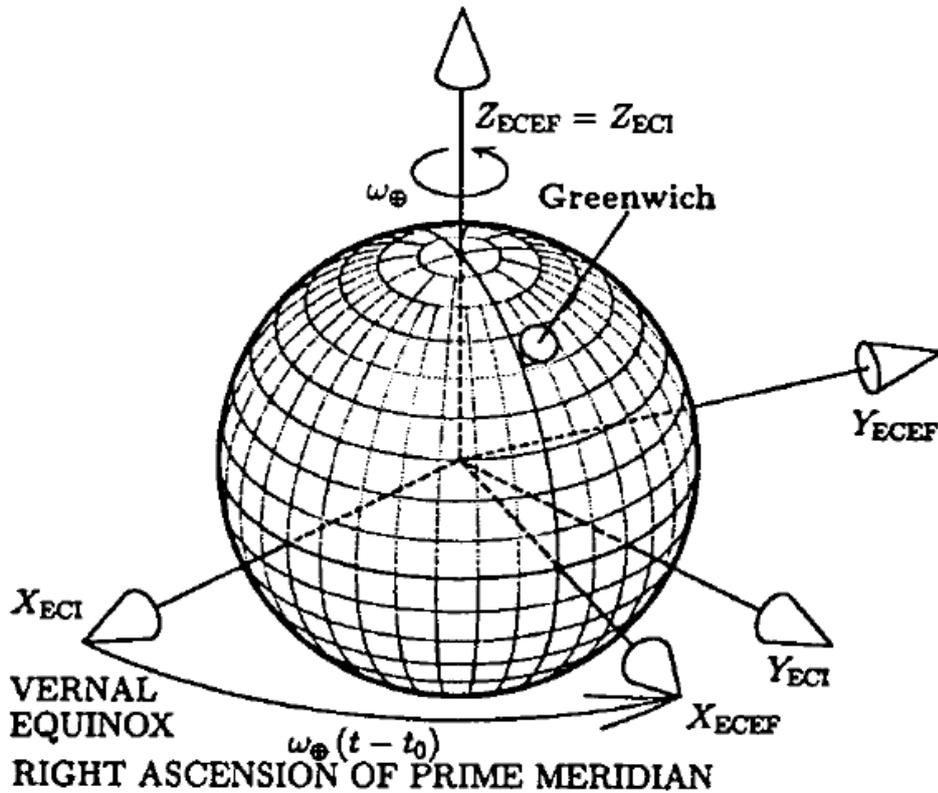


Figure 3.5: ECI - ECEF frames, [5].

### 3.2.1 Rotation angle

In order to transform the ECI frame in ECEF reference system (Fig. 3.5), we must know the Earth's current rotation ( $\theta$ ) [21]. This rotation is determined by:

$$\theta = \theta_0 + \omega_{Earth}(t - t_0)$$

Where:

- $\theta_0$  is rotation of the Earth at the reference time  $t_0$ ;
- $t$  is the current time;
- $\omega_{Earth} = 7,2921151467 \cdot 10^{-5} rad/s$  is the rotation speed of the Earth.

The rotation matrix [22] necessary for this transformation is the following:

$$\begin{bmatrix} x_{ECI} \\ y_{ECI} \\ z_{ECI} \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{ECEF} \\ y_{ECEF} \\ z_{ECEF} \end{bmatrix} \Leftrightarrow \begin{bmatrix} x_{ECEF} \\ y_{ECEF} \\ z_{ECEF} \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{ECI} \\ y_{ECI} \\ z_{ECI} \end{bmatrix}$$

The knowledge of the angle of rotation is not enough to change the reference system.

Two effects need to be considered: precession and nutation [23](Fig. 3.6).

### 3.2.2 Nutation

Discovered in 1728, by James Bradley, an English astronomer, nutation is a phenomenon due to the gravitational effects of secondary bodies that causes a spinning motion of the rotation axis of main body.

It's called as free nutation if it's not caused by external forces or Euler nutation (it acts on the second Euler angle).

For what concern the Earth, the main bodies which cause Nutation are the Sun and the Moon. In fact, the regression of the Moon's nodal line cause about  $\pm 17''$  in longitude and  $\pm 9.2$  in obliquity every 18.61 years.

### 3.2.3 Precession

Precession, instead, is a change for what concern the orientation of the rotation axis of the celestial body we are considering. Differently from the Nutation, the Precession regards the first Euler angle.

This phenomenon refers to a slow change in astronomical body's rotational axis which slowly traces out a cone.

For what concern the Earth, the Precession is also known as Precession of the equinoxes, Lunisolar precession or precession of the equator. The Earth complete a precessional cycle in approximately 25786 years and it completes about  $1^\circ$  every 72 years.

In fact:

- Lunisolar precession: cause  $0^\circ 0' 50.37''$  per year;
- Planetary precession: cause  $-0^\circ 0' 0.11''$  per year;
- TOTAL:  $0^\circ 0' 50.37''$  per year.

And so, after about 13000 years, the z axis won't be directed to Polaris, but it will be directed to Vega.

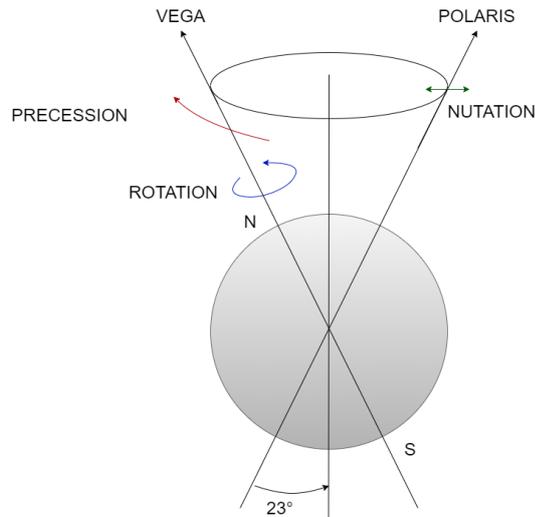


Figure 3.6: Precession and Nutation

In order to obtain the "*ECItoECEF.m*", the function we use in order to convert the ECI coordinates in ECEF coordinates, we use the "*dcmeci2ecef*" function implemented in MATLAB [14]. This function calculates the rotation matrix between the ECI and the ECEF frame using as input the date (UTC) and the ECI coordinates. In order to calculate the precession and nutation's effect, this function follows the procedure analyzed in [20], which is not presented in this thesis.

### 3.3 Latitude and Longitude

Latitude is an angle which define a range from  $0^\circ$  (Equator) to  $\pm 90^\circ$  at the poles. It's possible to define more than one typology of latitude [5]:

- Geocentric Latitude (the angle between the equatorial plane and the line which unites the center of the Earth with the orbiting object);
- Geodetic Latitude (the angle above between the equatorial plane and the normal component of the velocity of the orbiting object);
- Parametric Latitude (has no physical meanings, but it's necessary to calculate geodetic latitude).

In order to define the longitude, we must consider that the meridians (the lines that round the globe passing through the poles) have distance between them that measures exactly an arcdegree. In this way, the distance around the Earth measures  $360^\circ$ .

### 3.3.1 Cartesian and Polar coordinates

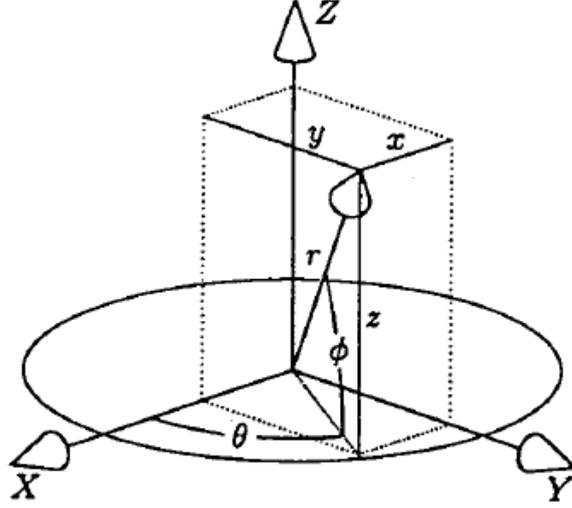


Figure 3.7: Cartesian and polar coordinates, [5].

The Cartesian (or Euclidean coordinates) and the Polar coordinates (Fig. 3.7) are related by these equations, as reported in [5]:

$$\begin{cases} x = r \cos \theta \cos \phi \\ y = r \sin \theta \cos \phi \\ z = r \sin \phi \\ r = \sqrt{x^2 + y^2 + z^2} \\ \phi = \arcsin\left(\frac{z}{r}\right) \quad \left(-\frac{\pi}{2} \leq \phi \leq \frac{\pi}{2}\right) \\ \theta = \arctan\left(\frac{y}{x}\right) \quad \left(-\pi \leq \theta \leq \pi\right) \end{cases} \quad (3.1)$$

### 3.3.2 Parametric Latitude

Let's consider  $z$  the Cartesian coordinate in polar direction and  $x_{meridional}$ , as in Fig. 3.8, the equatorial coordinate lying on the meridional plane [5].

The parametric solution for the ellipse is:

$$\frac{x_{meridional}^2}{a^2} + \frac{z^2}{b^2} = \frac{[a \cos(\phi_{parametric})]^2}{a^2} + \frac{[b \sin(\phi_{parametric})]^2}{b^2}$$

$$\begin{cases} x_{meridional} = a \cos \phi_{parametric} \\ z = b \sin \phi_{parametric} \end{cases} \quad (3.2)$$

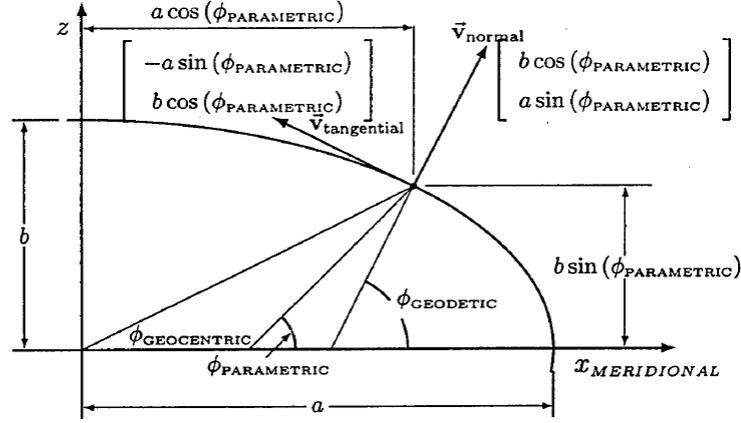


Figure 3.8: Geocentric, parametric and geodetic latitudes in meridional plane, [5].

### 3.3.3 Geodetic Latitude, Longitude and Altitude

In order to get Geodetic Latitude [5], we must calculate the tangent and normal direction of velocity:

$$v_{\text{tangential}} \propto \frac{\partial}{\partial \phi_{\text{parametric}}} \begin{bmatrix} a \cos(\phi_{\text{parametric}}) \\ b \sin \phi_{\text{parametric}} \end{bmatrix} = \begin{bmatrix} -a \sin(\phi_{\text{parametric}}) \\ b \cos \phi_{\text{parametric}} \end{bmatrix}$$

$$v_{\text{normal}} \propto \begin{bmatrix} b \cos(\phi_{\text{parametric}}) \\ a \sin \phi_{\text{parametric}} \end{bmatrix}$$

In this way we get:

$$\tan(\phi_{\text{geodetic}}) = \frac{a \sin \phi_{\text{parametric}}}{b \cos \phi_{\text{parametric}}} = \frac{a}{b} \tan \phi_{\text{parametric}}$$

And using trigonometric identities:

$$\sin \phi_{\text{geodetic}} = \frac{\tan \phi_{\text{geodetic}}}{\sqrt{1 + \tan^2(\phi_{\text{geodetic}})}} = \frac{a \sin \phi_{\text{parametric}}}{\sqrt{a^2 \sin^2(\phi_{\text{parametric}}) + b^2 \cos^2(\phi_{\text{parametric}})}}$$

$$\cos \phi_{\text{geodetic}} = \frac{1}{\sqrt{1 + \tan^2(\phi_{\text{geodetic}})}} = \frac{b \cos \phi_{\text{parametric}}}{\sqrt{a^2 \sin^2(\phi_{\text{parametric}}) + b^2 \cos^2(\phi_{\text{parametric}})}}$$

At this point, we obtain the cartesian coordinates x-z in terms of geodetic latitude:

$$x_{meridional} = a \cos \phi_{parametric} = \frac{a^2 \cos \phi_{geodetic}}{\sqrt{a^2 \cos^2(\phi_{geodetic}) + b^2 \sin^2(\phi_{geodetic})}}$$

$$z = b \sin \phi_{parametric} = \frac{b^2 \sin \phi_{geodetic}}{\sqrt{a^2 \cos^2(\phi_{geodetic}) + b^2 \sin^2(\phi_{geodetic})}}$$

And using the orthometric height  $h$ :

$$x_{meridional} = \cos \phi_{geodetic} \left( h + \frac{a^2}{\sqrt{a^2 \cos^2(\phi_{geodetic}) + b^2 \sin^2(\phi_{geodetic})}} \right)$$

$$z = \sin \phi_{geodetic} \left( h + \frac{b^2}{\sqrt{a^2 \cos^2(\phi_{geodetic}) + b^2 \sin^2(\phi_{geodetic})}} \right)$$

In ECEF coordinates, considering that the x axis passes through the equator at longitude ( $\theta$ ) = 0, we get:

$$\begin{cases} x_{ECEF} = \cos \theta x_{meridional} = \cos \theta \cos \phi_{geodetic} \left( h + \frac{a^2}{\sqrt{a^2 \cos^2(\phi_{geodetic}) + b^2 \sin^2(\phi_{geodetic})}} \right) \\ y_{ECEF} = \sin \theta x_{meridional} = \sin \theta \cos \phi_{geodetic} \left( h + \frac{a^2}{\sqrt{a^2 \cos^2(\phi_{geodetic}) + b^2 \sin^2(\phi_{geodetic})}} \right) \\ z_{ECEF} = \sin \phi_{geodetic} \left( h + \frac{b^2}{\sqrt{a^2 \cos^2(\phi_{geodetic}) + b^2 \sin^2(\phi_{geodetic})}} \right) \end{cases} \quad (3.3)$$

In terms of geodetic latitude ( $\phi_{geodetic}$ ), longitude ( $\theta$ ) and altitude  $h$ .

The inverse transformation can be express as:

$$\begin{cases} \theta = \text{atan2}(y_{ECEF}, x_{ECEF}) \\ \phi_{geodetic} = \text{atan2}\left(z_{ECEF} + \frac{e^2 a^2 \sin^3 \zeta}{b}, \xi - e^2 a \cos^3 \zeta\right) \\ h = \frac{\xi}{\cos \phi} - r_t \end{cases} \quad (3.4)$$

Where atan2 is the four-quadrant arctangent function in MATLAB and:

$$\begin{cases} \zeta = \text{atan2}(az_{ECEF}, b\xi) \\ \xi = \sqrt{x_{ECEF}^2 + y_{ECEF}^2} \\ r_t = \frac{a}{\sqrt{1 - e^2 \sin^2 \phi}} \end{cases} \quad (3.5)$$

Where:

- $r_t$  is the transverse radius of curvature on the ellipsoid;
- $a$  the equatorial radius;
- $b$  is the polar radius;
- $e$  is the elliptical eccentricity.

To calculate geodetic longitude, latitude and altitude is possible to use several MATLAB functions as: "*ecf2geodetic.m*" [15] or "*ecf2lla.m*" [16]. Both these functions require the ECEF position and the actual date as input.



## Chapter 4

# Electric Orbit - Raising

For what concern MAGNETO, the propulsion we use is the electric one. First, the electric propulsion is safer in respect of the chemical one. Furthermore, the engines are more efficient, and they require much less propellant to produce the same effect (even 20 times less).

The problem is that the electric propulsion can't be used as primary propulsion. In fact, even though the specific impulse can be much higher than the chemical one, the electric thruster flows are very small than the chemical ones, and they take much more times to achieve a particular speed.

On the other hand, the force produced by electric thruster can be applied continuously in order to reach a more accurate position.

There are some useful generalities in order to understand the Electric propulsion [7].

### 4.1 Generalities of Electric propulsion

The types of engine can be classified according to the source from which the energy or the type of acceleration is obtained.

- Chemical propulsion;
- Electric propulsion: is used exclusively in space and is characterized by a very low thrust-to-weight ratio and high (effective) exhaust output velocity.

#### Thrust

According to the law of action-reaction (similar to the Newton's law) we get:

$$m \frac{dV}{dt} = \dot{m}_p c$$

which allows us to obtain:

$$T = \dot{m}_p c$$

Where:

- $T$  = Thrust [N];
- $c$  = exhaust velocity;
- $\dot{m}_p$  = propellant mass flow rate.

Thrust is what we used in order to change spacecraft velocity.

### Thrust power

The thrust power is the energy per unit time we need to expel the propellant with  $c$ :

$$P_T = \frac{1}{2} \dot{m}_p c^2 = \frac{Tc}{2}$$

### Total impulse

$$I_t = \int_{t_0}^{t_f} T dt$$

If  $T = \text{constant}$ :

$$I_t = T \Delta t$$

### Total propellant mass

$$m_p = \int_{t_0}^{t_f} \dot{m}_p dt$$

If  $\dot{m}_p = \text{constant}$ :

$$m_p = \dot{m}_p \Delta t$$

### Specific Impulse

The specific impulse measures how efficiently the satellite consumes propellant mass.

$$I_{sp} = \frac{I_t}{m_p g_0} [s]$$

Where:

- $m_p g_0$ : propellant weight we would have consumed on Earth (at sea level).

If  $T$  and  $\dot{m}_p = \text{constant}$  (and so is  $c$ ):

$$I_{sp} = \frac{T\Delta t}{\dot{m}_p\Delta t g_0} = \frac{c}{g_0}$$

### 4.1.1 Rocket Equation

The characteristic velocity is defined as the velocity increment that a spacecraft with a variable mass  $m$  gets in the ideal case (no other forces considered and thrust parallel to the velocity) [7].

$$\Delta V = \int_{t_i}^{t_f} (T/m) dt$$

Where the mass variation is  $\dot{m} = \dot{m}_p$ .

If  $c = \text{constant}$ , the Rocket (Tsiolkovsky's) equation is:

$$\Delta V = c \ln\left(\frac{m_0}{m_f}\right)$$

or

$$\frac{m_f}{m_0} = e^{-\frac{\Delta V}{c}}$$

### 4.1.2 Velocity Losses

The ideal  $\Delta V$  is not the same as the velocity change because of thrust misalignment, gravity and aerodynamic force [7], as show in Fig. 4.1.

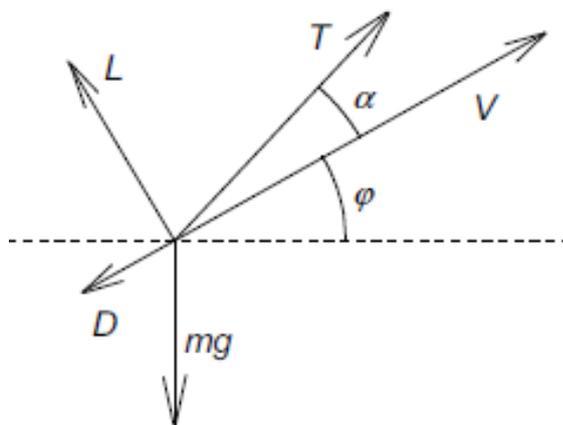


Figure 4.1: Internal and external forces acting on a spacecraft, [7].

$$\frac{d\vec{V}}{dt} = \frac{\vec{T}}{m} + \frac{\vec{L} + \vec{D}}{m} + \vec{g}$$

$$V_f - V_i = \int_{t_i}^{t_f} \frac{T}{m} dt - \int_{t_i}^{t_f} \frac{T}{m} (1 - \cos \alpha) dt - \int_{t_i}^{t_f} \frac{D}{m} - \int_{t_i}^{t_f} g \sin \phi dt$$

- Misalignment losses:  $\int_{t_i}^{t_f} \frac{T}{m} (1 - \cos \alpha) dt$  (thrust not parallel to velocity,  $\alpha \neq 0$ );
- Aerodynamic losses:  $\int_{t_i}^{t_f} \frac{D}{m}$  (absent outside atmosphere);
- Gravity losses:  $\int_{t_i}^{t_f} g \sin \phi dt$  (absent if  $\phi = 0$ ).

### 4.1.3 Electric propulsion maneuver

When we are considering electric propulsion maneuver, the hypotheses of impulsive maneuver fall as the electric propulsion is characterized by low acceleration and low thrust. Considering the Edelbaum problem hypotheses:

- almost circular orbit:  $r \approx a \approx p, e \approx 0, V^2 = \frac{\mu}{r^2}$ ;
- $E \approx \nu \approx M$ ;
- almost equatorial orbit  $i \approx 0, \cos i \approx 1, \sin i \approx 0$  (the final reference plane orbit is the same as the initial one);
- $\frac{T}{m} \ll \frac{\mu}{r^2}$ ;
- $A_V \approx A_T \ll \frac{\mu}{r^2}, A_R \ll \frac{\mu}{r^2}, A_W \ll \frac{\mu}{r^2}$ .

There is a gradual variation in speed which leads the spacecraft to perform a spiral trajectory.

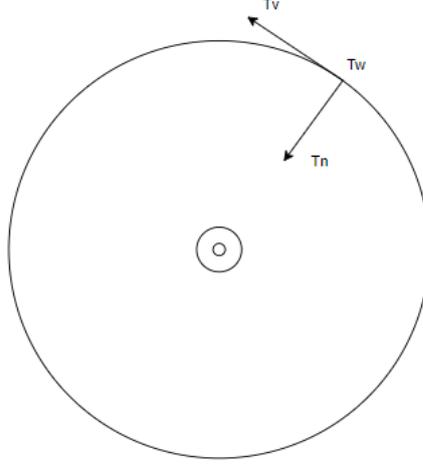


Figure 4.2: Thrust.

As we can see in Fig. 4.2, the thrust presents 3 components:

- $T_V \parallel V$  causes the variation of the energy ( $a, e, \omega$ );
- $T_N \perp V$  which doesn't change the energy;
- $T_W$  who causes changes in terms of  $i$  and  $\Omega$ .

In every point of the trajectory there is an optimal direction for what concern the thrust in order to modify the desired parameters. The Gauss planetary equations describe the temporal variation of the orbital parameters:

$$\frac{da}{dt}, \frac{de}{dt}, \frac{di}{dt}, \frac{d\omega}{dt}, \frac{d\Omega}{dt}, \frac{d\nu}{dt},$$

Using the Edelbaum model we can write the Gauss planetary equations as:

$$\begin{cases} \dot{a} = 2 \frac{T_V}{m} \frac{a}{V} \\ \dot{e} = [2 \cos \nu \frac{T_V}{m} - \sin \nu \frac{T_N}{m}] \frac{1}{V} \\ \dot{\omega} = -\dot{\Omega} + [2 \sin \nu \frac{T_V}{m} + \cos \nu \frac{T_N}{m}] \frac{1}{Ve} \\ \dot{i} = \cos(\omega + \nu) \frac{T_W}{m} \frac{1}{V} \\ \dot{\Omega} = \sin(\omega + \nu) \frac{T_W}{m} \frac{1}{iV} \\ \dot{M} = \dot{\nu} = \sqrt{\frac{\mu}{a}} \end{cases} \quad (4.1)$$

In case  $e \approx 0, i \approx 0, \dot{\omega}$  and  $\dot{\Omega} \rightarrow \infty$  because the ascending node and the periaspis are not defined. Using the Edelbaum hypotheses we can ignore the equations which present  $\dot{\omega}$  and  $\dot{\Omega}$  and so we get:

$$\begin{cases} \dot{a} = 2\frac{T_V}{m} \frac{a}{V} \\ \dot{e} = [2 \cos \nu \frac{T_V}{m} - \sin \nu \frac{T_N}{m}] \frac{1}{V} \\ \dot{i} = \cos(\omega + \nu) \frac{T_W}{m} \frac{1}{V} \end{cases} \quad (4.2)$$

In order to optimize the thrust direction, we introduce  $\alpha$  (in-plane angle, Fig. 4.3) and  $\beta$  (out-of-plane angle).

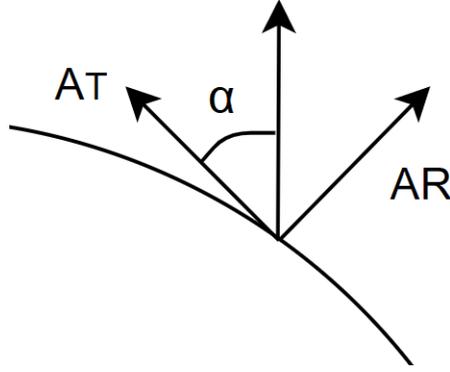


Figure 4.3: In-plane accelerations.

$$\begin{cases} T_V = T \cos \alpha \cos \beta \\ -T_N = T \sin \alpha \cos \beta = T_R \\ T_W = T \sin \beta \end{cases} \quad (4.3)$$

Edalbaum considers three problems, but for our case the only one that really matters is the one in which the change regards the semi-major axis  $a$ .

#### 4.1.4 Variation of $a$

In order to obtain the maximum increment of  $a$ , we must direct the thrust tangentially:

$$\alpha = \beta = 0 \rightarrow T \parallel V \rightarrow T_V = T, T_R = T_W = 0$$

As the thrust is applied,  $e$  change, but if the thrust is applied continuously, once the satellite complete a round around the Earth,  $e$  returns to cancel itself. In this way, we obtain again a circular orbit:

$$\Delta e = \Delta i = 0$$

## 4.2 Electromagnetic Theory

### 4.2.1 Electric Field

The Electric field is generated by charges, as shown in Fig. 4.4.

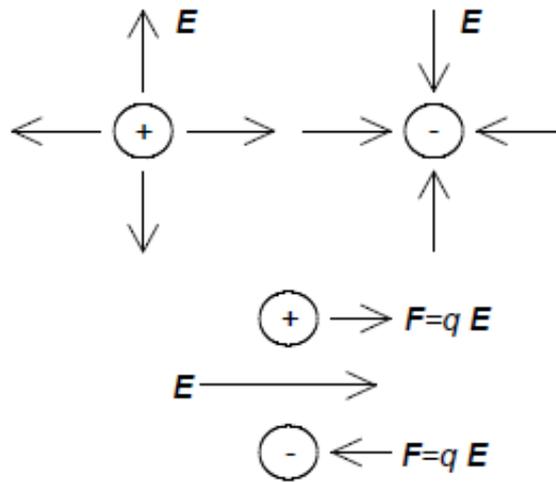


Figure 4.4: Electric Field, [7].

$$E = \frac{Qr}{4\pi\epsilon r^3} [N/C]$$

Where:

- $Q$  is the charge [C];
- $\epsilon$  is dielectric constant [F/m].

It creates the electrostatic force  $F = qE$  which is attractive for opposite charges and vice versa.

### 4.2.2 Currents

The current is a flux of moving charges:

$$j = nqv$$

Where:

- $n$  number of charges per unit volume;
- $q$  charge of a single particle;
- $v$  particle mean velocity.

### 4.2.3 Magnetic Field

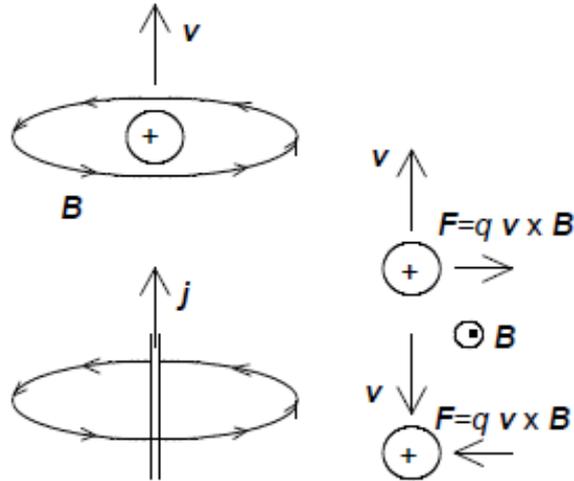


Figure 4.5: Magnetic Field, [7].

Currents generate the Magnetic Field  $B$  (Fig. 4.5).

$$B = \frac{\mu Q}{4\pi} \frac{v \times r}{r^3} [T]$$

Where  $\mu$  is magnetic permeability [Hr/m].

$\vec{B}$  creates a magnetic force  $F = qv \times B$  which acts on moving charges.

### 4.2.4 Maxwell Equations

General form:

$$\left\{ \begin{array}{l} \nabla \cdot D = q_{vol} \\ \nabla \times E = -\dot{B} \\ \nabla \cdot B = 0 \\ \nabla \times H = j + \dot{D} \end{array} \right. \quad (4.4)$$

Propulsion plasma:

$$\left\{ \begin{array}{l} \nabla \cdot E = q_{vol}/\epsilon_0 \\ \nabla \times E = -\dot{B} \\ \nabla \cdot B = 0 \\ \nabla \times B = \mu_0 j + \epsilon_0 \mu_0 \dot{E} \end{array} \right. \quad (4.5)$$

Where:

- $D$  = electric displacement;
- $H$  = magnetic field intensity;
- $q_{vol}$  charge density.

## 4.3 Particle Motion

### 4.3.1 $B = 0$

The Lorentz's force is:  $\vec{F} = q(\vec{E} + \vec{v} \times \vec{B})$ . If we are not considering the collision, the particles would accelerate their motion to infinity, in order to reach  $v = \infty$  at  $t = \infty$ . Instead, if we are taking count of the collisions, we must consider damping. In this way the mean particle equation becomes:

$$\vec{F} - \nu_c m \vec{v} = m \frac{d\vec{v}}{dt}$$

Where  $\nu_c$  is momentum collision frequency. If  $\vec{E} = \text{constant}$ :

$$\left\{ \begin{array}{l} \vec{v} = \frac{q}{(m\nu_c)} \vec{E} \\ \vec{j} = \frac{nq^2}{(m\nu_c)} \vec{E} \end{array} \right. \quad (4.6)$$

and the particles motions and the current are parallel to E.

### 4.3.2 $\mathbf{E} = 0$

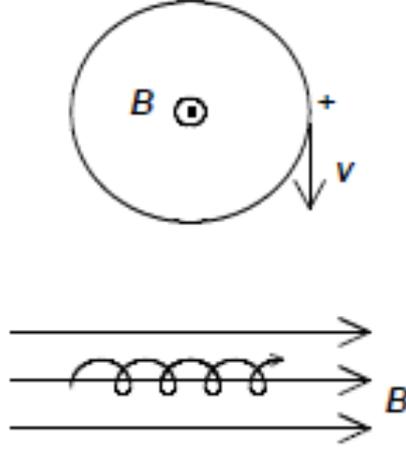


Figure 4.6: Particle motion -  $E = 0$ , [7].

If  $\nu_c = 0$  and  $B = \text{constant}$ , the motion is uniform along  $\vec{B}$  and  $v_{\parallel} = \text{constant}$ , Fig. 4.6. The circular motion in the plane is normal to  $B$  and the trajectory we get is a helix. The Larmor (or cyclotron or gyro radius) is :  $r_b = mv_{\perp}/(qB)$ , while the frequency is  $\omega_b = qB/m$ .

### 4.3.3 $\mathbf{E}, \mathbf{B} = \text{constant}$

If  $\nu_c = 0$  we have an accelerated motion along  $B$  ( $v_{\parallel} = qE_{\parallel}/m$ ) and a drift motion in direction  $E \times B$  ( $v_d = E \times B/B^2$ ).

If  $\nu_c \neq 0$ :

$$\begin{cases} v = \frac{\nu_c^2(q/m)\vec{E} + \nu_c(q/m)^2(\vec{E} \times \vec{B}) + (q/m)^3(\vec{E} \cdot \vec{B})\vec{B}}{\nu_c(\nu_c^2 + (qB/m)^2)} \\ j = \frac{\sigma_0}{1 + \Omega^2} [\vec{E} \pm \Omega(\vec{E} \times \vec{b}) + \Omega^2(\vec{E} \cdot \vec{b})\vec{b}] \\ j = \sigma_0 \vec{E}_{\parallel} + \frac{\sigma_0}{1 + \Omega^2} [\vec{E}_{\perp} \pm \Omega(\vec{E} \times \vec{b})] \end{cases} \quad (4.7)$$

Where:

- $\Omega = \omega_b/\nu_c = qB/(m\nu_c) = \text{Hall parameter}$ ;
- $\vec{b} = \vec{B}/b = \text{magnetic field unit vector}$ ;

- $\sigma_0 = nq^2/(m\nu_c) =$  scalar conductivity.

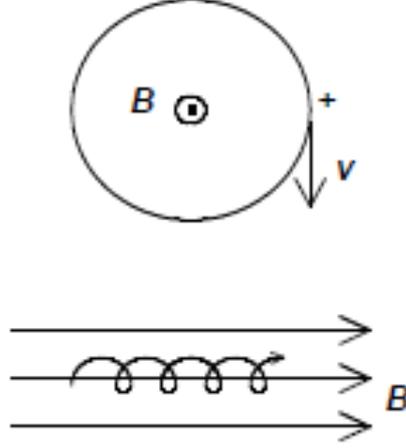


Figure 4.7: Particle motion -  $E, B = \text{constant}$ , [7].

- $\Omega \ll 1$ : there are many collisions,  $v$  is small and the effect of  $B$  is negligible;
- $\Omega \gg 1$ : there are few collisions,  $v$  is large and so the effect of  $B$ .

## 4.4 HT Thruster

It is a hybrid between an electrostatic and electromagnetic propulsion. It has the great advantage of not being subject to the thrust density limitation imposed by Child's law (ion thruster limit) thanks to the quasi neutral plasma present in the chamber.

The operating principle is based on the acceleration of an ionized working fluid (propellant) by the mutual action of the overlap of a magnetic field ( $\vec{B} = 0.1T$ ) and an electric field ( $\vec{E}$ ) orthogonal to each other and directed respectively radially and along the axis of the thruster. Usually they are made with cylindrical symmetry. Generally, the gas used is Xenon as it has high atomic mass and low ionization potential.

The magnetic field is generated by a magnetic circuit consisting of coils traversed by electric current and ferromagnetic elements in order to direct the magnetic field lines in the area of introduction of the propellant. The electric field is generated by an electrostatic potential ( $V_d \approx 300V$ ) held by a cathode (negatively polarized electrode) external to the motor and by an internal anode. In this way there is a low propellant density and a high Hall parameter ( $\Omega \approx 100$ ).

An electron current ( $-J_d$ ) is emitted from the cathode and flows axially towards the anode. When electrons enter the motor, they are affected by the magnetic field,

and they are trapped. In this way, their movement towards the anode is canceled, and they obtain an azimuth speed (orthogonal to the electric and magnetic field) creating a circumferential electronic current ( $J_\theta$ ) inside the thruster for the Hall effect. This discharge having a high density of high energy electrons allows the ionization of the propellant. The propellant is injected into the engine by the anode and it is ionized by the impact between the electrons and the neutral atoms of the working gas. At each collision the electron advances generating a current along the anode.

The mean free path of atoms is much smaller than the channel size and therefore each atom is ionized. The distribution of electrons generates a virtual cathode which generates an electrostatic potential with the anode allowing the acceleration of the ions produced. The ions accelerated by the electrostatic difference are not affected by the action of the magnetic field. In fact, due to their high atomic mass, they have a very large cyclotron radius and travel almost straight rectilinear paths along the motor axis. The ions are therefore accelerated by the electric field and generate thrust.

In tab. 4.1 we can see the performance of an HT Thruster.

Propellant	Xe	
$I_{sp}$	1500 – 2500	s
$P_E$	300 – 6000	W
$\eta$	0.5	
Voltage	200 – 600	V
Thruster mass	2 – 3	kg/kW
PPU mass	6 – 10	kg/kW
Feed System	regulated	
<i>lifetime</i>	> 7000	kg
<i>mission</i>	Transfer (med $\Delta V$ )	

Table 4.1: HT thruster performance, [7].

#### 4.4.1 SITAEEL HT20k Thruster

HT20k (Fig. 4.8) is a high-power Hall Effect Thruster (HET), [8]. This thruster has been designed to operate with nominal discharge power of 20kW. In tab. 4.2 there are the operative ranges of thrust, specific impulse and discharge voltage.

---

	MIN	MAX
Thrust [N]	0.86	1.08
$I_{sp}$ [s]	1908	2777
Discharge Voltage [V]	300	600

Table 4.2: Working Point characteristics (credit: SITAEL, [8]).

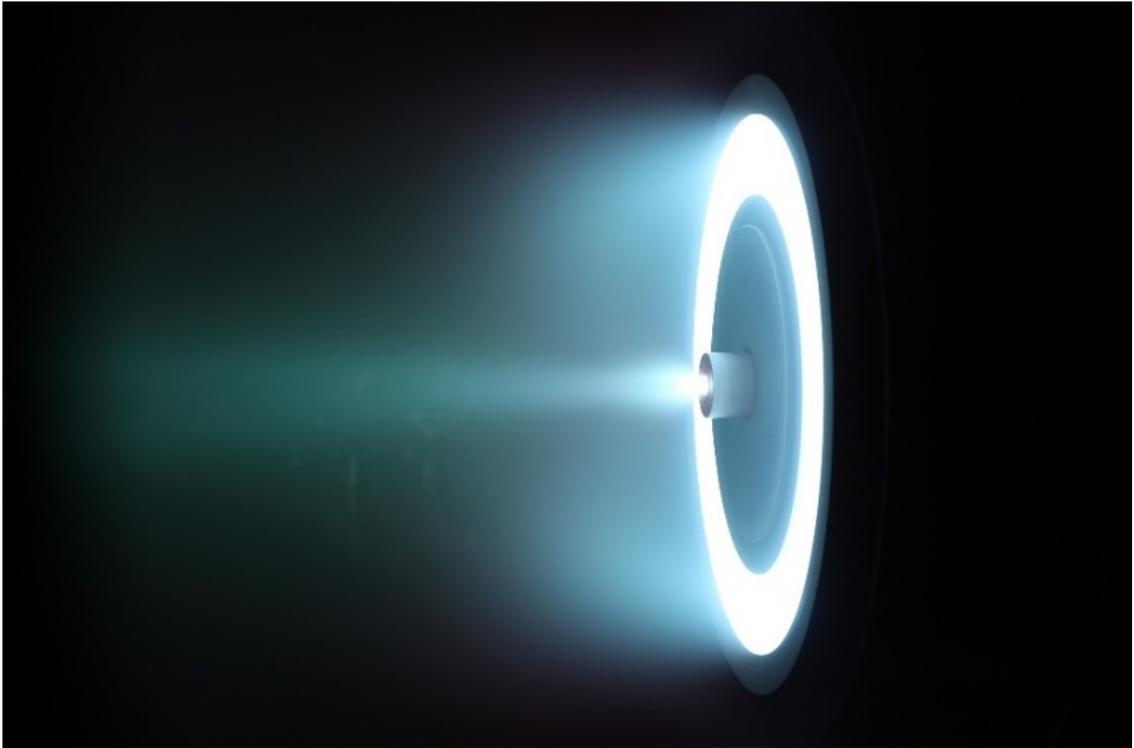


Figure 4.8: SITAEL HT20k Thruster (credit: SITAEL, [8]).



# Chapter 5

## EOR - MAGNETO

Orbit Raising is defined a type of orbital maneuver which causes a progressive increase the radius of the spacecraft's orbit. We can distinguish two types of maneuver:

- Low Thrust (high specific impulse and low thrust  $\rightarrow$  Electric propulsion);
- High Thrust (chemical propulsion, as described in chapter 2.6).

For what concern the Electric Orbit Raising (low thrust), the orbital maneuver is one of the simplest and most advantageous systems to use. In fact, it allows us to save propellant to be brought on board. The electric propulsion system used presents extremely reduced thrust (about  $1N$ ), low acceleration (about  $10^{-4}/10^{-6}g_0$ ) but also long transfer times.

The geometry of the trajectories is spiral.

For what concern the propellant consumption, Low Thrust maneuver represent the best possible choice thanks to the high Specific Impulse value.

To perform the EOR maneuver, MAGNETO needs JPL Spice.

### 5.0.1 JPL Spice

NASA's NAIF (The Navigation and Ancillary Information Facility) is a facility which offers "SPICE" observation geometry information in order to assist space agencies, scientist and engineers in solve space problems [9].

The SPICE system includes a software suite: the SPICE Toolkit. This toolkit is a collection of freely available user-level application program interfaces in which we may find some ready-to-use library and application offered in some languages such as C, FORTRAN, IDL, MATLAB and Java.

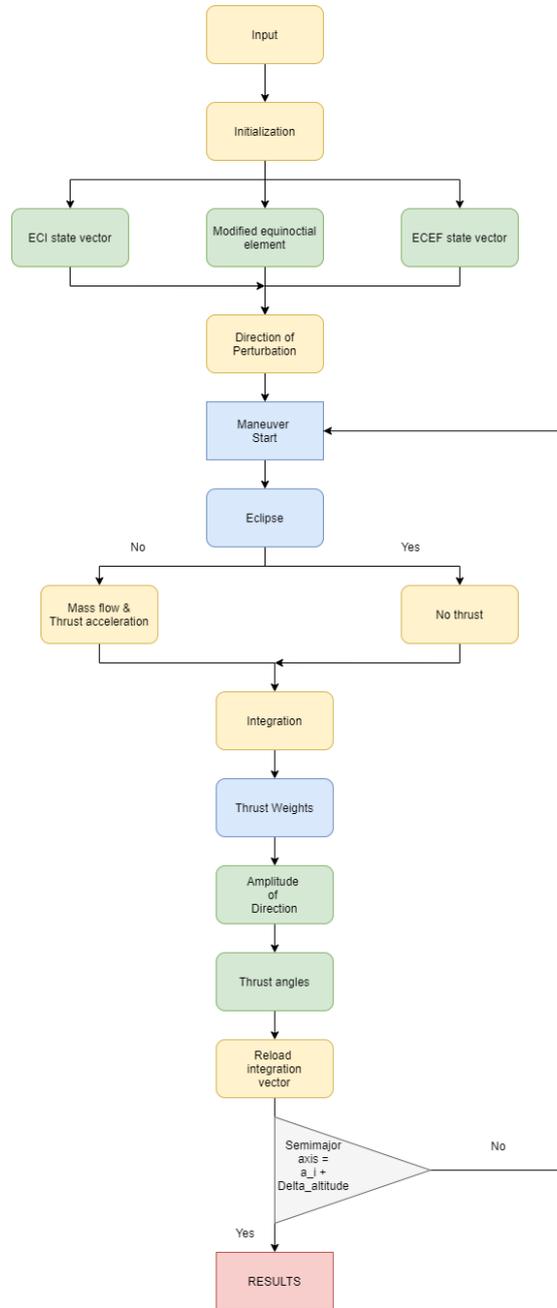


Figure 5.1: *EOR\_main.m*.

## 5.1 EOR main

To perform an EOR maneuver MAGNETO uses a function called "*EOR\_main.m*" function. As we can see in Fig. 5.1, the function needs as inputs: the date, the

orbital parameters, the number of thruster, the thrust for thruster, the specific impulse, the initial mass, the drag coefficient, the surfaces and some flag that are useful to control the orbital parameters change.

The function needs also the astrodynamics and utility constants.

First, using the JPL Spice function ("*cspice\_str2et.m*"), the function calculates the ephemeris time starting from the date given as an input. Then, the orbital element vector is initialized and then the function "*orb2mee.m*" converts it in a modified equinoctial orbital elements vector [26].

$$\begin{cases} p = a(1 - e^2) \\ f = e \cos(\omega + \Omega) \\ g = e \sin(\omega + \Omega) \\ h = \tan(i/2) \cos \Omega \\ k = \tan(i/2) \sin \Omega \\ L = \Omega + \omega + \nu \end{cases} \quad (5.1)$$

Where:

- $a$  = semi-major axis;
- $e$  = orbital eccentricity;
- $i$  = orbital inclination;
- $\omega$  = argument of perigee;
- $\Omega$  = right ascension of the ascending node;
- $\nu$  = true anomaly;
- $L$  = true longitude.

Using "*mee2eci.m*" it's possible to obtain the position and velocity of the space tug in ECI coordinates.

The relationships between ECI state vector and modified equinoctial ([10]) elements are:

$$r_{ECI} = \begin{bmatrix} \frac{r}{s^2}(\cos L + \alpha^2 \cos L + 2hk \sin L) \\ \frac{r}{s^2}(\sin L - \alpha^2 \sin L + 2hk \cos L) \\ \frac{2r}{s^2}(h \sin L - k \cos L) \end{bmatrix}$$

$$v_{ECI} = \begin{bmatrix} -\frac{1}{s^2} \sqrt{\frac{\mu}{p}} (\sin L + \alpha^2 \sin L - 2hk \cos L + g - 2fhk + \alpha^2 g) \\ -\frac{1}{s^2} \sqrt{\frac{\mu}{p}} (-\cos L + \alpha^2 \cos L + 2hk \sin L - f + 2ghk + \alpha^2 f) \\ \frac{2}{s^2} \sqrt{\frac{\mu}{p}} (h \cos L + k \sin L + fh + gk) \end{bmatrix}$$

Where:

- $\alpha^2 = h^2 - k^2$ ;
- $s^2 = 1 + h^2 + k^2$ ;
- $r = \frac{p}{w}$ ;
- $w = 1 + f \cos L + g \sin L$ .

It's now possible to calculate the ECEF positions and velocities using "*ECItoECEF.m*" function and the latitude, longitude and altitude using "*ecf2lla.m*" [16]. Both these functions follow the procedure described in chapter 3.

Imposing a  $\Delta t$  and the number of differential equation, the function plots all the variables that it needs. Then, it computes the directions of the perturbations checking all the orbital parameters from initial and final orbits.

We enter later in the while loop imposing the initial time to zero. The flow of the propellant and its mass is initially calculated in order to obtain the thrust and its acceleration in this way:

$$\begin{cases} \dot{m}_p = -\frac{T}{I_{sp}g} \\ m = m_i + \dot{m}_p(t_f - t_i) \\ a_T = \frac{T}{m} \end{cases} \quad (5.2)$$

### 5.1.1 Eclipse calculator

Another function necessary to define the eclipse is performed. Using the function "*eclipse\_calculator.m*" is possible to establish if the satellite is in umbra, penumbra or in sunlight.

The size and shape of umbra (Fig. 5.2) and penumbra regions (Fig. 5.3) are calculated using the planet and the Sun's size and their distance [27].

Umbra region is where the blockage of the solar light is total, while for what concern penumbra region it is partial. In order to control if the satellite we are considering is in umbra, penumbra or light, we must define the shadow cone surfaces. We consider the celestial bodies as spherical and their shadows are conical.

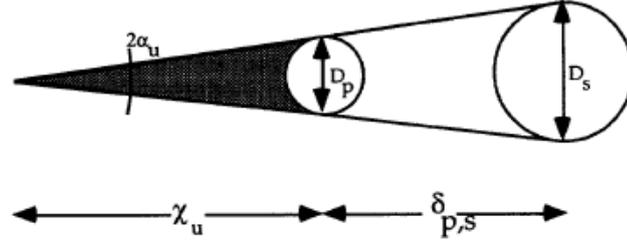


Figure 5.2: Umbral cone geometry [12].

This allows us to obtain:

$$\begin{cases} \chi_u = \frac{D_p \delta_{p-s}}{(D_s - D_p)} \\ \alpha_u = \sin^{-1} \left( \frac{D_p}{2\chi_u} \right) \end{cases} \quad (5.3)$$

Where:

- $D_p$  is the planet diameter;
- $D_s$  is the Sun diameter;
- $\delta_{p-s}$  is the distance between Sun and planet;
- $\chi_u$  umbra cone altitude;
- $\alpha_u$  half umbra cone angle.

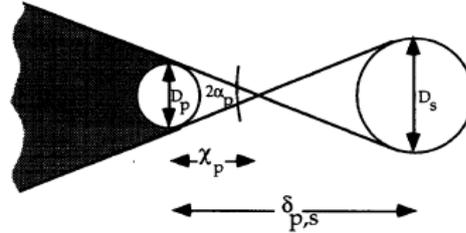


Figure 5.3: Penumbral cone geometry [12].

As the same way, we get:

$$\begin{cases} \chi_p = \frac{D_p \delta_{p-s}}{(D_s + D_p)} \\ \alpha_p = \sin^{-1}\left(\frac{D_p}{2\chi_p}\right) \end{cases} \quad (5.4)$$

Where:

- $\chi_u$  penumbra cone altitude;
- $\alpha_u$  half penumbra cone angle.

Then we must define the projection vector as:

$$\vec{r}_s = (\vec{r} \cdot \hat{s})\hat{s}$$

$\vec{r}_s$  and  $\vec{\delta} = \vec{r} - \vec{r}_s$  are shown in 5.4.

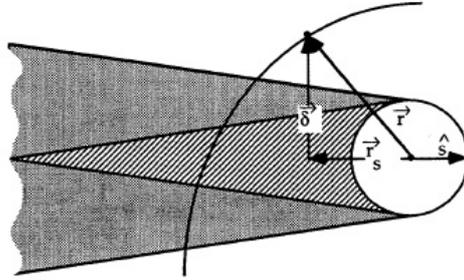


Figure 5.4:  $\vec{r}_s$  and  $\delta$  [12].

Then, we define the distance:

$$\begin{cases} \xi = (\chi_u - |\vec{r}_s|) \tan \alpha_u \\ \kappa = (\chi_p + |\vec{r}_s|) \tan \alpha_p \end{cases} \quad (5.5)$$

Where  $\xi$  is the distance between the center of the umbra cone and his end, and  $\kappa$  is the same as  $\xi$  but for the penumbra cone.

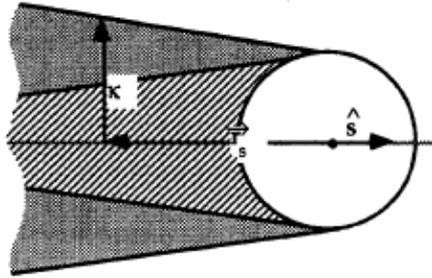


Figure 5.5: Umbra [12].

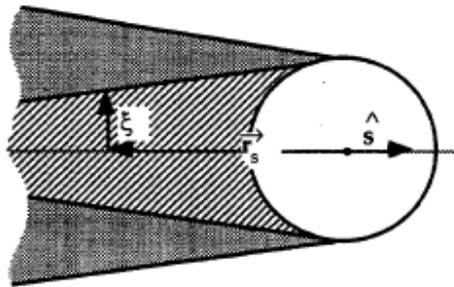


Figure 5.6: Penumbra [12].

- if  $(|\vec{\delta}| > \kappa)$  the spacecraft is in sunlight;
- if  $(\xi < |\vec{\delta}| < \kappa)$  the spacecraft is in penumbra (Fig. 5.6);
- if  $(\vec{\delta} < \xi)$  the spacecraft is in umbra (Fig. 5.5).

In MAGNETO, if the satellite is in umbra or in penumbra no thrust is provided.

## 5.2 Integration

Now the equations must be integrated.

We use the "*meeqm.m*" function which allows us to obtain the equations of motion as a function of the modified equinoctial elements. The current modified equinoctial orbital elements must be loaded and so even the ECI state vectors and the classical orbital elements. Then, the perturbations are calculated taking into account:

- Thrust perturbation;
- Non-spherical gravity Earth  $J_2$ ;
- Aerodynamic Drag;
- Secondary body perturbation;
- Solar pressure.

### 5.2.1 Propulsive thrust

The general formulas for thrust perturbation are:

$$\begin{cases} \Delta r_t = a_T \cdot \sin(\alpha) \cdot \cos(\beta) \\ \Delta t_t = a_T \cdot \cos(\alpha) \cdot \cos(\beta) \\ \Delta r_t = a_T \cdot \sin(\beta) \end{cases} \quad (5.6)$$

Where:

- $\alpha$  is the in-plane-angles or pitch angle;
- $\beta$  is the out-of-plane angle or yaw angle;
- $a_T = \frac{T}{m}u$  is the thrust acceleration;
- $T$  is the thrust;
- $m$  is the spacecraft mass;
- $u = [u_r \ u_t \ u_n]$ ;

### 5.2.2 Non-spherical gravity Earth

The potential gradient [10] is:

$$\begin{cases} \Delta r_{j2} = \frac{-3\mu J_2 r_{Earth}^2}{2r^4} \left[ 1 - \frac{12(h \sin L - k \cos L)^2}{(1+h^2+k^2)^2} \right]; \\ \Delta t_{j2} = \frac{-12\mu J_2 r_{Earth}^2}{r^4} \left[ \frac{(h \sin L - k \cos L)(h \cos L + k \sin L)^2}{(1+h^2+k^2)^2} \right]; \\ \Delta n_{j2} = \frac{-6\mu J_2 r_{Earth}^2}{r^4} \left[ \frac{(1-h^2-k^2)(h \sin L - k \cos L)^2}{(1+h^2+k^2)^2} \right]; \end{cases} \quad (5.7)$$

### 5.2.3 Aerodynamic Drag

The perturbations due to the aerodynamic drag are:

$$\begin{cases} \Delta D_r = \frac{1}{2}\rho S C_D v v_r; \\ \Delta D_t = \frac{1}{2}\rho S C_D v v_t; \\ \Delta D_n = 0; \end{cases} \quad (5.8)$$

Where:

- $\rho$  = atmospheric drag;
- $S$  = aerodynamic reference area;
- $C_D$  = drag coefficient;
- $v$  = velocity magnitude;
- $v_r = \sqrt{\frac{\mu}{p}}(f \sin L - g \cos L)$ ;
- $v_t = \sqrt{\frac{\mu}{p}}(1 + f \cos L + g \sin L)$ .

In the case analyzed, the aerodynamic drag is neglected because of the high altitude ( $> 15000km$ ) we consider [24].

### 5.2.4 Secondary Body perturbation

The equation that describes the second body perturbations [10] is:

$$\vec{t} = - \sum_{j=1}^n \mu_j \left[ \frac{\vec{d}_j}{d_j^3} + \frac{\vec{s}_j}{s_j^3} \right]$$

Where:

- $\vec{s}_j$  = vector from the primary to secondary body  $j$ ;
- $\mu_j$  = gravitational constant;
- $\vec{d}_j = \vec{r} - \vec{s}_j$ ;
- $r$  = position vector of the spacecraft relative to the primary body.

We use Battin's function in order to avoid numerical problems:

$$F(q_k) = q_k \left[ \frac{3 + 3q_k + q_k^2}{1 + (\sqrt{1 + q_k})^3} \right]$$

Where  $q_k = \frac{\vec{r}^T(\vec{r} - 2\vec{s}_k)}{s_k^T \vec{s}_k}$

In this way we get:

$$\vec{t} = \sum_{k=1}^n \frac{\mu_k}{d_k^3} [\vec{r} + F(q_k) \vec{s}_k]$$

The Secondary body perturbation, in modified equinoctial coordinate is:

$$a_{TB}^{\vec{}} = Q^T \vec{t}$$

Where  $Q = [\hat{i}_r \ \hat{i}_t \ \hat{i}_n]$

### 5.2.5 Solar - Radiation Pressure

The solar radiation pressure [25] is:

$$p_{SR} = 4.51 \cdot 10^{-6}$$

and the solar radiation force is:

$$F_{SR} = p_{SR} c_R A_{sun}$$

where:

- $c_R$  = reflectivity;
- $A_{sun}$  = exposed area to the Sun.

And the Solar Radiation acceleration is  $a_{SR} = F_{SR}/m$ . Normalizing:  $a_{SR}^{\vec{}} = a_{SR} \frac{\vec{d}_s}{d_s}$  ( $d_s$  = distance satellite-sun).

In this way we get:

$$a_{SolarRadiation}^{\vec{}} = Q^T a_{SR}^{\vec{}}$$

## 5.3 Modified equinoctial form of the orbital equations of motion

We get:

$$\begin{cases} \Delta_r = \Delta r_t + \Delta r_{j2} + \Delta D_r + \Delta r_{SBP} + \Delta r_{SR}; \\ \Delta_t = \Delta t_t + \Delta t_{j2} + \Delta D_t + \Delta t_{SBP} + \Delta t_{SR}; \\ \Delta_n = \Delta n_t + \Delta n_{j2} + \Delta D_n + \Delta n_{SBP} + \Delta n_{SR}; \end{cases} \quad (5.9)$$

As reported in [10], the Modified equinoctial form of the orbital equations of motion are as follow:

$$\begin{cases} \dot{p} = \frac{dp}{dt} = \frac{2p}{w} \sqrt{\frac{p}{\mu}} \Delta t \\ \dot{f} = \frac{df}{dt} = \sqrt{\frac{p}{\mu}} [\Delta r \sin L + [(w+1) \cos L + f] \frac{\Delta t}{w} - (h \sin L - k \cos L) \frac{g \Delta n}{w}] \\ \dot{g} = \frac{dg}{dt} = \sqrt{\frac{p}{\mu}} [-\Delta r \cos L + [(w+1) \sin L + g] \frac{\Delta t}{w} + (h \sin L - k \cos L) \frac{g \Delta n}{w}] \\ \dot{h} = \frac{dh}{dt} = \sqrt{\frac{p}{\mu}} \frac{s^2 \Delta n}{2w} \cos L \\ \dot{k} = \frac{dk}{dt} = \sqrt{\frac{p}{\mu}} \frac{s^2 \Delta n}{2w} \sin L \\ \dot{L} = \frac{dL}{dt} = \sqrt{\mu p} \left(\frac{w}{p}\right)^2 + \frac{1}{w} \sqrt{\frac{p}{\mu}} (h \sin L - k \cos L) \Delta n \end{cases} \quad (5.10)$$

Then the program uses the Runge-Kutta-Fehlberg 7(8) method to solve them.

The "*mee2eci.m*" function is used again for calculating the position and velocity in ECI coordinates and the "*mee2orb.m*" function for calculating the orbital parameters. Afterwards, the function calculates various weights in order to obtain the desired thrust angles.

The function "*ThrustWeights.m*" evaluates the weights for the initial guess thrust angles.

In this way, it's possible to evaluate the velocity vector and the thrust angle (in-plane angle  $\alpha$  and out-of plane angle  $\delta$ ).

The acceleration components [28] are expressed as:

$$\begin{cases} a_R = |\vec{a}| \cos \beta \sin \alpha \\ a_T = |\vec{a}| \cos \beta \cos \alpha \\ a_N = |\vec{a}| \sin \beta \end{cases} \quad (5.11)$$

Once this is done, the program plots again all the variables of interest and performs a final check on the semi-major axis.

This procedure is followed until the program "break" the "while" and that occurs when the semi-major axis reach the desired values imposed as input.

Once the program exits the while loop, it calculates the  $\Delta v$ , the final propellant mass and the mean thrust acceleration. In the end it plots the position in ECI and ECEF coordinates and all the parameters we want.

# Chapter 6

## EOR simulation

In this chapter will be presented some simulation for what concern the EOR maneuver.

The EOR maneuver it's necessary for both the OOSs presented in chapter 1. In fact, as we can see in Fig. 1.1 and 1.2, this maneuver it's the one necessary to bring our orbit object to its desired orbit.

Moreover, the maneuvers presented in this chapter are realistic maneuvers capable of simulating the launch of a telecommunication satellite with electric propulsion.

In order to understand the results, let's consider the first simulation (LEO to GEO).

### 6.1 LEO to GEO

In tab. 6.1 we can see the input used for this simulation.

$date$	2019 Feb 19 15 : 30 : 00	UTC;
$a_i$	$24421 \cdot 10^3$	[m]
$i_i$	0.1	[deg]
$e_i$	0.72	
$\Omega_i$	1	[deg]
$\omega_i$	0	[deg]
$\nu_i$	0	[deg]
$a_f$	$42168 \cdot 10^3$	[m]
$i_f$	0	[deg]
$e_f$	0.00045	
$\Omega_f$	0	[deg]
$\omega_f$	0	[deg]
$\nu_f$	0	[deg]
$thruster$	1	
$thrust\ per\ thruster$	$1000 \cdot 10^{-3}$	[N]
$I_{SP}$	2800	[sec]

Table 6.1: Input (EOR - LEO to GEO).

In Fig. 6.1, 6.2 and 6.3 we see the maneuver plot in 3-D, where in red we have the coordinates in ECI frame and in green the ECEF coordinates.

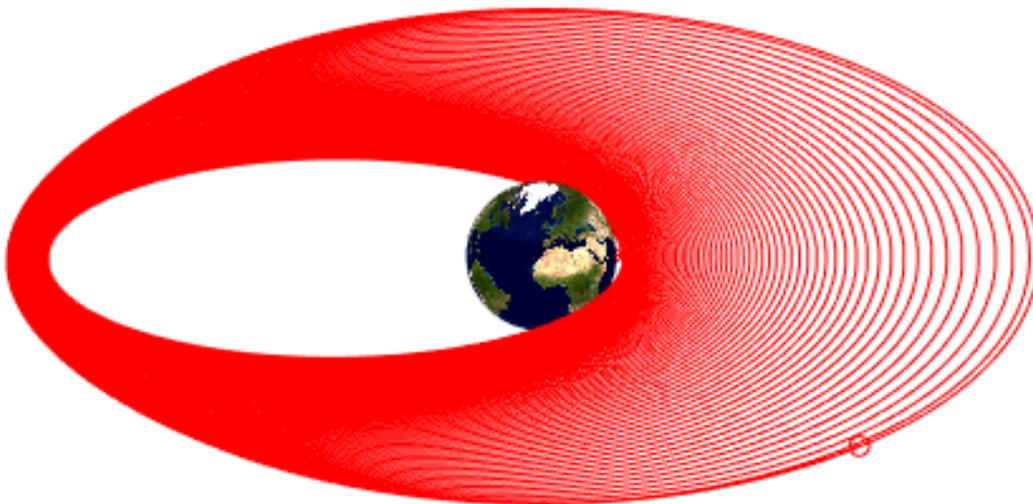


Figure 6.1: EOR maneuver in ECI (EOR - LEO to GEO).

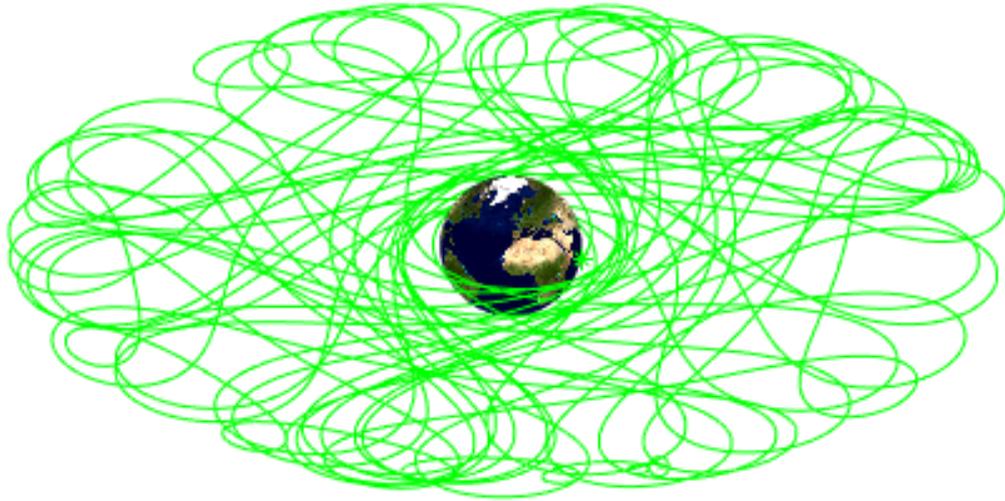


Figure 6.2: EOR maneuver in ECEF (EO - LEO to GEO).

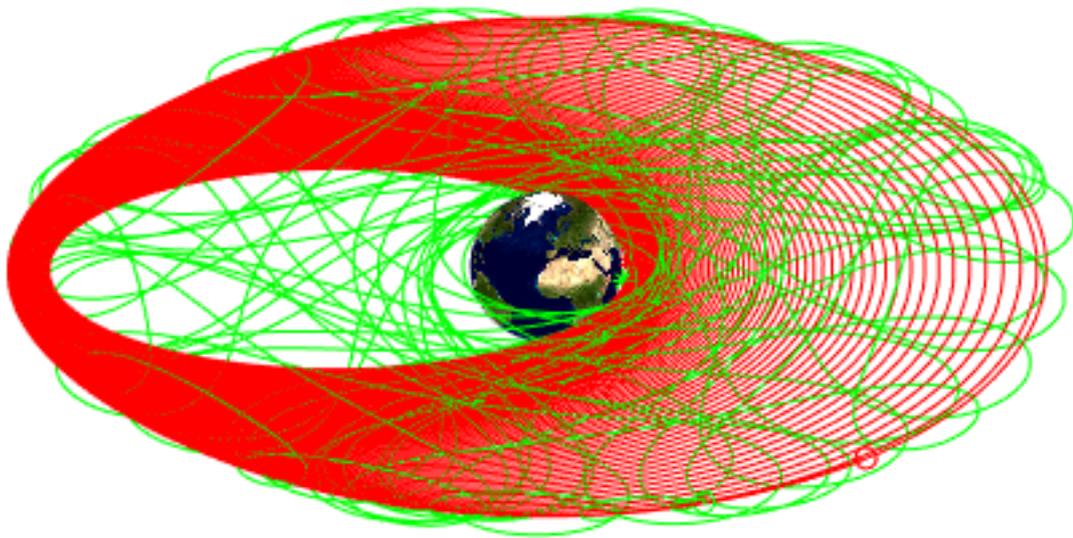


Figure 6.3: EOR maneuver (EO - LEO to GEO).

In Fig. 6.4, 6.5, 6.6 it's presented, respectively, how the semi-major axis, the orbital eccentricity and the orbital inclination change in function of time.

As we can see from the inputs and from the figures, in order to reach our goals, the semi-major axis must increase its values, while the orbital eccentricity and the

orbital inclination must be 0 at the end of the simulation.

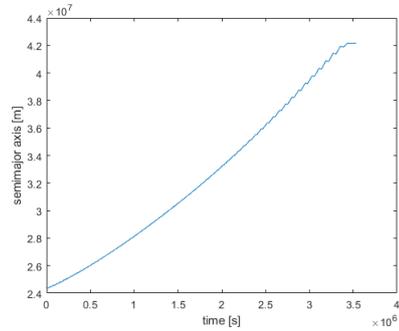


Figure 6.4: Time - semi-major axis (EOR - LEO to GEO).

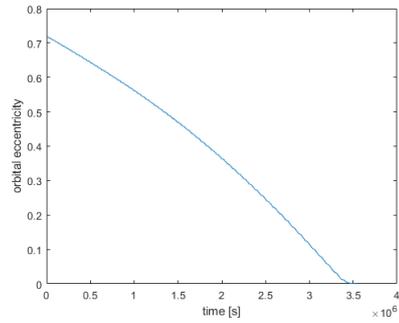


Figure 6.5: Time - Orbital eccentricity (EOR - LEO to GEO).

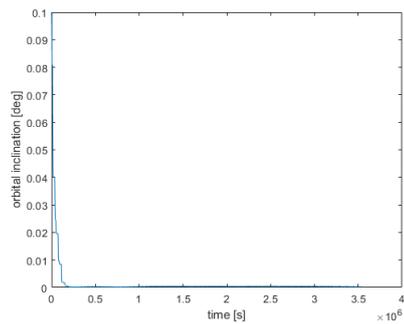


Figure 6.6: Time - Orbital inclination (EOR - LEO to GEO).

As the same way, in Fig. 6.7, 6.8, 6.9 we can see the  $\Delta V$ , the total mass of the satellite and the eclipse time.

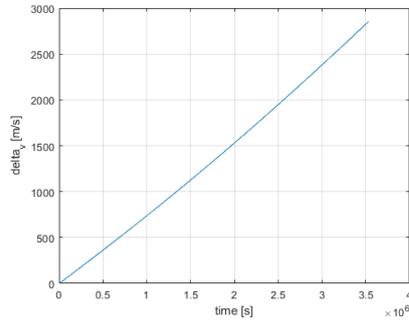


Figure 6.7: Time - Delta V (EOR - LEO to GEO).

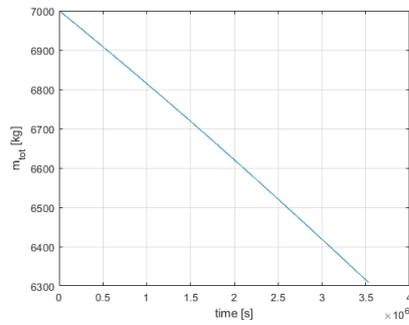


Figure 6.8: Time -  $m_{tot}$  (EOR - LEO to GEO).

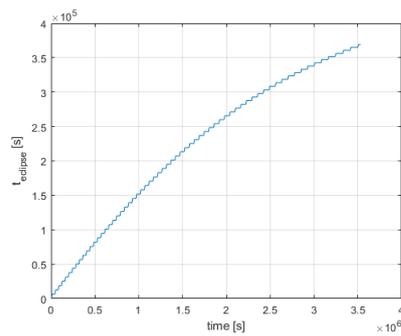


Figure 6.9: Time - Eclipse time (EOR - LEO to GEO).

In Fig. 6.10 it's possible to see the thrust angles in function of time. In green, we have  $\alpha$  (in-plane angle) and in red  $\beta$  (out-of-plane angle).

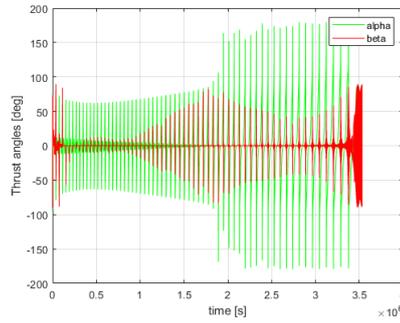


Figure 6.10: Time - Thrust angles (EOR - LEO to GEO).

In Fig. 6.11 and 6.12 there are presented the thrust weights and velocity change. The thrust weights are:

- $k_a$  (influences the semi-major axis) in blue;
- $k_e$  (influences the orbital eccentricity) in green;
- $k_i$  (influences the orbital inclination) in red;
- $k_\omega$  (influences the argument of periapsis) in cyan;
- $k_\Omega$  (influences the right ascension of ascending node) in purple.

The components of the velocity are:

- $u_{radial}$  (radial component) in blue;
- $u_{tangential}$  (tangential component) in green;
- $u_{normal}$  (normal component) in red.

For what concern the thrust weights, we can see that the  $k_i$  (thrust weights responsible for the inclination), is the first to become zero. That's highlighted even from the velocity, in fact, as the same way as  $k_i$ , even the  $u_{normal}$  reach the value "zero" before the other components.

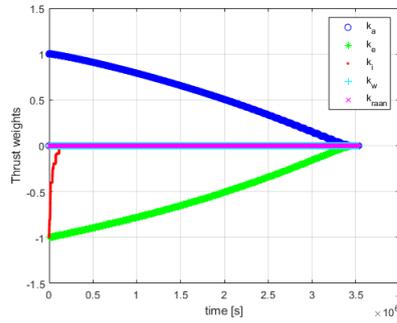


Figure 6.11: Time - Thrust weights (EOR - LEO to GEO).

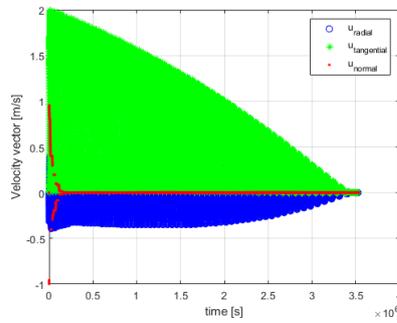


Figure 6.12: Time - Velocity (EOR - LEO to GEO).

In tab. 6.2 there are the values of:

- $t_{f_{maneuver}}$  = final time of the maneuver;
- $m_{prop}$  = propellant mass used;
- $\Delta V$ ;
- $t_{eclipse}$  = total eclipse time.

$t_{f_{maneuver}}$	40.887499999999996	days
$m_{prop}$	691.2190051583002	kg
$\Delta V$	2556.425985342690	m/s
$t_{eclipse}$	4.2743055555555555	days

Table 6.2: Results (EOR - LEO to GEO).

## 6.2 EOR - $\Delta$ altitude

In this section will be presented the results of simple EOR maneuvers when we raise the orbit respectively of 20, 50 and 100 *km*. The inputs are the same for both the simulations, and they are presented in 6.3. The only parameter that changes is the  $\Delta altitude$ .

In this case the maneuvers presented are useful only to change the semi-major axis value in order to make the first step of the phasing maneuver.

<i>date</i>	2019 Feb 19 15 : 30 : 00	<i>UTC</i> ;
<i>a<sub>i</sub></i>	$42165 \cdot 10^3$	[ <i>m</i> ]
<i>i<sub>i</sub></i>	0	[ <i>deg</i> ]
<i>e<sub>i</sub></i>	0	
$\Omega_i$	0	[ <i>deg</i> ]
$\omega_i$	0	[ <i>deg</i> ]
$\nu_i$	0	[ <i>deg</i> ]
<i>a<sub>f</sub></i>	$a_i + \Delta altitude$	[ <i>m</i> ]
<i>i<sub>f</sub></i>	0	[ <i>deg</i> ]
<i>e<sub>f</sub></i>	0	
$\Omega_f$	0	[ <i>deg</i> ]
$\omega_f$	0	[ <i>deg</i> ]
$\nu_f$	0	[ <i>deg</i> ]
<i>thruster</i>	1	
<i>thrust per thruster</i>	$1000 \cdot 10^{-3}$	[ <i>N</i> ]
<i>I<sub>SP</sub></i>	2800	[ <i>sec</i> ]

Table 6.3: Input (EOR).

In Fig. 6.13 we can see the trajectory of our satellite, where in red there are the ECI coordinates and in green the ECEF coordinates. As we can see, the ECEF position is basically constant. That's because we are considering a geostationary orbit and the satellite presents almost the same velocity as the rotation speed of the Earth.

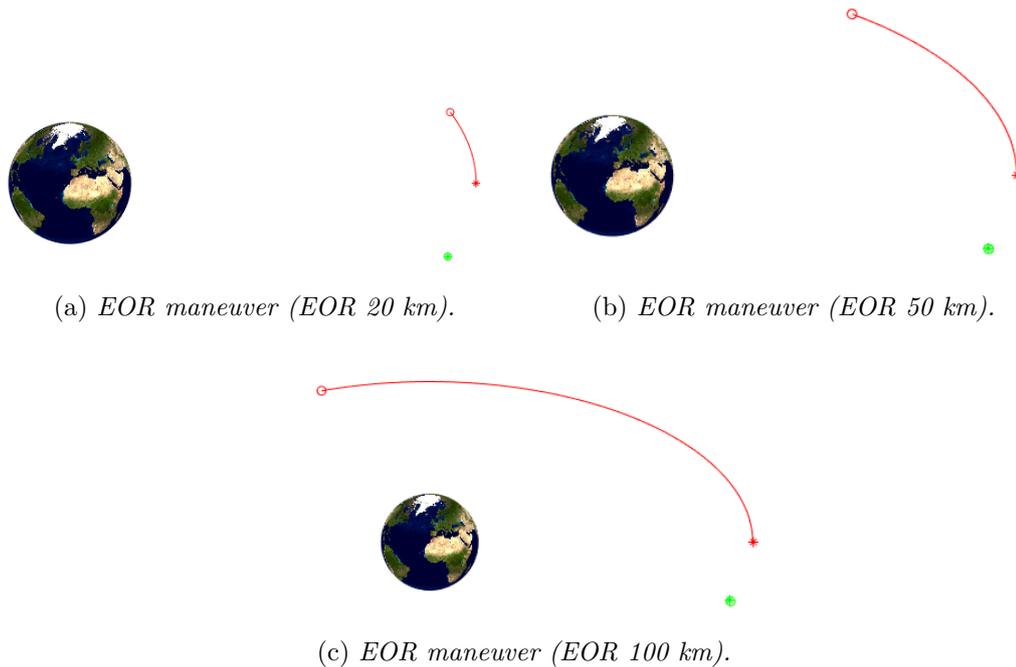
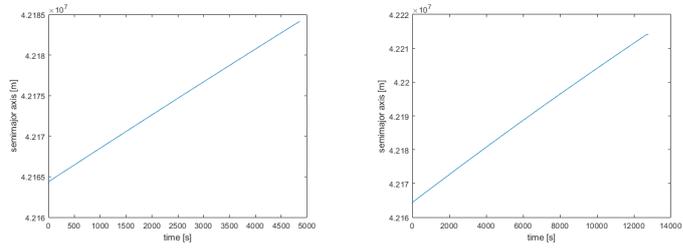


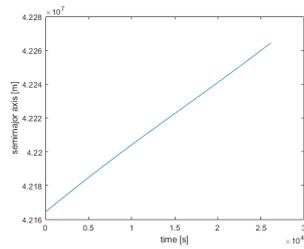
Figure 6.13: EOR maneuver

The differences for what concern the  $\Delta altitude$  are highlighted in Fig. 6.14. In fact, we can see that the first simulation ends when the semi-major axis increase its value of 20 km, while the other simulations continue. The trends of these curves are linear. That's because the satellite is never eclipsed during these maneuvers (Fig. 6.17). If it were, the parameters considered would have been constant during eclipse time.

Because of the finite maneuver and the constant thrust applied to the spacecraft, we can see, from tab. 6.4 that the maneuver time increase linearly with the  $\Delta altitude$  we want the satellite to reach and so the  $\Delta V$  and the  $m_{prop}$  used. In fact, if we consider Fig. 6.15 the cost of the maneuvers increases and focusing on the last two simulations, the  $\Delta V$  almost doubles its values as the same as the  $m_{prop}$  (see Fig. 6.16) and the  $t_{f_{maneuver}}$ .

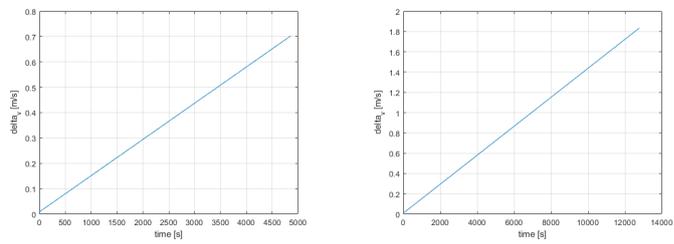


(a) *Time - semi-major axis (EOR 20 km).* (b) *Time - semi-major axis (EOR 50 km).*

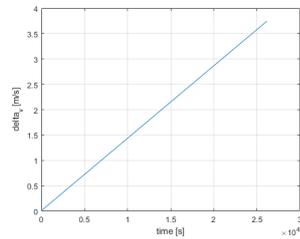


(c) *Time - semi-major axis (EOR 100 km).*

Figure 6.14: Time - semi-major axis.



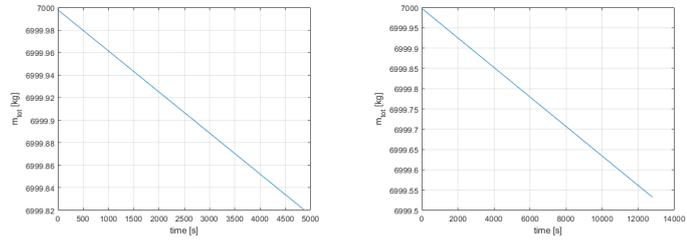
(a) *Time - Delta V (EOR 20 km).* (b) *Time - Delta V (EOR 50 km).*



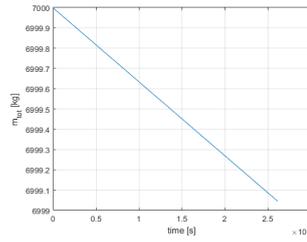
(c) *Time - Delta V (EOR 100 km).*

Figure 6.15: Time - Delta V.

6.2 – EOR -  $\Delta$  altitude

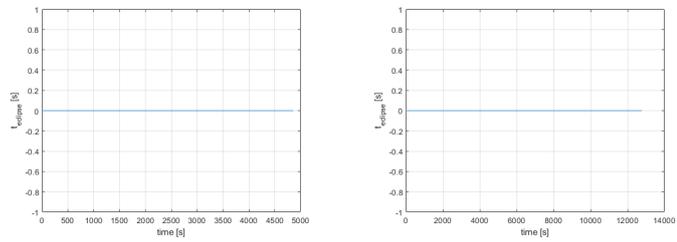


(a) *Time -  $m_{tot}$  (EOR 20 km).* (b) *Time -  $m_{tot}$  (EOR 50 km).*

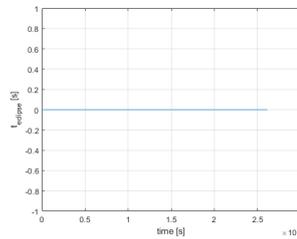


(c) *Time -  $m_{tot}$  (EOR 100 km).*

Figure 6.16: Time -  $m_{tot}$ .

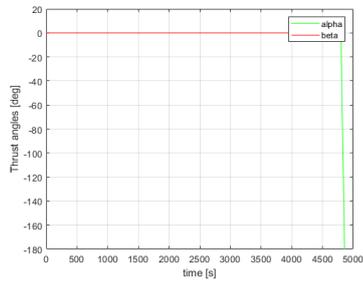


(a) *Time - Eclipse time (EOR 20 km).* (b) *Time - Eclipse time (EOR 50 km).*

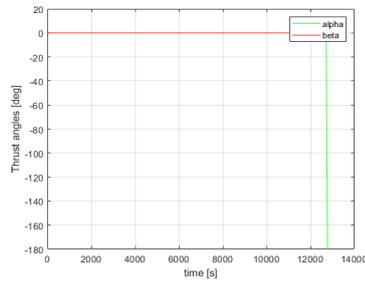


(c) *Time - Eclipse time (EOR 100 km).*

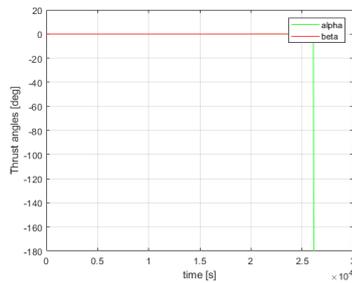
Figure 6.17: Time - Eclipse time.



(a) Time - Thrust angles (EOR 20 km).



(b) Time - Thrust angles (EOR 50 km).



(c) Time - Thrust angles (EOR 100 km).

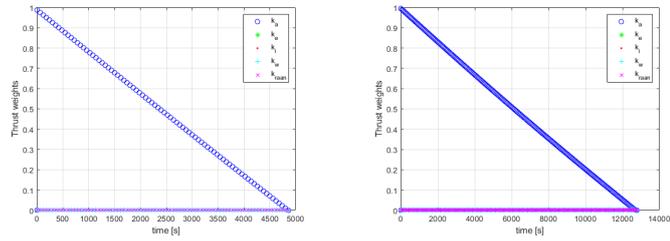
Figure 6.18: Time - Thrust angles.

Focusing on Fig. 6.18, in order to increase the semi-major axis, the in-plane angle ( $\alpha$ ) and out-of-plane angle ( $\beta$ ) are both equal to zero, according to the results of the Edelbaum problem.  $\alpha = 0$  means that the satellite is increasing just its tangential velocity. In fact, if  $\alpha = \pm\pi$  the semi-major axis would decrease its value. For what concern  $\beta$ , it always has a zero value because we never go out of the orbital plane.

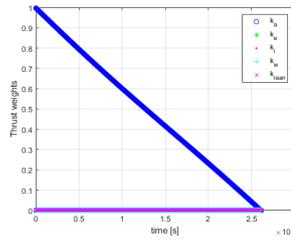
For what concern the Thrust weights (Fig. 6.19) and the Velocity (Fig. 6.20) the trends of the curves are basically the same.

As we can see, in order to increase  $a$ , its weights must present a positive value and so  $u_{tangential}$ . The more we are far from the orbit we want the satellite to reach, the more the thrust weights and the velocity have a higher value, while the more the satellite get closer to its objective, the more they decrease. The changes between these simulations are about how long we are going to apply the thrust.

6.2 – EOR -  $\Delta$  altitude

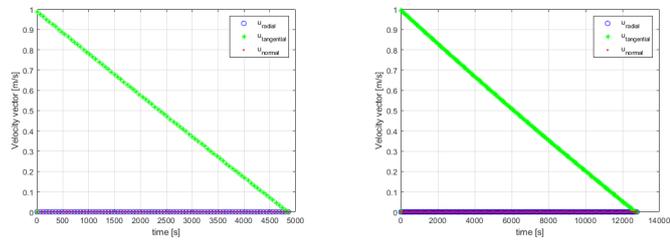


(a) *Time - Thrust weights (EOR 20 km).* (b) *Time - Thrust weights (EOR 50 km).*

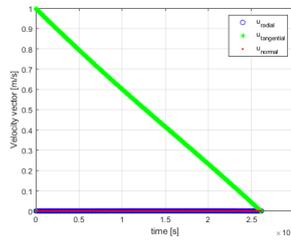


(c) *Time - Thrust weights (EOR 100 km).*

Figure 6.19: Time - Thrust weights.



(a) *Time - Velocity (EOR 20 km).* (b) *Time - Velocity (EOR 50 km).*



(c) *Time - Velocity (EOR 100 km).*

Figure 6.20: Time - Velocity.

In tab. 6.4 are reported the numerical values we obtain at the end of the simulations.

RESULTS	20 <i>km</i>	50 <i>km</i>	100 <i>km</i>	
$t_{f_{maneuver}}$	1.3666666666666667	3.5666666666666667	7.283333333333333	hours
$m_{prop}$	0.176993599855450	0.465427614434702	0.952706290579954	kg
$\Delta V$	0.702866248217103	1.834347270136513	3.745970375825520	m/s
$t_{eclipse}$	0.00	0.00	0.00	sec

Table 6.4: Results (EOR).

## Chapter 7

# Phasing maneuver with waiting phase

As the same way as the EOR maneuver, the phasing maneuver is important for what concern the Space Tug OOS (1.1) and the All-Electric platform (1.2), even though it refers to different mission phase in respect of the maneuver analyzed previously.

The phasing maneuver is that maneuver necessary to adjust the satellite's position within the same orbit. It's defined as an evasive maneuver in order to distance or nearer the spacecraft to the target.

Once the user defines the position he wants the satellite to reach, we must define the mission. In this way we get the phase angle to make, and so we must analyze the budget in terms of the  $\Delta V$ , propellant mass and maneuver time.

For what concern the chemical propulsion, the phasing maneuver can be divided into three phases, as described in 2.6. In this chapter we are going to follow that logic, even though we are considering electric propulsion.

So, the first phase consists of an EOR maneuver, the second is just a waiting phase and the third phase is a second EOR in order to get back to the initial orbit.

### 7.1 Maneuver logic

Before entering the function "*PHASING\_MANEUVER.m*", the program performs "*DELTA\_POSITION\_EOR2.m*". This function is composed of two consecutive EOR, performed in order to calculate the delta position performed by the satellite during the second EOR. The result of this function is the  $\Delta longitude(EOR2)$ . The EOR maneuvers are exactly the same we reported in chapter 5.

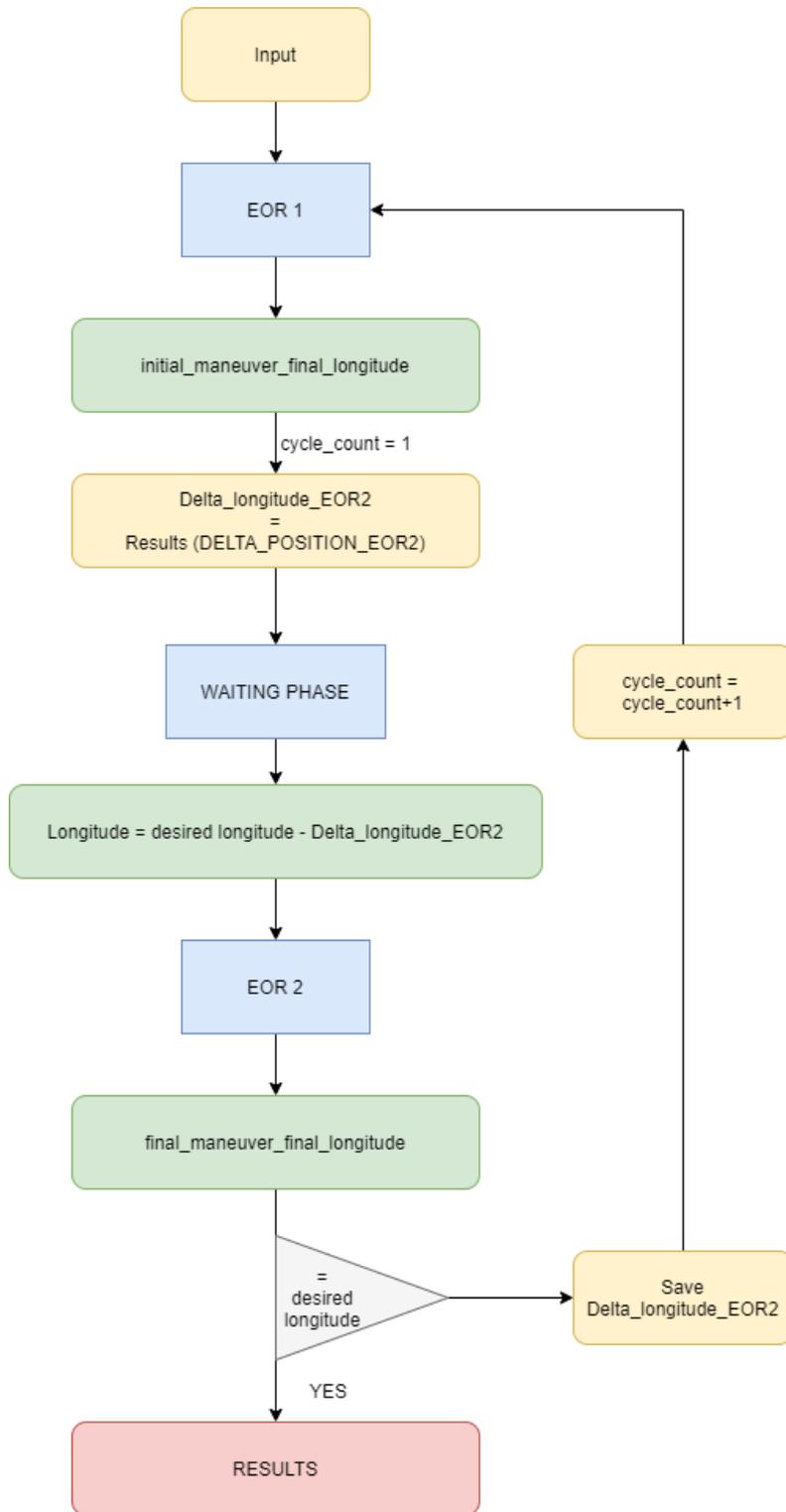


Figure 7.1: Phasing maneuver Logic.

Then we enter in "*PHASING\_MANEUVER.m*" (Fig. 7.1). Unlike the previous case (EOR), we have to initialize the position we want the satellite to reach (*True longitude* = input). In this chapter, the position we are referring is the longitude. Once the first maneuver (EOR) is done, the program saves the  $\Delta longitude(EOR1) = initial\ longitude - initial\ maneuver\ final\ longitude$  made by the spacecraft.

For what concern the waiting phase, the assumption is that the satellite will behave as if it's always eclipsed. In this way no thrust is produced. In order to exit this loop, we must wait that the longitude reached by the satellite is:

$$Longitude = desired\ Longitude - \Delta longitude(EOR2)$$

In the end we enter the final loop (second EOR) and once the satellite reaches the desired semi-major axis value, we control if it has reached the desired position. If not, we go back to the start of the simulation saving the  $\Delta longitude(EOR2)$  calculated at the end of the previous simulation. Otherwise, we exit the function saving all the data and plotting them.

## 7.2 Simulation - Geostationary orbit

In tab. 7.1 there are the input used for the following simulations.

<i>date</i>	2019 Feb 19 15 : 30 : 00	UTC;
<i>a<sub>i</sub></i>	$42165 \cdot 10^3$	[m]
<i>i<sub>i</sub></i>	0	[deg]
<i>e<sub>i</sub></i>	0	
<i>Ω<sub>i</sub></i>	0	[deg]
<i>ω<sub>i</sub></i>	0	[deg]
<i>ν<sub>i</sub></i>	21.5486	[deg]
<i>phasing<sub>altitude</sub></i>	$-100 \cdot 10^3$	[m];
<i>a<sub>f</sub></i>	$a_i + phasing_{altitude}$	[m]
<i>i<sub>f</sub></i>	0	[deg]
<i>e<sub>f</sub></i>	0	
<i>Ω<sub>f</sub></i>	0	[deg]
<i>ω<sub>f</sub></i>	0	[deg]
<i>ν<sub>f</sub></i>	21.5486	[deg]
<i>Initial Longitude</i>	$3.253111063935371e - 05$	[deg].
<i>thruster</i>	1	
<i>thrust per thruster</i>	$1000 \cdot 10^{-3}$	[N]
<i>I<sub>SP</sub></i>	2800	[sec]

Table 7.1: Input Phasing maneuver.

For what concern the waiting phase, every type of perturbation is considered: zonal harmonics, aerodynamic drag, solar radiation pressure, third body perturbation and eclipse.

As the same way as the previous chapter, in this one we will see the results MAGNETO gives us. We are going to use the same input but changing the position we want the satellite to reach, in terms of longitude.

### 7.2.1 Results - Geostationary orbit

In Fig. 7.2 we can see the phasing maneuver, respectively, for  $Longitude = 15^\circ$ ,  $30^\circ$ ,  $90^\circ$  and  $180^\circ$ . As the same way as the previous simulations, the red trajectory is the one in ECI coordinates, while in green we have the ECEF trajectory.

It has been decided to use  $phasing_{altitude} = -100 \cdot 10^3 m$  in order to increase the longitude. In fact, in this way, the satellite increases its velocity in respect of the Earth rotational speed. If we wanted the satellite to decrease its longitude, we would have increase its  $phasing_{altitude}$  in a way that the satellite would have been slower than the Earth.

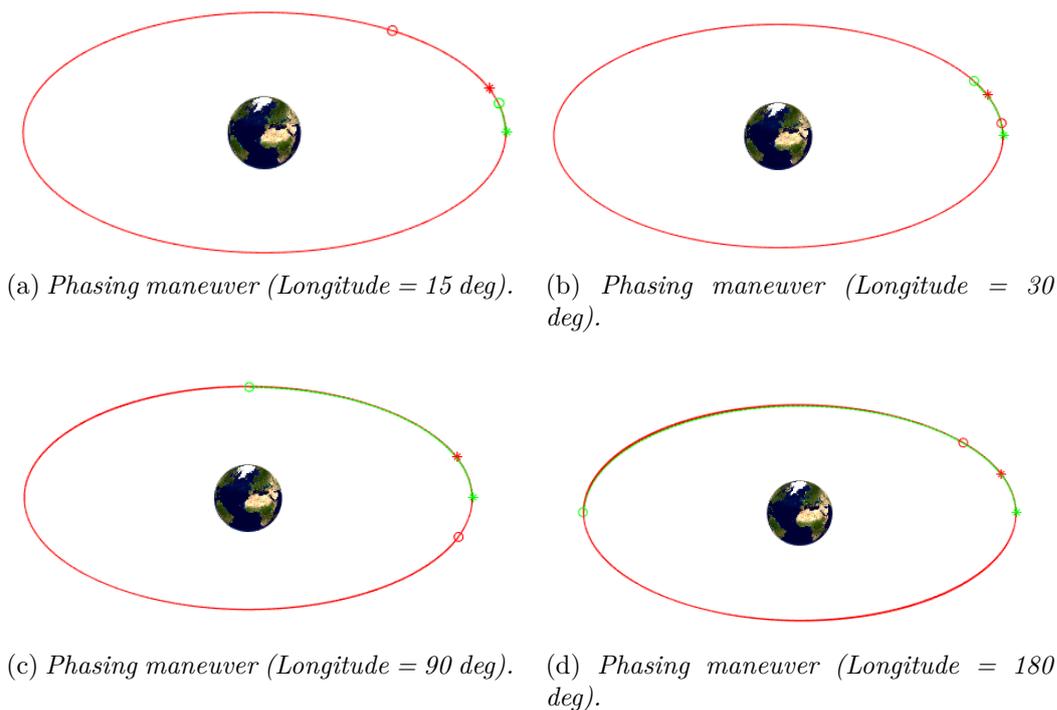
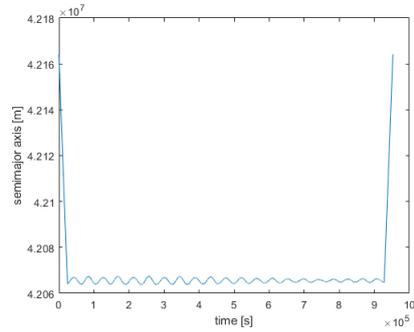
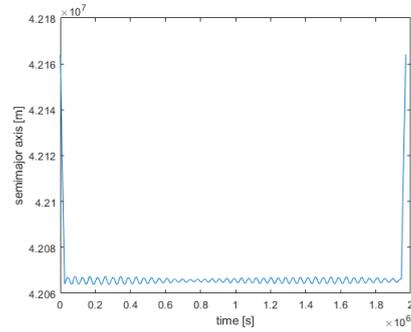


Figure 7.2: Phasing maneuver

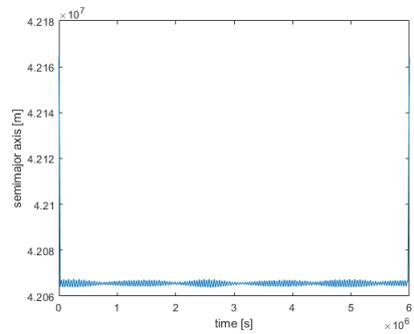
For what concern the semi-major axis, see Fig. 7.3, during the waiting phase is about constant, even if, because of the perturbation we are considering, it oscillates around its desired value.



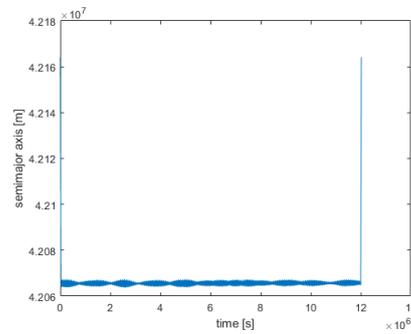
(a) Time - semi-major axis (Longitude = 15 deg).



(b) Time - semi-major axis (Longitude = 30 deg).



(c) Time - semi-major axis (Longitude = 90 deg).



(d) Time - semi-major axis (Longitude = 180 deg).

Figure 7.3: Time - semi-major axis.

In Fig. 7.5 there is the longitude in function of time, while in Fig. 7.4 there is the true longitude in function of time. The longitude is the parameter we are controlling and, in fact, the simulations end when it reaches the desired value.

Phasing maneuver with waiting phase

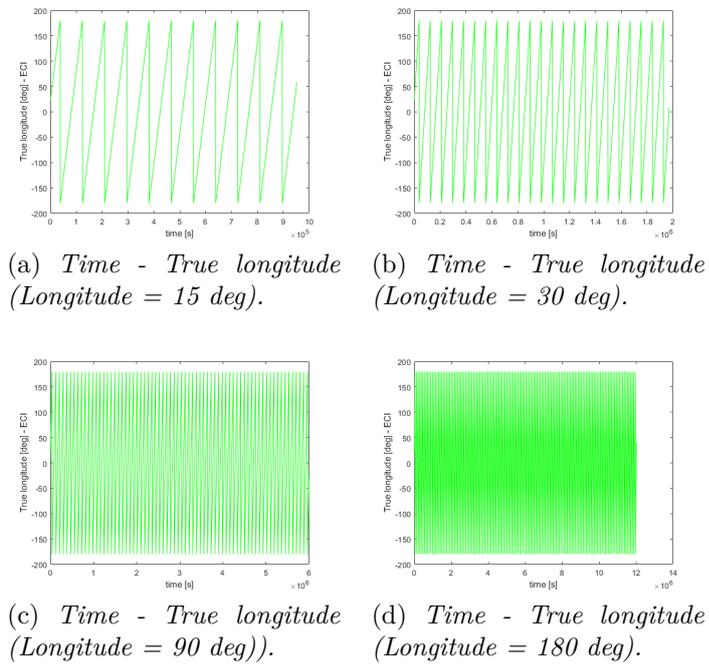


Figure 7.4: Time - True longitude.

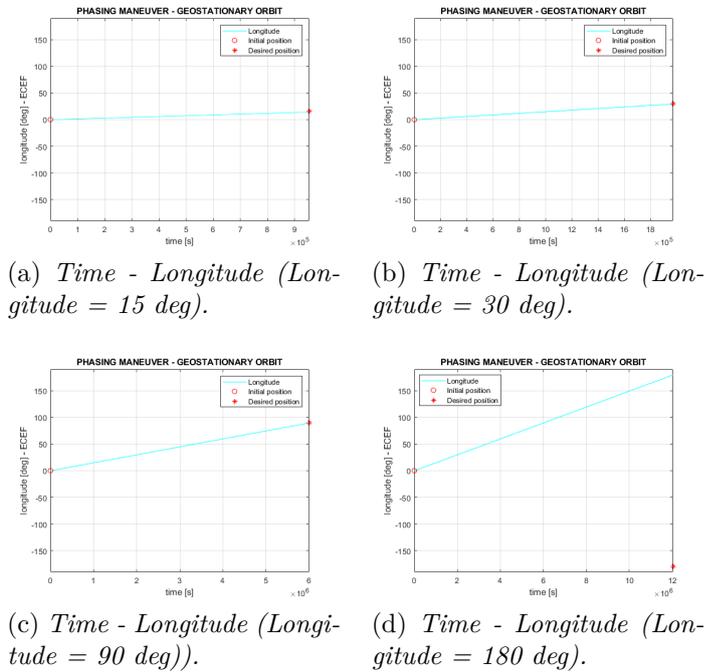
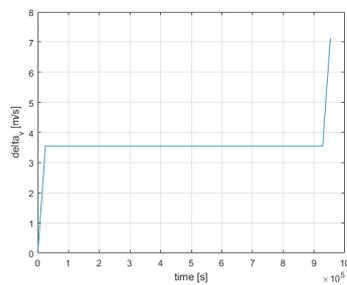


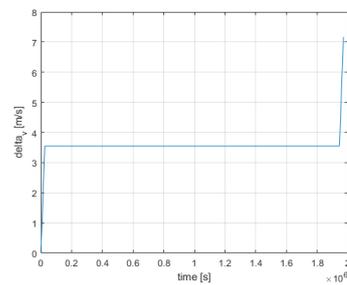
Figure 7.5: Time - Longitude.

As we can see in Fig. 7.6 and 7.7, when the satellite is in the waiting phase, the  $\Delta V$  and the  $m_{tot}$  are constant. Moreover, if we look at the results table 7.2, both the  $\Delta V$  and the mass of the propellant are practically equal. This is because during the waiting phase, the satellite behaves as if it were eclipsed and therefore no thrust is produced. In this way, the phases that have costs are the two EOR and both these maneuvers are performed to reach the same semi-major axis value.

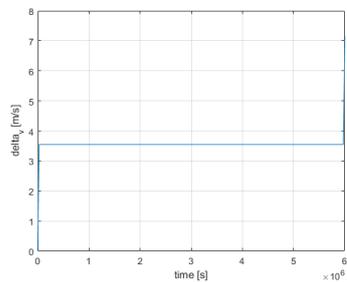
In the end, using this maneuver logic, the only parameter that changes is the time the satellite spends in waiting phase. Consequently, the eclipse time (7.8) increase its value.



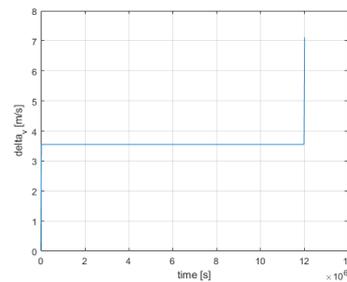
(a) *Time - Delta V (Longitude = 15 deg).*



(b) *Time - Delta V (Longitude = 30 deg).*



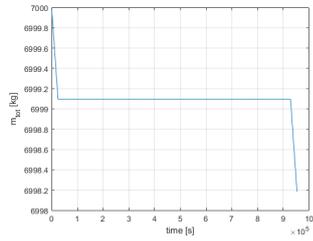
(c) *Time - Delta V (Longitude = 90 deg).*



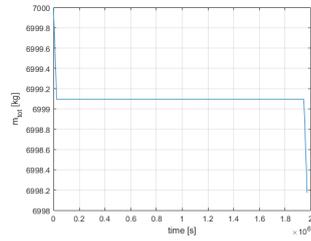
(d) *Time - Delta V (Longitude = 180 deg).*

Figure 7.6: Time - Delta V.

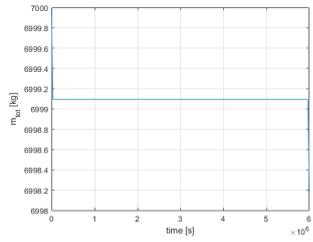
Phasing maneuver with waiting phase



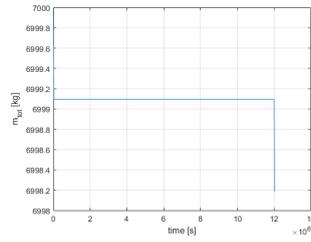
(a) Time -  $m_{tot}$  (Longitude = 15 deg).



(b) Time -  $m_{tot}$  (Longitude = 30 deg).

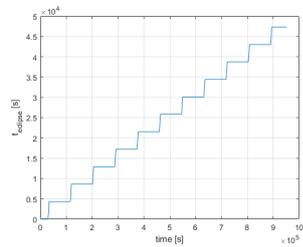


(c) Time -  $m_{tot}$  (Longitude = 90 deg).

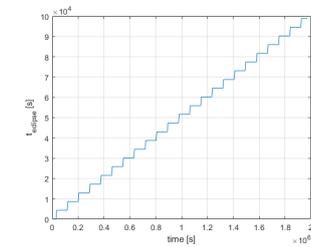


(d) Time -  $m_{tot}$  (Longitude = 180 deg).

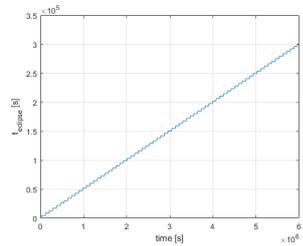
Figure 7.7: Time -  $m_{tot}$ .



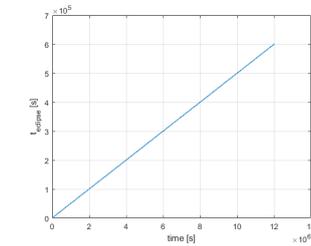
(a) Time - Eclipse time (Longitude = 15 deg).



(b) Time - Eclipse time (Longitude = 30 deg).



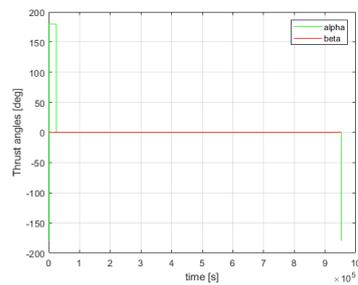
(c) Time - Eclipse time (Longitude = 90 deg).



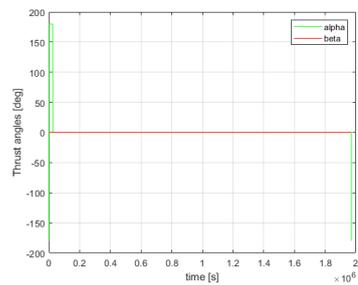
(d) Time - Eclipse time (Longitude = 180 deg).

Figure 7.8: Time - Eclipse time.

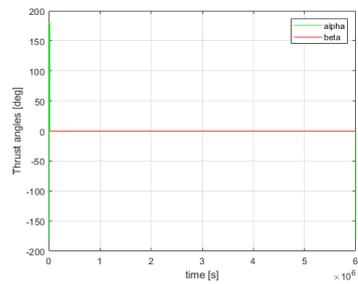
Fig. 7.9, 7.10 and 7.11 highlight the fact that the only parameter that change is the waiting time. In fact, if we zoom the figure when the satellite is performing the EOR, we can see that both the thrust angles, weights and velocity have identical trends. Even if it's not clear from these figures, the trends of  $k_a$  and  $u_{tangential}$  for what concern the ascending phase, are the same as the simulation EOR when we raise the orbit radius of  $\Delta altitude = 100 km$ . Instead, the descending phase (the first EOR of the phasing maneuvers considered) is practically opposite.



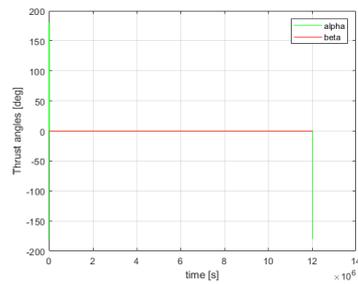
(a) *Time - Thrust angles (Longitude = 15 deg).*



(b) *Time - Thrust angles (Longitude = 30 deg).*



(c) *Time - Thrust angles (Longitude = 90 deg).*



(d) *Time - Thrust angles (Longitude = 180 deg).*

Figure 7.9: Time - Thrust angles.

Phasing maneuver with waiting phase

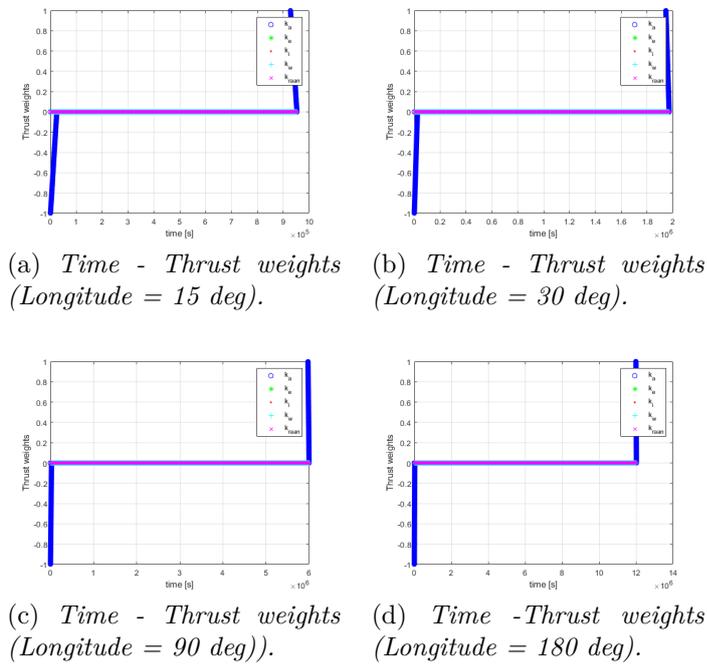


Figure 7.10: Time - Thrust weights.

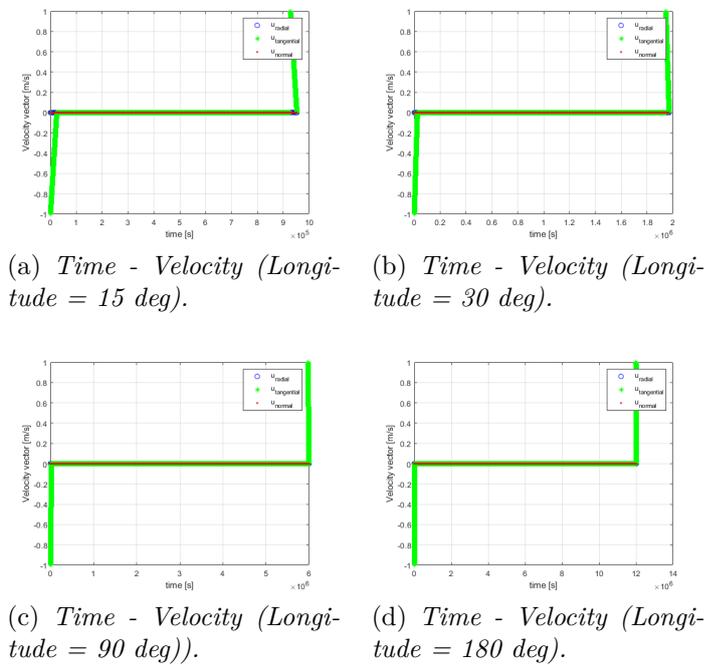


Figure 7.11: Time - Velocity.

RESULTS	15°	30°	90°	180°	
$true\ longitude_{phasing}$	14.11871	29.48327	89.69305	179.56187	deg
$t_{cycle}$	11.03194	22.81458	69.44514	139.17431	days
$\Delta v_{phasing\ cycle}$	7.12378	7.16665	7.14950	7.11521	m/s
$m_{prop\ cycle}$	1.81364	1.82456	1.82019	1.81145	kg
$t_{waiting\ cycle}$	10.45486	22.23403	68.86597	138.54861	days
$t_{eclipse}$	0.54792	1.14514	3.48125	6.96181	days

Table 7.2: Results.

As already said, the cost of the maneuvers presented is about the same for every simulation. That's normal because during the waiting phase no thrust is applied, and the satellite continues its motion with no thrust perturbation. So only the EOR phase presents costs in terms of  $\Delta V$  and  $m_{prop}$  and those phases are identical for all the simulations we made.

## 7.3 Generalization

Once MAGNETO was modified in such a way as to be able to perform the phasing maneuver for geostationary orbits, the goal was to generalize the maneuver for any type of orbit.

Since, if the satellite is in a different orbit than the geostationary one, the longitude changes continuously, the controls to be made during the maneuver have been changed.

### 7.3.1 Controls

As written before, in the event that the orbiting body is not on a geostationary orbit, longitude cannot be used as a control meter. In fact, we must have a control on parameters which have dependence on time.

For this reason, control methods have been studied which act on the various orbital parameters. A subdivision was made between the various types of orbits:

- Geostationary orbit;
- Circular and equatorial orbit ( $e = 0$  and  $incl = 0$ );
- Circular orbit ( $e = 0$ , but  $incl \neq 0$ );
- Equatorial orbit ( $incl = 0$ , but  $e \neq 0$ );
- Generic orbit ( $incl \neq 0$  and  $e \neq 0$ ).

For what concern the geostationary orbit, the control method is the same seen in the previous chapter.

### **Circular and equatorial orbit**

For a circular and equatorial orbit, the true longitude must be checked. In celestial mechanics, the true longitude is the position at which a satellite is when its inclination is  $i = 0$ . Using the inclination and the ascending node, the true longitude indicates the precise position from the central body at which the spacecraft would be located at a particular time. The true longitude is defined as:

$$l = \Omega + \omega + \nu$$

### **Circular and inclined orbit**

For a circular and inclined orbit, it's not possible to rely on true longitude. The choice of the parameter to be controlled falls on the argument of latitude ( $u$ ). The argument of latitude is an angular parameter useful to define the position of an orbiting object which is moving along a Keplerian orbit. In this case we are not going to consider the *RAAN*.

$$u = \nu + \omega$$

It defines the angle between the ascending node and the body.

### **Equatorial and eccentric orbit**

In this case the *RAAN* loses meaning. In fact, there is no a nodal line. Otherwise, it's still possible to rely on True longitude in order to obtain the position of the satellite. So, the controls in this case are the same used for *Circular and equatorial orbit*.

### **Generic orbit**

For what concern a generic orbit, there are no parameters that have no physical meaning. To identify the position of the satellite, we use the argument of latitude. In fact, because of the satellite's orbit which is inclined, the *RAAN* does not change. So, we can use the same control as Circular and inclined orbit.

## **7.4 Simulations - Generalization**

In this chapter will be presented the results of some simulations. All simulations have been made in order to have different kind of orbit. In this way has been

possible to evidence that the program will work using any type of control. In tab. 7.3 there are the common inputs for the orbit we are going to consider.

<i>date</i>	2019 Feb 19 15 : 30 : 00	UTC;
$a_i$	$32165 \cdot 10^3$	[m]
$\Omega_i$	0	[deg]
$\omega_i$	0	[deg]
$\nu_i$	0	[deg]
$a_f$	$a_i + phasing_{altitude}$	[m]
$\Omega_f$	0	[deg]
$\omega_f$	0	[deg]
$\nu_f$	0	[deg]
<i>thruster</i>	1	
<i>thrust per thruster</i>	$1000 \cdot 10^{-3}$	[N]
$I_{SP}$	2800	[sec]
<i>phasing<sub>altitude</sub></i>	$100 \cdot 10^3$	[m]
<i>true longitude<sub>phasing</sub></i>	180	[deg]

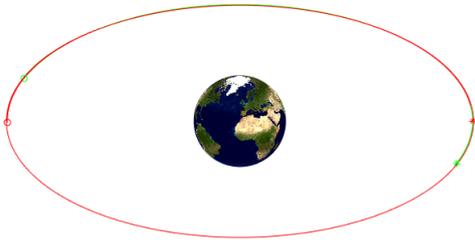
Table 7.3: Input Phasing maneuver (Generalization).

The simulations presented refers to the following type of orbit:

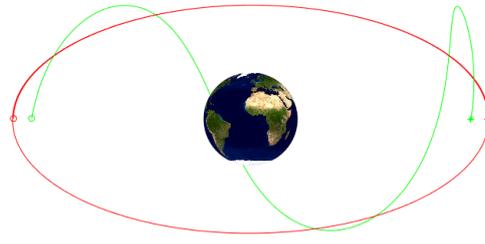
- Circular and equatorial orbit:  $i_i = i_f = 0$  and  $e_i = e_f = 0$ ;
- Circular and inclined orbit:  $i_i = i_f = 28.5$  and  $e_i = e_f = 0$ ;
- Equatorial and eccentric orbit:  $i_i = i_f = 0$  and  $e_i = e_f = 0.4$ ;
- Generic orbit (inclined and eccentric):  $i_i = i_f = 28.5$  and  $e_i = e_f = 0.4$ ;

### 7.4.1 Results - Generalization

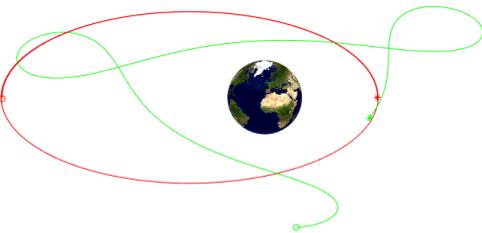
In Fig. 7.12 we can see the phasing maneuver, respectively, for circular and equatorial orbit, circular and inclined orbit, equatorial and eccentric orbit ad generic orbit. As the same way as the previous simulations, the red trajectory is the one in ECI coordinates, while in green we have the ECEF trajectory.



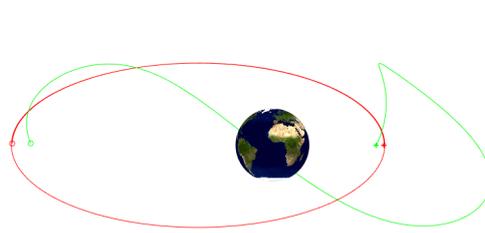
(a) *Phasing maneuver (Circular and equatorial orbit).*



(b) *Phasing maneuver (Circular and inclined orbit).*



(c) *Phasing maneuver (Equatorial and eccentric orbit).*



(d) *Phasing maneuver (Generic orbit).*

Figure 7.12: Phasing maneuver (Generalization).

As we can see in Fig. 7.13, in case of circular orbit, the semi-major axis in function of time behaves like the geostationary case. Obviously, if we are considering eccentric orbit, the semi-major axis is affected by the eccentricity. That's because of the non-linear trends.

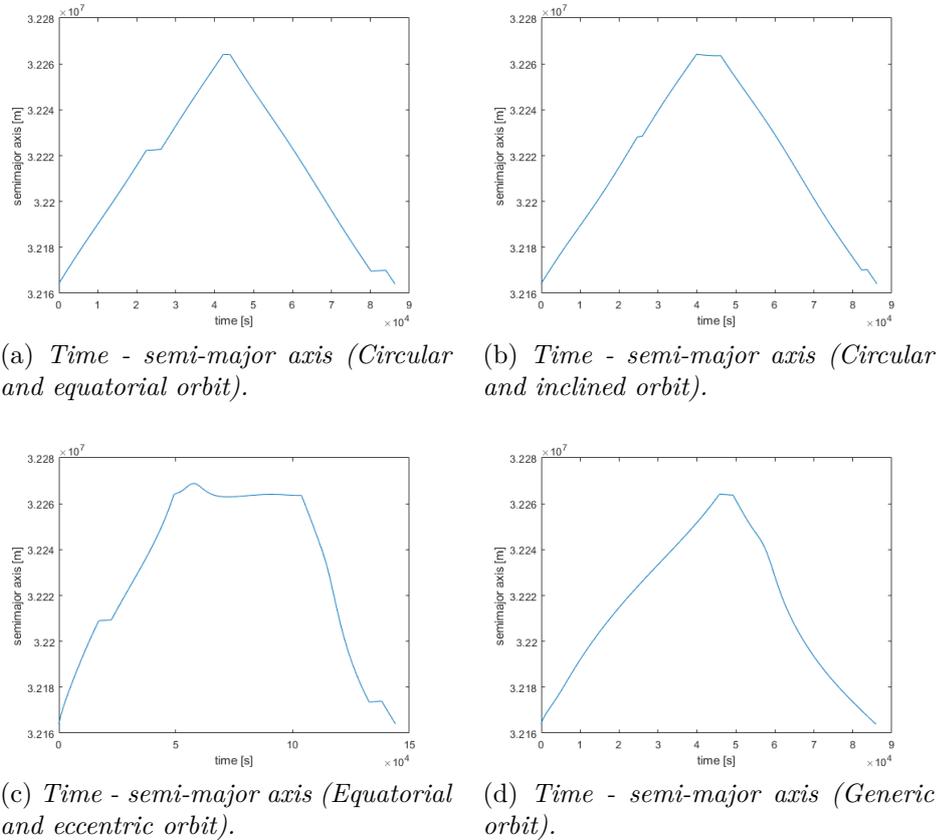
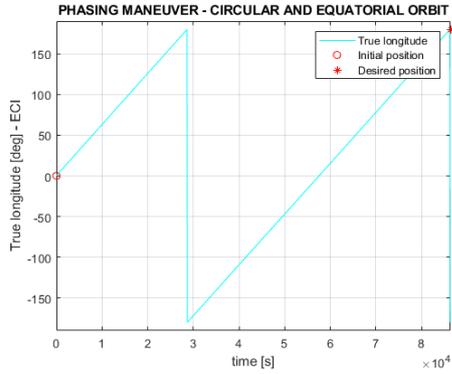
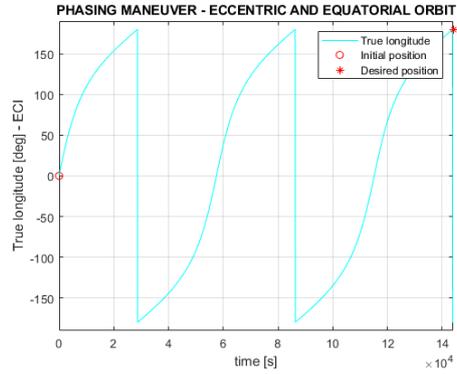


Figure 7.13: Time - semi-major axis (Generalization).

Fig. 7.14 and 7.15, puts in evidence the parameter we are controlling. In case of non-eccentric orbit, the positions change linearly with time. That's because in a circular orbit, the satellite has the same velocity in every point of the orbit. That's different for what concern an eccentric orbit, where the velocity of the satellite depends on its position along the orbit. In fact, the satellite has a higher speed if it's near to the perigee, and so, the trend of the position we are considering will have a greater inclination. That is highlighted even if we consider the longitude in function of time (see Fig. 7.16).

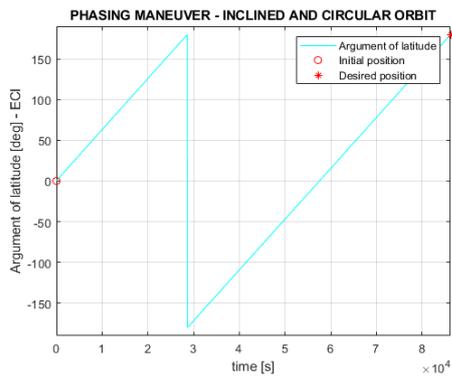


(a) Time - True longitude (Circular and equatorial orbit).

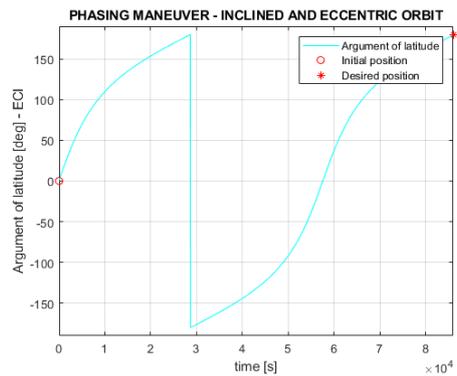


(b) Time - True longitude (Equatorial and eccentric orbit).

Figure 7.14: Time - True Longitude (Generalization).

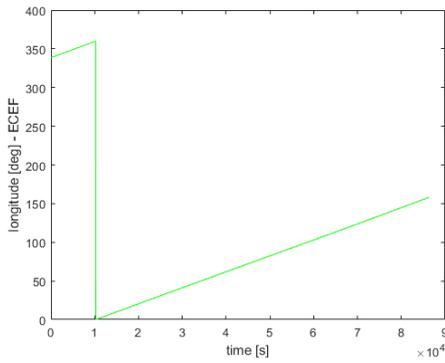


(a) Time - Argument of Latitude (Circular and inclined orbit).

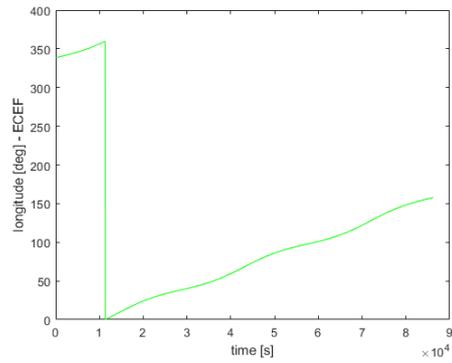


(b) Time - Argument of Latitude (Generic orbit).

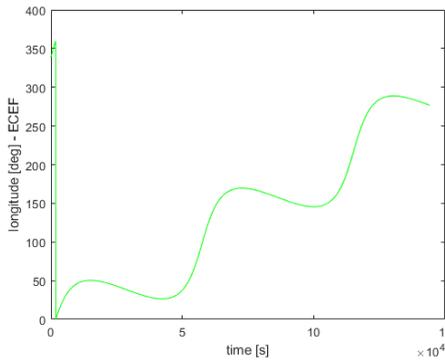
Figure 7.15: Time - Argument of Latitude (Generalization).



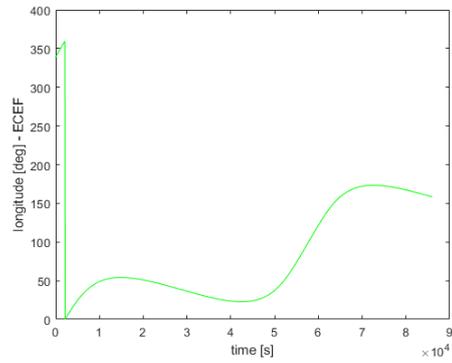
(a) *Time - Longitude (Circular and equatorial orbit).*



(b) *Time - Longitude (Circular and inclined orbit).*



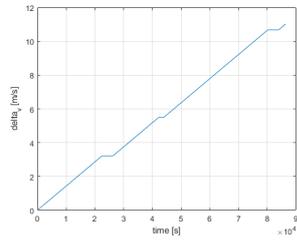
(c) *Time - Longitude (Equatorial and eccentric orbit).*



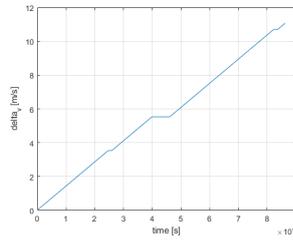
(d) *Time - Longitude (Generic orbit).*

Figure 7.16: Time - Longitude (Generalization).

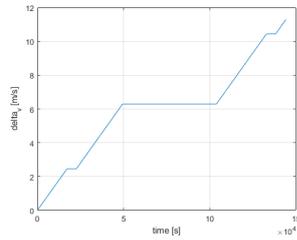
In the previous chapter it was highlighted that during the waiting phase and during the eclipse phase the  $\Delta v$  and the propellant mass does not vary (respectively Fig. 7.17 and 7.18). So, it is also in this case. However, as can be seen in Fig. 7.19, in the case of a generic orbit, the satellite never goes into eclipse, thanks to the inclination and eccentricity the orbit presents.



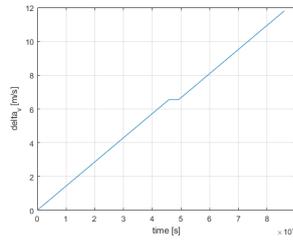
(a) Time - Delta V (Circular and equatorial orbit).



(b) Time - Delta V (Circular and inclined orbit).

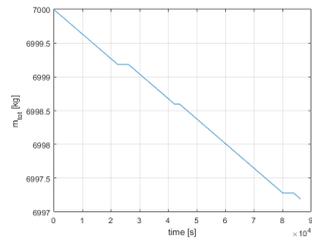


(c) Time - Delta V (Equatorial and eccentric orbit).

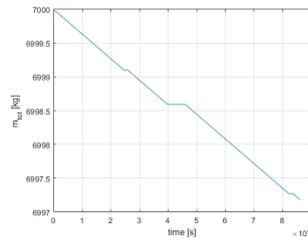


(d) Time - Delta V (Generic orbit).

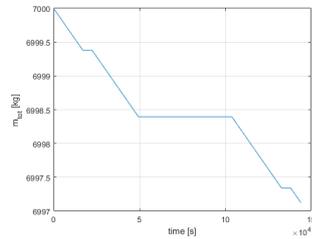
Figure 7.17: Time - Delta V (Generalization).



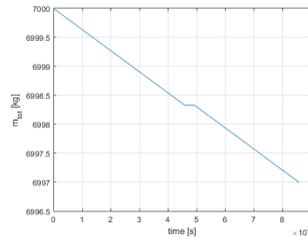
(a) Time -  $m_{tot}$  (Circular and equatorial orbit).



(b) Time -  $m_{tot}$  (Circular and inclined orbit).

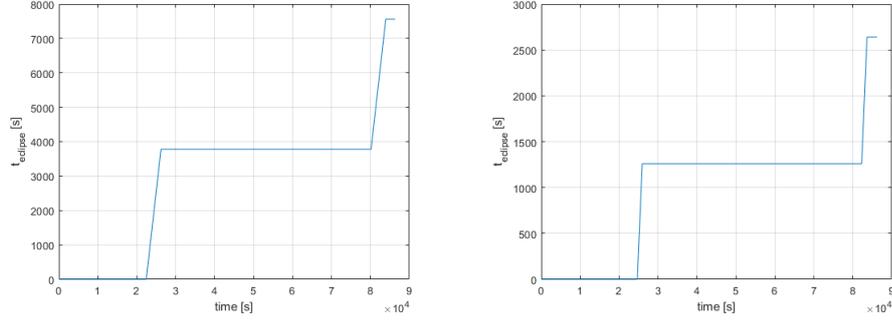


(c) Time -  $m_{tot}$  (Equatorial and eccentric orbit).

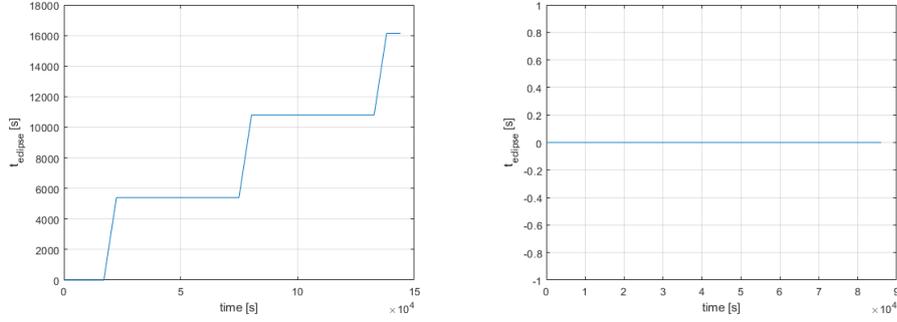


(d) Time -  $m_{tot}$  (Generic orbit).

Figure 7.18: Time -  $m_{tot}$  (Generalization).



(a) Time - Eclipse time (Circular and equatorial orbit). (b) Time - Eclipse time (Circular and inclined orbit).



(c) Time - Eclipse time (Equatorial and eccentric orbit). (d) Time - Eclipse time (Generic orbit).

Figure 7.19: Time - Eclipse time (Generalization).

Although the orbits that we are considering are different, if we consider only the circular orbits or only the eccentric ones, we can see how the trends of the thrust angles (7.20) and velocity (Fig. 7.22) are similar, while the thrust weights (Fig. 7.21) are similar to every case. For what concern the circular orbit, the trends are similar also to the geostationary case. Furthermore, even if we are considering inclined orbit, due to the fact that we are not changing the inclination,  $\beta$  and so the normal component of the velocity and  $k_i$  are 0. For what concern the eccentric orbit,  $\alpha \neq 0$  in order to increase semi-major axis and  $\alpha \neq \pm\pi$  to decrease it. That's because of the presence of  $e$ . It also affects the radial component of the velocity which is higher before the satellite gets to the apogee and become negative once it has been passed. When the satellite approaches to the apogee or the perigee,  $u_{radial} \rightarrow 0$ .

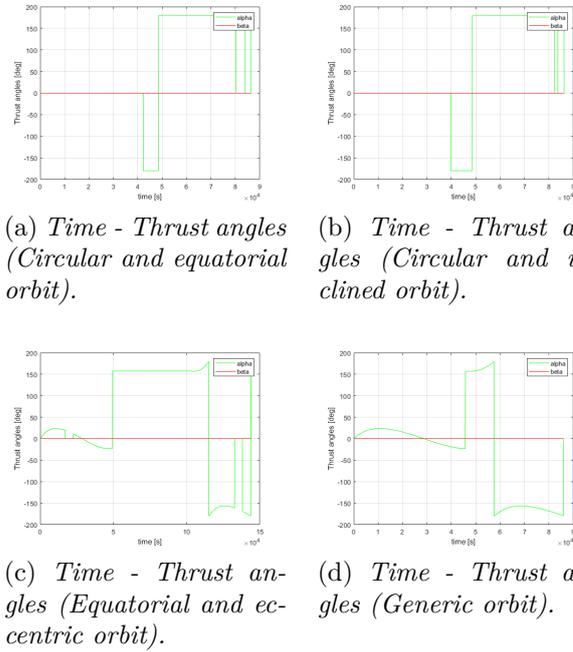


Figure 7.20: Time - Thrust angles (Generalization).

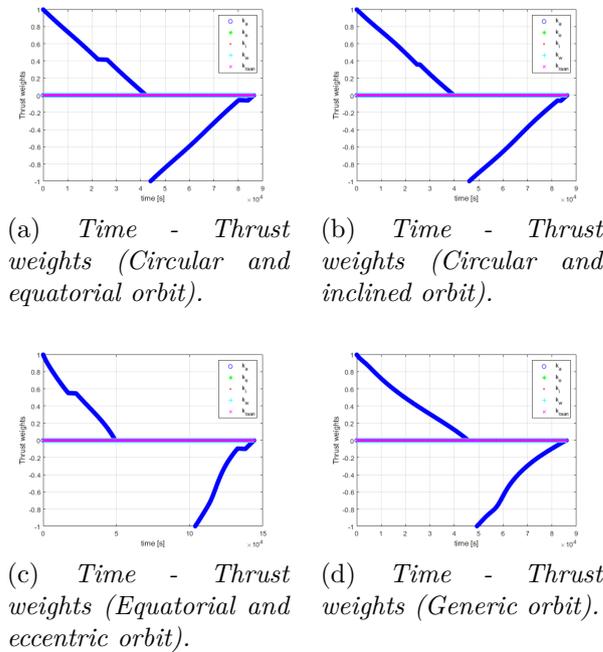
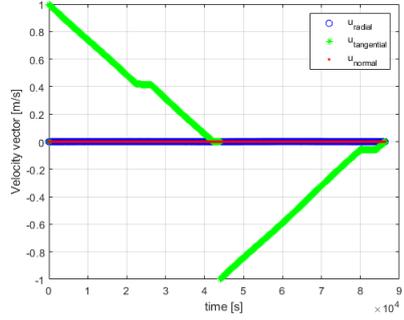
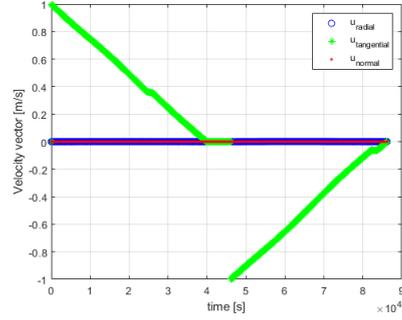


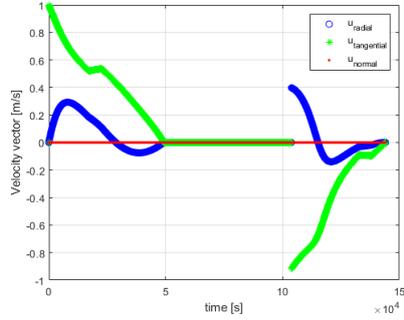
Figure 7.21: Time - Thrust weights (Generalization).



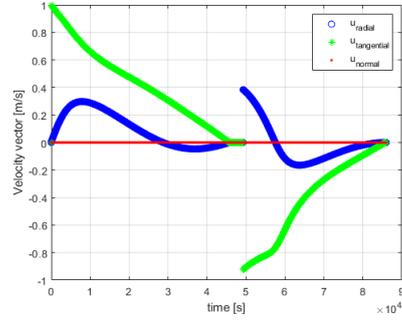
(a) Time - Velocity (Circular and equatorial orbit).



(b) Time - Velocity (Circular and inclined orbit).



(c) Time - Velocity (Equatorial and eccentric orbit).



(d) Time - Velocity (Generic orbit).

Figure 7.22: Time - Velocity (Generalization).

As reported in tab. 7.4, the costs of the maneuvers are about the same, even if we are considering different typology of orbit. The parameter that change are the final time and the  $t_{waiting\ cycle}$ .

RESULTS	Circular and equatorial	Circular	Equatorial	Generic	
$true\ longitude_{phasing}$	180.62494	179.79000	180.54545	179.25731	deg
$m_{prop\ cycle}$	2.80568	2.81878	2.87341	3.00671	kg
$t_{f\ cycle}$	1	0.99861	1.66806	0.99583	days
$\Delta v_{phasing\ cycle}$	11.01649	11.06795	11.28232	11.80539	m/s
$t_{waiting\ cycle}$	0.48333	1.71666	15.11667	0.95000	hours
$\Delta longitude_{waiting\ cycle}$	0.18932	0.672118	0.35799	0.36736	rad
$t_{eclipse}$	2.10000	0.733333	4.48333	0.00	hours

Table 7.4: Results (Generalization).



# Chapter 8

## No waiting

Using the same controls seen in the previous chapter, we tried to eliminate the waiting phase within the phasing maneuver. In this way, the phasing maneuver becomes the union of two consecutive EOR maneuvers. In order to do this, there were many possibilities such as modify the thrust of the satellite or the thrust weight, but in this chapter will be analyzed another way. In fact, it has been possible to reach our goals, modifying the phasing altitude in order to obtain it as output of the function. In this way, the phasing altitude we get is the one necessary to reach the desired position of the satellite, as we can see in Fig. 8.1.

In function of the orbit typology we are considering, the controls and the value of  $\Delta phasing_{altitude}$  are different (in Fig. 8.1 are presented the values for what concern a circular and equatorial orbit).

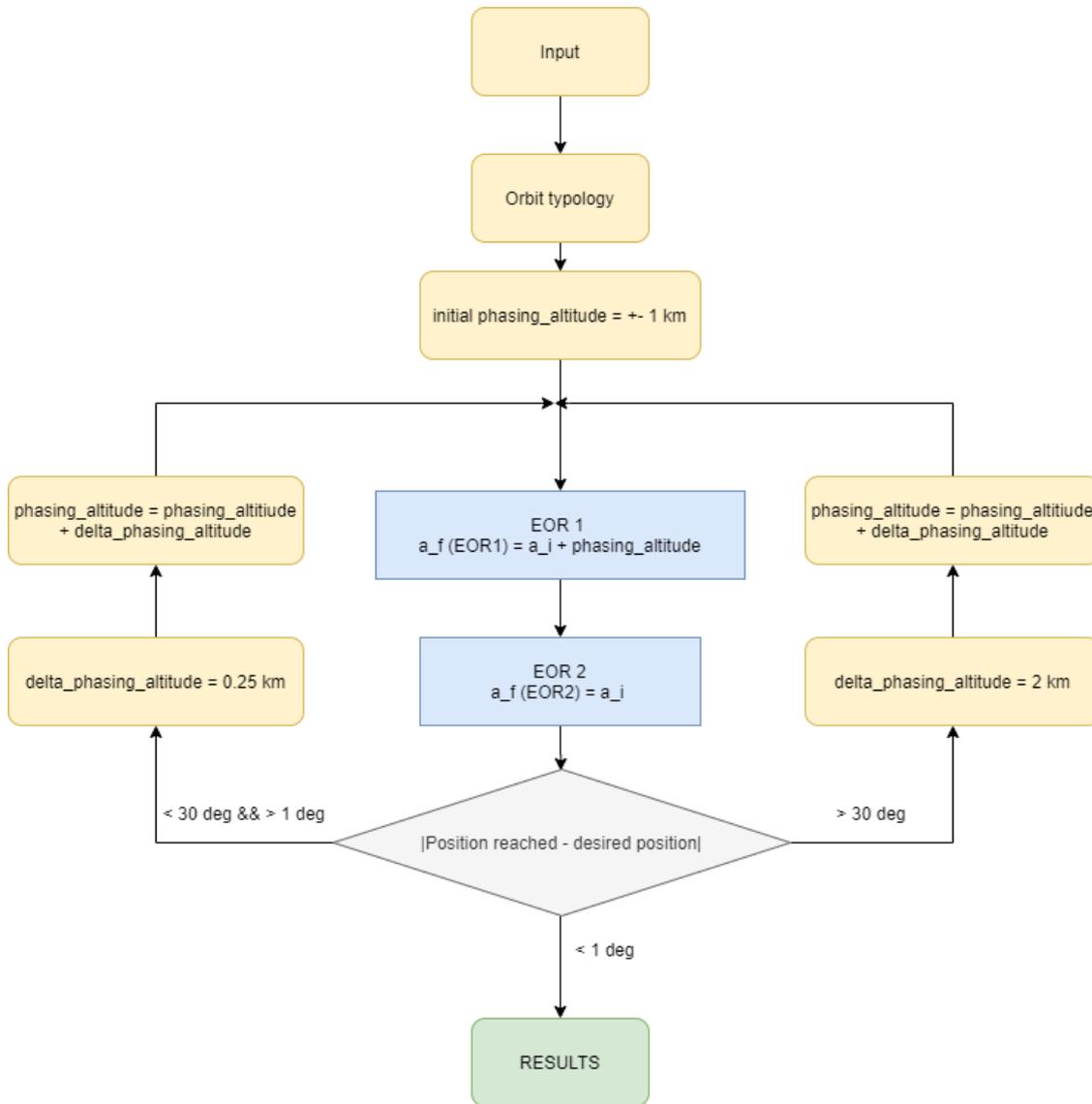


Figure 8.1: No Waiting - Logic.

## 8.1 Simulation - Geostationary orbit

In tab. 8.1 there are the input for phasing maneuver in case of geostationary orbit and the control regards the longitude.

It has been set an  $initial\ phasing_{altitude} = -1\ km$  and MAGNETO, performing two EOR, calculates the position reached by the satellite. If the angle the satellite reaches is lower than the desired position, the phasing altitude is increased by a certain value, otherwise it decreases. This procedure is also followed in reverse in case the initial phasing altitude is positive. That's decided by  $flag_{phasing\_altitude}$ ,

a flag imposed by the user in "MAIN.m":

- $flag\_phasing\_altitude = 0 \rightarrow phasing_{altitude} > 0$ ;
- $flag\_phasing\_altitude = 1 \rightarrow phasing_{altitude} < 0$ ;

<i>date</i>	2019 Feb 19 15 : 30 : 00	UTC;
<i>a<sub>i</sub></i>	$42157 \cdot 10^3$	[m]
<i>i<sub>i</sub></i>	0	[deg]
<i>e<sub>i</sub></i>	0	
$\Omega_i$	0	[deg]
$\omega_i$	0	[deg]
$\nu_i$	0	[deg]
<i>a<sub>f</sub></i>	$a_i + phasing_{altitude}$	[m]
<i>i<sub>f</sub></i>	0	[deg]
<i>e<sub>f</sub></i>	0	
$\Omega_f$	0	[deg]
$\omega_f$	0	[deg]
$\nu_f$	0	[deg]
<i>thruster</i>	1	
<i>thrust per thruster</i>	$1000 \cdot 10^{-3}$	[N]
<i>I<sub>SP</sub></i>	2800	[sec]
<i>initial phasing<sub>altitude</sub></i>	$-1 \cdot 10^3$	[m]
<i>Initial longitude<sub>phasing</sub></i>	338	[deg]
<i>Longitude<sub>phasing</sub></i>	350	[deg]

Table 8.1: Input Phasing maneuver (Geostationary orbit - No Waiting).

In Fig. 8.2, it's reported the phasing maneuver with both ECI (red trajectory) and ECEF (in green) coordinates, where the \* indicates the initial position, while *o* indicates the final position.

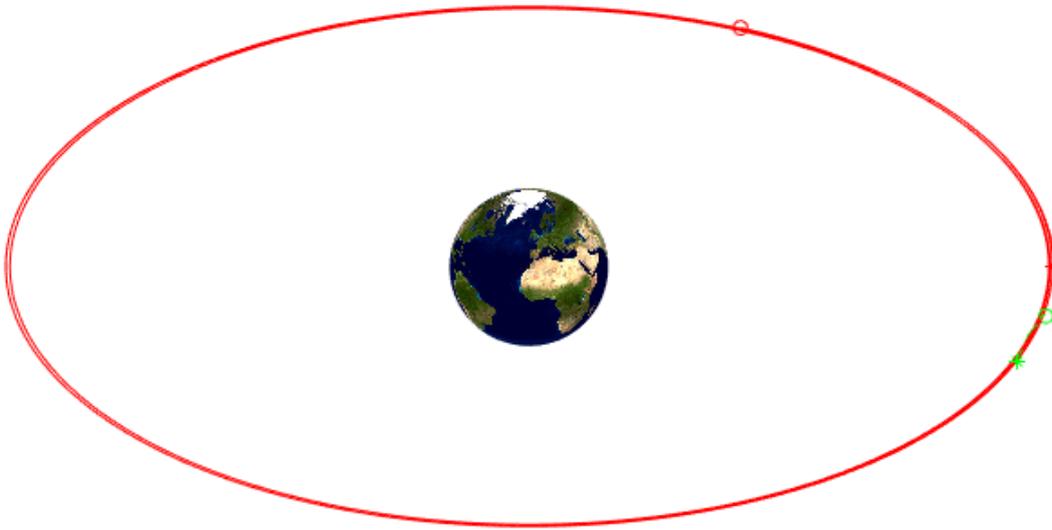


Figure 8.2: Phasing maneuver.

According to  $flag\_phasing\_altitude = 0$ , in Fig. 8.3, we can see the trend of the semi-major axis in function of time. It firstly decreases in order to reach  $a_f$ . In this way, the longitude increase its value (Fig. 8.4). In fact, assuming that the satellite speed is equal to the Earth rotational speed, if we decrease the semi-major axis value, the satellite increase its velocity. Then the satellite increases immediately its semi-major axis with no waiting phase.

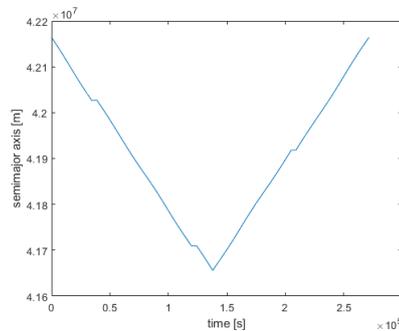


Figure 8.3: Time - semi-major axis.

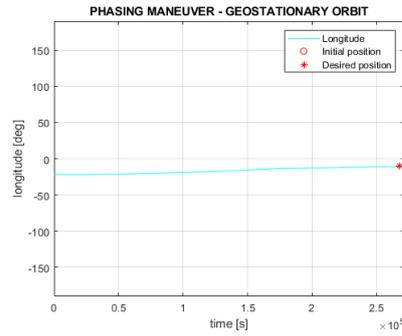


Figure 8.4: Time -Longitude.

As we said before, in this case there is no more the waiting phase. In this way, the maneuver is faster in respect of the maneuver analyzed in previous chapter, but it's also more expensive in terms of  $\Delta V$  and  $m_{prop}$  (Fig. 8.5 and 8.6), at least for what concern the geostationary orbit. Now these values are constant only if the satellite is in eclipse (Fig. 8.7). The linear trends are due to the fact that the engine we are considering presents constant value of  $Thrust$  and  $I_{sp}$ .

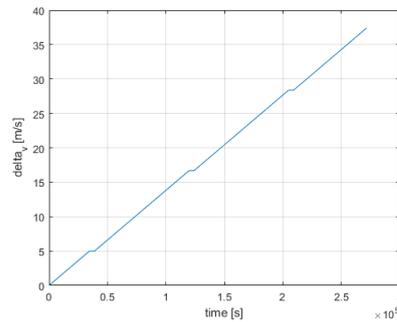


Figure 8.5: Time -  $\Delta V$ .

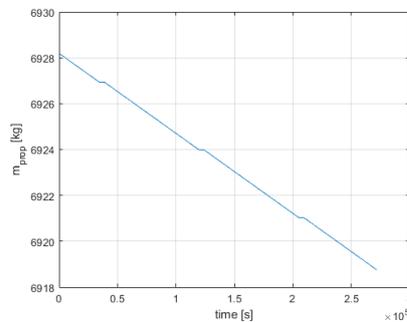


Figure 8.6: Time -  $m_{tot}$ .

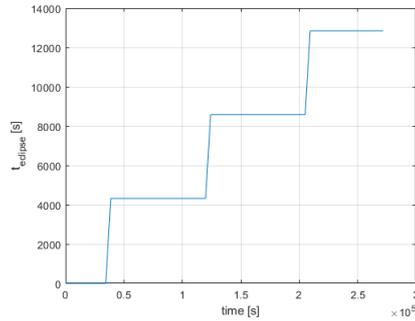


Figure 8.7: Time -  $t_{eclipse}$ .

In the analyzed case, we can use the hypotheses of the Edelbaum problem. In fact, in order to decrease the semi-major axis we get  $\alpha = \pi$ , while  $\beta = 0$ , while to get back to the initial orbit we have  $\alpha = 0$  and  $\beta = 0$ .

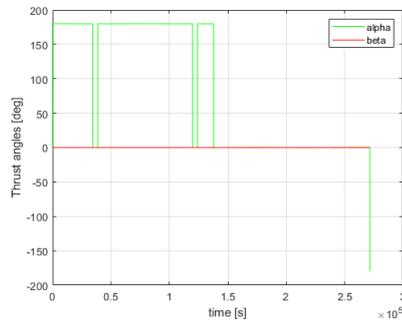
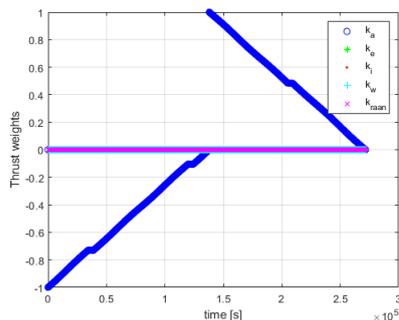
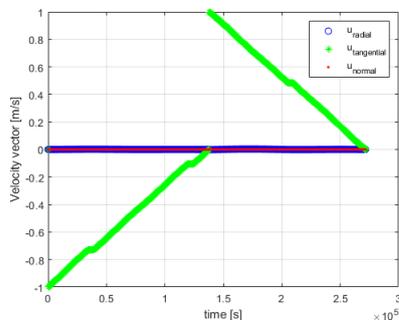


Figure 8.8: Time -  $\alpha$  and  $\beta$ .

$k_a$  is the only thrust weights that really matters in order to obtain a change for what concern the semi-major axis, and so for the  $u_{radial}$ . The fact that the waiting phase is no more presents in the maneuver logic it's highlighted also from Fig. 8.9 (Time - Thrust weights) and from Fig. 8.10 (Time - Velocity). In fact, considering  $k_a$  and  $u_{tangential}$ , the trends of these values are no more constant except when the satellite is in eclipse.


 Figure 8.9: Time - *Thrust weights*.

 Figure 8.10: Time - *Velocity vector*.

In tab. 8.2 there are the results of this simulation. The  $phasing_{altitude}$  presents a very high value in respect of the other simulation. That's because in order to make a change of  $\Delta Longitude \approx 1^\circ$ , it's necessary to increase the semi-major axis of about  $150 \text{ km}$ . Obviously, the more we go far from the initial orbit, the less we have to increase the semi-major axis value in order to obtain  $\Delta Longitude$ .

$phasing_{altitude}$	- 508	km
$true longitude_{phasing}$	349.4428566389121	deg
$m_{prop \ cycle}$	9.428733128102067	kg
$t_{f_{cycle}}$	3.094493637105124	days
$\Delta v_{phasing \ cycle}$	37.403180387831430	m/s
$t_{eclipse}$	0.1486111111111111	days

Table 8.2: Results (Geostationary orbit - No Waiting)

This maneuver logic has been able to solve a problem presents in [1]. In fact, as we can see in Fig. 8.11, in case we want the satellite to perform a phasing maneuver of  $15^\circ$ , this maneuvers take longer respect a phasing maneuver of  $60^\circ$ . That's

because when the satellite performs the first EOR with  $\Delta phasing_{altitude} = 20 \text{ km}$ , it may already pass through the desired position, and so it has to complete a full orbit before performing the second EOR.

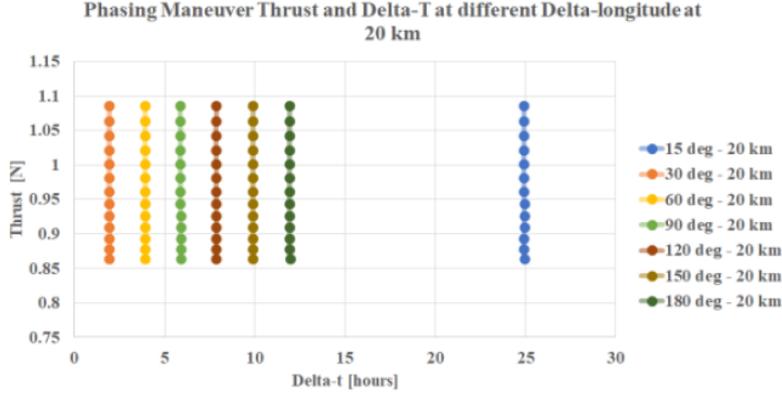


Figure 8.11: Results for the phasing maneuver at 20km, [1].

## 8.2 Simulations

In this chapter will be presented the results of some simulations. Every simulation has been made in order to have different kind of orbit. In this way has been possible to evidence that the program will work using any type of control. In tab. 8.3 there are the common inputs for the orbit we are going to consider.

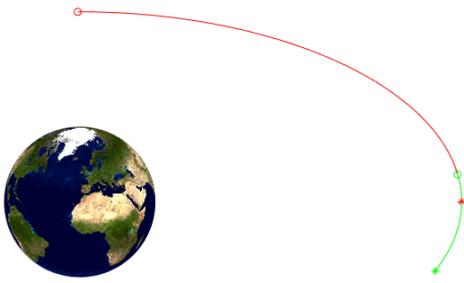
<i>date</i>	2019 Feb 19 15 : 30 : 00	UTC;
$a_i$	$32165 \cdot 10^3$	[m]
$\Omega_i$	0	[deg]
$\omega_i$	0	[deg]
$\nu_i$	0	[deg]
$a_f$	$a_i + phasing_{altitude}$	[m]
$\Omega_f$	0	[deg]
$\omega_f$	0	[deg]
$\nu_f$	0	[deg]
<i>thruster</i>	1	
<i>thrust per thruster</i>	$1000 \cdot 10^{-3}$	[N]
$I_{SP}$	2800	[sec]
<i>initial phasing<sub>altitude</sub></i>	$-1 \cdot 10^3$	[m]
<i>True longitude<sub>phasing</sub></i>	90	[deg]

Table 8.3: Input Phasing maneuver (No waiting).

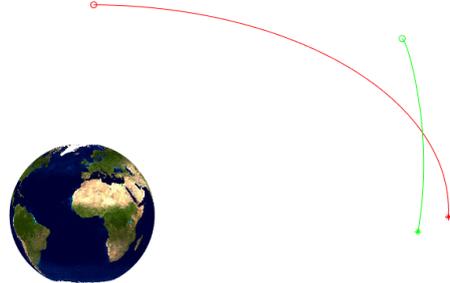
The simulations presented refers to the following type of orbit:

- Circular and equatorial orbit:  $i_i = i_f = 0$  and  $e_i = e_f = 0$ ;
  
- Circular and inclined orbit:  $i_i = i_f = 28.5$  and  $e_i = e_f = 0 \rightarrow True\ longitude_{phasing} = Argument\ of\ Latitude$ ;
  
- Equatorial and eccentric orbit:  $i_i = i_f = 0$  and  $e_i = e_f = 0.4$ ;
  
- Generic orbit (inclined and eccentric):  $i_i = i_f = 28.5$  and  $e_i = e_f = 0.4 \rightarrow True\ longitude_{phasing} = Argument\ of\ Latitude$ ;

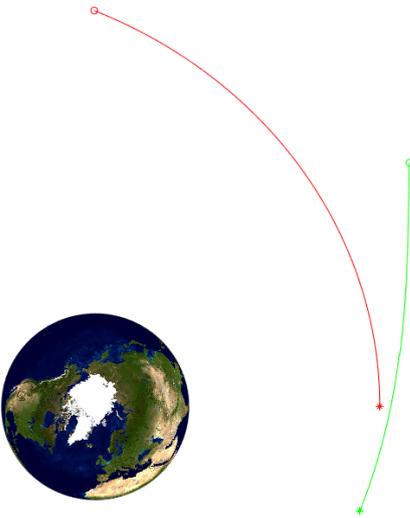
In Fig. 9.17 there are the interpolations we get. As we can see, these interpolations are obtained using  $flag\_phasing\_altitude = 1$  ( $initial\ phasing\_altitude < 0$ ). Fig. 8.12 there is the trajectory followed by the spacecraft. As the same way as all the previous simulations, in red there are ECI coordinates and in green the ECEF ones. In order to understand better the trajectory, we must consider Fig. 8.13, which highlighted the trends of the semi-major axis in function of time. We chose  $flag\_phasing\_altitude = 1$  ( $initial\ phasing\_altitude < 0$ ) and so during the first EOR,  $a$  decreases its value.



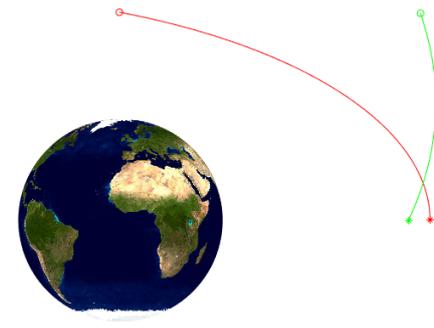
(a) *Phasing maneuver (Circular and equatorial orbit).*



(b) *Phasing maneuver (Circular and inclined orbit).*

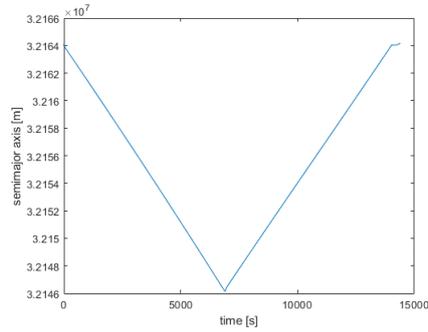


(c) *Phasing maneuver (Equatorial and eccentric orbit).*

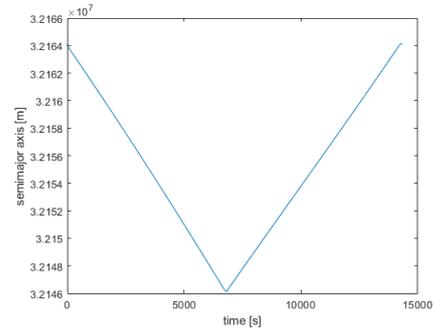


(d) *Phasing maneuver (Generic orbit).*

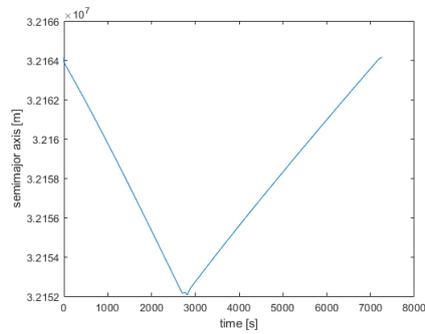
Figure 8.12: Phasing maneuver (No waiting).



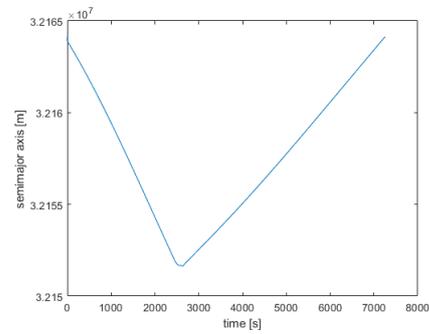
(a) Time - semi-major axis (Circular and equatorial orbit).



(b) Time - semi-major axis (Circular and inclined orbit).



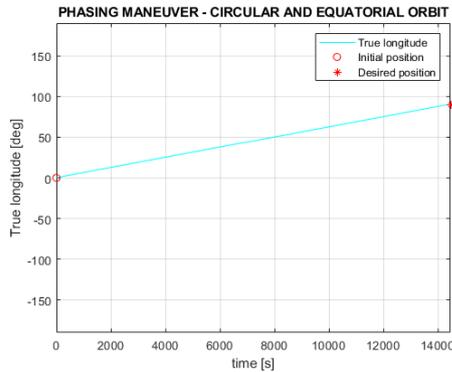
(c) Time - semi-major axis (Equatorial and eccentric orbit).



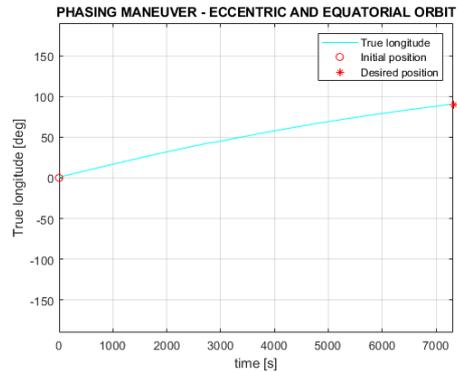
(d) Time - semi-major axis (Generic orbit).

Figure 8.13: Time - semi-major axis (No waiting).

As reported in chapter 7.3, in case of equatorial orbit, the maneuver ends when the satellite reaches the desired true longitude, while we consider the Argument of Latitude in case of inclined orbit (Fig. 8.14 and 8.15).

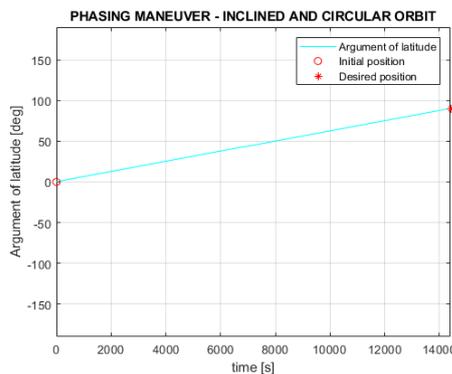


(a) Time - True longitude (Circular and equatorial orbit).

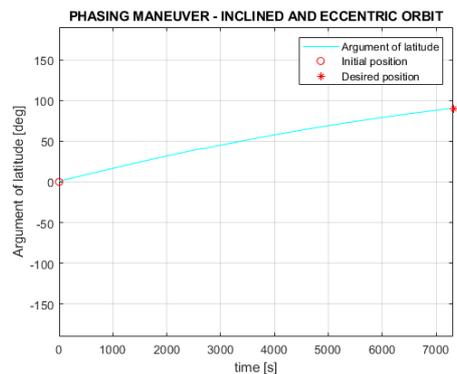


(b) Time - True longitude (Equatorial and eccentric orbit).

Figure 8.14: Time - True Longitude (No waiting).



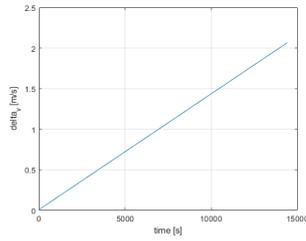
(a) Time - Argument of Latitude (Circular and inclined orbit).



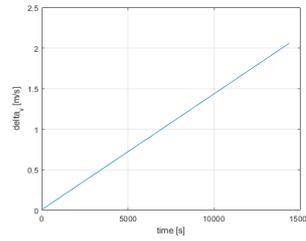
(b) Time - Argument of Latitude (Generic orbit).

Figure 8.15: Time - Argument of Latitude (No waiting).

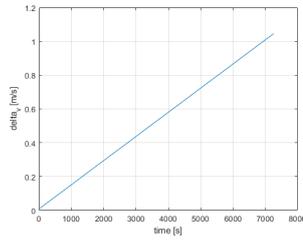
As always the costs are represented by the  $\Delta v$  (Fig. 8.16) and by the consumption of propellant (Fig. 8.17). Both these parameters, they don't have constant value in function of time because we have no waiting phase and the satellite is never eclipsed (Fig. 8.18). The cost is practically the same for what concern "Circular and equatorial orbit" and "Circular and inclined orbit" and the same is for "Equatorial and eccentric orbit" and "Generic orbit".



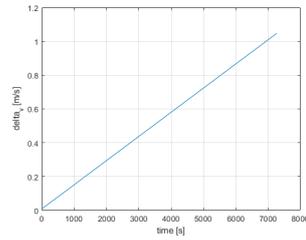
(a) *Time - Delta V (Circular and equatorial orbit).*



(b) *Time - Delta V (Circular and inclined orbit).*

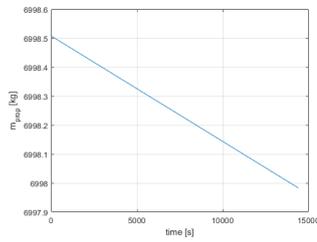


(c) *Time - Delta V (Equatorial and eccentric orbit).*

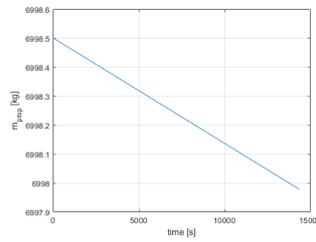


(d) *Time - Delta V (Generic orbit).*

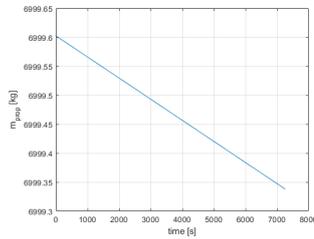
Figure 8.16: Time - Delta V (No waiting).



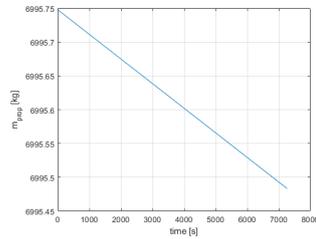
(a) *Time -  $m_{tot}$  (Circular and equatorial orbit).*



(b) *Time -  $m_{tot}$  (Circular and inclined orbit).*

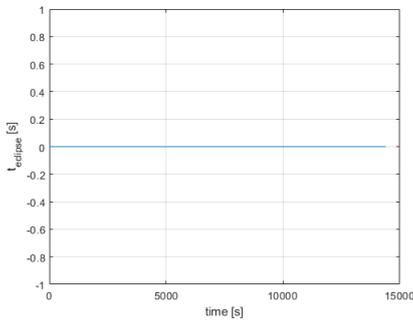


(c) *Time -  $m_{tot}$  (Equatorial and eccentric orbit).*

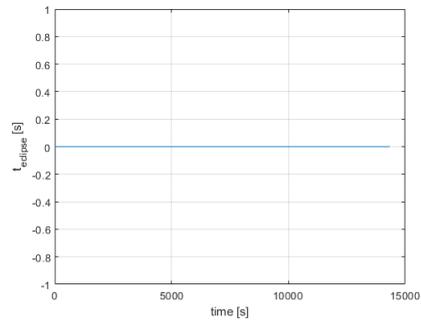


(d) *Time -  $m_{tot}$  (Generic orbit).*

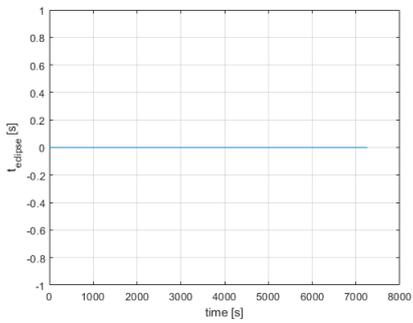
Figure 8.17: Time -  $m_{tot}$  (No waiting).



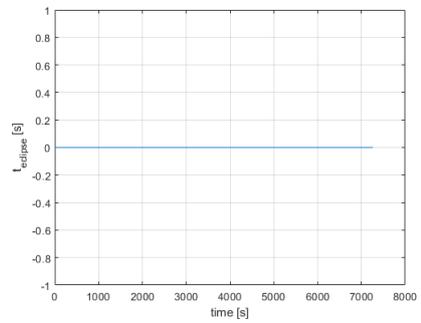
(a) *Time - Eclipse time (Circular and equatorial orbit).*



(b) *Time - Eclipse time (Circular and inclined orbit).*



(c) *Time - Eclipse time (Equatorial and eccentric orbit).*

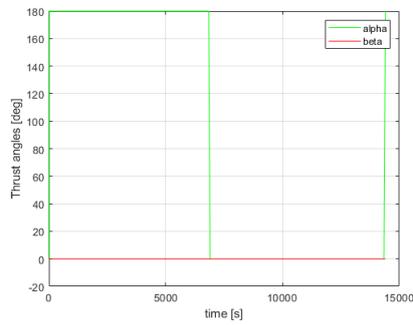


(d) *Time - Eclipse time (Generic orbit).*

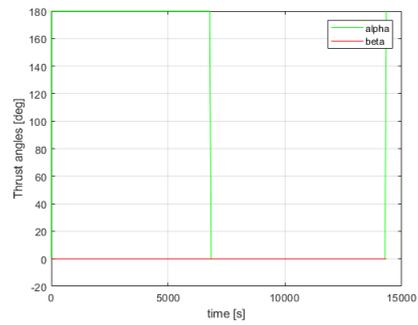
Figure 8.18: Time - Eclipse time (No waiting).

The  $\alpha$  and  $\beta$  trends are reported in Fig. 8.19, while the thrust weights in Fig. 8.20 and the velocity in Fig. 8.21. As we can see there is no waiting phase and the curves look the same if the orbits are circular or if they are eccentric.

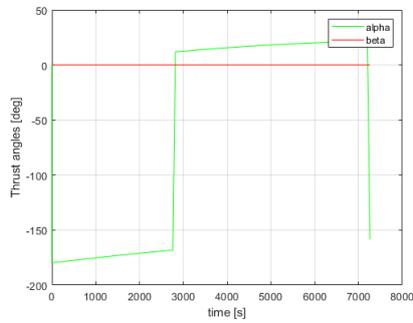
$\alpha \approx \pm\pi$  if we are going to decrease semi-major axis and  $\alpha \approx 0$  if the semi-major axis is about to increase its value, while the out-of plane angle is always 0.



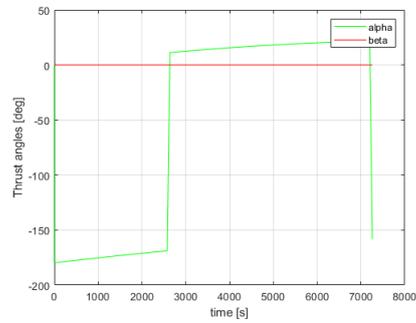
(a) *Time - Thrust angles (Circular and equatorial orbit).*



(b) *Time - Thrust angles (Circular and inclined orbit).*



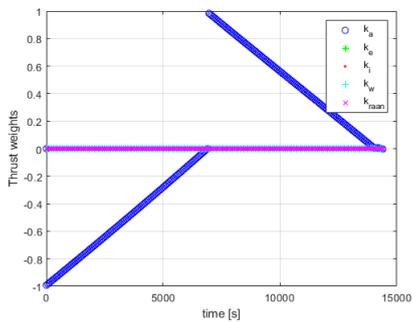
(c) *Time - Thrust angles (Equatorial and eccentric orbit).*



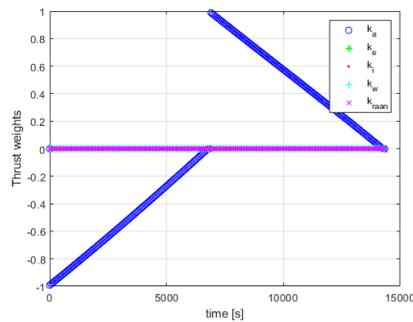
(d) *Time - Thrust angles (Generic orbit).*

Figure 8.19: Time - Thrust angles (No waiting).

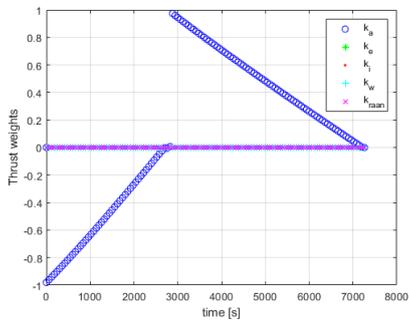
$k_a$  (the one connected to the semi-major axis) is the only thrust weight that changes. If we want the semi-major axis to decrease,  $k_a$  must be negative with linear trend with a positive slope, vice versa for what concern the increase of  $a$ .



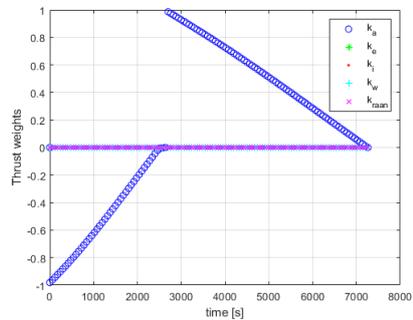
(a) *Time - Thrust weights (Circular and equatorial orbit).*



(b) *Time - Thrust weights (Circular and inclined orbit).*



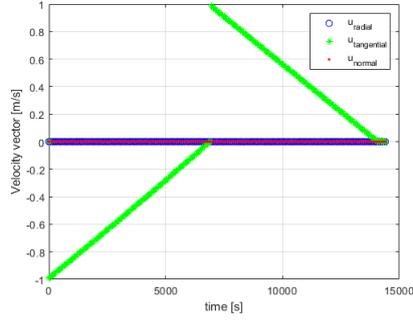
(c) *Time - Thrust weights (Equatorial and eccentric orbit).*



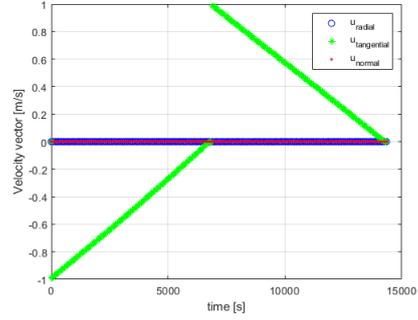
(d) *Time - Thrust weights (Generic orbit).*

Figure 8.20: Time - Thrust weights (No waiting).

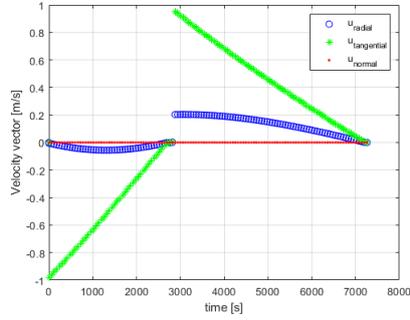
The curves of the velocity components is similar to the thrust weights one. In order to decrease the semi-major axis, for what concern circular orbit,  $u_{tangential}$  presents negative values. The tangential component increases its value linearly until the EOR maneuver is complete (when  $u_{tangential} = 0$ ). Vice versa when the semi-major axis decreases. For what concern the eccentric orbits, the discussion on the tangential component of speed is the same, while  $u_{radial}$  changes because of  $e$ .



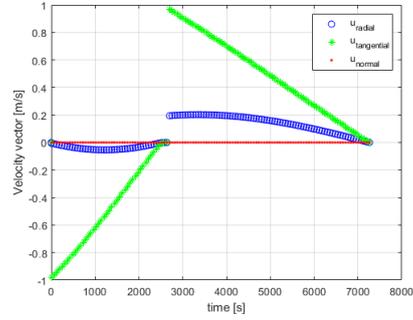
(a) Time - Velocity (Circular and equatorial orbit).



(b) Time - Velocity (Circular and inclined orbit).



(c) Time - Velocity (Equatorial and eccentric orbit).



(d) Time - Velocity (Generic orbit).

Figure 8.21: Time - Velocity (No waiting).

In tab. 8.4 are reported the numerical results we get at the end of the maneuvers.

RESULTS	Circular and equatorial	Circular	Equatorial	Generic	
$phasing_{altitude}$	-18	-18	-11.750	-12.5	km
$true\ longitude_{phasing}$	90.30897	90.31306	90.15217	90.64505	deg
$m_{prop\ cycle}$	0.52443	0.52224	0.26439	0.26439	kg
$t_{f\ cycle}$	0.16735	0.16666	0.08472	0.08472	days
$\Delta v_{phasing\ cycle}$	2.06623	2.05766	1.04579	1.04637	m/s
$t_{eclipse}$	0.00	0.00	0.00	0.00	hours

Table 8.4: Results (No waiting).



# Chapter 9

## Interpolation

The results obtained in the previous chapter turned out to be quite good. The maneuver logic in which only two EOR maneuvers are performed without a waiting phase has proved to be an excellent choice. We therefore chose to continue on this line.

Obviously, changes were made to the program in order to make the whole maneuver as accurate as possible.

### 9.1 Maneuver logic

We give the same input as before:

- $a_i$  Initial altitude [m] (semi-major axis altitude);
- $a_f$  Final altitude [m] (semi-major axis altitude);
- $e_i$  Initial eccentricity [adim];
- $e_f$  Final eccentricity [adim];
- $i_i$  Initial inclination [rad];
- $i_f$  Final inclination [rad];
- $\omega_i$  Initial Argument of the Perigee [rad];
- $\omega_f$  Final Argument of the Perigee [rad];
- $\Omega_i$  Initial RAAN [rad];
- $\Omega_f$  Final RAAN [rad];

- $\nu_i$  Initial true anomaly [rad];
- $\nu_f$  Final true anomaly [rad];
- number of thruster;
- thrust per thruster;
- $I_{sp}$  Specific impulse [s];
- True Longitude phasing [rad], position we want the satellite to reach.

For what concern the True longitude phasing, according to the type of orbit, it takes on a different meaning:

- Geostationary orbit:  $True\ Longitude_{phasing} = Longitude$  or  $True\ Longitude_{phasing} = True\ Longitude$  (decided by  $flag\_long$ , a flag imposed by the user in the "MAIN.m");
- Equatorial orbit:  $True\ Longitude_{phasing} = True\ Longitude$ ;
- Inclined orbit:  $True\ Longitude_{phasing} = Argument\ of\ Latitude$ .

In tab. 9.1 there are the input used for a simulation in case of geostationary orbit. The positions we are going to control is the longitude.

<i>date</i>	2019 Feb 19 15 : 30 : 00	UTC;
<i>a<sub>i</sub></i>	$42157 \cdot 10^3$	[m]
<i>i<sub>i</sub></i>	0	[deg]
<i>e<sub>i</sub></i>	0	
$\Omega_i$	0	[deg]
$\omega_i$	0	[deg]
$\nu_i$	21.55	[deg]
<i>a<sub>f</sub></i>	$a_i + phasing_{altitude}$	[m]
<i>i<sub>f</sub></i>	0	[deg]
<i>e<sub>f</sub></i>	0	
$\Omega_f$	0	[deg]
$\omega_f$	0	[deg]
$\nu_f$	21.55	[deg]
<i>thruster</i>	1	
<i>thrust per thruster</i>	$1000 \cdot 10^{-3}$	[N]
<i>I<sub>SP</sub></i>	2800	[sec]
<i>Longitude<sub>phasing</sub></i>	35	[deg]

Table 9.1: Input Phasing maneuver (Geostationary orbit - Interpolation).

### 9.1.1 Initial case

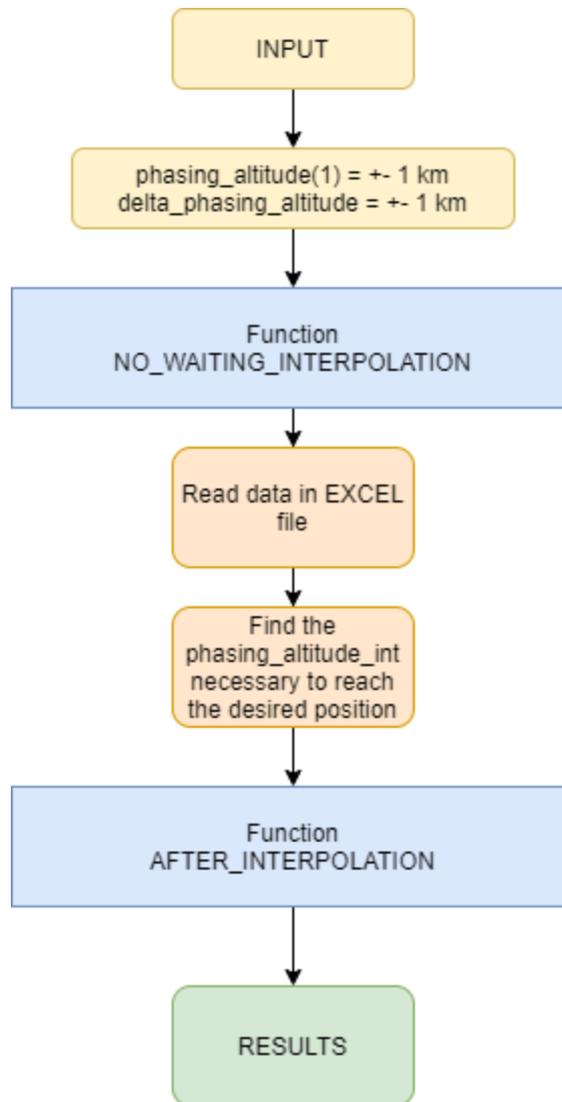


Figure 9.1: Initial maneuver logic.

As can be seen from Fig. 9.1, inputs are given to the program. We set an initial  $phasing\_altitude(1) = \pm 1 \text{ km}$  and a  $\Delta phasing\_altitude = \pm 1 \text{ km}$  according to a flag ( $flag\_phasing\_altitude$ ) set in the MAIN of the program. If  $flag\_phasing\_altitude = 0$ , the initial phasing altitude will be positive as well as the delta, vice versa if  $flag\_phasing\_altitude = 1$ . Then we enter the function "NO\_WAITING\_INTERPOLATION.m" (Fig. 9.2).

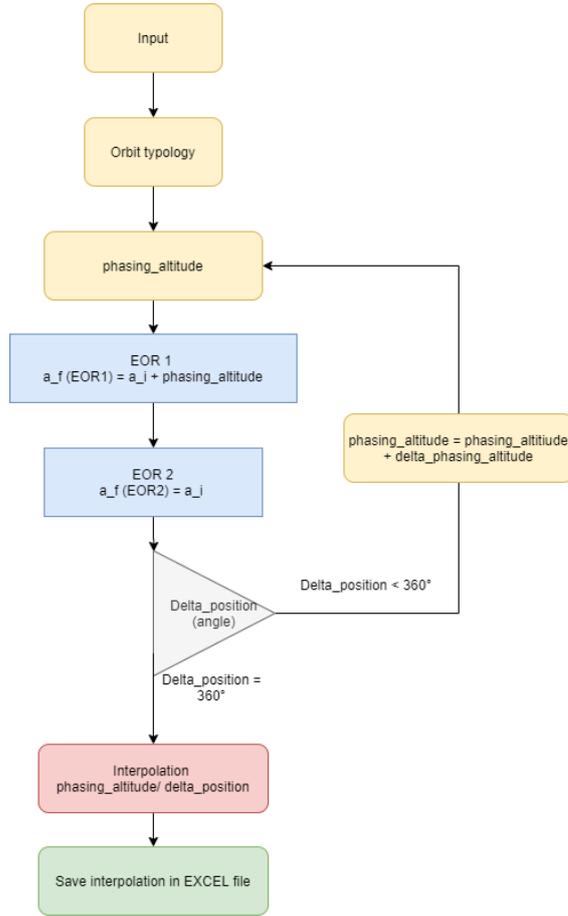


Figure 9.2: NO WAITING INTERPOLATION logic.

First we go to establish the type of orbit treated in order to carry out different checks.

The program then performs the EOR maneuvers with  $phasing_{altitude} = \pm 1km$  and once this is done, it calculates the position reached by the satellite in terms of angle.

The difference between the position reached and the initial position is therefore made. If this difference is  $< 360^\circ$  the program iterates the procedure with  $phasing_{altitude} = phasing_{altitude} + \Delta phasing_{altitude}$ , otherwise the program exits the loop.

For what concern the Geostationary orbit, if we are controlling the longitude, we set the initial  $phasing_{altitude} = \pm 150km$  and the  $\Delta phasing_{altitude} = \pm 20km$ . It has been decided to use this values in order to reduce the computational cost of the simulation and because under  $150km$  the longitude remain about the same as the initial one ( $\Delta Longitude \approx 1^\circ$ ), as reported in chapter 10.

Once the program exits the loop, it saves all the phasing altitudes used and all the delta positions made. Then, it interpolates the data obtained and saves them in an EXCEL file. The EXCEL file's folder is named as *date\_thrust\_I<sub>sp</sub>*, while the EXCEL file is named using the orbital elements which describe the initial orbit given as input. The EXCEL file looks like Fig. 9.3, where in the first column there are the *phasing<sub>altitude</sub>[m]*, while in the second one all the *Δposition[rad]*.

	A	B	C
1	1000	0,134377	
2	1010	0,135178	
3	1020	0,135978	
4	1030	0,136779	
5	1040	0,137579	
6	1050	0,13838	
7	1060	0,139181	
8	1070	0,139981	
9	1080	0,140782	
10	1090	0,141582	
11	1100	0,142383	
12	1110	0,143183	
13	1120	0,143984	
14	1130	0,144784	
15	1140	0,145585	
16	1150	0,146386	
17	1160	0,147186	
18	1170	0,147987	
19	1180	0,148787	
20	1190	0,149588	

Figure 9.3: EXCEL file examples.

Once the file EXCEL is created, the program reads and plot the interpolation.

In Fig. 9.4, we have the interpolation for what concern the Geostationary orbit with *flag<sub>phasing<sub>altitude</sub></sub>* = 1 (phasing altitude decrease step by step).

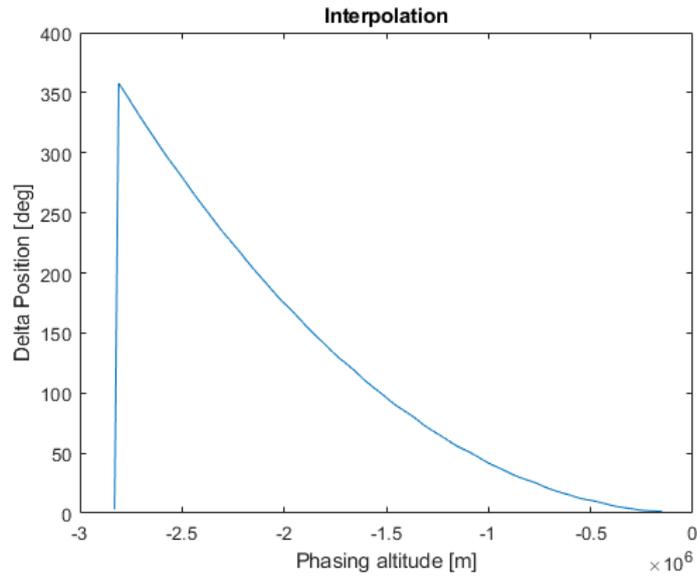


Figure 9.4: Interpolation  $phasing_{altitude}/\Delta position$ .

Then the program looks for the phasing altitude necessary to reach the delta angle we want to make. It creates a vector with all the phasing altitude useful to reach our purpose and it enters in "*AFTER\_INTERPOLATION.m*" function.

As shown in Fig. 9.5 the programs performs the two EOR using the first phasing altitude useful. If the satellite reaches the desired position, the program gives us the results, otherwise it tries another phasing altitude.

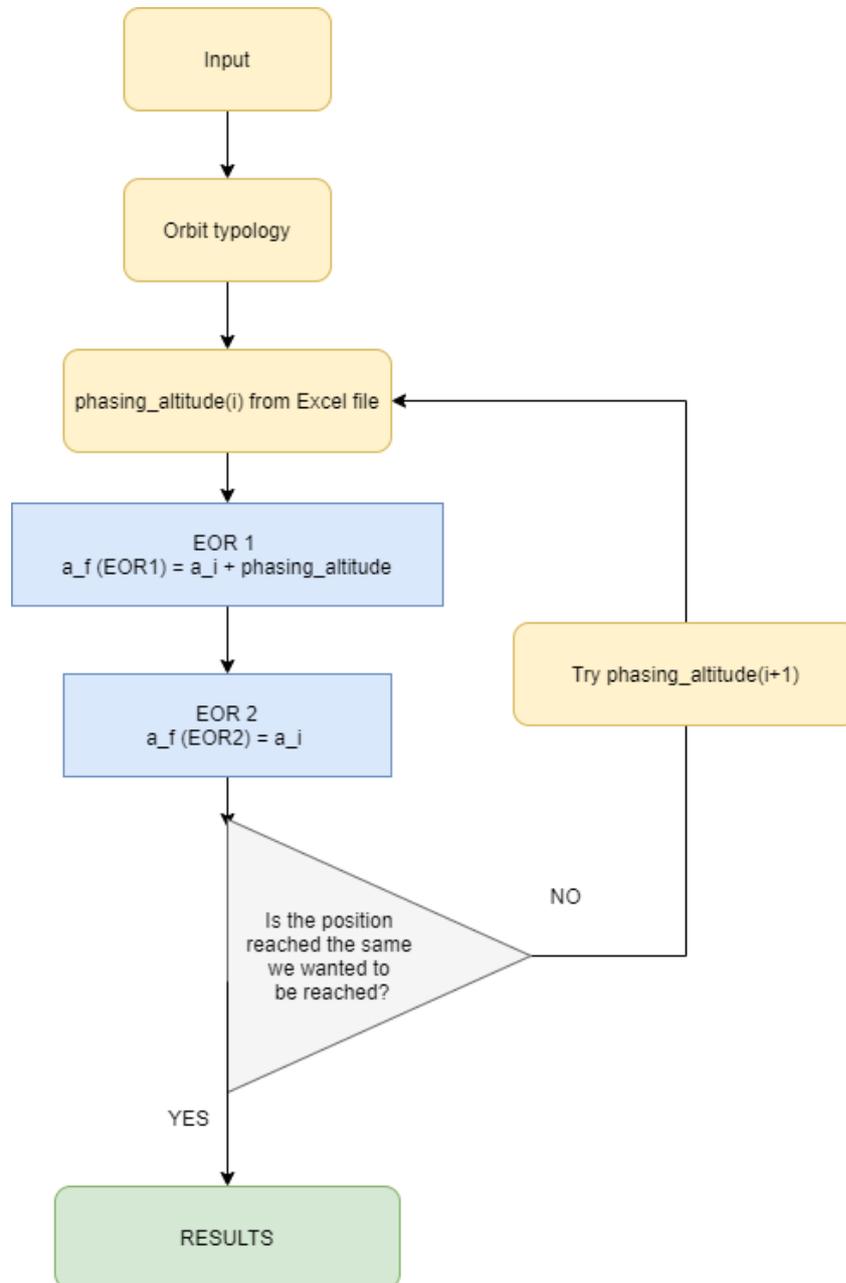


Figure 9.5: AFTER INTERPOLATION logic.

### 9.1.2 Results - Geostationary case

As the same way as previous simulations, in Fig. 6.13 we can see the trajectory of our satellite, where in red there are the ECI coordinates and in green the ECEF coordinates, where the \* indicates the initial position, while o indicates the final position. As we can see, the spacecraft decreases the semi-major axis, as reported in

Fig. 9.7. Its value (and so for any other parameters we are considering) is constant only when the satellite is in eclipse (Fig. 9.11).

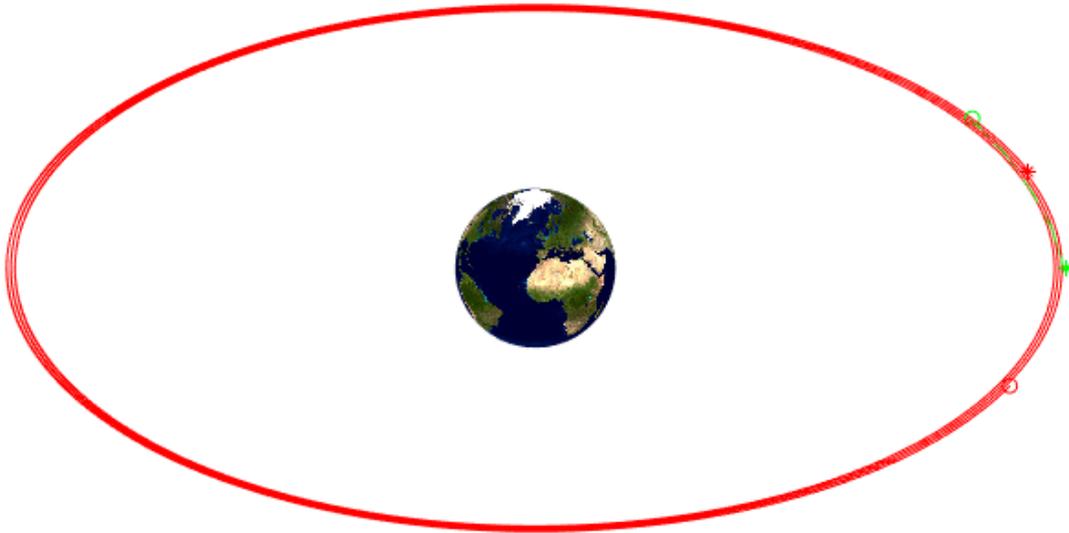


Figure 9.6: Phasing maneuver.

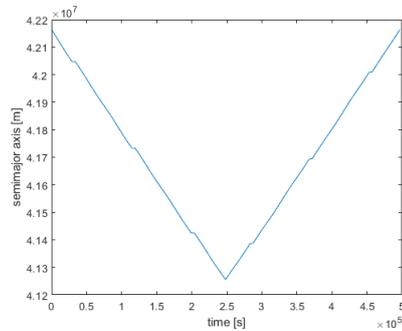


Figure 9.7: Time - semi-major axis.

In Fig. 9.8 there is longitude in function of time. We can see that the more the satellite is far from its initial orbit, the more the Longitude in function of time increases its inclination. In fact, as already said, the satellite velocity increases its value if the orbit's radius decrease and vice versa. That's why we firstly decrease semi-major axis in order to obtain an increase in terms of longitude.

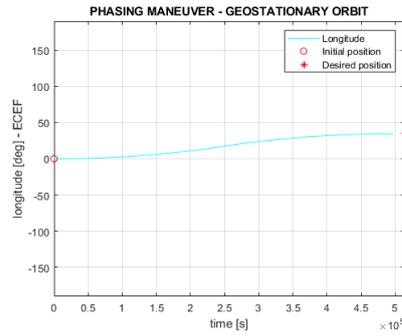


Figure 9.8: Time - Longitude.

As reported in Fig. 9.21, 9.22 and in chapter 8.1, the costs of this maneuver is higher in respect of the other simulations performed with different typology of orbit and also in respect of the first phasing maneuver logic (chapter 7) using the same inputs. That's because the satellite has to perform an EOR with a higher value of semi-major axis in respect of all the other case. Anyway the cost trends its almost linear because we are considering  $Thrust = 1\text{ N}$  and  $I_{sp} = 2800\text{ s}$ .

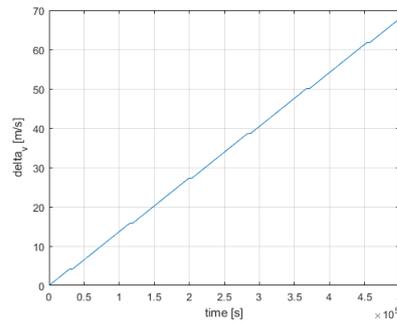


Figure 9.9: Time -  $\Delta V$ .

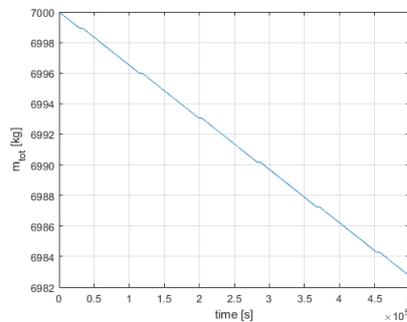


Figure 9.10: Time -  $m_{tot}$ .

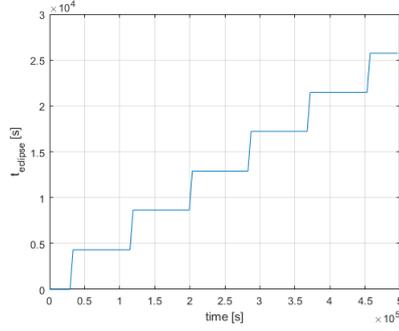


Figure 9.11: Time -  $t_{eclipse}$ .

In the same way, in Fig. 9.12, 9.13 and 9.14 we can see respectively the thrust angles, the thrust weights and the velocity in function of time. The trends of those value are about the same every simulation we made for what concern non-eccentric orbit. In fact,  $\alpha = \pi$  in order to decrease  $a$ ,  $\alpha = 0$  if  $a$  must increase its value and  $\beta = 0$  because we are not maneuvering out of the orbital plane. However, those figures highlight that there is no more a waiting phase and if we are increasing or decreasing the semi-major axis during the first or the second EOR.

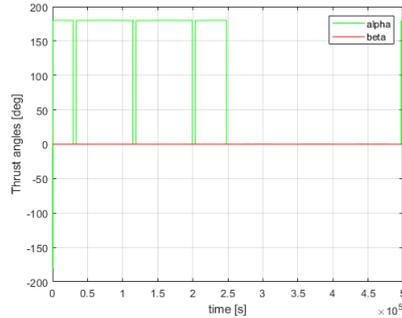


Figure 9.12: Time -  $\alpha$  and  $\beta$ .

$k_a$  is negative according to the decrease in the orbital parameter to which it refers, and it increases linearly its values until the end of the first EOR. For what concern the second EOR, the trends are reversed.

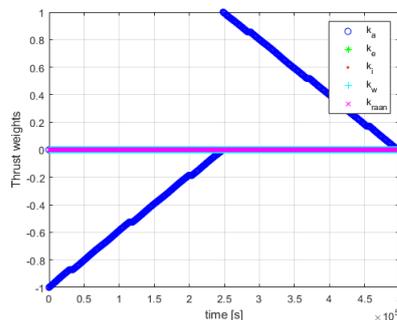


Figure 9.13: Time - *Thrust weights*.

The same for  $u_{tangential}$ , the only velocity component which changes during the phasing maneuver. Its trend is approximately the same of  $k_a$ .

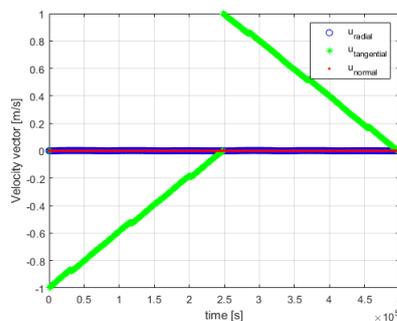


Figure 9.14: Time - *Velocity vector*.

In tab. 9.2 there are the results of this simulation.

$phasing_{altitude}$	- 908.55	km
$initial_{longitude}$	3.253111069638516e-05	deg
$longitude_{phasing}$	34.504905887605766	deg
$m_{prop\ cycle}$	17.168379185978665	kg
$t_{f_{cycle}}$	5.754861111111111	days
$\Delta v_{phasing\ cycle}$	67.437039525049613	m/s
$t_{eclipse}$	0.297916666666667	days

Table 9.2: Results (Geostationary orbit - Interpolation).

The great advantages we have using this maneuver logic is the maneuver time. In fact, if we consider that in this case we are performing a phasing maneuver with  $\Delta Longitude \approx 35^\circ$  in about 6 days, in the simulation performed in chapter 7, the satellite performs a lower  $\Delta Longitude (\approx 30^\circ)$  in about 23 days (see 7.2). So, using

this logic allow us to perform the maneuver in less time and make our satellite operational earlier, which brings great advantages.

One of the problems is the great computational cost that a simulation of this kind brings. In fact, for what concern a geostationary orbit, if the check is on longitude (not True Longitude), the simulation takes several hours in order to converge.

### 9.1.3 Following case

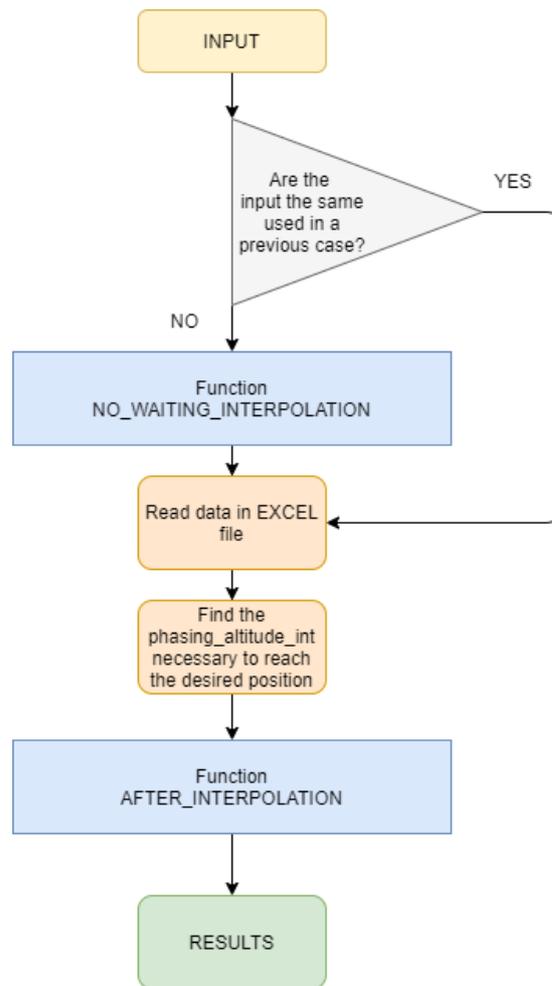


Figure 9.15: Maneuver logic.

As we can see in Fig. 9.15, if we are analyzing an identical orbit with respect to one already analyzed in a previous case, the program reads the EXCEL file directly without passing through the "*NO\_WAITING\_INTERPOLATION.m*" function.

In this way, if we have already analyzed the case in question, the computational cost of the operation decrease. That's a great advantage, at least for what concern the geostationary orbit. In fact, if the orbit considered is another one, the simulation usually doesn't take too much time.

## 9.2 Simulation

In tab. 9.3 there are the common inputs to the following simulations. For what concern inclined orbit, we are not going to consider the *true longitude*, but the *argument of latitude*, as seen in previous chapter "Generalization".

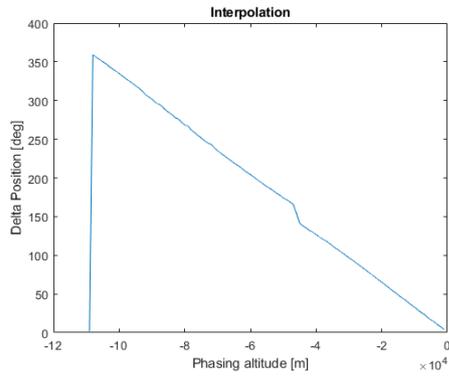
<i>date</i>	2019 Feb 19 15 : 30 : 00	UTC;
$a_i$	$37164 \cdot 10^3$	[m]
$\Omega_i$	0	[deg]
$\omega_i$	0	[deg]
$\nu_i$	0	[deg]
$a_f$	$a_i + phasing_{altitude}$	[m]
$\Omega_f$	0	[deg]
$\omega_f$	0	[deg]
$\nu_f$	0	[deg]
<i>thruster</i>	1	
<i>thrust per thruster</i>	$1000 \cdot 10^{-3}$	[N]
$I_{SP}$	2800	[sec]
<i>phasing<sub>altitude</sub></i>	$100 \cdot 10^3$	[m]
<i>inital true longitude</i>	0	[deg]
<i>true longitude<sub>phasing</sub></i>	35	[deg]

Table 9.3: Input Phasing maneuver (Interpolation).

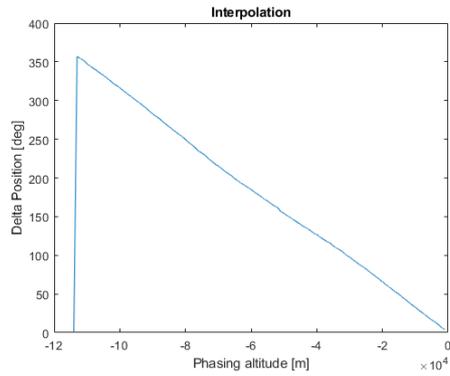
The simulations presented refers to the following type of orbit:

- Circular and equatorial orbit:  $i_i = i_f = 0$  and  $e_i = e_f = 0$ ;
- Circular and inclined orbit:  $i_i = i_f = 28.5$  and  $e_i = e_f = 0$ ;
- Equatorial and eccentric orbit:  $i_i = i_f = 0$  and  $e_i = e_f = 0.4$ ;
- Generic orbit (inclined and eccentric):  $i_i = i_f = 28.5$  and  $e_i = e_f = 0.4$ ;

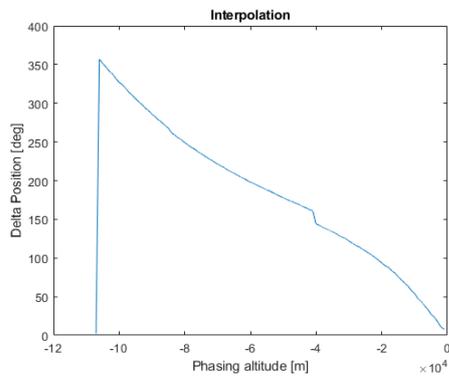
In Fig. 9.17 there are the interpolations we get. As we can see, these interpolations are obtained using *flag\_phasing\_altitude* = 1 (we decrease the semi-major axis step by step). As reported, for what concern some interpolations, there is a range of positions that the satellite is not able to reach. The reasons why will be explained in 9.3.



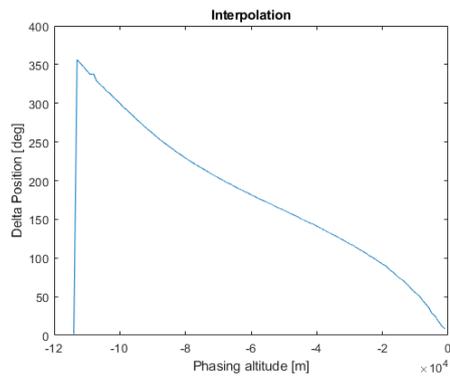
(a) *Interpolation (Circular and equatorial orbit).*



(b) *Interpolation (Circular and inclined orbit).*



(c) *Interpolation (Equatorial and eccentric orbit).*



(d) *Interpolation (Generic orbit).*

Figure 9.16: Interpolation  $phasing_{altitude}/\Delta position$ .

Fig. 9.17 there is the trajectory followed by the spacecraft. As the same way as all the previous simulations, in green we have the ECEF positions, while in red the ECI ones. In order to understand better the trajectory, we must consider Fig. 9.18, which highlighted the trends of the semi-major axis in function of time.

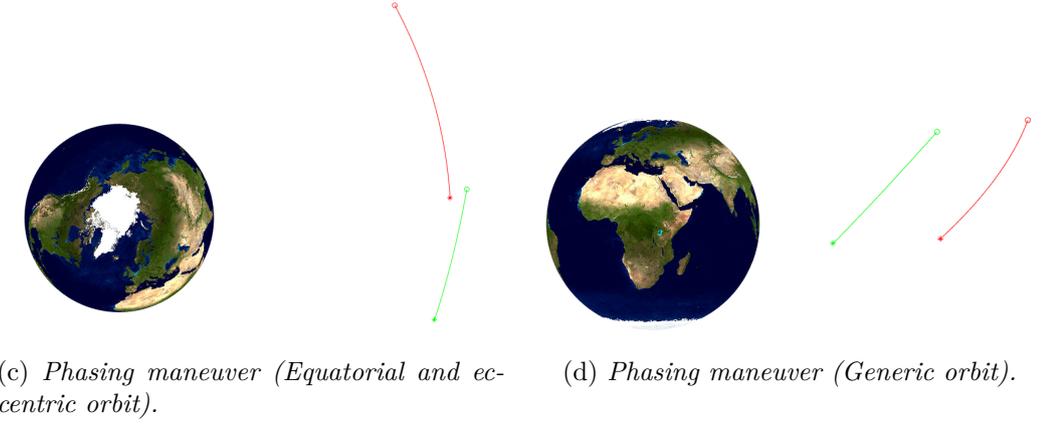
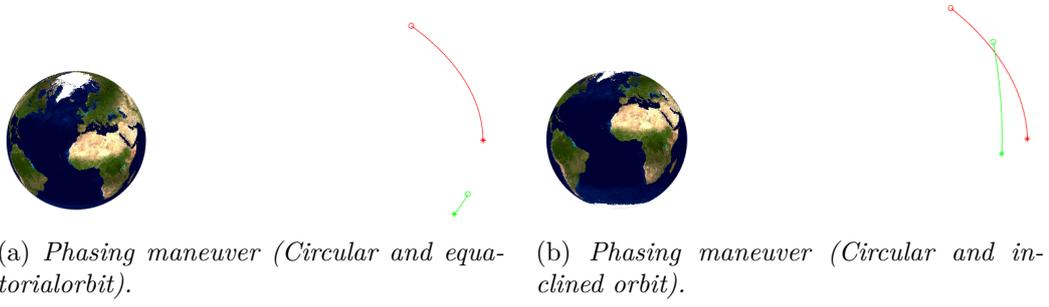
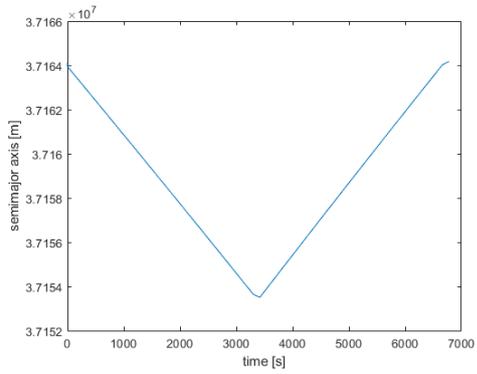


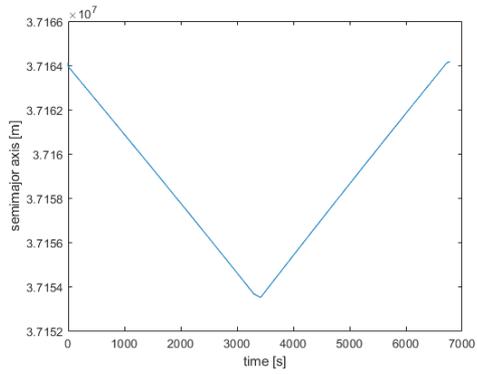
Figure 9.17: Phasing maneuver (Interpolation).

## Interpolation

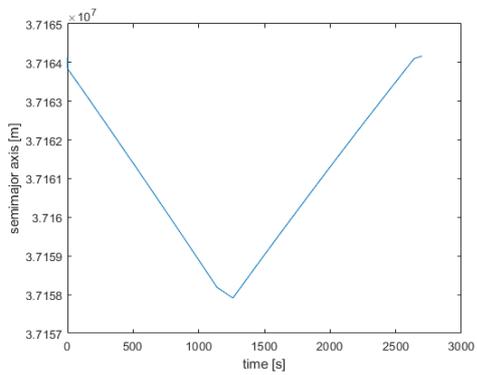
---



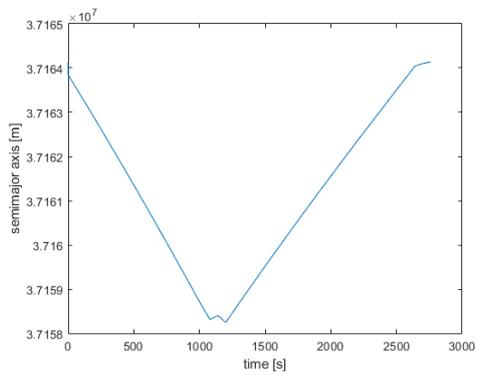
(a) *Time - semi-major axis (Circular and equatorial orbit).*



(b) *Time - semi-major axis (Circular and inclined orbit).*



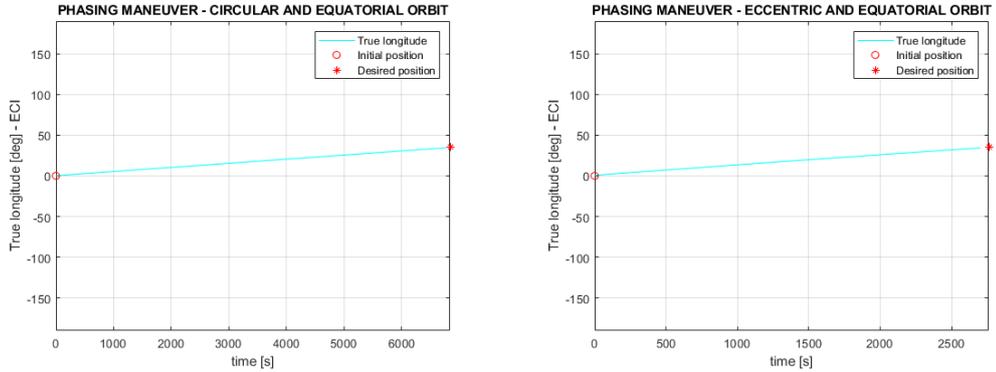
(c) *Time - semi-major axis (Equatorial and eccentric orbit).*



(d) *Time - semi-major axis (Generic orbit).*

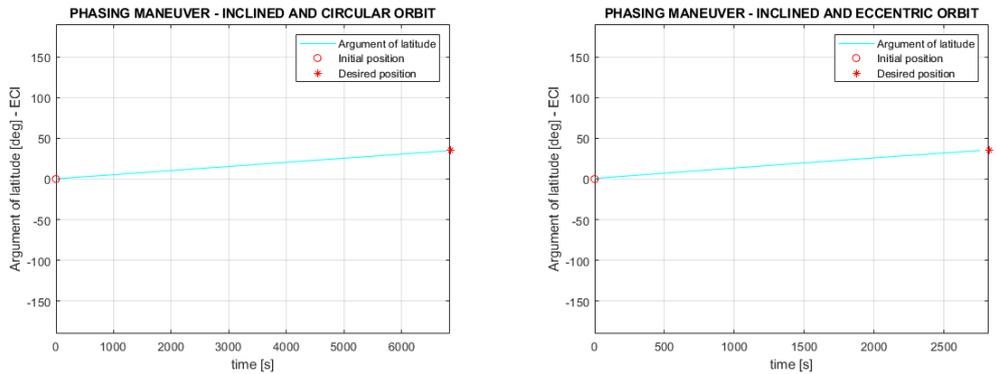
Figure 9.18: Time - semi-major axis (Interpolation).

Considering the control presented in chapter 7.3 ("Generalization"), we check the True Longitude for what concern the Equatorial orbits, while the Argument of Latitude in case of inclined orbit (Fig. 9.19 and 9.20).



(a) *Time - True Longitude (Circular and equatorial orbit).* (b) *Time - True Longitude (Equatorial and eccentric orbit).*

Figure 9.19: Time - True Longitude (Interpolation).



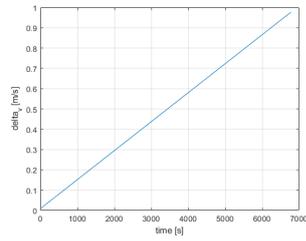
(a) *Time - Argument of Latitude (Circular and inclined orbit).* (b) *Time - Argument of Latitude (Generic orbit).*

Figure 9.20: Time - Argument of Latitude (Interpolation).

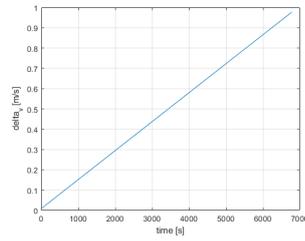
As always the costs are represented by the  $\Delta v$  (Fig. 9.21) and the decrease in  $m_{tot}$  and therefore by the consumption of propellant (Fig. 9.22). Both these parameters, in every case considered, don't show plateaus in their trends. That's because we have no waiting phase and the satellite is never eclipsed during the maneuvers presented (Fig. 9.23). Moreover, as we can see, the cost is practically the same for what concern "Circular and equatorial orbit" and "Circular and inclined orbit" and the same is for "Equatorial and eccentric orbit" and "Generic orbit".

## Interpolation

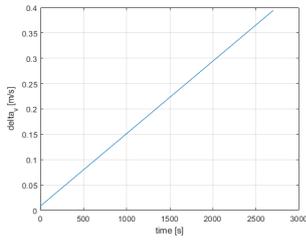
---



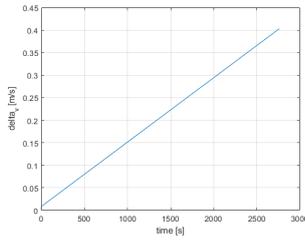
(a) *Time - Delta V (Circular and equatorial orbit).*



(b) *Time - Delta V (Circular and inclined orbit).*

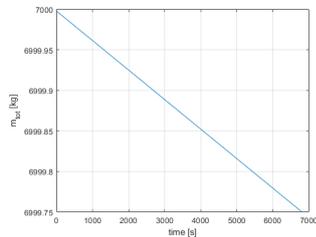


(c) *Time - Delta V (Equatorial and eccentric orbit).*

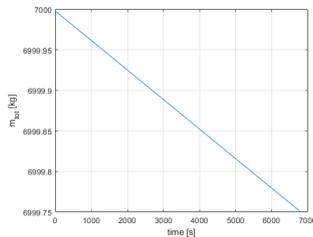


(d) *Time - Delta V (Generic orbit).*

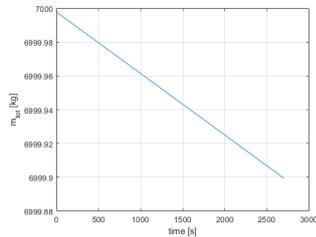
Figure 9.21: Time - Delta V (Interpolation).



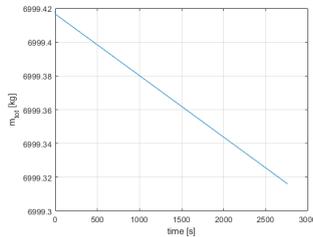
(a) *Time -  $m_{tot}$  (Circular and equatorial orbit).*



(b) *Time -  $m_{tot}$  (Circular and inclined orbit).*

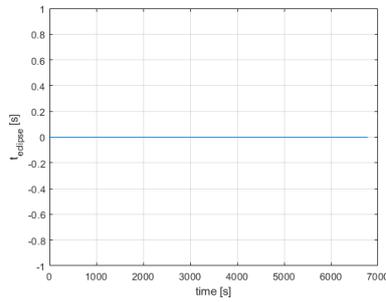


(c) *Time -  $m_{tot}$  (Equatorial and eccentric orbit).*

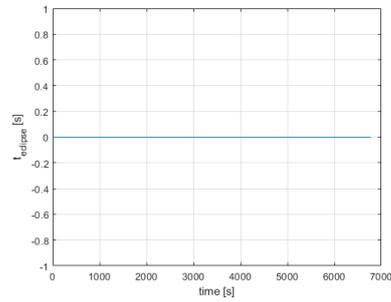


(d) *Time -  $m_{tot}$  (Generic orbit).*

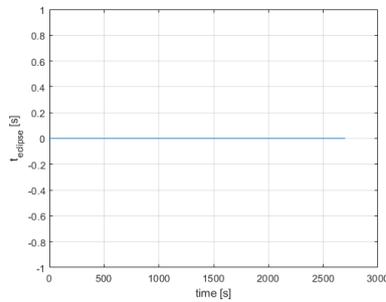
Figure 9.22: Time -  $m_{tot}$  (Interpolation).



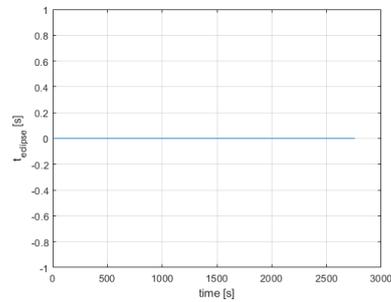
(a) *Time - Eclipse time (Circular and equatorial orbit).*



(b) *Time - Eclipse time (Circular and inclined orbit).*



(c) *Time - Eclipse time (Equatorial and eccentric orbit).*

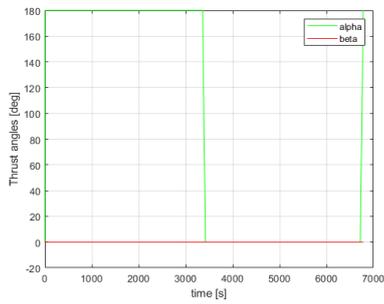


(d) *Time - Eclipse time (Generic orbit).*

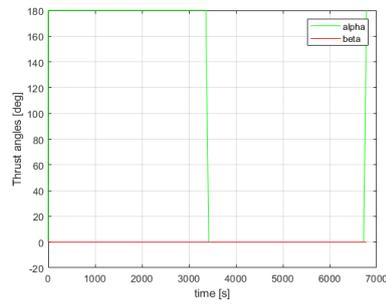
Figure 9.23: Time - Eclipse time (Interpolation).

The  $\alpha$  and  $\beta$  trends are reported in Fig. 9.24, while the thrust weights in Fig. 9.25 and the velocity in Fig. 9.26. The trends are about the same as the previous chapter. As we can see there is no waiting phase and the trends are similar if the orbits are circular or if they are eccentric.

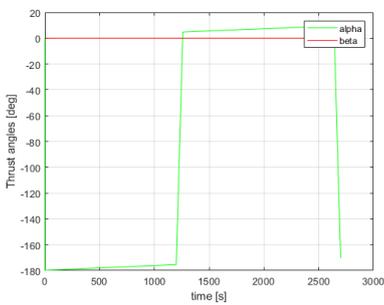
$\alpha \approx \pm\pi$  if we are going to decrease semi-major axis and  $\alpha \approx 0$  if the semi-major axis is about to increase its value, while  $\beta = 0$  in every case. That's because we are not maneuvering outside the orbital plane.



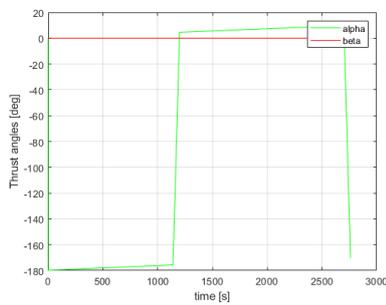
(a) *Time - Thrust angles (Circular and equatorial orbit).*



(b) *Time - Thrust angles (Circular and inclined orbit).*



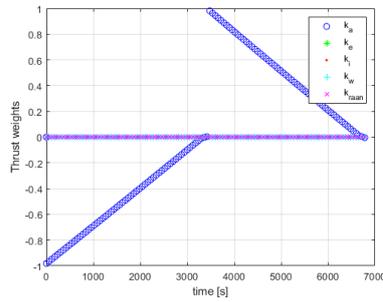
(c) *Time - Thrust angles (Equatorial and eccentric orbit).*



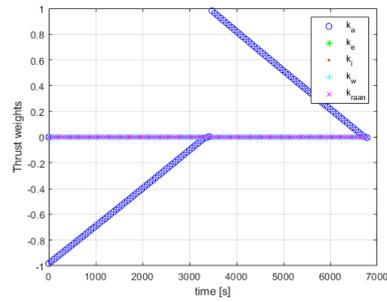
(d) *Time - Thrust angles (Generic orbit).*

Figure 9.24: Time - Thrust angles (Interpolation).

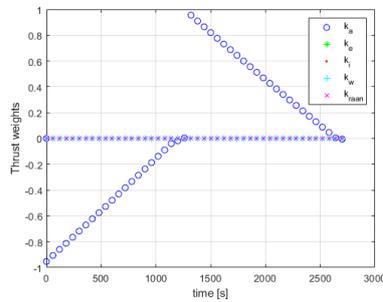
As we can see, the only thrust weights that change is  $k_a$  (the one connected to the semi-major axis). In fact, we are not changing any other orbital parameter. In order to decrease the semi-major axis, its value is negative with linear trend with a positive slope, vice versa as regards the increase of  $a$ .



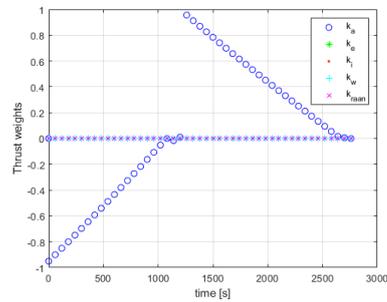
(a) *Time - Thrust weights (Circular and equatorial orbit).*



(b) *Time - Thrust weights (Circular and inclined orbit).*



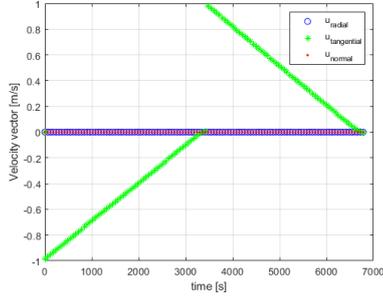
(c) *Time - Thrust weights (Equatorial and eccentric orbit).*



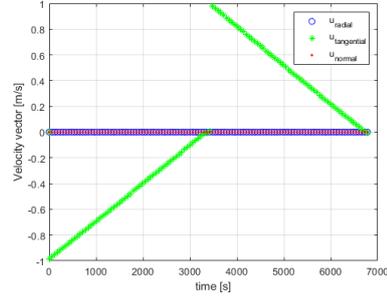
(d) *Time - Thrust weights (Generic orbit).*

Figure 9.25: Time - Thrust weights (Interpolation).

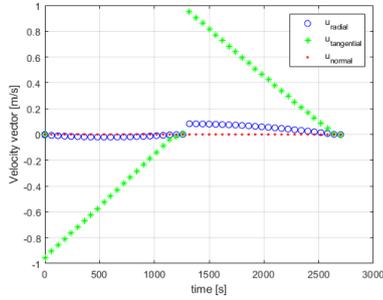
The trend of the velocity is about the same as the thrust weights. In order to decrease the semi-major axis, for what concern circular orbit,  $u_{tangential}$  presents negative values. The tangential component increases its value linearly until it reaches the 0 (when the EOR maneuver is complete). Vice versa when the semi-major axis has to decrease. As for the eccentric orbits, the discussion on the tangential component of speed does not change. What changes is  $u_{radial}$  that changes because of the presence of  $e$ .



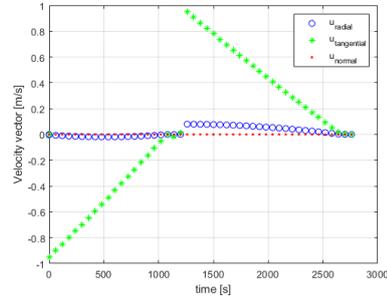
(a) Time - Velocity (Circular and equatorial orbit).



(b) Time - Velocity (Circular and inclined orbit).



(c) Time - Velocity (Equatorial and eccentric orbit).



(d) Time - Velocity (Generic orbit).

Figure 9.26: Time - Velocity (Interpolation).

In tab. 9.4 are reported the results of the simulations described earlier.

RESULTS	Circular & Equatorial	Circular	Equatorial	Generic	
$phasing_{altitude}$	-10.59	-10.59	-6.2	-5.83	km
$true\ longitude_{phasing}$	34.53359	34.53359	34.27574	34.97157	deg
$m_{prop\ cycle}$	0.24692	0.24692	0.09823	0.10051	kg
$t_{f\ cycle}$	1.9	1.9	0.76667	0.78333	hours
$\Delta v_{phasing\ cycle}$	0.97716	0.97716	0.39429	0.40289	m/s
$t_{eclipse}$	0.00	0.00	0.00	0.00	hours

Table 9.4: Results (Interpolation).

### 9.3 Eclipsed positions

Considering Fig. 9.16 there is a range of position the satellite is not able to reach in case of "Circular and Equatorial orbit" and "Equatorial and eccentric orbit". For what concern the circular and equatorial orbit, this range goes from  $143^\circ$  to  $163^\circ$  and

that's because the satellite is in eclipse when it passes through these positions. In fact, by doing a simulation with the same inputs as this case, if we want the satellite to reach a  $true\ longitude_{phasing} > 163^\circ$  (for example:  $true\ longitude_{phasing} = 180^\circ$ ), we obtain the trajectory presented in Fig. 9.27, where in yellow there are the eclipsed positions.



Figure 9.27: Phasing maneuver (Circular and equatorial orbit  $\rightarrow true\ longitude = 180^\circ$ ).

In order to highlight the true longitude eclipsed, see Fig. 9.28.

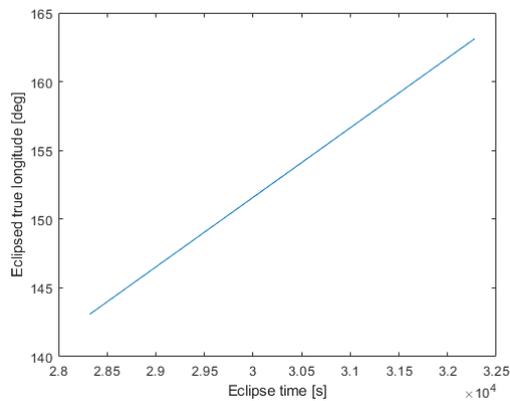


Figure 9.28: Eclipse time - Eclipsed true longitude.

### 9.3.1 Eclipsed true longitude

The problem of eclipsed positions concerns only the checks that are performed on true longitude and argument of latitude. This is because when we calculate these two parameters, the reference system is ECI and therefore the rotation of the Earth

is not considered.

We went for a simulation in which the orbit was only propagated for 365 days in order to save all the positions eclipsed according to the various days. Using the same input of the previous simulation for what concern the initial orbit, we get the results in Fig. 9.29, where  $day1 = 2019 Feb 19 15 : 30 : 00 UTC$ .

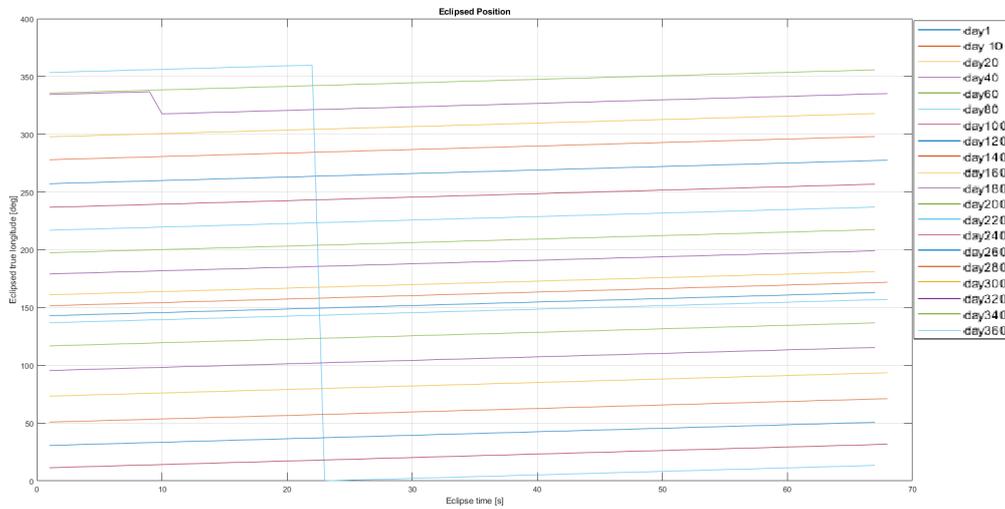


Figure 9.29: Eclipsed position - day.

The Penumbra cone moves about  $0.958^\circ$  per day (mean value). These results have been obtained performing a simulation where we propagate the orbit for about 220 days and saving the  $\Delta angle$  between the first eclipsed position of two consecutive days, as shown in Fig. 9.30.

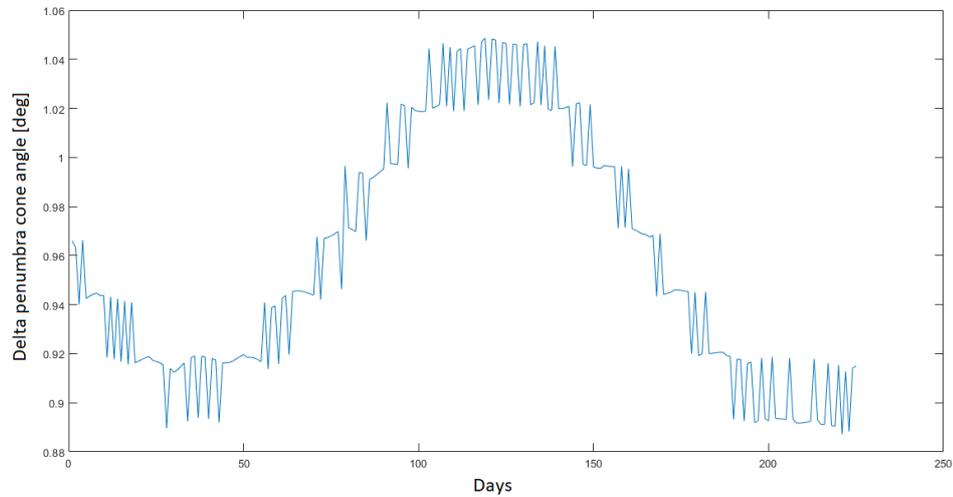


Figure 9.30: Penumbra cone angle - day.

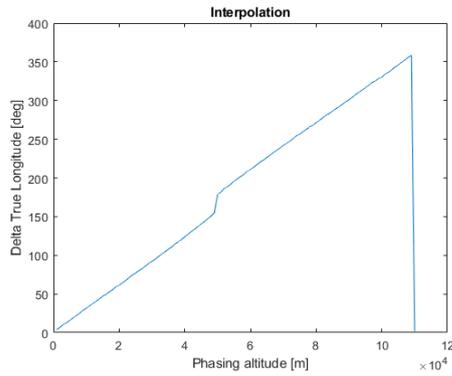
So, if we want the satellite to reach  $TrueLongitude_{phasing} = 155^\circ$  using the same input of the "Circular and equatorial orbit" case, we must wait approximately 13 days.

### 9.3.2 Circular and equatorial orbit - 13 days after

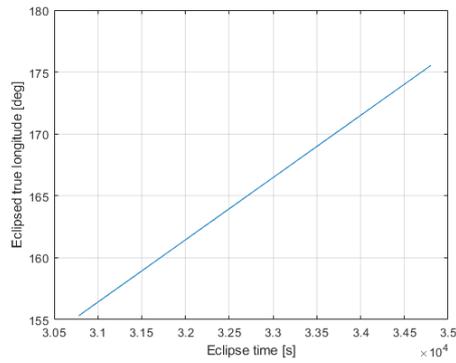
As said before, in order to perform this simulation, we are going to use the same input as the previous one (see tab. 9.3, Circular and equatorial orbit case) changing the date and the position (true longitude) we want the satellite to reach:

- $date = 2019\ Mar\ 4\ 15 : 30 : 00\ UTC;$
- $True\ Longitude_{phasing} = 155^\circ.$

Unlike previous simulations, now we use  $flag_{phasing\_altitude} = 0$  in order to increase the  $phasing_{altitude}$ . In Fig. 9.31 are reported the interpolation and the eclipsed position we get when we are considering the same inputs, but 13 days after. As we can see, the eclipse effectively moves and it's now possible to reach positions that weren't available before.



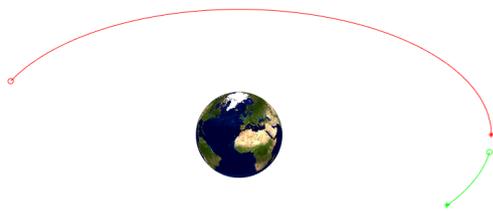
(a) *Interpolation phasing<sub>altitude</sub>/Δposition (Circular and equatorial orbit - 13 days after).*



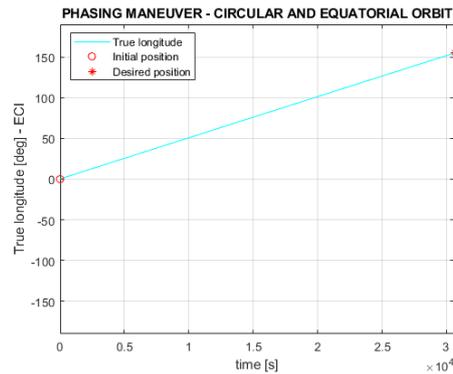
(b) *Eclipse time - Eclipsed true longitude (Circular and equatorial orbit - 13 days after).*

Figure 9.31: Interpolation + Eclipsed true longitude (Circular and equatorial orbit - 13 days after).

In Fig. 9.32 we can see the phasing maneuver and the true longitude in function of time, while in tab. 9.5 the results of this simulation.



(a) *Phasing maneuver (Circular and equatorial orbit - 13 days after).*



(b) *Time - True Longitude (Circular and equatorial orbit - 13 days after).*

Figure 9.32: Phasing maneuver / Time - True Longitude (Circular and equatorial orbit - 13 days after).

$phasing_{altitude}$	49	km
$initial\ true\ longitude$	0	deg
$true\ longitude_{phasing}$	154.6933643678069	deg
$m_{prop\ cycle}$	1.114404147238020	kg
$t_{f_{cycle}}$	8.516666666666667	hr
$\Delta v_{phasing\ cycle}$	4.380350053811925	m/s
$t_{eclipse}$	0.0	days

Table 9.5: Results (Circular and equatorial orbit - 13 days after - Interpolation).



# Chapter 10

## Validation

In order to validate the results MAGNETO gives, STK has been used to simulate two phasing maneuvers.

These maneuvers were simulated using the same inputs both for MAGNETO and STK.

In order to simulate the maneuver as accurate as possible, we set the same engine (constant *Thrust* and  $I_{sp}$ ) and using the results MATLAB gives to use, we impose the STK simulation time to be equal to the one MAGNETO gives us.

The propagator used is "Earth J2" which simulate the perturbing forces given by non-spherical gravity of the Earth.

The simulation has been done also using "Earth Point Mass" propagator to control the percentage difference between the results of the two simulations with results given by MATLAB.

### 10.0.1 STK

Systems Tool Kit (or Satellite Tool Kit) [11], known as STK, is a software from Analytical Graphics, Inc. It's able to perform complex analyses of space platforms and share results in one integrated environment. This program has been developed since 1989 as a commercial tool in order to find solutions for what concern the involving Earth-orbiting satellites problems.

In AGI's lists of clients there are organizations such as NASA, ESA, CNES, DLR, Boeing, JAXA, ISRO, Lockheed Martin, Northrop Grumman, Airbus, DOD, and Civil Air Patrol.

## 10.1 Simulation 1

The inputs used for the first simulation are shown in tab. 10.1 (MATLAB) and in Fig. 10.1 (STK). In this simulation, instead of considering the phasing maneuver with interpolation logic, we imposed the  $phasing_{altitude} = 100 \text{ km}$ , and we saved the results that are going to be shown.

$date$	2019 Feb 19 15 : 30 : 00	UTC;
$a_i$	$32164.1363 \cdot 10^3$	[m]
$i_i$	0	[deg]
$e_i$	0	
$\Omega_i$	0	[deg]
$\omega_i$	0	[deg]
$\nu_i$	0	[deg]
$phasing_{altitude}$	100	[km]
$a_f$	$a_i + phasing_{altitude}$	[m]
$i_f$	0	[deg]
$e_f$	0	
$\Omega_f$	0	[deg]
$\omega_f$	0	[deg]
$\nu_f$	0	[deg]
$thruster$	1	
$thrust \text{ per thruster}$	$1000 \cdot 10^{-3}$	[N]
$I_{SP}$	2800	[sec]
$True \text{ longitude}$	169	deg

Table 10.1: Input first simulation for Validation.

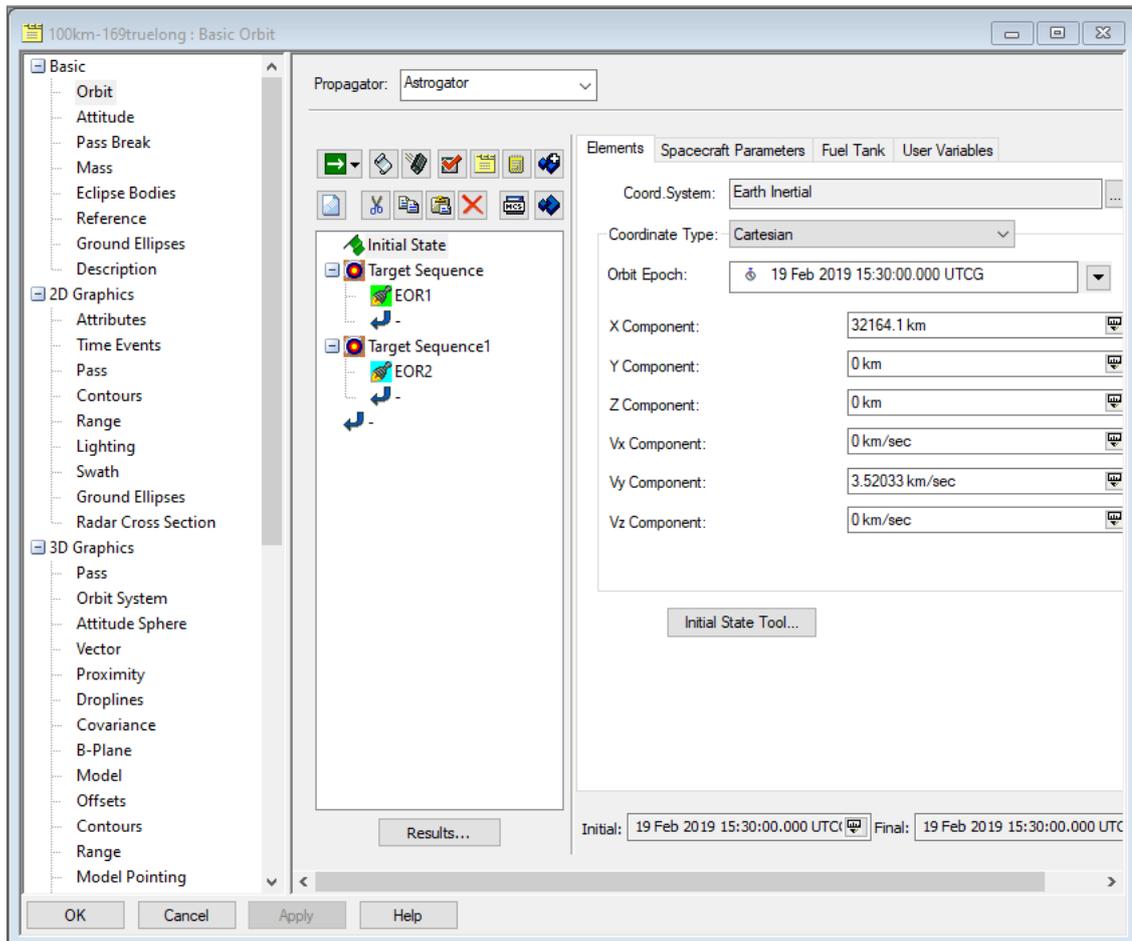
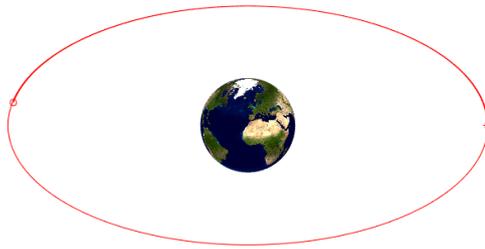


Figure 10.1: Input - STK.

In Fig. 10.2 there is the trajectory followed by the satellite. The coordinates presented are in ECI frame with an overview on initial and final position. For what concern MAGNETO (a), the initial position is highlighted with the "\*", while the final position with the "o", while, for what concern STK we can see the trajectory followed by the satellite and in (b) there is the initial position and in (c) the final position.



(a) *Phasing maneuver in ECI (MATLAB 1).*



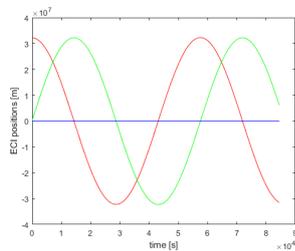
(b) *Phasing maneuver in ECI (initial position - STK 1).*



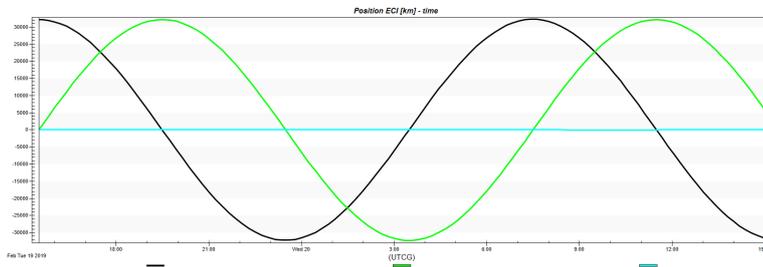
(c) *Phasing maneuver in ECI (final position - STK 1).*

Figure 10.2: Phasing maneuver (Simulation 1).

In Fig. 10.3 and in 10.4 there are the trends, respectively of the ECI and of the ECEF coordinates. As we can see the trends are the same.

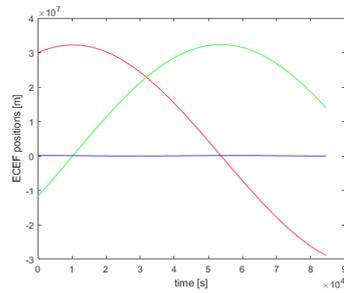


(a) *Time - ECI Position (MATLAB 1).*

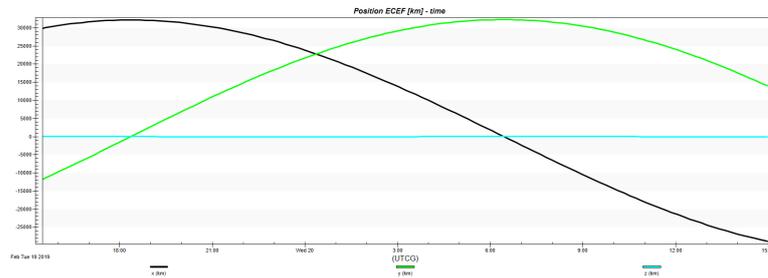


(b) *Time - ECI Position (STK 1).*

Figure 10.3: Time - ECI Position (Simulation 1).



(a) Time - ECEF Position (MATLAB 1).



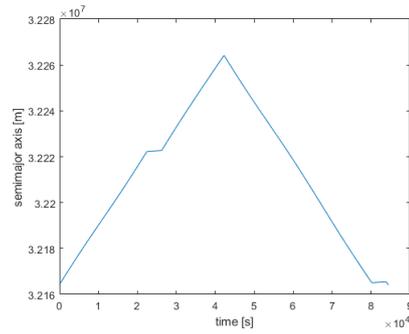
(b) Time - ECEF Position (STK 1).

Figure 10.4: Time - ECEF Position (Simulation 1).

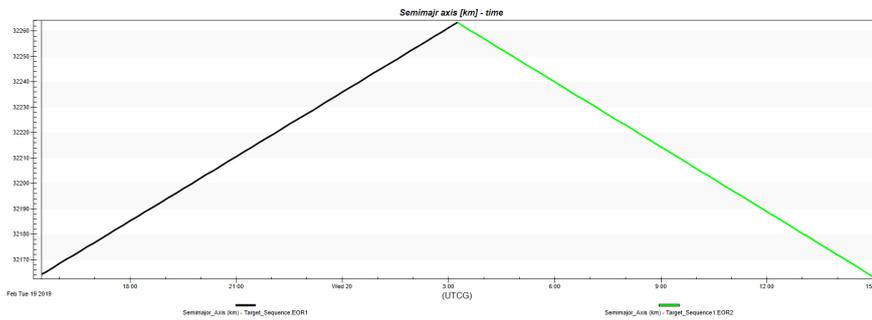
For what concern the coordinates, we have:

- MAGNETO (red) → x-coordinate (both ECI and ECEF);
- MAGNETO (green) → y-coordinate;
- MAGNETO (blue) → z-coordinate;
- STK (black) → x-coordinate;
- STK (green) → y-coordinate;
- STK (cyan) → z-coordinate;

In Fig. 10.5 there is the semi-major axis trend in function of the time. For what concern the figures relative to STK simulation, in black we have the parameters change due to the first EOR, while in green the ones of the second EOR.



(a) Time - semi-major axis (MATLAB 1).

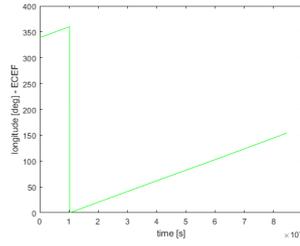


(b) Time - semi-major axis (STK 1).

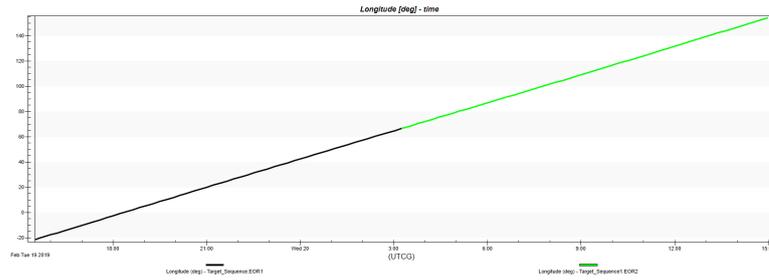
Figure 10.5: Time - semi-major axis (Simulation 1).

As for the semi-major axis, also for the Longitude (Fig. 10.6) and the True longitude (10.7) the trends are the same. Both these parameters increase their values until the True Longitude reaches its desired values (the controls are the same reported in earlier chapter "Generalization").

10.1 – Simulation 1

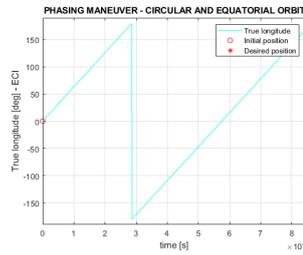


(a) Time - Longitude (MATLAB 1).

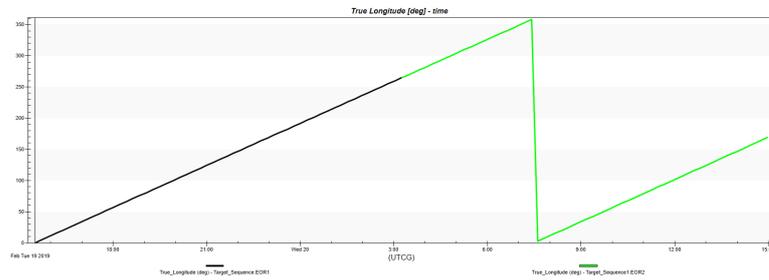


(b) Time - Longitude (STK 1).

Figure 10.6: Time - Longitude (Simulation 1).



(a) Time - True Longitude (MATLAB 1).

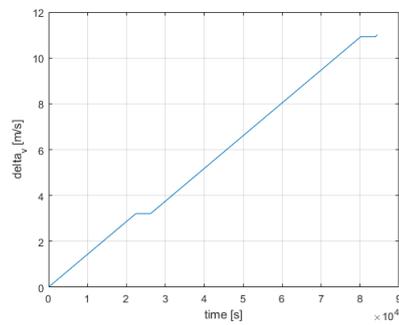


(b) Time - True Longitude (STK 1).

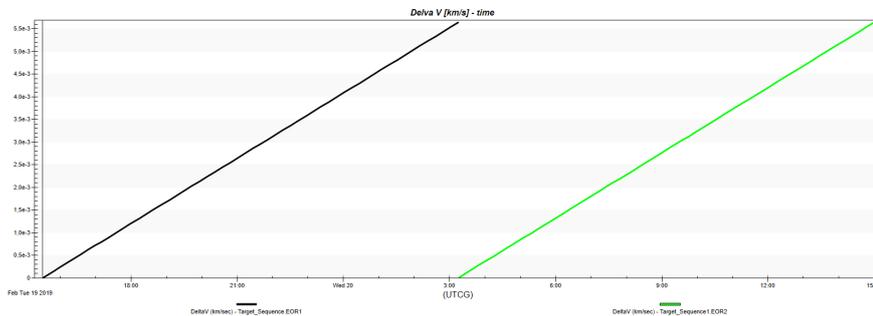
Figure 10.7: Time - True Longitude (Simulation 1).

For what concern the  $\Delta V$  (Fig. 10.8), the STK simulation has been made as the

union of two consecutive EOR, such as MAGNETO, but the program recognized the phasing maneuver as two different maneuvers. So the  $\Delta V$ , once the first EOR is over, restarts from 0. For what concern the sum of the two value at the end of the simulation, it's identical to the  $\Delta V$  calculated in MATLAB (see tab. 10.2). Instead, the propellant mass (Fig. 10.9) doesn't present this problem.

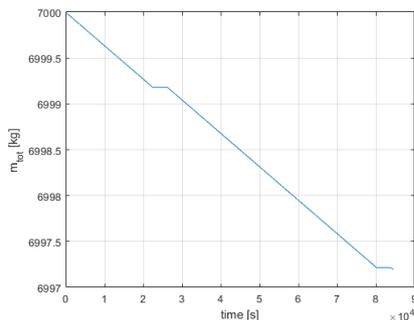


(a) *Time -  $\Delta V$  (MATLAB 1).*

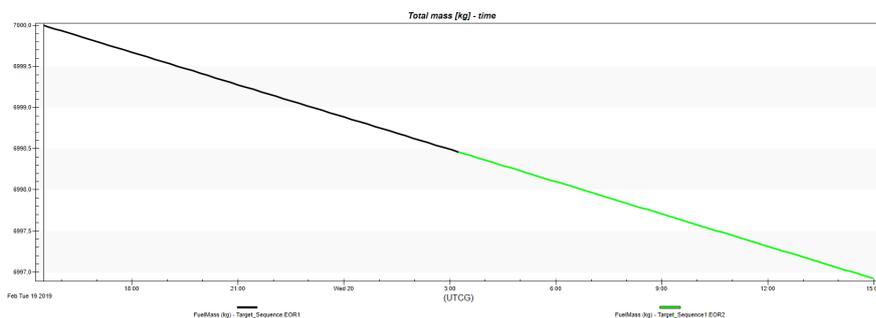


(b) *Time -  $\Delta V$  (STK 1).*

Figure 10.8: Time -  $\Delta V$  (Simulation 1).



(a) Time -  $m_{tot}$  (MATLAB 1).



(b) Time -  $m_{tot}$  (STK 1).

Figure 10.9: Time -  $m_{tot}$  (Simulation 1).

Tab. 10.2 report the results obtained by the simulations performed by MAGNETO and STK ("Earth J2" & "Earth Point Mass").

As we can see, the results are very similar for what concern every parameter considered and that's highlighted by the tab. 10.3.

RESULTS	MATLAB	STK (Earth J2)	STK (Earth Point Mass)	
$ECIPosition_x$	$-3.1517 \cdot 10^7$	$-3.1444 \cdot 10^7$	$-3.1388 \cdot 10^7$	m
$ECIPosition_y$	$6.1299 \cdot 10^6$	$6.5143 \cdot 10^6$	$6.7997 \cdot 10^6$	m
$ECIPosition_z$	636	325.237	-361.236	m
$ECEFPosition_x$	$-2.8914 \cdot 10^7$	$-2.8860 \cdot 10^7$	$-2.8810 \cdot 10^7$	m
$ECEFPosition_y$	$1.3959 \cdot 10^6$	$1.4081 \cdot 10^6$	$1.4193 \cdot 10^6$	m
$ECEFPosition_z$	$-5.7216 \cdot 10^4$	$-5.8053 \cdot 10^4$	$-5.7991 \cdot 10^4$	m
$Truelongitude_{phasing}$	168.9934	168.296	167.77	deg
$longitude$	154.2304	153.992	153.774	deg
$m_{tot\ cycle}$	6997.1965	6996.93	6996.93	kg
$t_{f\ cycle}$	84540	84540	84540	sec
$\Delta v_{phasing\ cycle}$	11.0079	11.2296	11.1910	m/s

Table 10.2: Results - simulation 1.

The percentage differences between MATLAB simulation and STK (both using "Earth J2" and "Earth Point Mass") are very little, except for what concern the z-position in ECI frame. Even if in this case the percentage difference is huge ( $\approx 49\%$  in "Earth J2" case and  $\approx 157\%$  in "Earth Point Mass" case), the result is not so problematic. The high percentage is due to the fact that the x-coordinate value is very small (6 or 7 order of magnitude) in respect of all the other coordinates values.

Percentage difference	MATLAB - STK (Earth J2)	MATLAB - STK (Earth Point Mass)
$ECIPosition_x$	-0.2307%	-0.4096%
$ECIPosition_y$	6.2702%	10.9266%
$ECIPosition_z$	-48.8624%	-156.798%
$ECEFPosition_x$	-0.1868%	-0.3600%
$ECEFPosition_y$	0.8739%	1.6728%
$ECEFPosition_z$	1.4622%	1.3550%
$Truelongitude_{phasing}$	-0.4127%	-0.7239%
$longitude$	-0.1546%	-0.2959%
$m_{tot\ cycle}$	-0.0038%	-0.00381%
$tf_{cycle}$	0%	0%
$\Delta v_{phasing\ cycle}$	2.0137%	1.6631%

Table 10.3: Percentage difference - Simulation 1.

## 10.2 Simulation 2

As the same way as the previous case, in tab. 10.4 are shown the inputs used for MATLAB simulation, while in Fig. 10.10 the STK ones. Differently from the previous simulations, in this case the  $phasing_{altitude}$  is an output for MATLAB, in fact we followed the "interpolation" logic, while, for what concern the STK's simulation, it's an input.

<i>date</i>	2019 Feb 19 15 : 30 : 00	UTC;
<i>a<sub>i</sub></i>	$32164.1363 \cdot 10^3$	[m]
<i>i<sub>i</sub></i>	1	[deg]
<i>e<sub>i</sub></i>	0.01	
<i>Ω<sub>i</sub></i>	0	[deg]
<i>ω<sub>i</sub></i>	0	[deg]
<i>ν<sub>i</sub></i>	0	[deg]
<i>a<sub>f</sub></i>	<i>a<sub>i</sub> + phasing<sub>attitude</sub></i>	[m]
<i>i<sub>f</sub></i>	1	[deg]
<i>e<sub>f</sub></i>	0.01	
<i>Ω<sub>f</sub></i>	0	[deg]
<i>ω<sub>f</sub></i>	0	[deg]
<i>ν<sub>f</sub></i>	0	[deg]
<i>thruster</i>	1	
<i>thrust per thruster</i>	$1000 \cdot 10^{-3}$	[N]
<i>I<sub>SP</sub></i>	2800	[sec]
<i>True longitude</i>	350	deg

Table 10.4: Input simulation for Validation- 2.

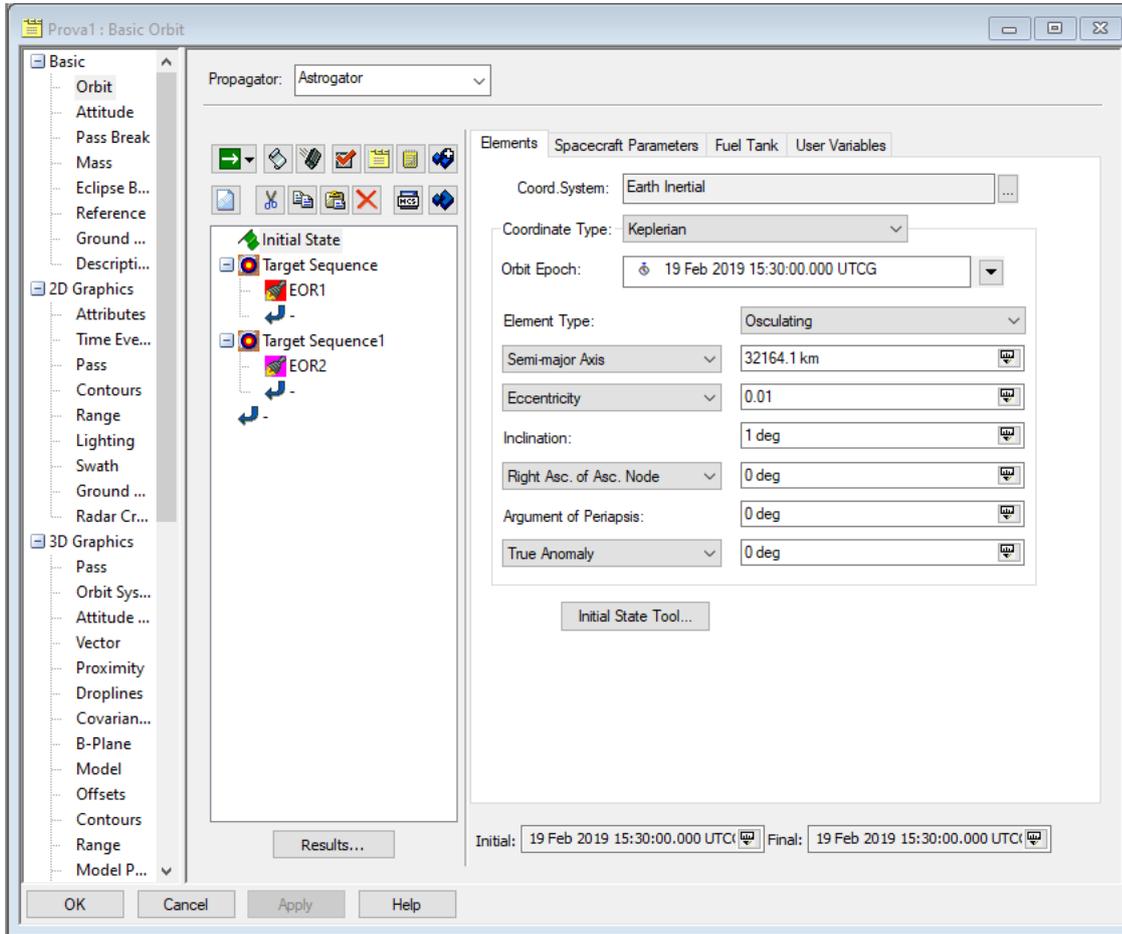
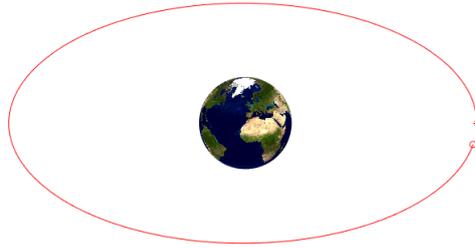
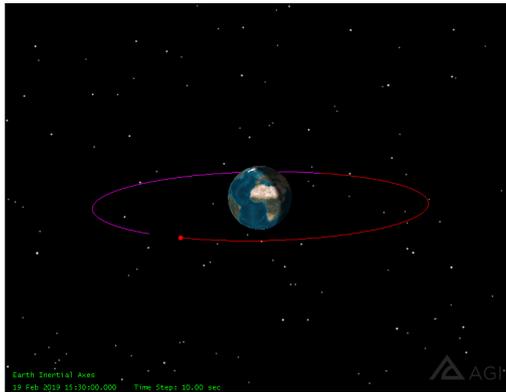


Figure 10.10: Input - STK.

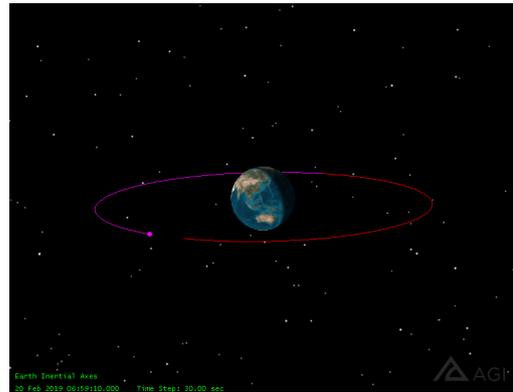
In Fig. 10.11 there is the trajectory followed by the satellite. The coordinates presented are in ECI frame with an overview on initial and final position. For what concern MAGNETO, the initial position is highlighted with the "\*", while the final position with the "o".



(a) *Phasing maneuver in ECI (MATLAB 2).*



(b) *Phasing maneuver in ECI (initial position - STK 2).*

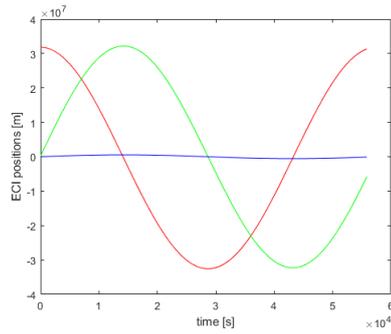


(c) *Phasing maneuver in ECI (final position - STK 2).*

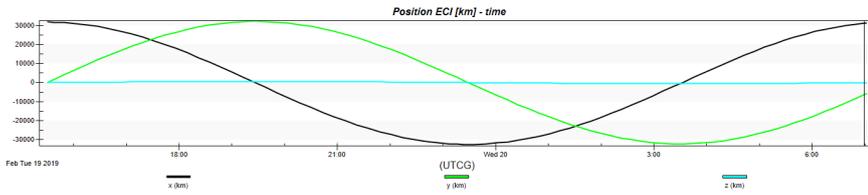
Figure 10.11: Phasing maneuver (Simulation 2).

In Fig. 10.12 and in 10.13 there are the trends, respectively of the ECI and of the ECEF coordinates. As we can see, the trends are the same.

Validation

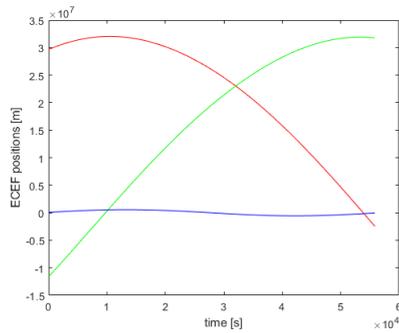


(a) Time - ECI Position (MATLAB 2).

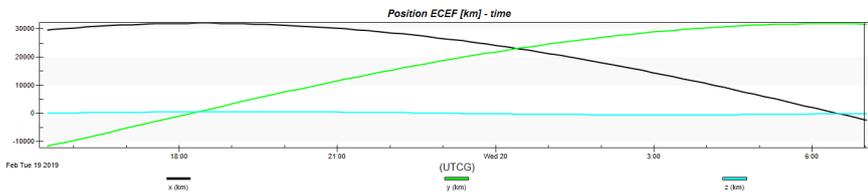


(b) Time - ECI Position (STK 2).

Figure 10.12: Time - ECI Position (Simulation 2).



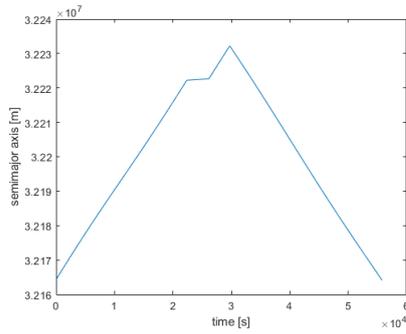
(a) Time - ECEF Position (MATLAB 2).



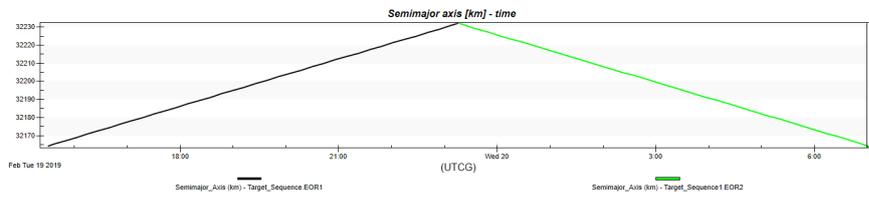
(b) Time - ECEF Position (STK 2).

Figure 10.13: Time - ECEF Position (Simulation 2).

In Fig. 10.14 there is the semi-major axis trend in function of the time. The  $phasing_{altitude}$  necessary in order to reach  $True\ longitude = 350^\circ$  is  $67.98\ km$ .



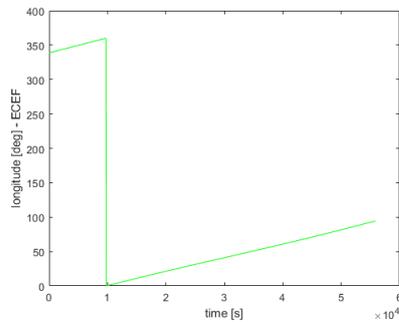
(a) Time - semi-major axis (MATLAB 2).



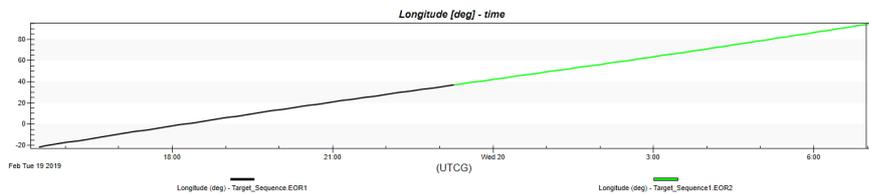
(b) Time - semi-major axis (STK 2).

Figure 10.14: Time - semi-major axis (Simulation 2).

As reported in Fig. 10.15 and in 10.16, imposing that value of *phasing<sub>altitude</sub>* allow us to obtain practically the same results for what concern both the simulations.

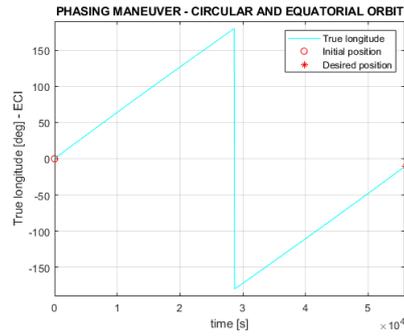


(a) Time - Longitude (MATLAB 2).

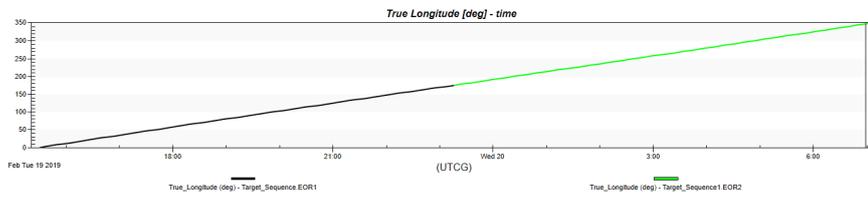


(b) Time - Longitude (STK 2).

Figure 10.15: Time - Longitude (Simulation 2).



(a) *Time - True Longitude (MATLAB 2).*

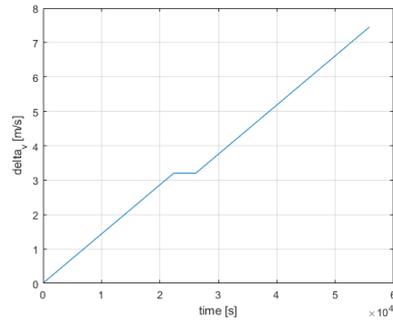


(b) *Time - True Longitude (STK 2).*

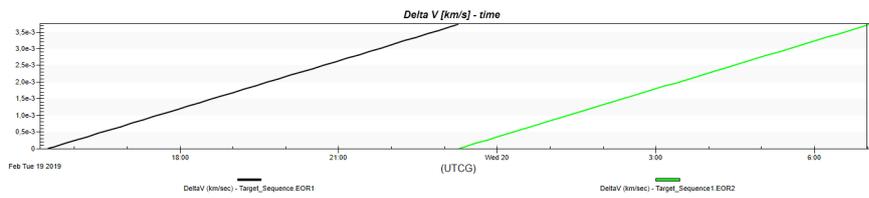
Figure 10.16: Time - True Longitude (Simulation 2).

As for the costs ( $\Delta V$  in Fig. 10.17 and  $m_{tot}$  in Fig. 10.18), the discussion made for the previous simulation is still valid. The  $\Delta V$  is seen by stk as two different maneuvers, but the sum is equal to the result given by MATLAB, while the propellant mass doesn't present this problem.

10.2 – Simulation 2

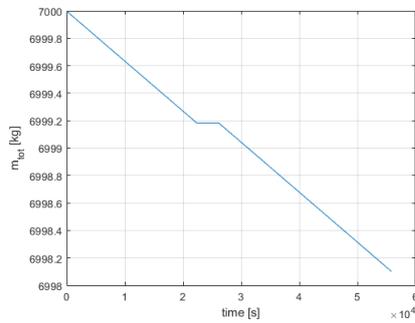


(a) Time -  $\Delta V$  (MATLAB 2).

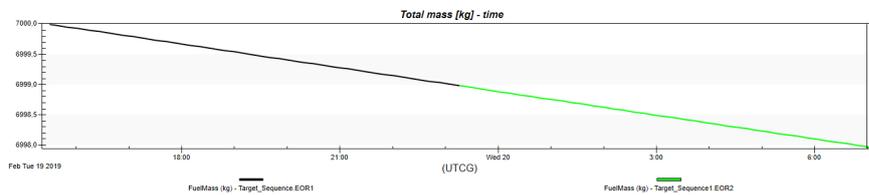


(b) Time -  $\Delta V$  (STK 2).

Figure 10.17: Time -  $\Delta V$  (Simulation 2).



(a) Time -  $m_{tot}$  (MATLAB 2).



(b) Time -  $m_{tot}$  (STK 2).

Figure 10.18: Time -  $m_{tot}$  (Simulation 2).

Tab. 10.5 reports the results obtained.

RESULTS	MATLAB	STK (Earth J2)	STK (Earth Point Mass)	
$ECIPosition_x$	$3.1334 \cdot 10^7$	$3.12795 \cdot 10^7$	$3.13319 \cdot 10^7$	m
$ECIPosition_y$	$-5.7441 \cdot 10^6$	$-6.03671 \cdot 10^6$	$-5.75122 \cdot 10^6$	m
$ECIPosition_z$	$-1.0047 \cdot 10^5$	$-1.032 \cdot 10^5$	$-9.84361 \cdot 10^4$	m
$ECEFPosition_x$	$-2.4579 \cdot 10^6$	$-2.36761 \cdot 10^6$	$-2.45068 \cdot 10^6$	m
$ECEFPosition_y$	$3.1761 \cdot 10^7$	$3.17802 \cdot 10^7$	$3.17611 \cdot 10^7$	m
$ECEFPosition_z$	$-4.2964 \cdot 10^4$	$-4.57842 \cdot 10^4$	$-4.09302 \cdot 10^4$	m
$Truelongitude_{phasing}$	349.6102	349.075	349.597	deg
$longitude$	94.4252	94.2622	94.4122	deg
$m_{tot\ cycle}$	6998.1033	6997.97	6997.96	kg
$t_{f_{cycle}}$	55920	55920	55920	sec
$\Delta v_{phasing\ cycle}$	7.44513	7.43063	7.511126	m/s

Table 10.5: Results - simulation 2.

Unlike previous simulation, in this case the results are more similar in the case we don't take count of the  $J_2$  perturbations. In fact, in tab. 10.6 there are the percentage differences of STK solutions using "Earth J2" and "Earth Point Mass" as propagator. However, the results obtained are good. In fact, as the same way as the previous simulation, we can see that the percentage differences are acceptable for both the propagator used.

Percentage difference	MATLAB - STK (Earth J2)	MATLAB - STK (Earth Point Mass)
$ECIPosition_x$	-0.17262%	-0.00538%
$ECIPosition_y$	5.093263%	0.12316%
$ECIPosition_z$	2.713867%	-2.02759%
$ECEFPosition_x$	-3.67537%	-0.29572%
$ECEFPosition_y$	0.060723%	0.00587%
$ECEFPosition_z$	6.563176%	-4.73456%
$Truelongitude_{phasing}$	-0.15308%	-0.00378%
$longitude$	-0.17267%	-0.01382%
$m_{tot\ cycle}$	-0.00191%	-0.00205%
$t_{f_{cycle}}$	0%	0%
$\Delta v_{phasing\ cycle}$	-0.05978%	0.826126%

Table 10.6: Percentage difference - Simulation 2.

# Chapter 11

## Conclusions

MAGNETO is a software able to perform a preliminary analysis and design of an electric platform. The main goal of this thesis is to add new simulation capabilities to MAGNETO, allowing the possibility to perform a phasing maneuver. The first activities concerned the introduction of a time-dependent reference system through a conversion method, from ECI (Earth-Centered Inertial) to ECEF (Earth-Centered, Earth-Fixed).

The input given to MAGNETO are the orbital parameters, the date, the thruster performance, the satellite dimensions, the *phasing<sub>altitude</sub>* and the longitude we want the satellite to reach.

The output are the trajectory (in ECI and ECEF coordinates), the orbital parameters changes, the  $\Delta V$ ,  $m_{prop}$ , latitude, longitude, the Thrust weights and angles and the velocity components during all the maneuver.

Initially, the phasing maneuver was set according to the chemical propulsion methodology: first, the satellite performs an EOR-like maneuver (Electric Orbit Raising), second, it waits on the phasing orbit and, in the end, it performs another EOR-like maneuver to get back to the initial orbit.

Subsequently, the maneuver was modified in such a way to make it available for every typology of orbits. In this way, MAGNETO controls the longitude in case of geostationary orbit, the true longitude ( $l = \nu + \omega + \Omega$ ) in case of equatorial orbit and the argument of latitude ( $u = \nu + \omega$ ) if the orbit considered is inclined.

Then, the maneuver has been modified in order to complete the phasing maneuver without the waiting phase. In this way, it was composed by two consecutive EORs, one ascending and one descending, or vice versa. In order to do so, the phasing altitude became a variable to be determined. To obtain the desired altitude value, MAGNETO performs the EORs increasing or decreasing the altitude (in function of a flag imposed by the user) step by step. Once all the positions on the orbit are covered, the program interpolates the data obtained and saves them. In the end, MAGNETO finds all the phasing altitudes available to reach the desired position and performs the maneuver.

Using this logic, a minimum result is obtained in terms of time, but the maneuver results more expensive in terms of  $\Delta V$  and  $m_{prop}$ , at least for what concern the geostationary orbit. Performing a phasing maneuver with  $\Delta longitude = 35^\circ$ , it takes 5 days, while the same maneuver with the waiting phase takes 26 days. For what concerns the cost, obviously it is higher than the one we get with the first maneuvre logic because the thrust is applied continuously.

For what concern any other types of orbit, the maneuver is faster and often less expensive. In fact, using fixed values of  $phasing_{altitude}$ , it is possible that the satellite would have passed the position we wanted it to reach. If the  $\Delta true longitude = 15^\circ$ , once the satellite performs the first EOR, it has to wait in the waiting phase for almost a complete orbit.

This problem disappears using the interpolation logic.

In the end, the results obtained have been validated using STK (Systems or Satellite Tool Kit). The percentage differences result very low ( $< 1\%$ ) for what concern longitude, true longitude,  $\Delta V$  and  $m_{prop}$  with a maximum of 5 – 6% for the coordinates (both in ECI and ECEF frame). The only exception is the z-coordinates in ECI frame of the first simulation in chapter 10. In that case the percentage differences result  $\approx 49\%$  in case of "Earth J2" propagator and  $\approx 157\%$  if the propagator is "Earth point mass". This is due to the fact that the order of magnitude of this value is about 6 – 7 lower than all the other coordinates. In fact, the value of the difference is about 300 meters, which is almost negligible compared to the value of all the other coordinates.

The results we obtained are acceptable but they can be improved in the future. So, even if they are derived with an error of few percentage points, a further improved could reduce the error in particular on maneuver time and consequently the  $m_{prop}$  used and the  $\Delta V$ .

The trajectory represents a sub-optimal propagation due to the direct integration of the equations of motion with a pre-determined weighed method. The derived Thrust Weights, calculated each integration step during the maneuver, are the results of a normalization between the orbital parameter we have and the one we want.

In order to optimize them, we could implement MGNETO with a genetic code in order to calculate the thrust weights. The implementation of this approach would allow to optimize the maneuver in terms of either  $m_{prop}$  or  $\Delta V$  or maneuver time. We could also implement a variable thrust for what concern the engine. In this way, we could improve the reliability of MAGNETO, even though the operative points are limited.

# Bibliography

- [1] Christopher A. Paissoni, Jasmine Rimani, Nicole Viola, Tommaso Andreussi, Geo telecommunication satellite: new opportunities enabled by a 20kW class Hall thruster, 21-25 October 2019.
- [2] J. Rimani, C.A. Paissoni, N. Viola, G. Saccoccia, J.G. del Amo, “Multidisciplinary Mission and System Design Tool for a Reusable Electric Propulsion Space Tug”, *11th IAA Symposium on the Future of Space Exploration*, Torino, Italy, June 2019.
- [3] Martina Mammarella, A Comprehensive Modeling Framework for Integrated Mission Analysis and Design of a Reusable Electric Space Tug, November 2018.
- [4] Roger R. Bate, Donald D. Mueller, Jerry E. White, *Fundamentals of Astrodynamics*, 1971.
- [5] Mohinder S. Grewal, Lawrence R. Weill, Angus P. Andrews, *Global Positioning Systems, Inertial Navigation, and Integration*, 2001.
- [6] AGI site, url: <https://www.agi.com/home>
- [7] Lorenzo Casalino, *Space Propulsion notes*, 2019.
- [8] SITael site, url: <https://www.sitael.com/space/advanced-propulsion/electric-propulsion/hall-effect-thrusters/>
- [9] NAIF site, url: <https://naif.jpl.nasa.gov/naif/>
- [10] Modified Equinoctial Orbital Elements,  
url: [https://spsweb.fltops.jpl.nasa.gov/portaldataops/mpg/MPG\\_Docs/Source%20Docs/EquinoctialElements-modified.pdf](https://spsweb.fltops.jpl.nasa.gov/portaldataops/mpg/MPG_Docs/Source%20Docs/EquinoctialElements-modified.pdf).
- [11] AGI site, product: STK, url: <https://www.agi.com/products>, 2020.
- [12] Carlos R. Ortiz Longo, Steven L. Rickman, *Method for the Calculation of Spacecraft Umbra and Penumbra Shadow Terminator Points*, April 1995
- [13] url: <https://adcsforbeginners.wordpress.com/tag/earth-centred-inertial-frame/>
- [14] MATLAB function `dcmece2ecef`,  
url: <https://it.mathworks.com/help/aerotbx/ug/dcmeci2ecef.html>
- [15] MATLAB function `ecef2geodetic`,  
url: <https://it.mathworks.com/help/map/ref/ecef2geodetic.html>
- [16] MATLAB function `ecef2lla`,  
url: <https://it.mathworks.com/help/aerotbx/ug/ecef2lla.html>

- [17] David A. Vallado, *Fundamentals of Astrodynamics and applications*, 1997.
- [18] Nasa, Johannes Kepler, <https://www.nasa.gov/kepler/education/johannes>
- [19] url: <https://www.newton.ac.uk/about/isaac-newton/life>
- [20] Transformation between the International Terrestrial Reference System and the Geocentric Celestial Reference System (IERS Conventions (2010) Gérard Petit and Brian Luzum).
- [21] Oliver Montenbruck, Eberhard Gill, *Satellite Orbits, Models, Methods, and Applications*, Springer, 2000.
- [22] Howard D. Curtis, *Orbital Mechanics for Engineering Students*, Elsevier, 2005.
- [23] Explanatory supplement to the *Astronomical almanac*, Edited by P. Kenneth Seidelmann, University Science Books, 1992.
- [24] An investigation into satellite drag modeling performance, Stephen Russell Mance.
- [25] Solar radiation pressure, drag and gravitational effects on a dust particle in Earth orbit, Bindu Bhargavi Jagannatha.
- [26] Sven Schäff, Andreas Wiegand, *Advanced Electric Orbit-Raising Optimization for Operational Purpose*.
- [27] Dayne G. Cook, *Solar radiation pressure modeling issues for high altitude satellites*, March 2001.
- [28] A. Ruggiero, P. Pergola, S. Marcuccio, M. Andrenucci, *Low-Thrust Maneuvers for the Efficient Correction of Orbital Elements*, September 11 – 15, 2011.

# Ringraziamenti

Desidero ringraziare innanzitutto mio padre, mia madre e le mie sorelle per aver sempre creduto in me.

Vorrei ringraziare Alessandra Gravili per esserci stata sempre per me e avermi sopportato e supportato.

Ringrazio i miei cugini Giorgio, Luigi, Marco, Luca, Matteo, Elisa e Claudia per tutti i momenti passati insieme.

Ringrazio tutta la mia famiglia.

Ringrazio Federico Di Fazio, Matteo Ripa e Leonardo Barraco per avermi reso la vita semplice a Torino. Allo stesso modo ringrazio Gianluca Monaco, Cataldo Moschetti, Francesco Minosi, Federica Nasr, Filippo Sanangelantoni e Daniele Nitti. Ringrazio i miei compagni di corso Ilario Cavallera, Federico Favalli, Daniel Giusa, Paolo Aime, Donato Chirulli, Nicolò Benigno ed Emanuel Loprete.

Ringrazio i miei amici "di giù" Anselmo Mirante, Marco De Vincenzis, Matteo Gelfusa, Claudia Alborino, Erica Zonfrilli, Giulia Zonfrilli, Francesco Rampini, Alessio Panetta e Valeria Di Schiavi.

Vorrei ringraziare infine la Prof.ssa Viola per avermi permesso di eseguire questo lavoro e Jasmine Rimani e Christopher Andrea Paissoni per avermi seguito in questi ultimi mesi.