

Politecnico di Milano

Project: Orbital Mechanics

Academic year 2023-2024

Professor: Camilla Colombo

Authors: Group 2346 - Apollo credici			
Person Code	Matriculation number	Surname	Name
10726940	244805	Bachini	Pier Francesco A.
10721571	252781	Belletti	Stefano
10704282	242297	Giardini	Chiara
10979819	242895	Gómez Sánchez	Carolina



**POLITECNICO
MILANO 1863**

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Submission date: Friday 16th February, 2024

Contents

1	Interplanetary Explorer Mission	1
1.1	Introduction	1
1.2	Problem-solving Strategy	1
1.2.1	Synodic period	1
1.2.2	Contour plots analysis	2
1.3	Optimizations	3
1.3.1	Grid search	3
1.3.2	Genetic Algorithm (GA)	4
1.3.3	Multi Start	4
1.4	Conclusions	4
2	Planetary Explorer Mission	6
2.1	Introduction	6
2.2	Orbit characterization	6
2.3	Perturbations	6
2.3.1	Zonal harmonics: J_2 effect	7
2.3.2	Air drag	7
2.4	Ground track	7
2.4.1	Repeating ground track	7
2.5	Evolution of the orbit	8
2.6	Filtering	10
2.7	Comparison with the real data	10
2.8	Conclusions	13
3	Academic Honesty	13

1 Interplanetary Explorer Mission

1.1 Introduction

The PoliMi Space Agency is carrying out a transfer mission between a planet and a body in the Solar System with a powered gravity assist flyby at an intermediate planet, the aim of this report is to perform a preliminary mission analysis to find the best possible solution of the mission cost in terms of delta velocity using the Patched Conics method.

The minimum requirements for this mission are:

- the Spacecraft (S/C) must depart from Saturn
- the S/C must perform a flyby in Jupiter
- the S/C must arrive on Asteroid N.79
- the S/C earliest departure can be 00:00:00 01/01/2028
- the S/C latest arrival can be 00:00:00 01/01/2058

Lambert's problem was implemented to find the optimal heliocentric orbit to perform a transfer between two given points and its time of flight (ToF).

The method of patched conics was used in order to simplify the problem, as prescribed in the project requirements: elliptic heliocentric cruise for both the first and second leg of the mission. [1]. The Jupiter encounter consists of a planned *gravity assisted flyby*: this allows the S/C to earn a Delta-v needed to complete the mission and helps also reducing the costs of a possible plane change manoeuvre.

Furthermore we did not consider planetary departure and insertion as prescribed by the project requirements: initial heliocentric orbit is equal to the one of the departure planet and final heliocentric orbit is equal to the one of the arrival asteroid. [2]

1.2 Problem-solving Strategy

The PoliMi Space Agency provided a time window of 30 years:

Earliest Departure	00:00:00 01/01/2028
Latest Arrival	00:00:00 01/01/2058

Our goal was to define a more confined sub-window to search for an optimum Delta-v, aiming for a more efficient utilization of both time and computational resources.

1.2.1 Synodic period

In first approximation, the *synodic period* was taken into account: this is the amount of time required for an object to reappear at the same point in relation to two or more other objects. Because Lambert's arcs depend on their position vectors in the Sun-centred inertial reference frame, the optimal solution will occur *every synodic period* since the planets will assume the same reciprocal location.

We calculated that the synodic period between the departure and flyby planet is approximately **19 years and 305 days** and the one between flyby planet and arrival object is approximately **1 year and 337 days**.

This result can be also seen into Figure 1, where the Delta-time between the local minima are approximately equal to the synodic periods found.

1.2.2 Contour plots analysis

An effective tool to optimize the problem is the study of the contour plot (of Delta-v) of the first interplanetary mission's part between Saturn and Jupiter. Our Delta-v plots were created with all the possible combinations for departure time and flyby time: defining the total cost in terms of Delta-v with Lambert. Every point of the plot represents a Delta-v that was calculated with departure planet's position, flyby's planet position and time of flight: these elements can be defined with just 2 time instants, leading to a simple implementation of the algorithm which relies on two nested for loops that analyze the entire time window.

This procedure is often used for a *celestial body to celestial body* transfer, since we are calculating two delta velocities: one for the insertion on the transfer arc from the departure orbit and one for the insertion from the transfer arc into the arrival orbit. We decided not to include the second Delta-v in our plots since we are performing a flyby and we are not interested in inserting into Jupiter's orbit. This is an approximation but it was considered acceptable as the Delta-v needed at the flyby is almost always very small compared to the whole cost of the mission.

Same process can be carried out for the Jupiter-Asteroid N.79 part of the mission, solving the Lambert's problem for flyby and arrival times, thus giving another *2-degrees of freedom* plot. This time we also decided to ignore the Delta-v needed at the flyby. The results are even more evident since the solution is located exactly in a minimum of the plot.

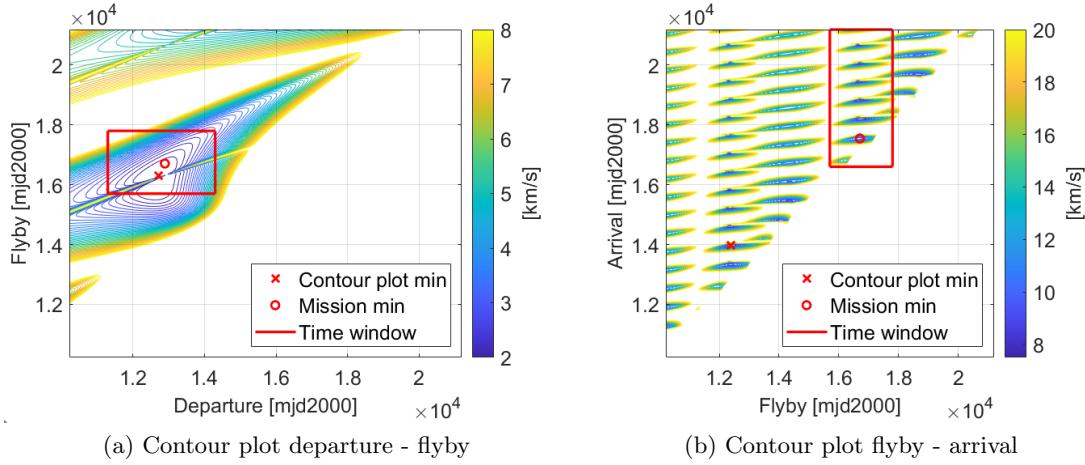


Figure 1: Contour plots (Δv plots) used to define the smaller time window

The plots shown in Figure 1 are bonded together with 1 degree of freedom: indeed, the flyby instant must be equal for both arrival (at the flyby) of the first Lambert Arc and for departure (from the flyby) of the second one. This means that the total number of degrees of freedom for the new time interval is 3.

Analyzing the first plot: the minimum Delta-v for the first Lambert Arc must be located inside the single *blue area* (corresponding to the lowest Delta-v values) of the plot. This assumption was made since the *blue area* is visible only once: Saturn - Jupiter synodic period will be taken into account but is obvious that this number is going to be determinant for the mission.

A square region around the *lowest Delta-v area* is located and carried on for the analysis of the other plot: by drawing the same line from the flyby axis (*y-axis*) of Figure 1 (a) onto the flyby axis (*x-axis*) of Figure 1 (b) we can now take all the values from $t_{arrival} > t_{flyby}$ and $t_{arrival} \leq t_{latest\ arrival}$ for the arrival time.

By doing so, we can see that a better refinement can be done by reducing the flyby time interval to cover only a smaller part of the visible minima of Figure 1 (b): the left portion of the graph, near the delimited area of the red rectangle, has some minima that have higher values of

Delta-v, for this reason they can be removed from the time window.

The final chosen windows are:

	Initial Date	Final Date
Departure Window	00:00 09/12/2030	00:00 25/02/2039
Flyby Window	00:00 26/12/2042	00:00 25/09/2048
Arrival Window	00:00 13/06/2045	00:00 01/01/2058

The plots in Figure 1 show that the minimum Delta-v found (red dot) was, indeed, inside the region highlighted with this strategy.

1.3 Optimizations

In order to refine the model and choose the optimal departure and arrival dates several optimization models were implemented, considering the lowest mission cost in terms of delta velocity as the most important parameter to optimize.

The team decided to implement these optimization techniques:

- Grid search
- Genetic Algorithm
- Multi Start

The most important part of all the optimization algorithms is the *cost function*: this function takes three inputs such as the three event times (departure, flyby and arrival instants) and gives, as a single output, the total Delta-v cost of that computed mission: $\text{DeltaV} = \text{costFunction}(t_1, t_2, t_3, \text{options})$.

This function is used in every algorithm and all the three methods rely on the same one, tweaked each time for a better implementation. If a solution is not physically possible, to avoid having to handle *NaN* values, the output of the function will return an extremely high number (in terms of Delta-v), thus not considering the solution as feasible.

Other parameters were taken into account: a minimum perigee radius was defined to avoid collision and/or drag perturbations with Jupiter's atmosphere, setting it to the Jupiter's diameter, giving a minimum altitude at the closest approach of its radius.

Later implementations of these algorithms proved to find consistent results with the whole time window, thus proving our refined time window was effective. Furthermore, a more complex multi start algorithm was tested, transforming the 3-inputs function into a 6-inputs one: the number of orbits for the Lambert function was unlocked to search for a minor Delta-v solution (otherwise set to 0). This was not found and it was decided not to include in the final version of the project.

1.3.1 Grid search

The first optimization algorithm we tested was the grid search: an easy-to-implement solution that gives good results on relatively small windows. Our code relies on a standard grid search created with 3 nested for loops that cycle through the given time windows for departure, flyby and arrival dates. These loops were also refined with an outer while loop that changes the *active search window* to be centered on the "previous iteration" solution and also narrows the total time interval. The while loop breaks when the chosen discretization is met.

The main difference, from a standard grid search, was implemented after this workflow: an *fmincon* is used after a decided number of while loops (or when the discretization is met) to find the exact solution of the local minimum found. This value is then stored to be compared later with other results.

For our project, we decided to create different time windows to be analyzed: the defined Figure 1 and few smaller ones that are located inside of it. This choice was made to better explore all the minima of the contour plot: the grid search, in fact, is not capable of analyzing the full problem and can easily fall in solutions far from the global minimum if the search interval is too large. This behaviour can also be seen from the *grid size*: the total number of points is calculated multiplying the 3 dimensions of the grid. For a grid of $n \times n \times n$ elements, the total number of points analysed will be n^3 . However, as n increases, the computation time required grows exponentially, but the same cannot be said for the solution found: a different and far value from the best solution could be outputted since the points of the function evaluation will fall on specific locations of the function resulting in misleading minima finding. In our case, if we increase the number of points we only get a higher running time, most of the cases finding the wrong solution. Analyzing the whole window, we tested a reduced the number of points and the global solution was found in a matter of seconds: we were lucky enough to define the grid in such a way to get a point to be located extremely close the global solution.

To avoid these problems, we decided to divide the window in smaller parts to better analyze all the possible minima shown in the contour plot.

1.3.2 Genetic Algorithm (GA)

The second optimization was created with a *genetic algorithm*, giving a faster minimum finder and a different approach to compare results. The implementation led to a simple code with important constrains, to better run the simulation. The algorithm proved to be consistent with the results found with grid search and, if we chose a narrow window, the script was able to find the global minimum, almost every time it run.

1.3.3 Multi Start

The last and most powerful algorithm implemented is the *multi start*: a simple core mechanics relying on random points evaluation, suitable for a 3 degrees of freedom problem. The algorithm relies on *fmincon* for the solution calculation and on parallel computing for faster data evaluation. This algorithm is able to find the global minimum with as few as 100 random starting points on the restricted time window.

1.4 Conclusions

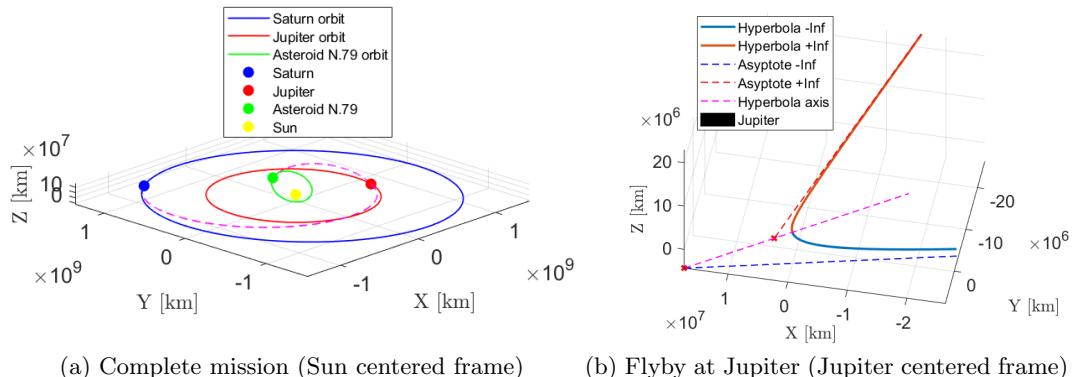


Figure 2: Plots of the mission

	Value	Unit
Δv_{tot}	11.4626	[km/s]
- Δv_{dep}	1.8865	[km/s]
- Δv_{arr}	8.7021	[km/s]
- Δv_{ga}	0.8740	[km/s]
- $\Delta v_{flyby\ tot}$	8.2068	[km/s]
- $\Delta v_{ga}/\Delta v_{flyby\ tot}$	0.1065	[‐]
Δt_{tot}	≈ 12.73	[years]
- t_{dep}	2035 04 20	[date]
- t_{flyby}	2045 09 28	[date]
- t_{arr}	2048 01 13	[date]

Table 1: Complete mission results

	Value	Unit
e^-	1.0393	[‐]
e^+	1.2344	[‐]
r_p	738835.92	[km]
δ	128.2950	[deg]
Δt_{SoI}	190.9614	[days]
h_{min}	668924.92	[km]
Type	Trailing	[‐]
V_∞^-	2.5971	[km/s]
V_∞^+	6.3402	[km/s]
V_{planet}	13.5988	[km/s]
v_∞^-	15.1062	[km/s]
v_∞^+	9.4355	[km/s]

Table 2: Flyby results

Lambert arcs	a [km]	e [‐]	i [rad]	Ω [rad]	ω [rad]	θ [rad]
First	1.0429e+09	0.3067	0.0235	1.8182	3.5923	3.0573
Second	4.9814e+08	0.5246	0.3854	5.9836	2.9139	5.2866

Table 3: Delta-v and Delta-t results table

In terms of time efficiency, all the optimization methods tested take similar amount of time. Our solution proved to be the same output of all the three optimizations implemented. This was later tested with the whole time window to understand if a better solution was possible: all three optimizations, running with the larger time window, found the same result as found before, thus validating our strategy. The global minimum that was found is assumed to be the actual global minimum for our mission: *global optimization* is not simple and is not always possible to find the best minima.

The Delta-v calculated can be divided in the change of velocity performed by the S/C and in the change of velocity given by the flyby planet: this allows for a better understanding of all the parts of the mission. Starting from the flyby, it can be seen that the Delta-v needed to perform the powered gravity assist is much smaller if compared with the one given by the planet. The fraction of the two values is 0.1065, showing that the flyby was indeed effective: the smaller this fraction, better is the exploitation of the flyby planet’s velocity. Furthermore, the flyby was effective for a plane change since the difference of inclination between the two Lambert arcs is 20.7 deg and the exit heliocentric velocity defines a plane with an inclination similar to the inclination of the final orbit of asteroid N.79: 22.0791 deg, compared with 22.9012 deg of the NEO. In our case we also have an increase of heliocentric velocity (before and after the flyby), defining a *trailing* flyby.

Both departure and arrival Delta-v are as expected from the Figure 1, where it can be seen how the final change of velocity needed is much higher than the departure one.

The time of flight of the whole mission is approx. 12.73 years, this is not comparable with missions like *Voyager 1* that took as little as 3 years and 2 month to perform a similar duration mission (from Earth to Saturn) but this can be traced back to multiple factors including, as we suppose, a better alignment of the planets and a different mission analysis goal.

2 Planetary Explorer Mission

2.1 Introduction

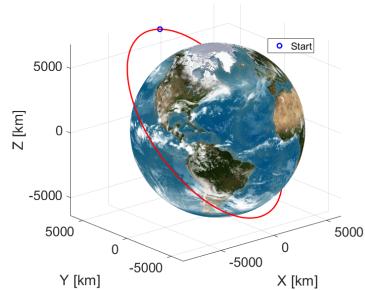
The PoliMi Space Agency is carrying out a Planetary Explorer Mission, to perform Earth observation. The aim of this report is to perform the orbit analysis and ground track estimation for this mission taking into account different perturbations and to propose an orbit modification in order to better the communications with ground stations by imposing a repeating ground track.

The minimum requirements for this mission are:

- The orbit's semi major axis must be 8016 km
- The orbit's eccentricity must be 0.1678
- The orbit's inclination must be 50.3342 degrees
- The orbit's Ground Track repeating ratio must be $k=12$ and $m=1$
- The orbit must consider J2 and Drag perturbations
- The S/C's drag coefficient must be 2.1
- The S/C's area over mass ratio must be $0.0171 \text{ m}^2/\text{kg}$

To further improve the analysis, the model is compared using data from a real object with similar orbit parameters: the object selected is a Russian satellite named Express-MD2.

2.2 Orbit characterization



The orbit is geocentric and its keplerian elements were already partly assigned: the semi major axis, the eccentricity and the inclination. In order to complete the definition of the orbit the right ascension of the ascending node (RAAN), the argument of the pericentre and the starting mean anomaly were all taken to be the same value as the similar object analyzed: Express-MD2.

Figure 3: Nominal orbit [3]

a [km]	e [-]	i [deg]	Ω [deg]	ω [deg]	θ [deg]
8016	0,1678	50,3342	27,2290	315,4032	122,0796

Table 4: Orbital Parameters

2.3 Perturbations

The perturbations modelled, as indicated by the minimum requirements for the mission, are the air drag and oblateness of the earth. The perturbations are introduced into the model as accelerations \vec{a}_p added to the two body problem formulation

$$\ddot{\vec{r}} = -\frac{\mu_{\oplus}}{r^3} \vec{r} + \sum \vec{a}_p$$

2.3.1 Zonal harmonics: J_2 effect

It represents the perturbation on the orbit due to the Earth's oblateness. The perturbing acceleration is:

$$\vec{a}_{J_2} = -\frac{3}{2} \frac{J_2 \mu_{\oplus} R_{\oplus}^2}{r^4} \begin{Bmatrix} \frac{x}{r} \left(\frac{5z^2}{r^2} - 1 \right) \\ \frac{y}{r} \left(\frac{5z^2}{r^2} - 1 \right) \\ \frac{z}{r} \left(\frac{5z^2}{r^2} - 3 \right) \end{Bmatrix}$$

Where J_2 is the second zonal harmonic, μ_{\oplus} and R_{\oplus} are the Earth's gravitational parameter and mean radius and $\vec{r} = x\hat{i} + y\hat{j} + z\hat{k}$ is the vector that points the position of the spacecraft on the orbit from the centre of the Earth.

2.3.2 Air drag

It represents the perturbation on the orbit due to the air molecules present on the atmosphere for orbits lower than 400km. The perturbing acceleration is:

$$\vec{a}_{drag} = -\frac{1}{2} \frac{A_{cross} C_D}{m} \rho(h, t) v_{rel}^2 \frac{\vec{v}_{rel}}{\|\vec{v}_{rel}\|}$$

Where m is the spacecraft's mass, C_D is the drag coefficient, A_{cross} is the cross-sectional area perpendicular to \vec{v}_{rel} which is the air-relative speed and $\rho(h, t)$ is the atmospheric density.

2.4 Ground track

The projection of satellite's path onto the earth's surface is computed without perturbations for 1 and 36 orbits in Figure 4.a. The number of orbits was chosen as multiple of 12 in accordance to the ratio of satellite revolutions and earth rotations to better understand the satellite's coverage.

The effects of the two perturbations mentioned before on the ground track can be seen in the following figure:

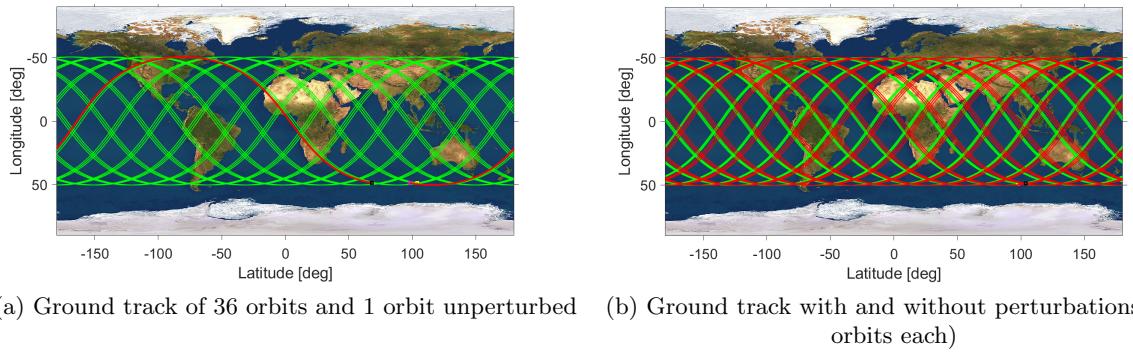


Figure 4: Ground Tracks

Where in Figure 4.a the red line represents the the ground track of one orbit and the green one the ground track of 36 orbits; while in Figure 4.b both red and green represents the ground track of 36 orbits but the green are unperturbed orbits and the red are perturbed.

2.4.1 Repeating ground track

In order to improve the communications with ground stations, a repeating ground track is proposed. As such, the semi-major axis of the orbit is modified given the desired number of revolutions of the satellite and rotation of the Earth, with the following formula:

$$a_{rep} = \sqrt[3]{\mu \cdot \left(\frac{m}{\omega_E \cdot k}\right)^2} \quad (1)$$

In which μ represents the planetary constant of earth, m the number of earth rotations, k the number of satellite revolutions and ω_E the earth angular velocity. The semi-major axis needed to comply with a repeating ratio a 12:1 can be seen in Table 5. This new value is only 0.358% larger than the original one given, meaning the maneuver to accomplish this wont represent big ordeal. The repeating semi-major axis was only used in this subsection, the rest of calculations on the project will be done using the original value given of 8016km.

k [-]	m [-]	a [km]	a_{rep} [km]
12	1	8016	8044,7019

Table 5: Repeating parameters & Change in semi-major axis

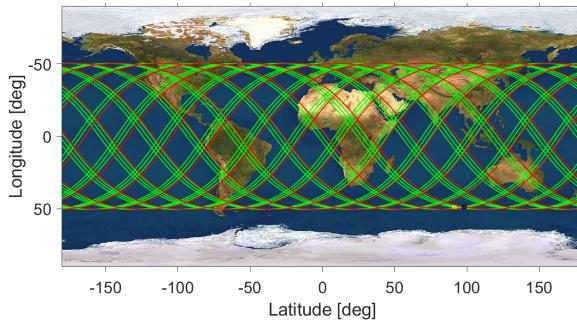


Figure 5: Repeating ground track with and without perturbations 36 orbits [3]

Where in Figure 5 the red line represents the the repeating ground track of the unperturbed orbit while the green is for the perturbed orbit.

The semi-major axis, required to achieve a repeating ground track was calculated and then the orbit was evaluated with and without perturbations, but it can be seen that for the perturbed ground track it doesn't repeat exactly its path, this is due to the resonance perturbations the satellite is subjected to. Since the semi-major axis is changing, the period of the orbit as well as the other Keplerian elements are changing too, thus effectively giving us a completely different orbit. Therefore, the repeating ground track algorithm won't work because the orbit is drifting with time, hence satellites with repeating ground track missions must undergo periodic orbit maintenance maneuvers to keep the desired path [4].

2.5 Evolution of the orbit

To simulate the evolution of the orbit, we numerically integrated the Cartesian equations of motion as a propagating method. The propagating period chosen is 60 nominal orbits, so it is possible to appreciate the changes that the two perturbations cause to the orbit on a span of around 5 days.

The Keplerian elements behaviour over the same period of time were calculated using both the numerical integration of the Gauss planetary equations and the Cartesian equations:

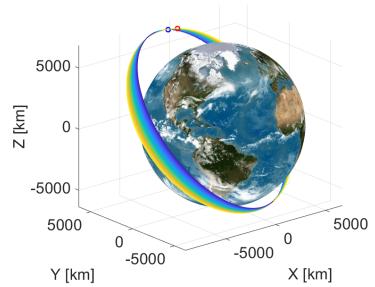


Figure 6: Orbit propagation perturbed

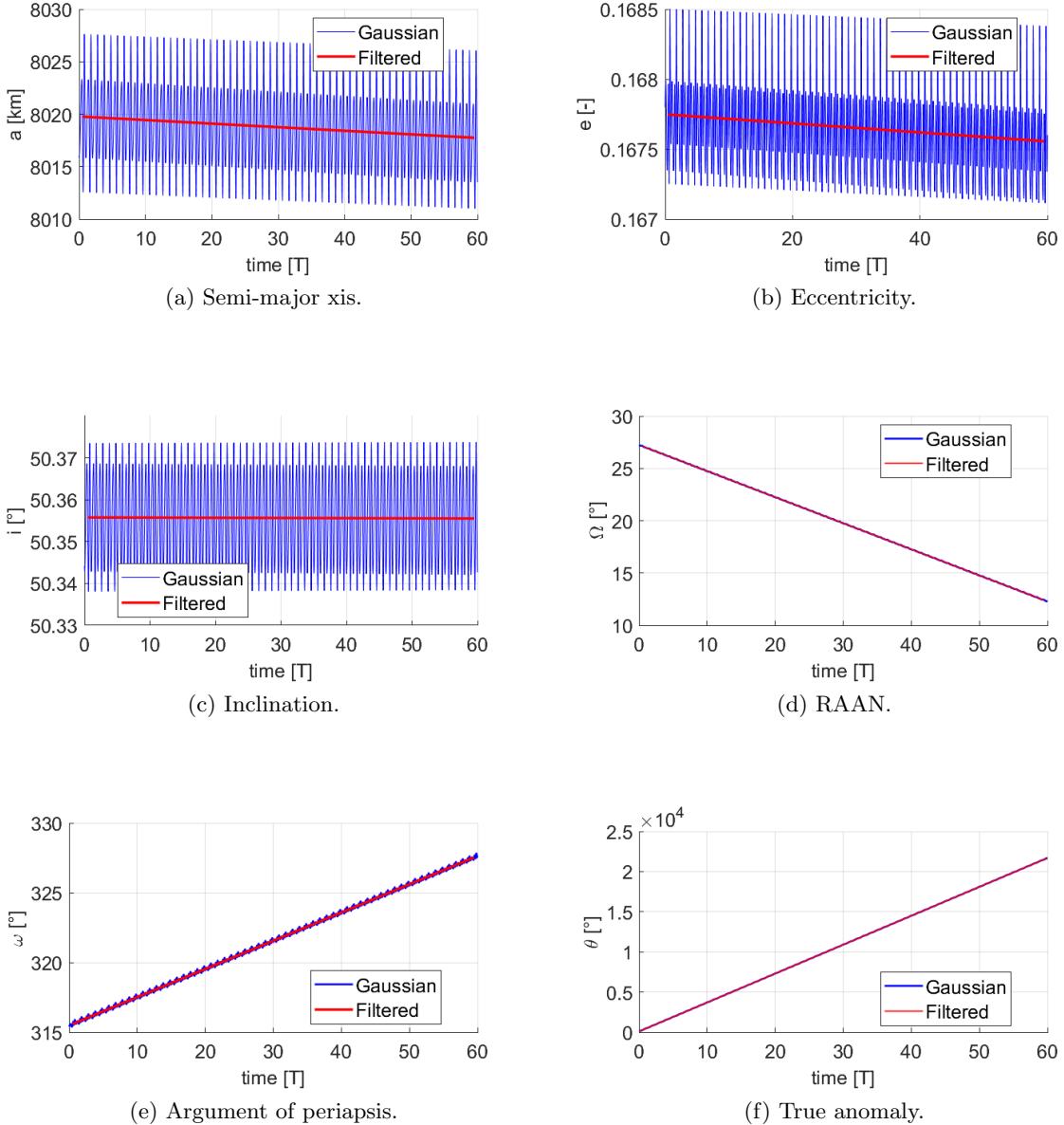
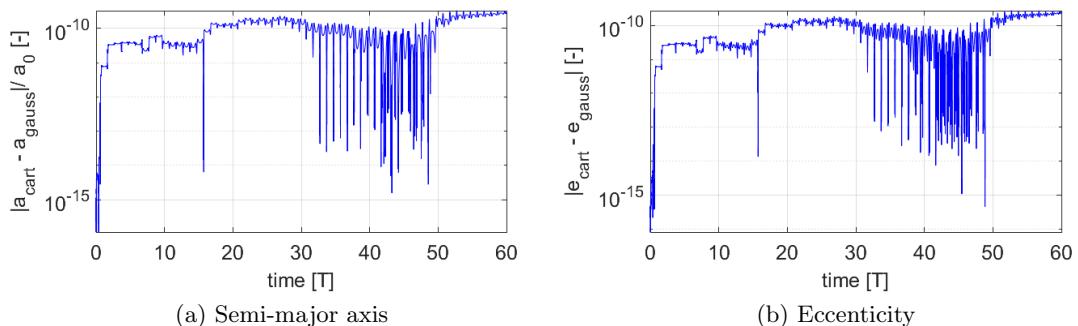


Figure 7: Evolution of Keplerian elements with Guass equations and their filtered versions

Where in Figure 7 the blue line represents the numerical integration of the Guassian equation for each keplerian element, and the red line is the filtered version of these same values.

The error difference between both propagation methods for the calculation of the evolution of each Keplerian element can be seen in the following figures:



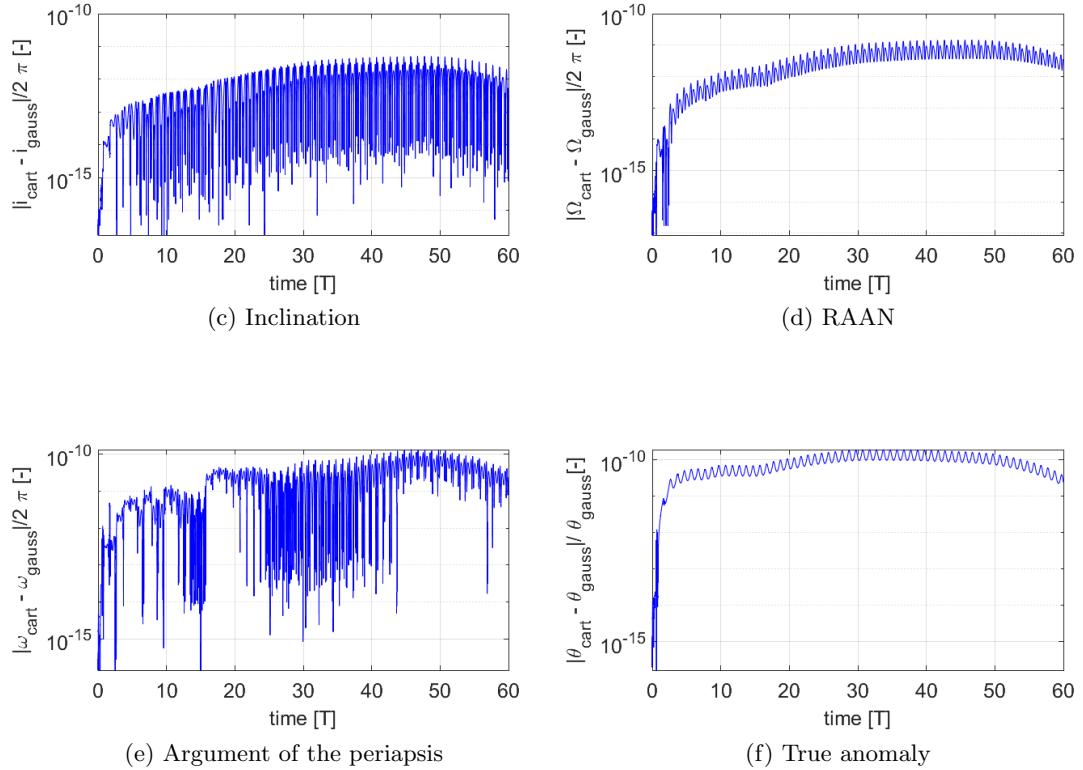


Figure 8: Evolution propagation method difference

Where in Figure 8 the blue line represents the error difference, and the red line is the filtered version of these same values.

In Figure 7 only the Gaussian equations of motion are present, that is because the difference between the Cartesian equations of motion are almost indistinguishable: Figure 8, confirms this showing that the biggest difference between these values is in ranges of 10^{-10} . Which means that both methods can be considered reliable since they described similar and reasonable trends in accordance with the expected effects from the perturbations analysed, as previously explained. Moreover the time required to compute this results is similar.

Since both methods give almost identical results, from this point forward all comparisons for the evolution of keplerian elements will be done using the numerical integration of the Gaussian equations of motions.

2.6 Filtering

In order to better appreciate the evolution of the Keplerian elements, a low-pass filter is implemented using Matlab's tool *movmean*, to lower the amount of noise, the filtering can be seen in Figure 7.

The number of data available for each keplerian element over the course of 5 days was divided by the number of orbits, these way we have an estimation of how much data is available for each orbit. This was the parameter chosen for the moving window of the filter, ensuring a good reduction of the noise and good visibility of the secular trend of the graphics.

2.7 Comparison with the real data

As previously said, the object selected to compare the model was the Russian satellite Express-MD2. It has to be noted that there seems to be some misleading data with this satellite, since some sources say the satellite was lost shortly after it's launch on 2012, or that it was left on

an unusable orbit [5], and meanwhile the NASA horizon system still provides its ephemerides up to this date [6]. Having known this, we still decided to use the satellite, since we are mainly interested in finding an object with similar keplerian elements to our case and the data provided by NASA is reliable enough for us to carry out this analysis. The orbital parameters of this satellite are the following:

a [km]	e [-]	i [deg]	Ω [deg]	ω [deg]	θ [deg]
8256,9	0,1956	49,9077	27,2290	315,4032	122,0796

Table 6: Orbital Parameters [6]

The evolution of the Keplerian elements of the Express-MD2 are calculated using the same models implemented for our satellite and compared to the real data obtained from the ephemerides [6] of the satellite.

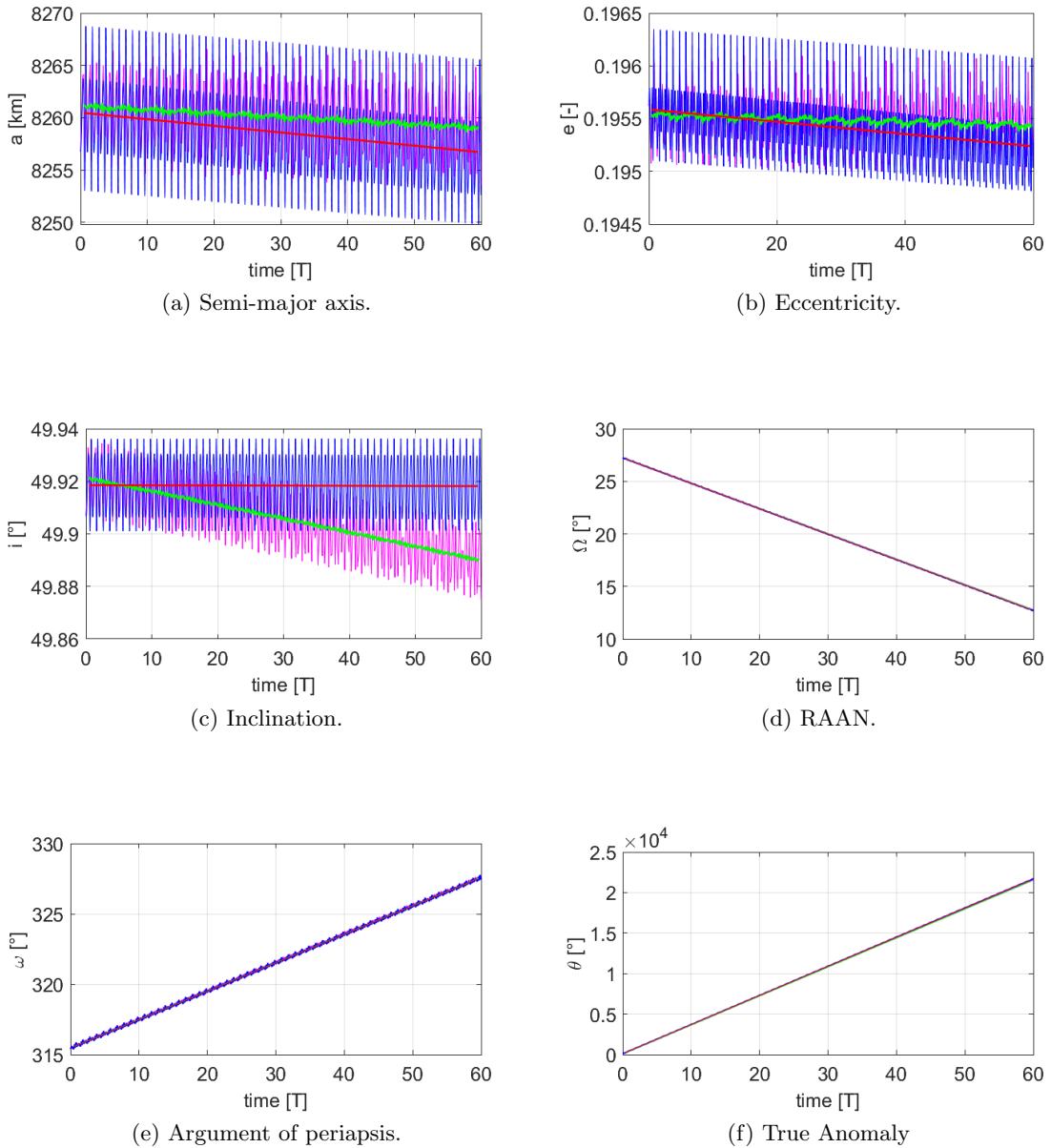


Figure 9: Comparison Real Data and Gaussian with filtered Keplerian elements of Express-MD2

The difference between the real data from the ephemerides [6] and the values calculated with the numerical integration of the gaussian equations of motion can be seen in the following figures:

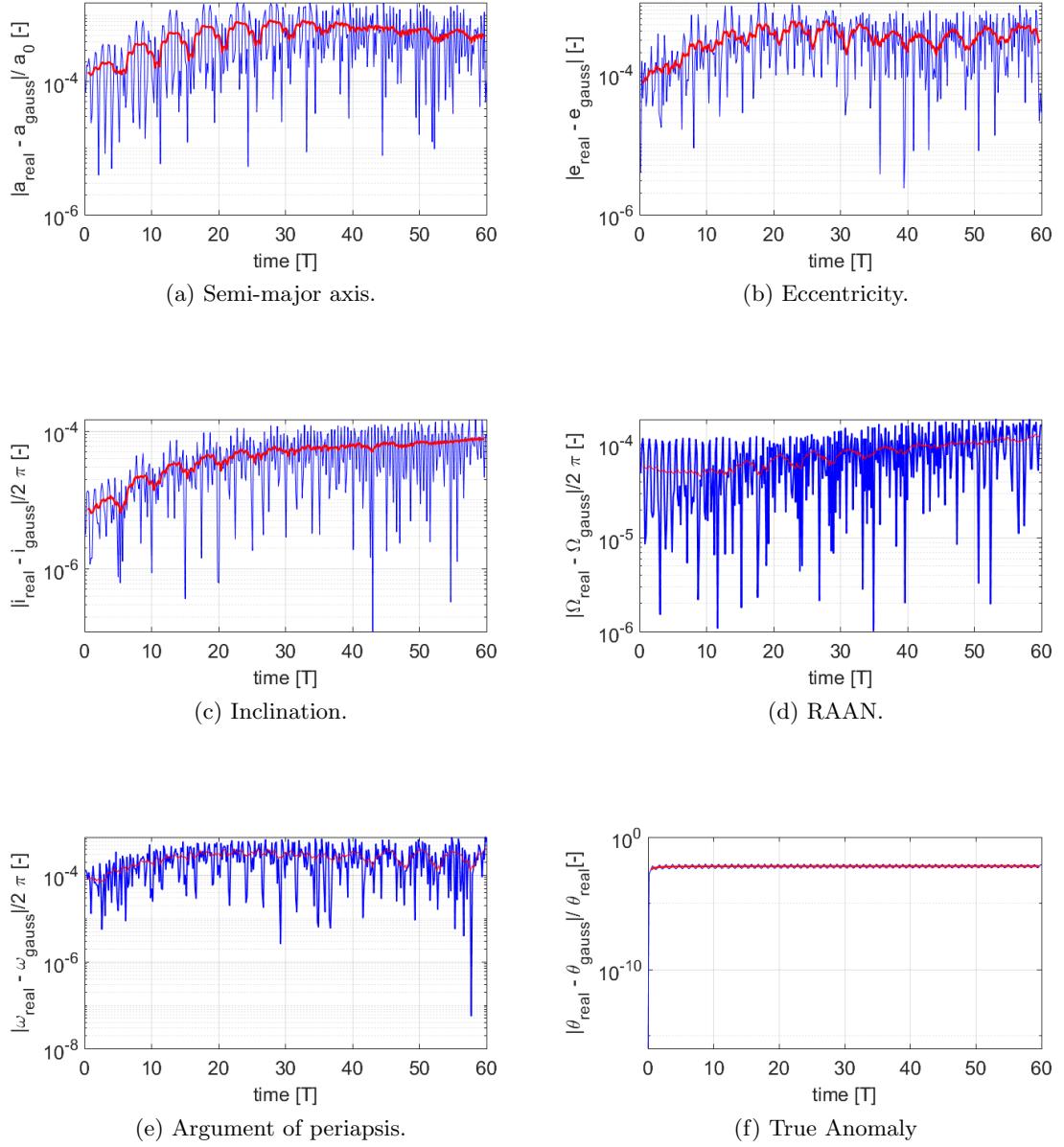


Figure 10: Evolution Real Data and Propagation Method Difference

On Figure 9 when comparing the models implemented with the real data from the Express-MD2 ephemerides [6], it can be seen that the values calculated have the same tendency as the real data, which is indication of the accuracy of the models. But there are of course some discrepancies within the values as seen on Figure 10. One of the causes of these variations is the precision of the models, when propagating the orbit some tolerances and parameters are assumed that can affect how accurate the models are. Another cause, is the fact that in the models only two major perturbations were taken into account, while in reality the satellite is affected by a greater range of perturbations, whose effect might not be as prominent as the air drag and J2 perturbations, but can still affect all keplerian elements of the orbit.

2.8 Conclusions

From the evolution semi-major axis and eccentricity of our satellite on Figure 7, it can be seen that both the eccentricity and the semi-major axis of the orbit reduce with time, this is a major influence of the air drag perturbation because of the decrease on the s/c's velocity whenever it passes through the orbits pericentre.

On the other hand, the oblateness of the earth can affect a lot of parameters of the orbit, for example, as it can be seen on Figure 4, the orbit's ascending node (RAAN) move westward since it has an inclination between 0 degrees and 90 degrees. another noticeable perturbation, is the way it seems the orbit moves to the side, on Figure 6, this is known as the "Hula-hooping" effect an its due to the perigee precession that makes the apse line rotate with time.

3 Academic Honesty

Earth's flat image used for 3d representations and/or for animations was downloaded from [Nasa website](#) as there's no restrictions under "[Still Images, Audio Recordings, Video, and Related Computer Files for Non-Commercial Use](#)" for our use. We do not own this image: NASA/Goddard Space Flight Center Scientific Visualization Studio The Blue Marble Next Generation data is courtesy of Reto Stockli (NASA/GSFC) and NASA's Earth Observatory. The Earth image used for the *Groud Track Plots* was taken from the *WeBeep* function folder. Moreover we did use MatLab® install, *Global Optimization Toolbox* and the *Optimization Toolbox*.

References

- [1] H. D. Curtis, *Orbital Mechanics for Engineering Students*. Oxford: Butterworth-Heinemann, 3rd ed., 2014.
- [2] Colombo, Camilla, "Lecture Notes," 2023.
- [3] "NASA Scientific Visualization Studio," 2009.
- [4] Wakker, Karel, "Fundamentals of Astrodynamics," 2015.
- [5] "NSSDC Spacecraft ID 2012-044B," 2022.
- [6] "JPL NASA Horizons Web Application," 2023.