



POLITECNICO DI MILANO
M.Sc Space Engineering

ORBITAL MECHANICS

A.Y. 2020 2021

Professor Camilla Colombo

Alkhawashke Suhailah

ID number 10772675 - 966950

Salvi Ciro

ID number 10568573 - 969080

Spada Fabio

ID number 10611057 - 967678

Staffolani Alessandro

ID number 10578461 - 966903

Contents

1	Interplanetary Transfer	3
1.1	Description	3
1.2	Leg Approach	4
1.2.1	Considerations	5
1.3	Global Approach	6
1.3.1	Global GA Analysis	6
1.3.2	Refinement	8
1.3.3	Refinement Method Comparison	10
1.4	Conclusion	11
2	Planetary Mission	14
2.1	Description	14
2.1.1	Data	14
2.1.2	Hypothesis	14
2.2	Ground Track	15
2.2.1	Repeating GT	15
2.3	Numerical Propagation	18
2.3.1	Results	18
2.4	Real Satellite Comparison	21
2.4.1	Observations	21
2.5	Conclusions	22

Chapter 1

Interplanetary Transfer

1.1 Description

The request of the interplanetary assignment is to perform a preliminary mission analysis of an Explorer Mission visiting three planets in the solar system with the patched conics method. The transfer from the initial planet to the final planet is assisted by a powered Flyby. The figure of merit assessed in the transfer options is the total cost in terms of the Δv . The time of flight is neglected as a figure of merit, as manoeuvres are considered as impulsive. The sequence of the mission is as follows:

- Departure from Mercury
- Flyby at Venus
- Arrival to Jupiter

Earliest Departure: 2027/Jun/01

Latest Arrival: 2067/Jun/01

The following assumptions were taken into account:

- Initial heliocentric orbit is equal to that of the departure planet
- Final heliocentric orbit is equal to that of the arrival planet.
- Other planets presence and perturbations during the interplanetary cruise are ignored (3rd body perturbations, SRP and others)
- The asteroid belt presence was ignored.
- The lowest acceptable height of pericentre at the flyby was set to 100 km.

To attempt the feasibility study, the interplanetary transfer was approached by two separate methods: The Leg Approach and the Global Approach. The prior attempts to find the optimal solution by minimizing the leg related costs and initially discarding the flyby, while the latter studies the minimum of the total Δv accounting for the effects of the flyby from the beginning. The results of both methods are compared, and the optimal Δv is then chosen. In both approaches a powered flyby was accounted for; the custom script used for gravity assist analysis, *powFlyby.m*, embeds the constraints on the minimum perigee height required to avoid planet impact; the working principle of this function is described below.

powFlyby.m

The entry and exit velocities are fixed with the interplanetary legs choice, thus the turning angle required by the flyby is fixed too. Called δ the real turning angle, the radius of the pericentre of the hyperbola is between the radiiuses associated to two unpowered flybys with deflection angles equal to δ . It is possible to briefly verify this: rp_{Guess_1} is the pericentre radius of an hyperbola characterized by a deflection angle equal to δ and by an infinite velocity equal to real entry velocity; rp_{Guess_2} is the pericentre radius of an hyperbola characterized as well by a deflection angle δ , but by an infinite velocity equal to real exit velocity. Without

losing generality it is hypothesized that entry excess velocity is lower than exit excess velocity. This implies that $rp_{Guess_1} > rp_{Guess_2}$; indeed, to get a fixed deflection angle, the smaller the orbiter velocity, the smaller the gravity pull required, so the higher the associated pericentre radius. Moreover, an acceleration is required at the pericentre, as the exit excess speed is higher than the entry one. The actual trajectory is made up of two different branches, and the real deflection angle is equal to the sum of the semideflection angles of both branches. Calling δ^- the entry branch deflection angle and δ^+ the exit branch deflection angle, it results $\delta = \delta^-/2 + \delta^+/2$; the two infinite velocities are different, and to get the previously mentioned sum equal to δ , it is necessary to choose a proper pericentre radius. Choosing rp_{Guess_1} as pericentre radius, at the pericentre the orbiter velocity would have been deflected of $\delta^-/2$ since its entry in planet's SOI, thus of $\delta/2$; in the second branch, however, the gravity pull would not be enough to deflect the trajectory of further $\delta/2$, as the orbiter would be characterized by higher velocities with respect to the first branch. δ^- would indeed be smaller than $\delta/2$. Totally the planet would not deflect of δ , but of an angle minor than δ . The pericentre must be lower than rp_{Guess_1} . Fixing pericentre radius at rp_{Guess_2} , instead, the orbiter trajectory after the pericentre would be deflected of $\delta^+/2$, so of $\delta/2$. However, satellite is slower along the first leg with respect to the second: the gravitational pull would be enough to force $\delta^-/2$ to be higher than $\delta/2$. The pericentre must indeed be higher than rp_{Guess_2} . As rp_{Guess_1} is higher than rp_{Guess_2} , the pericentre radius must lie between these two values.

1.2 Leg Approach

The first analysis performed was based on the following idea: the search for a global acceptable solution can be made by focusing on specific time windows, namely the ones associated to the minimum cost of each leg, at first discarding the gravity assist manoeuvre; once windows are defined, the two legs are patched together according to the best flyby option possible.

A general algorithm was developed to get a first glance at how the porkchop plots of the two legs are shaped: restricting the analysis on the optimal zones allows swift computations and provides reasonable results at the same time. The porkchop plots are based on the whole launch window for both first and second leg; the costs of first impulse and second impulse, respectively of the first and second leg, are represented in the space '**DepartureDate - ToF - Cost**' for a reduced departure window in Figure 1.2.

It can be noticed that the optimum costs region lies, for both porkchop plots, in a zone where time of flight values are subdued, and the higher the time of flight, the higher the cost associated to a departure instant. A pattern can be observed for each of the transfer legs: the repeating porkchop plots are similar and a minimum can be found for each of these; the time distance between each repetition is approximately equal to the synodic period of the two planets involved. This feature is exploited in *porkchopRefined2Leg1.m* and in *porkchopRefined2Leg2.m*, where the low ToF zone is efficiently analyzed: a rectangular time window, in which a refined analysis is performed, automatically shifts along the departure date axis with a step equal to the synodic period. Computational costs are thus minimized. Once local minima for both legs are located, time windows around minima are generated such that inside these windows, transfer costs do not exceed the local minimum up to a certain fixed tolerance.

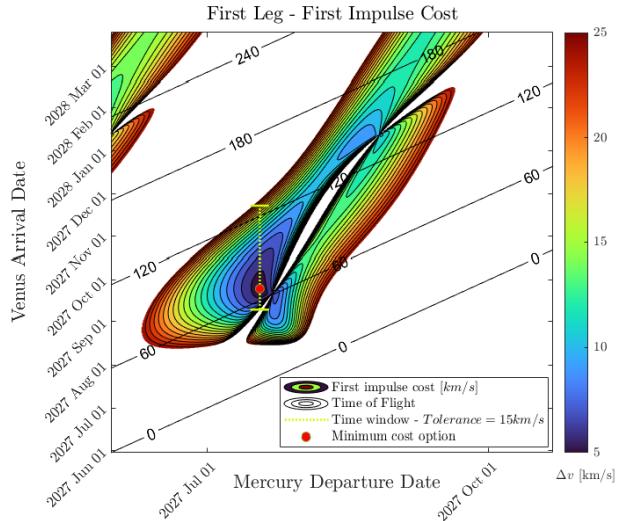


Figure 1.1: Example of window generation, given a fixed tolerance

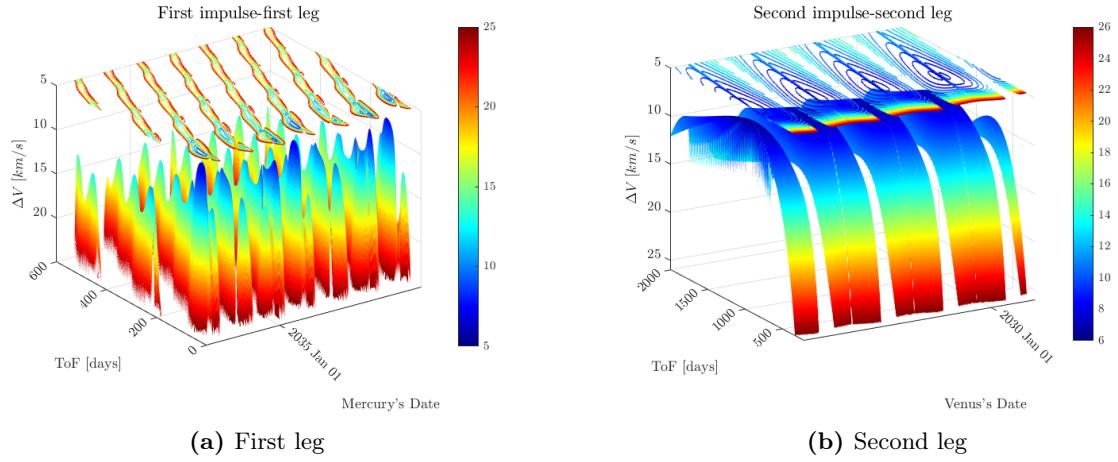


Figure 1.2: 3D Porkchop Plots

These windows are built allowing only Venus passage date to vary, fixing Mercury and Jupiter dates equal to the ones of each minimum (figure 1.1). Two different sets of time windows for the two legs are then found; what is left is patching the two legs together to obtain global transfer options. This is done with an overlap of the two sets of windows over Venus passage date that allows to find the intersections between time windows. The process is shown in figure 1.9. The final result is a very restricted set of time windows in which a genetic algorithm based optimization process is implemented: Venus date is allowed to shift within each window found with the overlap.

The Leg Approach proves its inefficiency when the two legs are patched: to allow for possible solutions, tolerances over minimum costs must be set as relatively high, and the most solutions found require high transfer costs and so do not make for realizable transfers. E.g. allowing for 8 km/s and 2 km/s respectively as tolerance over first impulse and second impulse cost, two available options are found over the full window; the optimal one costs 29.75 km/s and Mercury, Venus and Jupiter dates are respectively 2056-Jan-28, 2056-Feb-21 and 2058-May-18. For the couple of tolerances equal to 15 km/s and 5 km/s a higher number of solutions is found, and the best solution accounts for a cost of 27.26 km/s. The dates related to this transfer are, following transfer sequence, 2056-Jan-28, 2056-Feb-18 and 2058-May-18. This last solution is represented in figure ???. It is interesting noticing that the final trajectory resembles a Hohmann-like transfer, and the fly-by modifies the heliocentric trajectory as an approximately tangent manoeuvre; flyby altitude of this transfer, as seen in figure ?? is not the minimum allowed by the constraints and, as well, the plane of flyby trajectory is almost perpendicular to Venus' velocity: fly by potential is not fully used. Trying to find a more efficient solution, at this point, is mandatory.

1.2.1 Considerations

It was observed that local minima of each leg differ from each other and their evolutions with respect to time are periodic (figure 1.4). A frequency decomposition of these discrete signals was implemented to find the most influent harmonic contributions, being aware that the previously shown graphs coincide with real costs evolution only in highlighted points; therefore, the points lying on the dashed graph line do not correspond to real possible transfers.

It is first noticeable that Jupiter position dominates the costs of the second leg, as the most influent harmonic contribution is associated to Jupiter's period. Mercury's orbit is characterized by higher eccentricity and inclination with respect to the other two orbits. Furthermore, Mercury and Venus periods are more comparable than Venus and Jupiter ones. For these reasons, the first leg frequency analysis is more complex and frequency spectrum is harder to analyze.

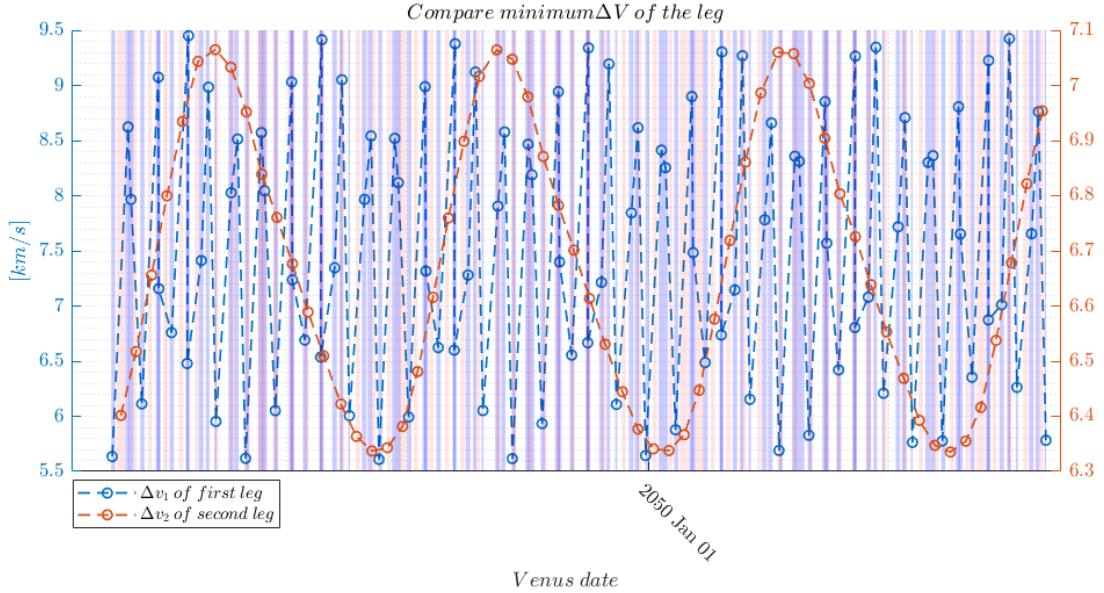


Figure 1.3: Sets of Minima and overlap of windows, for 8 km/s tolerance over first leg's cost and 2km/s over second leg's cost

There are two problems in this approach. The first problem is that the *discrete* signal sampling time is close to the synodic period of each leg, so the sampling frequency (f_s) is always lower than the higher revolution frequency among the two planets ($f_{Revolution} = 1/T_{Revolution}$). According to Nyquist theorem, the harmonic contribution associated to such a frequency will appear folded in the frequency spectrum, as well as all the frequencies above $\frac{f_s}{2}$. Regarding the second leg, Jupiter's frequency is not affected by this problem and it is easily visible, while in the first leg, Venus frequency is folded, as it is higher than Nyquist frequency. The second problem of this approach is that minima are not equally spaced in time, so the sampling time of the discrete signal is not constant; it is indeed harder to associate folded frequencies to the original ones, as Nyquist frequency is not directly related to the synodic period.

1.3 Global Approach

The global approach does not solely account for the minimum costs of each leg, yet instead studies the minimum cost of the whole maneuver accounting for the powered flyby effects, trying to find the minimum cost transfer in the global window. The function evaluating the total cost of the transfer, being given the dates at the three planets, is *dv_optMod2.m*: this function uses *powFlyby.m*, thus embedding the constraint over minimum pericentre height for the gravity assist. For the Global approach assessment, several MATLAB optimization tools were exploited to obtain the optimal solution, such as *GA*, *fmincon*, and *fminunc*. *Dv_optMod2.m* is fed as objective function to all the MATLAB built-in optimizers used. The Global Approach was executed on two consecutive stages: an **initial global analysis** with GA on the whole window, and a **refined analysis** around the best points found in the previous stage.

1.3.1 Global GA Analysis

A **window** corresponds to a pair of dates that define the time boundaries of the problem, namely the earliest departure date from Mercury and latest arrival on Jupiter. Note that in this three dofs problem a window defines a cube where each of the three dimensions corresponds to a dof; thus, the window defines boundaries

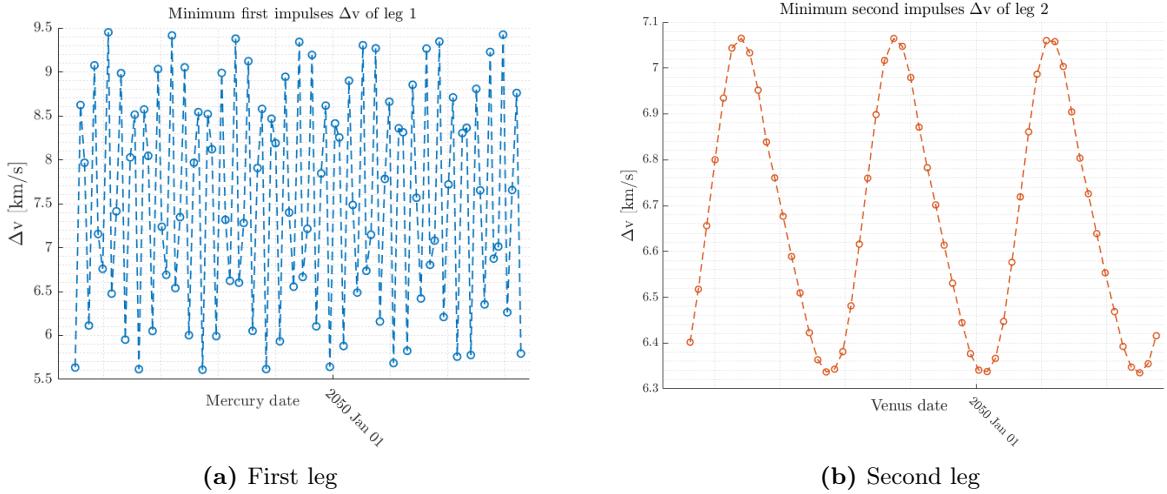


Figure 1.4: Sets of Minima on the whole 40 years window, for separated legs

for all three planets dates.

For the first global GA Analysis, the overall window was divided into two main overlapped smaller windows – 2027-06-01 to 2054-06-01 and 2046-06-01 to 2067-06-01 – which were separately assessed by GA. It was noticed that the density of the population generated is higher on the origin of the time axis, and for this reason, it is necessary to furtherly divide the above-mentioned windows into smaller ones to efficiently exploit the functionality of GA. Each of these sub-windows of 6 years is shifted to the next one by 30 days, and the GA assessment is done for 10 iterations for each window. It was chosen to leave the *PopulationSize* parameter at its default value of 50 to allow for swifter calculations.

ga.m function

GA is a genetic algorithm optimizer and it is based on randomly chosen guesses. Genetic algorithms follow the concept of natural selection inspired by nature: the algorithm starts generating an initial population and for each individual a fitness function is evaluated; the individuals with the best fitness values are chosen. With selection, mating and mutation strategies, the population evolves to a new one where each individual is again assessed, until the function converges.

This technique is definitely relevant for our purpose since it assesses several points at each iteration. On the other hand, GA optimizer is non-deterministic so it is expected to converge to a different value each time it is run; to ensure the best minimum to be found, the GA optimizer is run several times and the final minimum is extracted at the end of the process. This multiple run is performed in two custom functions, namely *GA_FlybyMod2.m* and *GA_Flyby_con.m*. In both, constraints were added on the departure and arrival date limits and on the correct temporal sequence of planet related dates; in the latter, time of flights were limited too, to better analyze the low time-of-flight zone: coarse tolerances over coasting phase duration were used, obtaining in the end similar results with respect to the unconstrained case.

Local minima are distributed in the whole window and it is noticeable that they are denser in the region immediately over the diagonal. In the **departure date - times of flight** space (figure 1.5), the optima are associated to high ToF options, with the two phases (Mercury to Venus transfer and Venus to Jupiter transfer) approximately equally lasting. It is worth noticing, as well, that local minima are organized in groups and each group is made of optimal transfer options aligned with respect to the second leg ToF axis. The meaning of this pattern is further analyzed in section 1.4.

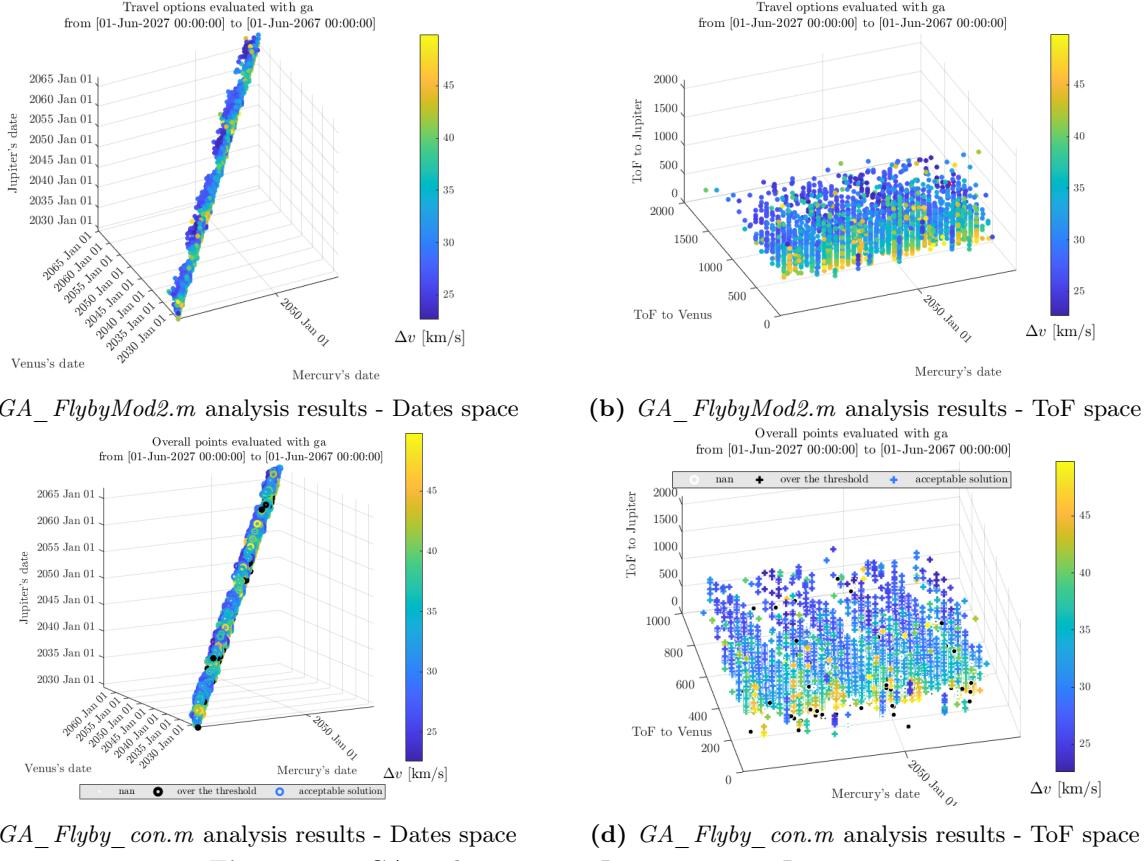


Figure 1.5: GA evaluations, 01-Jun-2027 to 01-Jun-2067

1.3.2 Refinement

Among the previously found local minima, the best options were chosen for a further refinement. The overall minimum cost is 22.55 km/s, and the team accounted for a tolerance over this cost of about 1km/s: the threshold of 23.5 km/s was then set and the transfer options with a total cost under this threshold were selected for the refinement. With such a threshold the analysis satisfies both the choice of a variety of options, both a reasonable margin of improvement. At this point, three different refinement approaches were used:

- Grid Search
- Gradient Based Optimizers (*fmincon.m/fminunc.m*)
- Genetic Algorithm (*ga.m*)

To compare the three strategies' efficiency, they were applied to the same windows built around the previously selected minima. The semi amplitude of the windows was set to 10 days to avoid exploring other local minima inside the window, thus wasting computational power aimed at refining the sole initial guess.

Grid Search

The most robust strategy for refining the global optimal solution is a grid search approach; a uniformly spaced 3D grid of dates is built around each minimum and for any possible combination of dates of the three planets, the cost of the whole transfer is evaluated. The grid is defined by three parameters: the vector of three dates associated to the minimum, the time step to discretize the grid and its semi-amplitude.

Initial cost: 24.00256 km/s, dep: 2048-07-06, flyby: 2051-04-08, arr: 2053-04-03			
Window Semiamplitude [days]	Time step [h]	Δv optimized [km/s]	Computational time [s]
10	24	23.9972	2.8
10	12	23.9972	19.9
2	4.8	24.0004	2.64
2	2.4	23.9996	20
2	1.2	23.9839	159
1	0.6	23.9893	160
1	0.24	23.9900	2562

Table 1.1: Grid search analysis around one sample local minimum

The importance of using a low time step to refine the minimum is noticeable. Besides, increasing in refinement higher exponentially the computational costs, and a monotonical convergence is not guaranteed. (table 1.1)

Gradient Algorithms Optimizers

Fminunc and *fmincon* are gradient based optimizers which converge to a local minimum, given an initial guess. The prior freely converges to the optimum, while the latter is constrained: departure and arrival times have boundaries for every planet, and in particular the boundaries of one planet's dates are shifted from the dates of the consecutive planet of a time equal to the minimum ToF associated to the respective transfer leg.

If the two previous algorithms are directly applied to the local optimum, the iteration converges to the local optimum itself whatever optimizer tolerances are set. This is due to the non-smoothness of the objective function. To avoid this problem, an automatic procedure was implemented to shift the initial guess, namely the three planets' dates, within the refinement window. Nevertheless, both these methods need a trial-and-error procedure to set each time the best options to get meaningful results, and for this reason they are less automatable than GA.

Genetic algorithm

GA was used to refine the minima found by the global GA analysis, where the windows were restricted to a semi-amplitude of 10 days, taken by the other methods. To account for the expected probability of landing on a local minimum, GA was optimized further by altering its options. The approach taken was to force a greater diversity in each GA generation through studying the three GA options: **EliteCount**, **CrossoverFraction**, **PopulationSize**. By changing these, the diversity and mutation of the populations are altered. To decide on the GA options to be chosen, the effect of their change on our problem was studied on a sample minimum.

First, a customized version of the MATLAB built-in function *deterministicstudy* was used for the comparison, where it figuratively presents the behavior of GA over a range of one GA option. (figure 1.6)

The values of **CrossoverFraction** compared are: [0:0.2:1]. It was observed that the best values were obtained at *CrossoverFraction* of: 0.2, 0.6, 0.8.

The values of **EliteCount** compared are: [1:3]. Provided that the default *EliteCount* is 3 when the default population of 50 is set. It was observed that the best values were obtained at an *EliteCount* of 1 & 2.

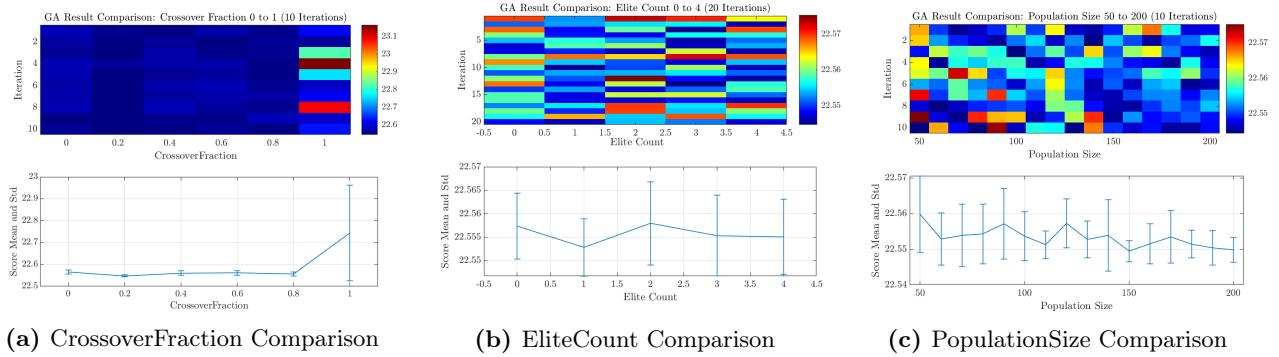


Figure 1.6: Deterministic Study of GA options

The values of **PopulationSize** compared are: [50:10:200]. As the population increases the mean and standard deviation starts to decrease. Yet, the results are not predictable and the decrease of standard deviation is not significant.

A further assessment was made by changing two options at once and searching for the optimum values for 10 GA iterations. The options considered were: *CrossoverFraction* = [0.6:0.05:1] and *EliteCount* = [1:3]. Note that the population was kept as default due to the high computational cost.

From this evaluation, the best results were obtained from the following GA options:

- **Unconstrained** *CrossoverFraction* = 0.8 *EliteCount* = 2
- **Constrained** *CrossoverFraction* = 0.8 *EliteCount* = 1

With the information provided by the above analysis, the GA refinement was made with the following GA option combinations:

Minimum Δv [km/s]	EliteCount: 1	EliteCount: 2
Double constraint	22.5443941142542	22.5442790309273
Single constraint	22.6283810145068	22.6277862325289

Table 1.2: Best results from GA refinement

The results obtained from *EliteCount* = 2 were the optimal solutions. Yet, the difference between the values is in the order of 10^{-3} or 10^{-4} km/s, thus, both GA options are acceptable. For this problem and with the relatively small time span imposed for the refinement, it is thought that the use of default GA options would provide reasonable values.

1.3.3 Refinement Method Comparison

Figure 1.10 presents a comparison of the results obtained from the three refinement methods as well as the Global GA approach. It is observed that the GA refinement provides the best Δv values. The methods were also compared in terms of computational time; the Grid search seems to achieve the best result, yet this is due to the necessity of using a low refinement to have the computation made for all minima in a reasonable time; the gradient based optimizer is actually the most efficient in terms of computational time (see Table 1.3).

Computational time comparison [s]				
	Grid Search	Fmincon	Fminunc	GA
Single Constraint	313.868	1260.065	1395.002	6276.633
Double Constraint	139.385	251.020	294.631	1733.984

Table 1.3: Total computational time for refinement methods used

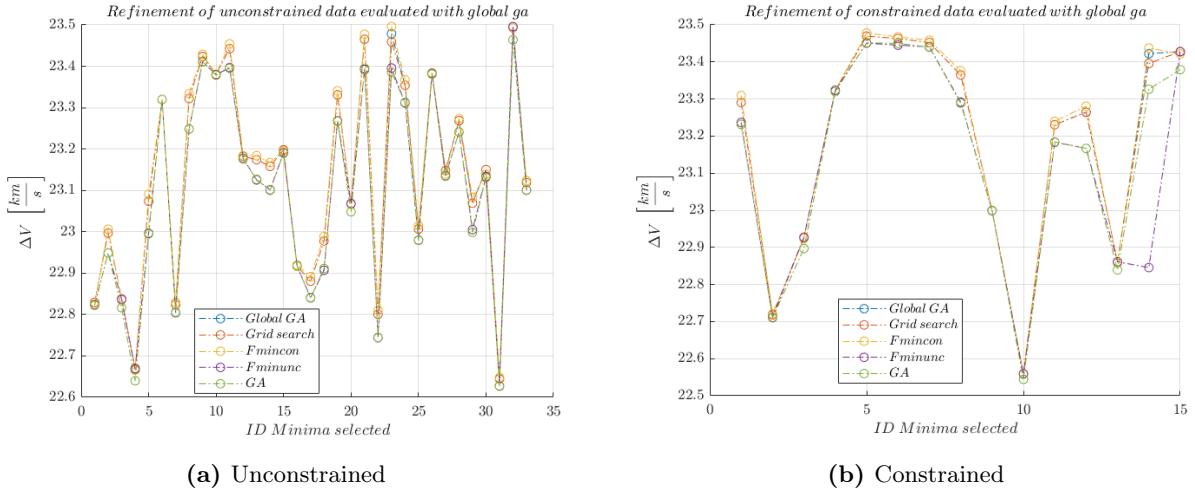


Figure 1.7: Comparison of refinement methods

1.4 Conclusion

After assessing both the Leg Approach and the Global Approach, it was concluded that the Leg Approach is not preferred for finding the optimal solution with the constraints that were set for our problem. Instead, the Global Approach would be the chosen strategy for solving the interplanetary transfer as it presented a significantly better result for our problem. Thus, it would be the chosen strategy for further assessments. The main challenge with the Leg Approach was due to fixing Mercury and Jupiter dates, which stiffened the problem and reduced it to a single degree of freedom. Further implementations of this approach could be realized by relaxing the constraints on Mercury and Jupiter dates, returning to a three degrees of freedom problem, while keeping the computational requirements low.

The frequency analysis allowed the team to have a deeper insight into the physics of the problem. For the mathematical issues already depicted, however, using such an analysis for design purposes is not convenient: the sampling period is imposed by the planets' orbital parameters, thus aliasing related phenomena are not avoidable.

As for the Global Approach, the improvement in the total Δv was significant, compared to that obtained from the Leg Approach. The method of the overlapping windows used in the Global GA approach has proven to be effective for the assessment of a large departure and arrival window. Another potential method for reaching similar values would be increasing the population of the GA optimizer, provided that the computational cost is not of concern during the preliminary design, since increasing this parameter significantly increases the computational cost. Furthermore, the values obtained from the refinement stage were within proximity to the evaluated points, thus supporting the conclusion in favor of the performance of the Global GA Approach.

Comparing the three methods of refinement, the GA one is preferred over the others in terms of the obtained optimal Δv value, yet the Gradient Based refinement method overcomes the others in terms of computational duration.

The grid search approach, despite its low efficiency, is the more robust: robustness is crucial for problems characterized by a high number of local minima, such as the one of this assignment; in turn, however, such a strategy requires high computational costs: these grow with the cube of the number of dates that make up the latus of the refinement grid. The grid search approach can be indeed used for assessment purposes, while heuristic algorithms prove to be more efficient for a whole window analysis.

Concerning the almost vertical disposition of local minima, outlined in Global GA analysis section, it is possible to say that the solution of the interplanetary transfer problem is much more sensitive to Venus and Mercury related dates rather than to Jupiter's one. This can be explained by considering that Venus and Mercury are faster orbiting planets than Jupiter: the temporal scales of these planets' motion are far different, and the smaller temporal scales, namely Mercury and Venus dates, dominate the results.

The choice of Venus as flyby planet has furthermore proven crucial for our problem: such little massive planet requires the gravity assisted satellite to have a low perigee to effectively benefit from the flyby; in turn, the constraint over minimum perigee radius is demanding, because of Venus' small mass. The overall best transfer solution found is, as expectable, exploiting the flyby to the utmost, as the related flyby perigee height is practically equal to the minimum height imposed for avoiding interaction between satellite and Venus's atmosphere.

The optimal result obtained from the preliminary design of the interplanetary study from Mercury to Jupiter, passing by Venus is reported below, as well as the optimal transfer solutions obtained from both Leg and Global Approaches.

	Departure	Fly-by	Arrival
Date and Time	30-Mar-2049 20:13:26	15-Apr-2051 14:29:29	13-Dec-2053 13:08:07
Δv [km/s]	14.32496	0.21563	8.00368
Total ToF:	1718.705 days	Total Δv :	22.54428 km/s

Table 1.4: The final best solution (*flyby time: 21.15 h*)

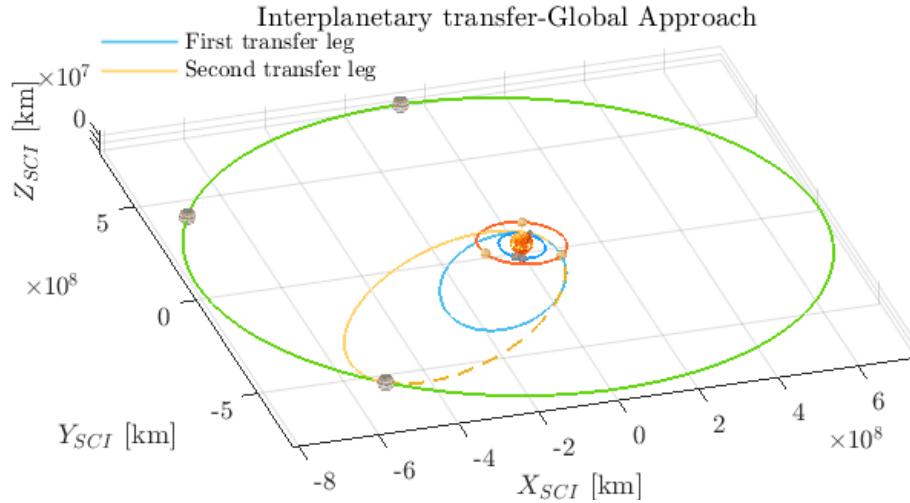


Figure 1.8: Optimal transfer solution, *Global analysis*

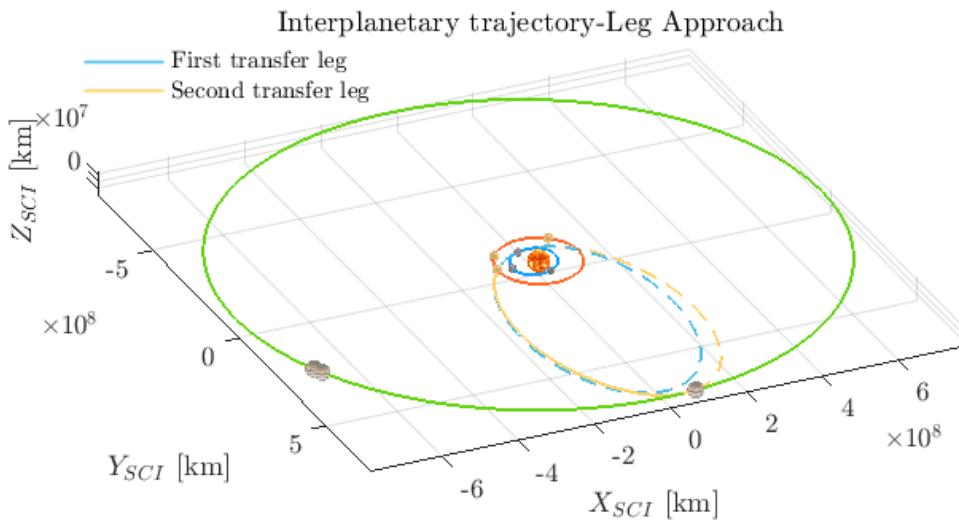


Figure 1.9: Optimal transfer solution, *Leg approach*

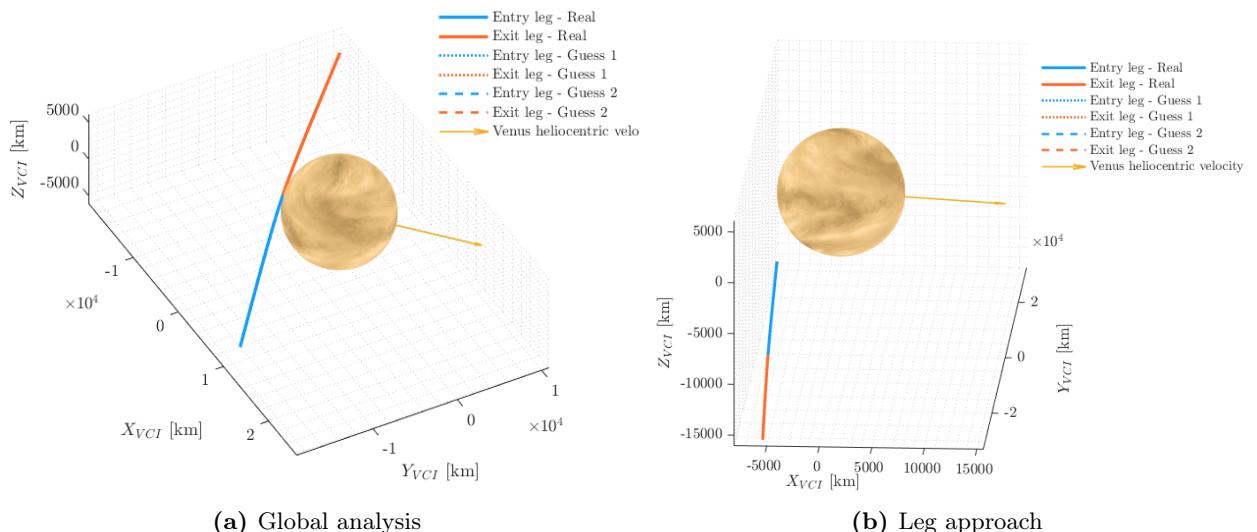


Figure 1.10: Flyby of the best solution, coming from each method

Chapter 2

Planetary Mission

2.1 Description

The request of the assignment is to carry out the orbit analysis and ground track estimation of a Highly Elliptical Orbit. The following study considers the zonal J_2 and the air drag perturbations on the orbit and defines the ground track of the satellite, in case of unperturbed orbit and in case of just J_2 perturbation. A solution to obtain a repeating ground track, both for perturbed and unperturbed case, is proposed.

The orbit evolution is then predicted through two different methods: the direct time integration of the cartesian equations of motion in ECI and the computation of Keplerian elements' variation through Gauss equations (for non-conservative perturbations). The propagation models are used in the end on a real satellite and the results are compared with real life data.

2.1.1 Data

The Earth – centred orbit is characterized by the following parameters, and the data used in the assignment are:

a [km]	e [-]	i [°]	Ω [°]	ω [°]	θ [°]
23210	0.7091	12.6865	0	0	0

h_p [km]	k_m [-]	A_m $\left[\frac{m^2}{kg} \right]$	c_d [-]
6751.789	$12/5$	0.060	2.1

2.1.2 Hypothesis

In all the analysis other non-mentioned perturbations (such as Moon's gravity, Sun's gravity, SRP, higher order gravity perturbations, magnetic field, ...) are not considered.

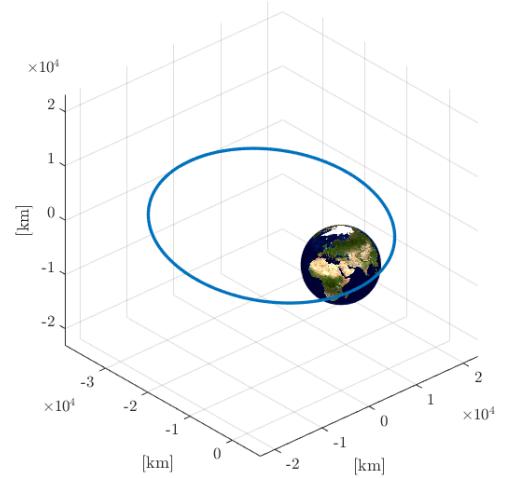


Figure 2.1: Initial Orbit

2.2 Ground Track

The ground track of the satellite is shown in the following figures, for different cases:

- **Unperturbed orbit**, over one revolution, one day and ten days;
- **Perturbed orbit (only J_2 considered)**, over one revolution, one day and ten days.

In all these cases the satellite does not provide a repeating ground track.

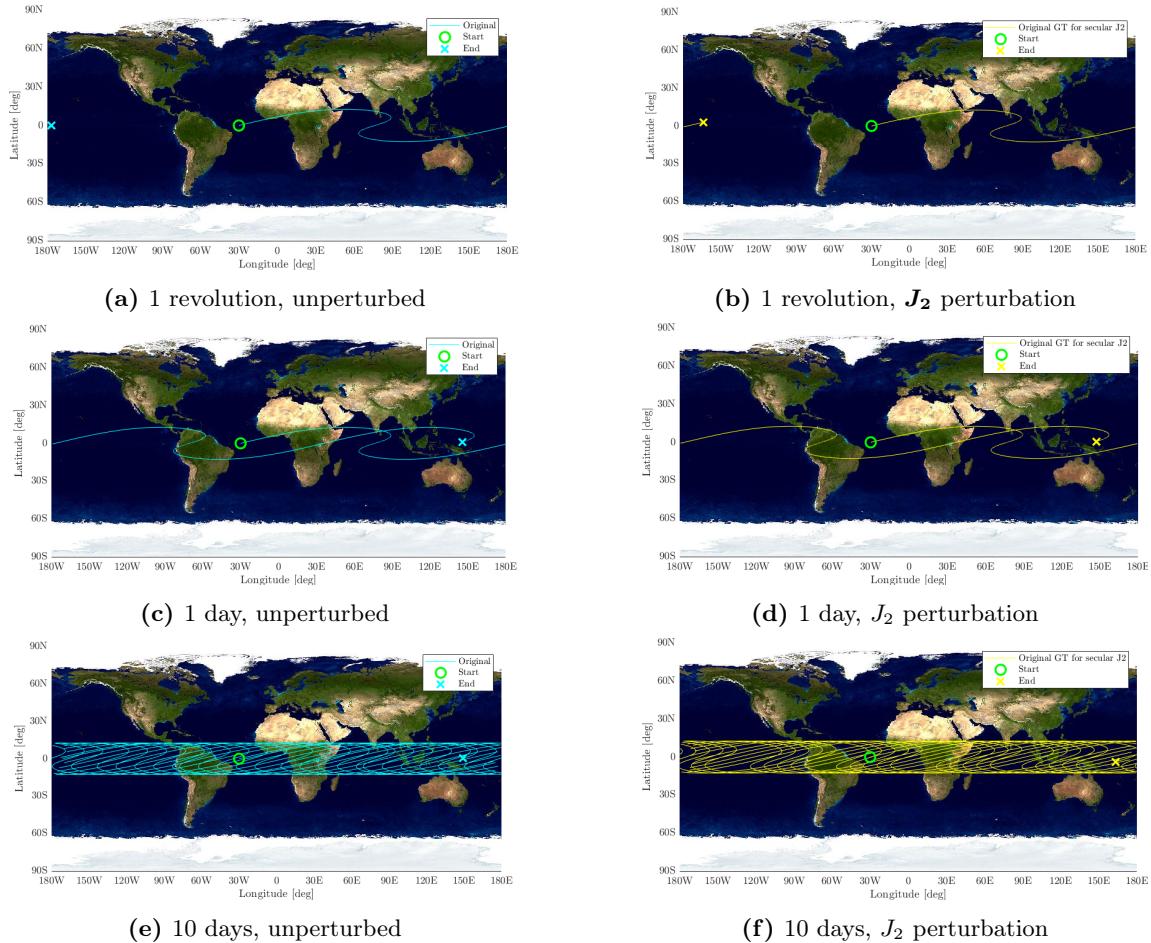


Figure 2.2: Ground Tracks of Perturbed and Unperturbed case

2.2.1 Repeating GT

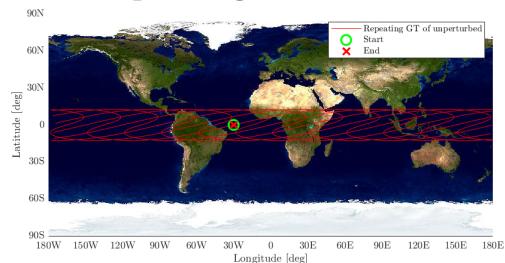


Figure 2.3: Repeating GT, unperturbed

To obtain a repeating ground track, in the unperturbed case, it is sufficient to change the semimajor axis of the orbit accordingly to equation 2.1. As shown in the following calculations, the new semimajor axis only depends on the ratio k/m (as the Earth's rotation rate and the parameter μ are considered constant).

$$a_{rep} = \left(\mu \left(\frac{m}{k \omega_{\oplus}} \right)^2 \right)^{1/3} = 23522.851 \text{ km} \quad (2.1)$$

For the J_2 -perturbed orbit it is possible to vary the semimajor axis in a similar way as the one described before, but the nodal regression, the perigee precession, and the effect on the mean anomaly of the orbit due to the perturbation must be considered:

$$\frac{m}{k} = \frac{\omega_{\oplus} - \dot{\omega}(a_{rep}, e, i)}{n(a_{rep}) + \dot{\omega}(a_{rep}, e, i) + \dot{M}_0(a_{rep}, e, i)} \quad (2.2)$$

This non-linear equation (that comes from the modification of the Greenwich nodal period due to the regression of the node, and the modification of the satellite nodal period due to the perigee precession and \dot{M}_0) is to be solved with respect to a_{rep} to find the semimajor axis for which the ground track will repeat.

$$a_{rep}^{J2} = 23512.338 \text{ km} \quad (2.3)$$

The equation used, though, accounts only for secular variations of the elements, and the propagation with this new semimajor axis still does not provide a repeating ground track. The J_2 perturbation has a significant effect on a HEO, especially if its inclination differs from the critical one (63.4°); the evolution of the elements has a highly oscillating behaviour, as, for example, the semimajor axis osculates of about 80 km around its mean value. For this reason, the choice of an initial guess to start the propagation of the orbit is crucial for all the evaluations. In fact, if the orbit propagation starts with initial conditions of $a = a_{rep}^{J2}$ and $\theta_0 = 0$, the average value of a after the propagation differs from a_{rep}^{J2} of about 71 km (figure 2.5).

For this reason, the propagated orbit with the above-mentioned initial conditions is not repeating yet.

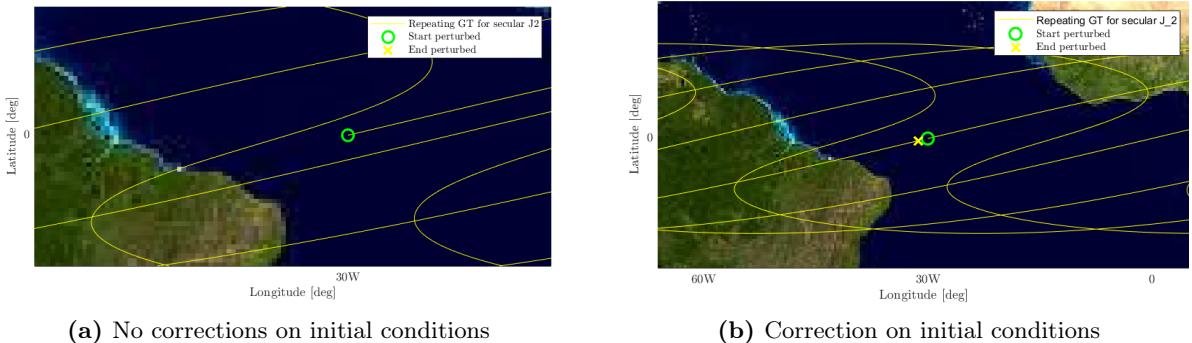


Figure 2.4: Repeating GT, J_2 perturbation

The search for the right initial conditions can be made with two approaches:

- Search for the **initial semimajor axis**, keeping θ_0 constant
- Search for the **initial θ_0** , keeping the initial semimajor axis constant, equal to a_{rep}^{J2} .

In both cases, the initial conditions found are such that the average of the semimajor axis in the first twelve revolutions of the orbit is equal to a_{rep}^{J2} .

The first method was chosen to find the proper initial semimajor axis, for a fixed θ_0 , as it converges faster: the function *iterative3.m* was called iteratively inside a ‘for’ cycle, taking as input the initial semimajor axis (that varies for every iteration) and the other Keplerian elements (kept constant); for every iteration it propagates the orbit with the initial conditions given. It then evaluates the average a_{avg} of the propagation and defines $\Delta a_k = a_{k,avg} - a_{rep}^{J2}$. The output is $a_{k+1}^0 = a_k^0 - \Delta a$, which is given as input as the initial condition of the next iteration. The initial semimajor axis found after ten iterations is $a^0 = 23582.717 \text{ km}$.

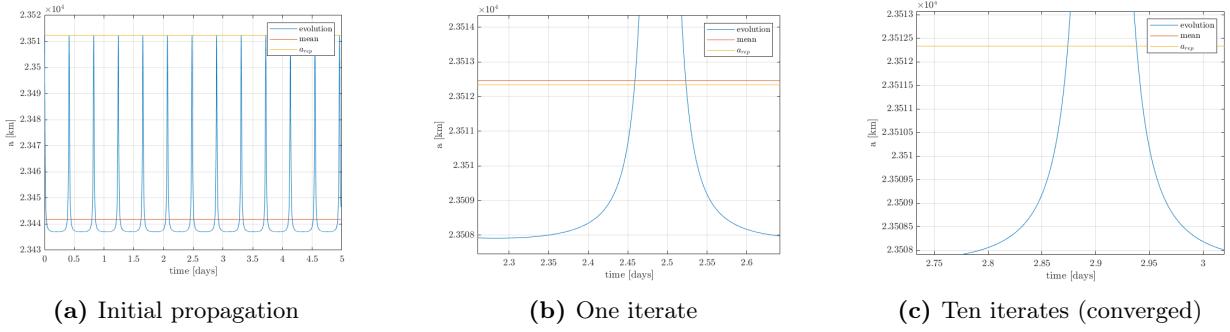


Figure 2.5: Graphical representation of the iterative method

The ground track is approximately repeating: it changes its shape due to the perigee precession, but the starting and the ending point – approximately – coincide. To achieve a highly accurate repeating ground track it would be necessary to also act on the other elements (for example: use the critical inclination to obtain a null perigee precession).

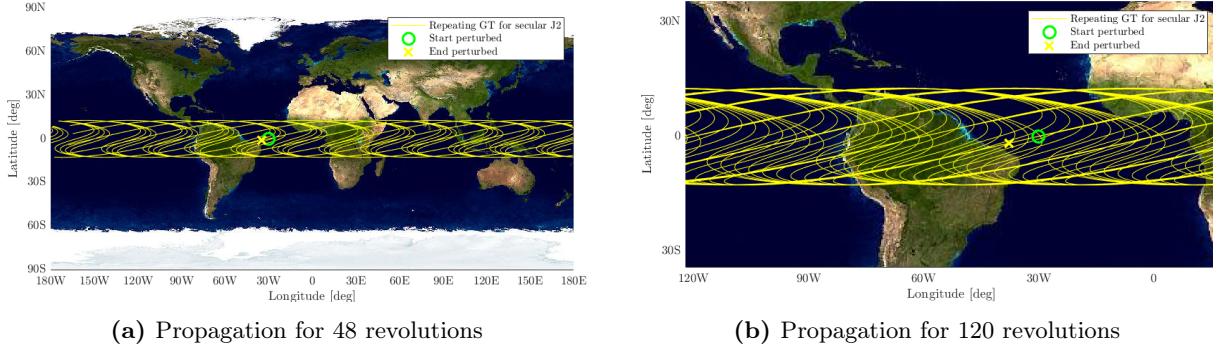


Figure 2.6: Ground track evolution for $\theta_0 = 0^\circ$

The ground track repeats on the ascending nodes: after twelve revolutions it always passes from the same equatorial points, on the same longitudes, even if the shape of the ground track varies whatsoever. Using the right initial conditions, the effect of Ω regression on the ground track is cancelled thanks to a correct evaluation of a_{rep}^{J2} . On the other hand, it is not possible to counteract the argument of periapsis precession, unless the critical inclination is chosen.

Note that whatever initial condition on θ_0 is given, the GT always passes from the same twelve equatorial points, coinciding with the ascending nodes: given as initial condition, for example, $\theta_0 = 180^\circ$, the GT does not repeat on this initial point (descending node), but it will repeat on the same longitudes, that can be seen in figure 2.6 as diagonal, equator-crossing, straight lines.

The initial value of the argument of pericentre has a large influence on the ground track shape, and the initial choice on this Keplerian element leads to different qualitative results; but, because of the secular variation due to J_2 , these different qualitative results will repeat after a perigee precession of 360° (corresponding in our case to about 428 days).

The fact that the GT does not end on the starting point is due to a wrong estimation of the period on which the propagation is done, caused by numerical errors, high oscillating behaviour of the semimajor axis, and perigee precession. One possible way to proceed in evaluating the right period of the propagation would be a semi-analytical method to search for an expression including analytical approximations of the oscillations

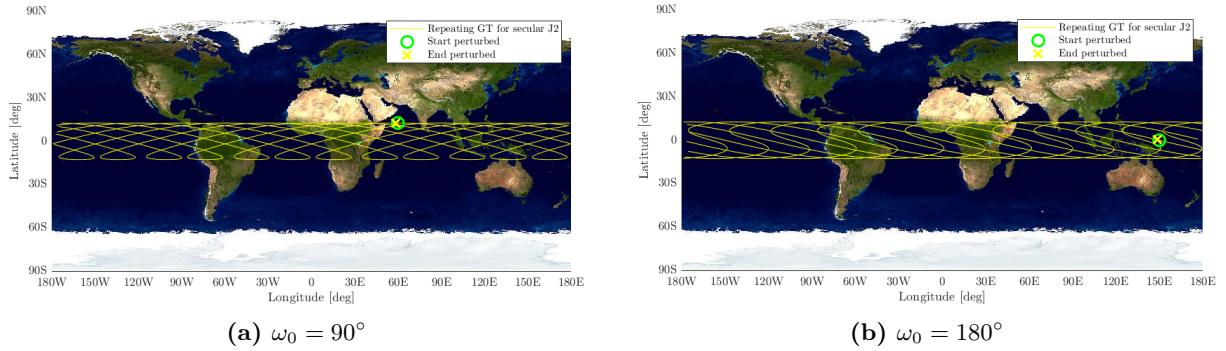


Figure 2.7: Different initial conditions on ω

of the semimajor axis around its mean value. In the propagation, the nodal period was used, according to the following equation:

$$T = \frac{2\pi}{n + \dot{\omega} + \dot{M}_0} \quad (2.4)$$

2.3 Numerical Propagation

The satellite position evolution was predicted using two techniques:

- the numerical integration of the Cartesian equations of motion
- the numerical integration of the Gauss equations for unconservative perturbations that directly evaluate the variation of the Keplerian elements

Both integrations were made with *ode113*, with a relative tolerance of $1e-13$ and an absolute tolerance of $1e-14$. In both cases the propagation was made for a predetermined time, equal to 2000 times the initial orbital period, which corresponds to about 815 days. The Karman line was chosen as the upper limit of the atmosphere, defining the zone where the satellite should not enter, to prevent its crash; the propagator automatically stops if the satellite reaches the altitude of 100 km.

2.3.1 Results

The Keplerian elements evolutions obtained are shown in the following graphs; the normalized error between the two propagations is small: it never reaches an order higher than 10^{-7} . Some elements were filtered to better understand their behaviour (figure 2.10).

To estimate the accuracy of the two methods, through the whole propagation some quantities were calculated. From theory,

- $\mathbf{e} \cdot \mathbf{h}$ is a quantity that should always be null
- The specific energy should be constant when the perturbations are conservative

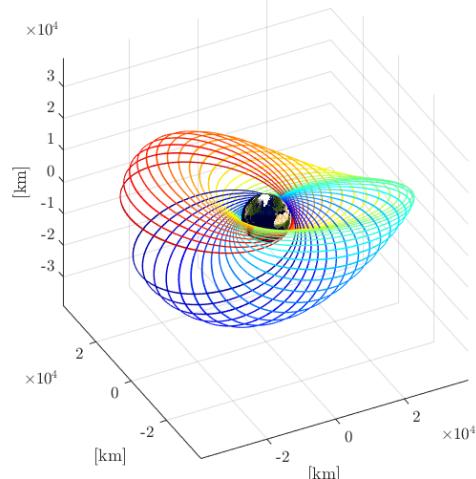


Figure 2.8: Orbit evolution (*blue is the initial orbit*)

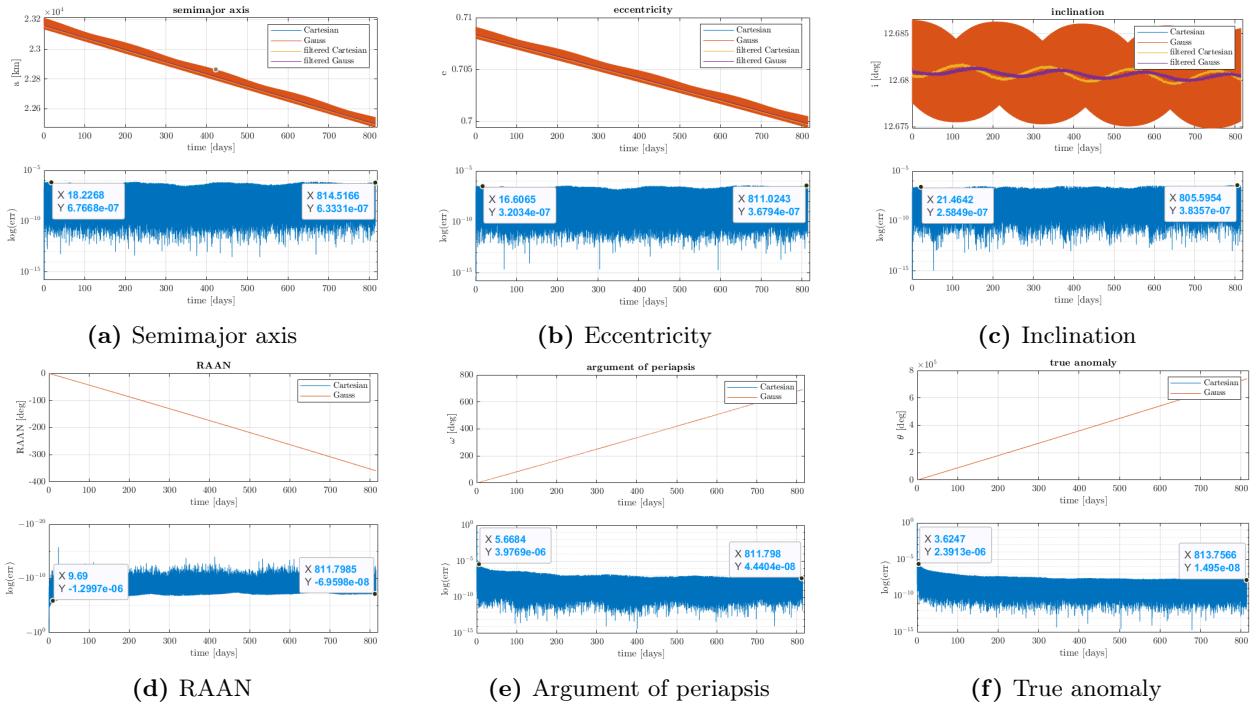


Figure 2.9: Evolution of keplerian elements evaluated with both methods and relative errors (normalized)

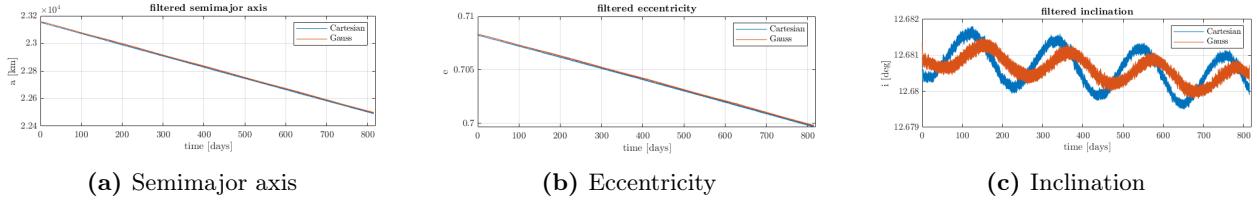


Figure 2.10: Filtered Keplerian elements, highlight

Note that the specific energy must account for both the primary gravitational potential and the J_2 potential; the value of the latter is defined by:

$$R_{ZH}(r, \phi) = J_2 \left(\frac{R_e}{r} \right)^2 P_2(\cos \phi) \quad (2.5)$$

Theoretically, specific energy - if only J_2 is considered - is constant, while it decreases if drag is accounted for, this is because drag is a non-conservative perturbation. Energy decreases rapidly each time the satellite passes through its perigee: in this zone the air density and the velocity of the satellite increase, so the energy dissipation due to drag is higher than in other points. This gives the energy graph a “stepped” behaviour, and in a long term period, the overall trend is approximately linearly – decreasing (figure 2.12).

Comparison between the two methods of propagation was made by evaluating the differences between the Keplerian elements propagated with Cartesian and Gauss schemes at each time step, normalized over the initial value of each element (figure 2.9). As shown in the graphs the error is low, except for the inclination (figure 2.10), where a different filtered behaviour between the two propagations is noticeable (only when drag is considered). The filtering was made with the MATLAB function *movmean.m*, over a four orbital period-long window, to isolate the long term evolution of the elements. Four periods were chosen as the

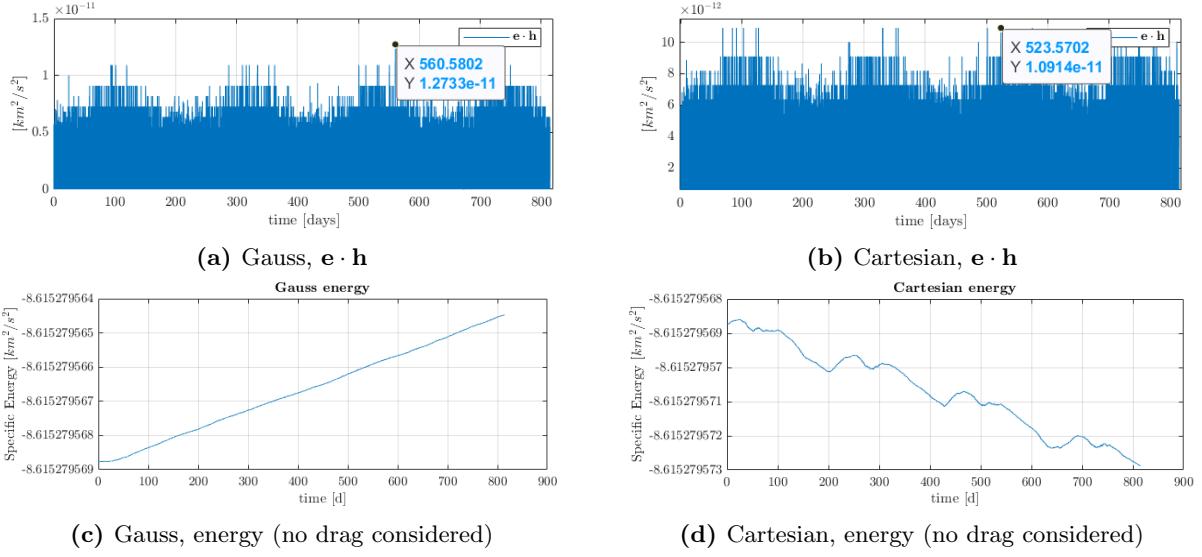


Figure 2.11: Conserving quantities

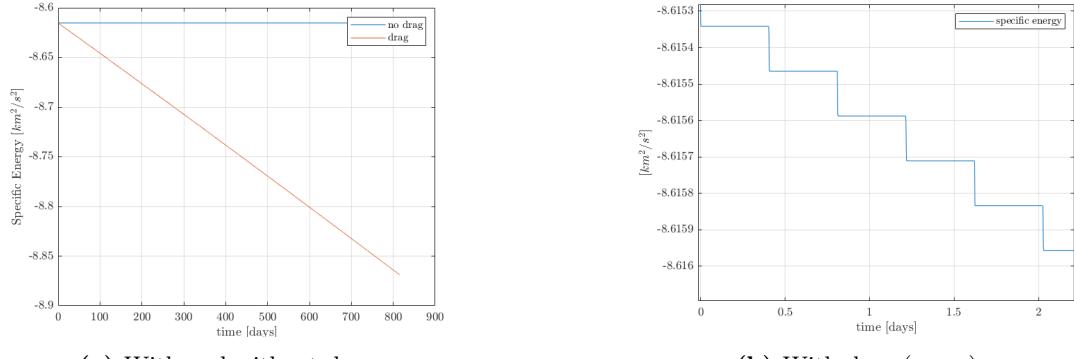


Figure 2.12: Specific energy graphs

minimum time span allowing both for filtered results' smoothness and filtering of the undesired high frequency harmonic contributions.

Gauss propagation was found to be the most efficient method of propagation, as its computational time over long propagations is significantly lower: for a 814 days propagation, Gauss' computational time is 84 minutes, while Cartesian's is 168 minutes, also because the last one evaluates at the end the Keplerian elements from position and velocity.

Both were found to be quite accurate, even if the specific energy in the conservative case varies; the scalar product between eccentricity vector and specific angular momentum is only two orders above the *epsilon machina* (figure 2.11). The highest errors between the two propagation methods were found at the perigee passage of each orbit. To assess a more accurate propagation the team would try and reduce the absolute and relative tolerances in *ode113*'s options.

The J_2 perturbation provides a considerable effect on the orbit - which has quite a low perigee altitude of 380 km - and constantly varies its Keplerian elements: the actions of the J_2 perturbations (figure 2.13) are indeed a summation of a high-frequency contribution (the elements vary constantly, even in just one orbit) and a secular contribution. The high frequency variations affect all elements, while the secular variation is linear and affects only Ω , ω and M_0 (if only J_2 is considered).

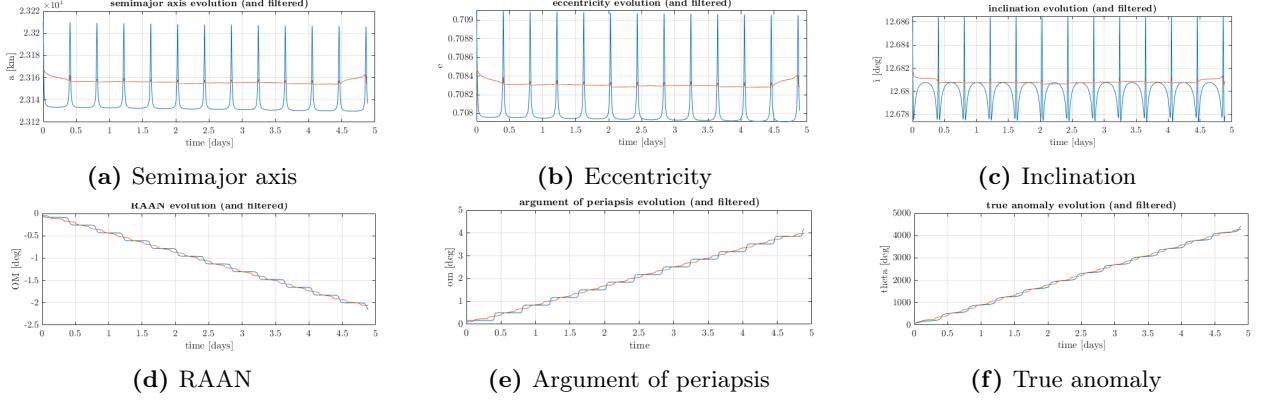


Figure 2.13: High Frequency oscillations and filtering

Filtering the high frequency evolution using an average-based method (figure 2.10) permits to distinguish the long term and secular variations of the elements. Specifically, semimajor axis and eccentricity decrease linearly due to drag, and Ω and ω have a secular linear variation due to J_2 perturbation. Inclination has a particular behaviour, presenting drag-related secular evolution, and J_2 related long term and short term evolution patterns. Figure 2.14 shows the filtered evolution of this element when no drag is considered: the linear secular evolution is cancelled. This is thought to be due to the atmosphere's rotation (fixed to Earth), which contributes with an out-of-plane perturbing acceleration.

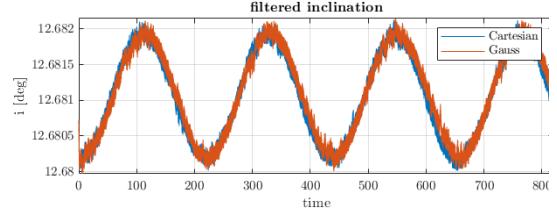


Figure 2.14: Filtered inclination, no drag

2.4 Real Satellite Comparison

To check the accuracy of the model used for this assignment, a comparison between numerical integration and real data was made. The Molniya 1/36 is a satellite launched in 1977 and whose orbit is very similar to the one analysed in this assignment apart from i , as no similar inclinations were found for HEOs with similar a and e .

a [km]	e [-]	i [°]	h_p [km]
26538	0.7437	62.97	425

Its known orbital parameters, extracted from the TLEs, were compared to the orbital parameters obtained from the Gauss propagation method. The propagation was performed for a period of 1200 days, starting from the same initial condition of the real case, in a selected window of time where the elements of the real satellite were found to be more regular (to avoid the presence of manoeuvres or general irregularities) (figure 2.15).

2.4.1 Observations

As shown in the graphs – besides the argument of perigee and the RAAN – the model does not suit reality: many other perturbations – not accounted for in this assignment – must be considered to get closer to the real behaviour of the satellite, such as the third body perturbation, SRP, magnetic field, etc. For this reason,

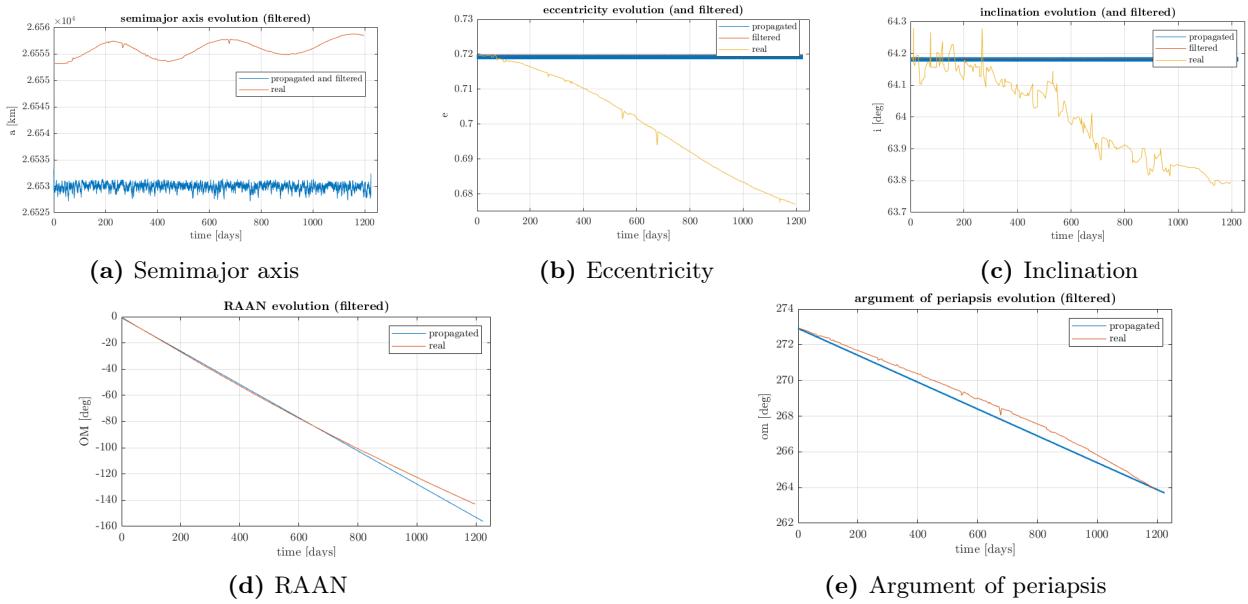


Figure 2.15: Comparison between real data and Propagated data

the real and propagated graphs differ considerably.

The true anomaly is not represented, because the TLEs are not continuous sets of data: in this case, the satellite sent its ephemeris everytime it passed around the angular position of 70° , so the sampling time of this keplerian element was too high to have a comparable behaviour with the Gauss propagation.

2.5 Conclusions

The planetary assignment was a didactical way to understand the complexity of the space world, where many different phenomena must be considered for control and prediction purposes. The regular, Keplerian models used to reduce the complexity of many possible space-related tasks are to be integrated with the variety of perturbations present in the real world. For example, just by considering one zonal perturbation, many problems arise in the design of the ground track of a satellite, whose perturbed orbit differs from the simplistic Keplerian one. All elements vary, transforming the steady orbit into an osculating one, and turning the tasks – such as GT design – into multifield problems, where often the solution can only be found with semi-analytical methods, or even with optimization techniques, such as genetic algorithms.

Because of the complexity of the problem, an analytical approach is impossible, and an accurate prediction of the trajectory of the satellite can be performed only with numerical methods such as the previously mentioned Gauss and Cartesian propagation. To have the best representation of reality, it is necessary to implement and consider all the possible perturbations that one orbit can be affected by.

This Highly Elliptical Orbit is subjected to high variations of its elements with just one perturbation considered. The propagated results considerably differ from reality because such an orbit is sensitive to other perturbations; having a high apogee radius makes it necessary to consider Solar radiation pressure and third body gravity perturbations (Moon and Sun). To get closer to reality starting by the model already implemented, the team would proceed by modelling all the other known perturbations (SRP, higher order zonal and tesseral harmonics, moon and sun perturbations, magnetic field). The final result accuracy, though, would be limited by all the errors arising during the analysis: numerical errors, modelling errors (e.g. atmosphere, unknown phenomena), and statistical variability.