

SGN – Assignment #1

Edoardo Mensi Weingrill, 244706

1 Periodic orbit

Exercise 1

Consider the 3D Earth–Moon Circular Restricted Three-Body Problem with $\mu = 0.012150$. Note that the CRTBP has an integral of motion, that is, the Jacobi constant

$$C(x, y, z, v_x, v_y, v_z) := 2\Omega(x, y, z) - v^2 = C$$

where $\Omega(x, y, z) = \frac{1}{2}(x^2 + y^2) + \frac{1-\mu}{r_1} + \frac{\mu}{r_2} + \frac{1}{2}\mu(1-\mu)$ and $v^2 = v_x^2 + v_y^2 + v_z^2$.

- 1) Find the coordinates of the five Lagrange points L_i in the rotating, adimensional reference frame with at least 10-digit accuracy and report their Jacobi constant C_i .

Solutions to the 3D CRTBP satisfy the symmetry

$$\mathcal{S} : (x, y, z, \dot{x}, \dot{y}, \dot{z}, t) \rightarrow (x, -y, z, -\dot{x}, \dot{y}, -\dot{z}, -t).$$

Thus, a trajectory that crosses perpendicularly the $y = 0$ plane twice is a periodic orbit.

- 2) Given the initial guess $\mathbf{x}_0 = (x_0, y_0, z_0, v_{x0}, v_{y0}, v_{z0})$, with

$$\begin{aligned} x_0 &= 1.068792441776 \\ y_0 &= 0 \\ z_0 &= 0.071093328515 \\ v_{x0} &= 0 \\ v_{y0} &= 0.319422926485 \\ v_{z0} &= 0 \end{aligned}$$

Find the periodic halo orbit having a Jacobi Constant $C = 3.09$; that is, develop the theoretical framework and implement a differential correction scheme that uses the STM, either approximated through finite differences **or** achieved by integrating the variational equation.

Hint: Consider working on $\varphi(\mathbf{x} + \Delta\mathbf{x}, t + \Delta t)$ and $J(\mathbf{x} + \Delta\mathbf{x})$ and then enforce perpendicular cross of $y = 0$ and Jacobi energy.

The periodic orbits in the CRTBP exist in families. These can be computed by ‘continuing’ the orbits along one coordinate or one parameter, e.g., the Jacobi energy C . The *numerical continuation* is an iterative process in which the desired variable is *gradually* varied, while the rest of the initial guess is taken from the solution of the previous iteration, thus aiding the convergence process.

- 3) By gradually decreasing C and using numerical continuation, compute the families of halo orbits until $C = 3.04$.

(8 points)

1.1 Lagrangian Points Coordinates

The computation of the five Lagrangian points L in the rotating adimensional reference frame begins with the scalar potential $\Omega(x, y, z)$ for the 3D CRTPBP. Given μ and defining $r_1 = \sqrt{(x + \mu)^2 + y^2 + z^2}$ and $r_2 = \sqrt{(x + \mu - 1)^2 + y^2 + z^2}$, the scalar potential $\Omega(x, y, z)$ is defined accordingly. The Lagrangian Points, which represent the equilibrium points of the CRTBP, are determined by setting the partial derivatives of the scalar potential $\Omega(x, y, z)$ to zero. Due to the planar nature of the CRTBP, the z components is zero. Indeed, the equations that need to be solved are:

$$\frac{\partial U}{\partial x} = \frac{\partial U}{\partial y} = 0 \quad (1)$$

The three co-linear Lagrangian points (L_1, L_2, L_3) are given by the solution of $\frac{\partial U}{\partial x} = 0$, as they lie on the x -axis of the Earth-Moon rotating adimensional reference frame, hence even y is zero. For the triangular Lagrangian points (L_4, L_5), the two partial derivatives need to be satisfied simultaneously.

The solver used is MATLAB's `fsoolve`, with function and optimality tolerance both of 1e-10 to guarantee an accuracy until 10-digits, as required for the results. The optimality tolerance ensures a stable and highly accurate solution not affected by small numerical errors. For each point, an initial guess in the surrounding of the expected solution is given to the solver to ensure convergence.

In Fig. I are reported the coordinates of the five Lagrangian points computed in the Earth-Moon rotating frame (EMRF) $x-y$ plane, along with Earth and Moon position. Moreover, the unitary circumference centered respectively on Earth and Moon as well as the curves representing the partial derivative over x are plotted to highlight the characteristics of the Lagrangian points.

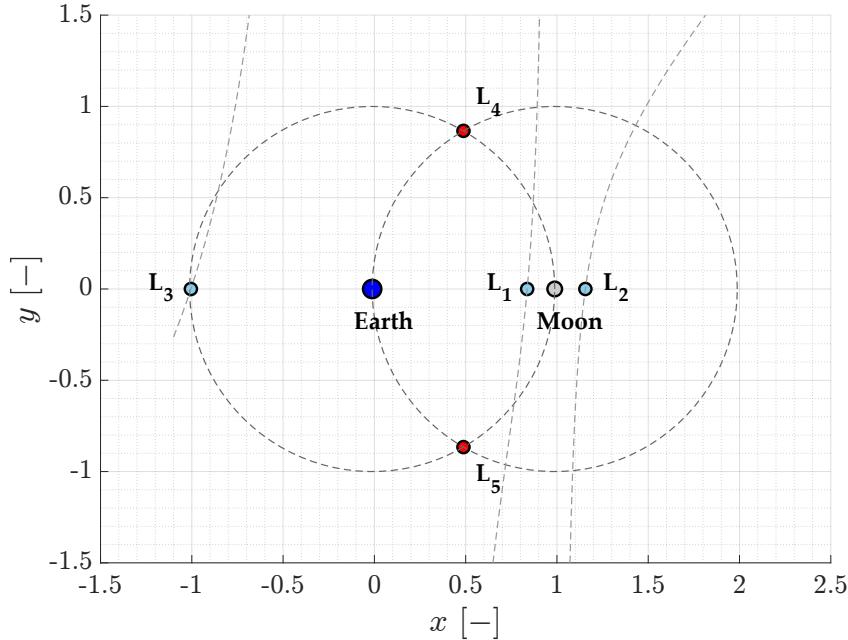


Figure 1: Lagrangian Points Coordinates @ (EMB) Earth-Moon Rotating Frame

Since the Lagrangian points are the equilibrium points of the system, their velocity is zero. Therefore the Jacobi constant can be easily retrieved as: $C_i = 2 U(x_i, y_i)$ for the i-th Lagrangian point.

In Table I for each Lagrangian point are reported the coordinates and the associated Jacobi Constant.

	L_1	L_2	L_3	L_4	L_5
x	+0.8369180073	+1.1556799131	-1.0050624018	+0.4878500000	+0.4878500000
y	0.00000000000	0.00000000000	0.00000000000	+0.8660254038	-0.8660254038
C	+3.2003380950	+3.1841582164	+3.0241489429	+3.00000000000	+3.00000000000

Table 1: Lagrangian points coordinates and Jacobi constants

1.2 HALO Orbit with $C = 3.09$

The objective of this point is to compute a periodic orbit around Lagrangian point L_2 in the Earth-Moon CRTBP with a target Jacobi Constant. Due to the simmetry of the system, the condition for an orbit in CRTBP to be periodic is that it crosses the $y = 0$ plane twice.

Starting from an initial guess \mathbf{x}_0 , and including the event function for the $y = 0$ condition, the propagation time must be set sufficiently large to ensure the periodic orbit condition is met; otherwise, the integration may terminate prematurely. Then, the orbit has been propagated by numerically integrating the equations of motion for the Circular Restricted Three Body Problem:

$$\mathbf{f} = \begin{cases} \dot{x} &= v_x \\ \dot{y} &= v_y \\ \dot{z} &= v_z \\ \dot{v}_x &= 2v_y + \frac{\partial U}{\partial x} \\ \dot{v}_y &= -2v_x + \frac{\partial U}{\partial y} \\ \dot{v}_z &= \frac{\partial U}{\partial z}, \end{cases} \quad (2)$$

In order to correct the initial guess to obtain the requested HALO orbit, an iterative differential correction scheme has been implemented, based on the first-order Taylor expansion of the flow with respect to the initial state and the final time, presented in Eq. (3). In the scheme, the event function and the value of the current Jacobi Constant are monitored during integration to verify whether the required periodicity condition and target energy level have been achieved.

$$\varphi(\mathbf{x}_0 + \delta\mathbf{x}_0, t + \delta t) \approx \varphi(\mathbf{x}_0, t_f) + \frac{\partial \varphi}{\partial \mathbf{x}_0} \delta\mathbf{x} + \frac{\partial \varphi}{\partial t} \delta t, \quad (3)$$

The derivative of the flow with respect to the initial state can be computed introducing the State Transition Matrix (STM or $\Phi(t)$). The STM provides a linear mapping of small perturbations in the initial state to the final state, effectively linearizing the flow around the current trajectory. The STM has been computed through the variational approach with the equations detailed in Eq. (4), as it is more accurate with respect to finite difference method.

$$\begin{cases} \dot{\Phi} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}(t)\Phi \\ \Phi(0) = I_{6 \times 6} \end{cases} \quad (4)$$

The derivative of the flow with respect to the time is exactly the derivative of the state at the final time $\dot{\mathbf{x}}(t_f)$.

To complete the basis of the correction scheme implemented, also the Jacobi Constant has been linearized around the initial state \mathbf{x}_0 through the first-order Taylor expansion:

$$C(\mathbf{x}_0 + \delta\mathbf{x}_0) = C(\mathbf{x}_0) + \frac{\partial C}{\partial \mathbf{x}_0} \delta\mathbf{x}_0 \quad (5)$$

With all the theoretical framework defined, the next step is to implement the differential correction scheme. The purpose of this scheme is to iteratively refine initial state \mathbf{x}_0 and final time t_f

in order to match the conditions required (periodic orbit and target Jacobi Constant C_{target}). First of all, the error vector, which measures the deviations from the expected solution in terms of periodicity and target energy level, needs to be defined. Starting from an augmented linear system $[7 \times 7]$, comprising the six components of the state and time, the error vector has been computed as:

$$\mathbf{e} = \mathbf{A}\boldsymbol{\delta}, \quad \mathbf{A}_{[7 \times 7]} = \begin{bmatrix} \Phi(t_f)_{[6 \times 6]} & \dot{\mathbf{x}}_f \\ \nabla C_{[1 \times 6]} & 0 \end{bmatrix} \quad (6)$$

where $\boldsymbol{\delta}$ are the corrections required, ∇C is the gradient of the Jacobi Constant with respect to the state variables, while the zero value comes from the fact that C does not depend explicitly on time.

The global $[7 \times 7]$ system can be reduced to a $[4 \times 4]$ system as the constrained variables are only y, v_x, v_z , and C . The first three variables are directly linked with the periodicity condition, as they all need to be zero, while C needs to match C_{target} . Solving the full system would lead to infinite solutions, as it is under-constrained. The reduction of the system to the constrained variables and limiting the corrections to the non-constrained variables $(x_0, z_0, v_{y,0}, t_f)$, which represent the degrees of freedom available to correct the trajectory, ensures a well-posed system with a unique solution. To perform this reduction, only the rows of \mathbf{A} matrix associated with the constrained variables are considered. Regarding the corrections $\boldsymbol{\delta}$, only the columns of \mathbf{A} associated with the non-constrained variables need to be used. The resulting system becomes:

$$\begin{bmatrix} \delta x_0 \\ \delta z_0 \\ \delta v_{y,0} \\ \delta t_f \end{bmatrix}_{k+1} = \begin{bmatrix} x_0 \\ z_0 \\ v_{y,0} \\ t_f \end{bmatrix}_k + \begin{bmatrix} \Phi_{21} & \Phi_{23} & \Phi_{25} & \dot{y}(t_f) \\ \Phi_{41} & \Phi_{43} & \Phi_{45} & \dot{v}_x(t_f) \\ \Phi_{61} & \Phi_{63} & \Phi_{65} & \dot{v}_z(t_f) \\ \frac{\partial C}{\partial x}(\mathbf{x}_0) & \frac{\partial C}{\partial z}(\mathbf{x}_0) & \frac{\partial C}{\partial v_y}(\mathbf{x}_0) & 0 \end{bmatrix}_k^{-1} \begin{bmatrix} 0 - y_f \\ 0 - v_{x,f} \\ 0 - v_{z,f} \\ J_{target} - J_f \end{bmatrix}_k \quad (7)$$

This reduction does not affect the validity of the solution, while ensuring higher computation efficiency by considering only the corrections required to satisfy the constraints.

The system reported was solved iteratively using a pseudo-Newton method, updating initial state and final time until all the conditions are satisfied with a tolerance of 10^{-12} . In Table 2 the corrected initial state is reported.

x	y	z	v_x	v_y	v_z
1.0590402077	0.0000000000	0.07392773779	0.0000000000	0.3469245709	0.0000000000

Table 2: Corrected initial state of the halo orbit with $J = 3.09$

The corrected initial state has been propagated to visualize the periodic orbit around L_2 presented in Fig. 2.

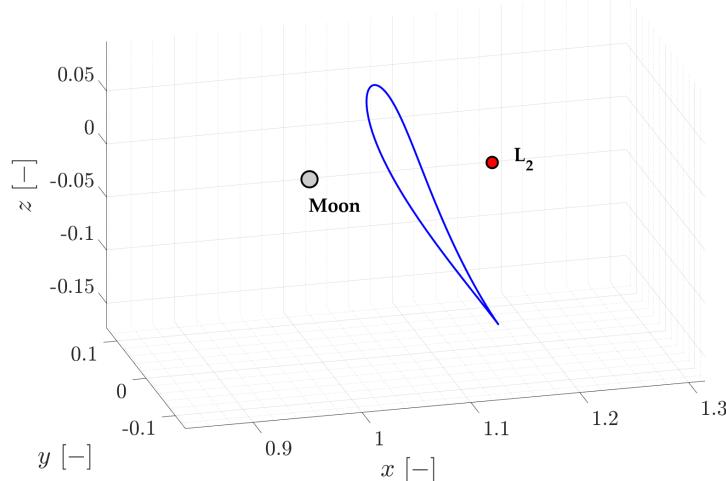


Figure 2: HALO Orbit with $C = 3.09$ @ (EMB) Earth-Moon Rotating Frame

1.3 Families of HALO Orbits

The objective is to compute the HALO orbit having $C = 3.04$ exploiting the numerical continuation method. It consists in varying a parameter (C in this case) incrementally, leveraging the solution at one iteration as the initial guess for the next iteration. In this way, convergence is ensured by keeping the initial guess always close to the solution.

In the context of the exercise, the Jacobi Constant has been reduced from $C = 3.09$ to the target value $C = 3.04$ by a small step ($\Delta C = -0.005$). For each new C , the same differential correction scheme as in point 1.2 has been applied to compute the HALO orbit. The process is repeated iteratively until the final condition is reached. In Table 3 the corrected initial state of the HALO orbit with $C = 3.04$ is reported.

x	y	z	v_x	v_y	v_z
1.0125655235	0.0000000000	0.0672339583	0.0000000000	0.5103251959	0.0000000000

Table 3: Corrected initial state of the halo orbit with $C = 3.04$

The Fig. 3 illustrates the family of HALO orbits around L_2 , obtained by propagating the corrected initial state at each iteration of the differential correction scheme, while varying the Jacobi constant C incrementally.

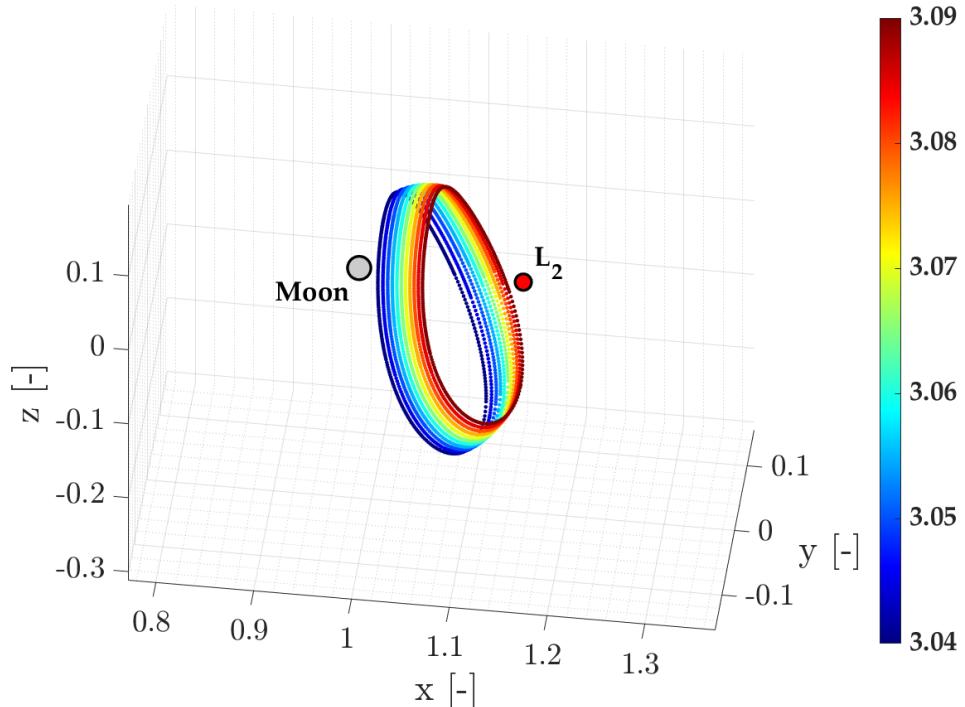


Figure 3: Families of HALO Orbits with C decreasing from $C = 3.09$ to $C = 3.04$. The colorbar on the right indicates the Jacobi Constant associated with each orbit. @ (EMB) Earth-Moon Rotating Frame

As could be noticed from the plot, while the Jacobi Constant C decreases the HALO orbits are larger and more elongated. Indeed, the decrement of C corresponds to a higher kinetic energy, enabling the trajectory to reach regions farther from the equilibrium point L_2 .

2 Impulsive guidance

Exercise 2

Consider the two-impulse transfer problem stated in Section 3.1 (Topputo, 2013)^{*}

- 1) Using the procedure in Section 3.2, produce a first guess solution using $\alpha = 0.2\pi$, $\beta = 1.41$, $\delta = 4$, and $t_i = 2$. Plot the solution in both the rotating frame and Earth-centered inertial frame (see Appendix 1 in (Topputo, 2013)). Consider the parameters listed in Table 4 and extract the radius and gravitational parameters of the Earth and Moon from the provided kernels and use the latter to compute the parameter μ .

Symbol	Value	Units	Meaning
m_s	3.28900541×10^5	-	Scaled mass of the Sun
ρ	3.88811143×10^2	-	Scaled Sun-(Earth+Moon) distance
ω_s	$-9.25195985 \times 10^{-1}$	-	Scaled angular velocity of the Sun
ω_{em}	$2.66186135 \times 10^{-1}$	s^{-1}	Earth-Moon angular velocity
l_{em}	3.84405×10^8	m	Earth-Moon distance
h_i	167	km	Altitude of departure orbit
h_f	100	km	Altitude of arrival orbit
DU	3.84405000×10^5	km	Distance Unit
TU	4.34256461	days	Time Unit
VU	1.02454018	km/s	Velocity Unit

Table 4: Constants to be considered to solve the PBRFBP. The units of distance, time, and velocity are used to map scaled quantities into physical units.

- 2) Considering the first guess in 1) and using $\{\mathbf{x}_i, t_i, t_f\}$ as variables, solve the problem in Section 3.1 with simple shooting in the following cases
 - a) without providing any derivative to the solver, and
 - b) by providing the derivatives and by estimating the state transition matrix with variational equations.
- 3) Considering the first guess solution in 1) and the procedure in Section 3.3, solve the problem with multiple shooting taking $N = 4$ and using the variational equation to compute the Jacobian of the nonlinear equality constraints.
- 4) Perform an n-body propagation using the solution $\{\mathbf{x}_i, t_i, t_f\}$ obtained in point 2), transformed in Earth-centered inertial frame and into physical units. To move from 2-D to 3-D, assume that the position and velocity components in inertial frame are $r_z(t_i) = 0$ and $v_z(t_i) = 0$. To perform the propagation it is necessary to identify the epoch t_i . This can be done by mapping the relative position of the Earth, Moon and Sun in the PCRTBP to a similar condition in the real world:
 - a) Consider the definition of $\theta(t)$ provided in Section 2.2 to compute the angle $\theta_i = \theta(t_i)$. Note that this angle corresponds to the angle between the rotating frame x -axis, aligned to the position vector from the Earth-Moon System Barycenter (EMB) to the Moon, and the Sun direction.
 - b) The angle θ ranges between $[0, 2\pi]$ and it covers this domain in approximately the revolution period of the Moon around the Earth.

*F. Topputo, “On optimal two-impulse Earth–Moon transfers in a four-body model”, Celestial Mechanics and Dynamical Astronomy, Vol. 117, pp. 279–313, 2013, DOI: [10.1007/s10569-013-9513-8](https://doi.org/10.1007/s10569-013-9513-8)

- c) Solve a zero-finding problem to determine the epoch at which the angle Moon-EMB-Sun is equal to θ_i , considering as starting epoch 2024 Sep 28 00:00:00.000 TDB.

Hints: Exploit the SPK kernels to define the orientation of the rotating frame axes in the inertial frame for an epoch t . Consider only the projection of the EMB-Sun position vector onto the so-defined x-y plane to compute the angle (planar motion).

Plot the propagated orbit and compare it to the previously found solutions.

(11 points)

2.1 First Guess Solution

The aim of this section is to produce a first guess solution for the two-impulse Earth-Moon transfer within the Planar Bicircular Restricted Four-Body Problem (PBRFBP) using the given parameters $(\alpha, \beta, \delta, t_i)$. The procedure followed is the one described in Section 3.1 (Topputto, 2013). First, some initial parameters need to be computed, as the scaled departure orbit radius r_0 and velocity v_0 along with the gravitational constants of Moon and Earth and the mass parameter μ . Knowing Earth's radius, the altitude of the departure orbit, and the distance unit DU , r_0 can easily retrieved. The velocity is computed as: $v_0 = \beta \sqrt{(1 - \mu)/r_0}$. Once defined all the parameters, the initial state has been computed as:

$$\mathbf{x}_0 = \begin{bmatrix} x_0 \\ y_0 \\ v_{x,0} \\ v_{y,0} \end{bmatrix} = \begin{bmatrix} r_0 \cos(\alpha) - \mu \\ r_0 \sin(\alpha) \\ -(v_0 - r_0) \sin(\alpha) \\ (v_0 - r_0) \cos(\alpha) \end{bmatrix}. \quad (8)$$

The resulting initial guess is reported in Table 5.

$r_{x,0}$ [DU]	$r_{y,0}$ [DU]	$v_{x,0}$ [VU]	$v_{y,0}$ [VU]
0.001624	0.010008	-6.302738	8.674975

Table 5: Initial guess in Earth-Moon rotating frame.

Then, the initial guess \mathbf{x}_0 obtained has been propagated by numerically integrating the equations of motion of PBRFBP from t_i to $t_i + \delta$. The trajectory obtained has also been rotated in the Earth-Centered Inertial (ECI) Frame. The trajectory, either in the Earth-Moon Rotating Frame (EMRF) and in the ECI Frame, is presented in Fig. 4.

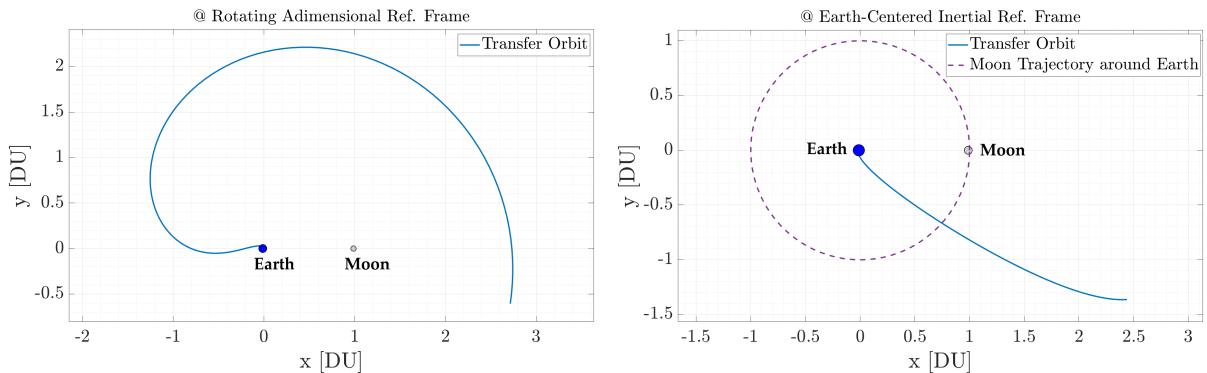


Figure 4: Trajectory in @(EMB) Earth-Moon Rotating Frame (left) and in @(Earth) ECI frame (right)

As can be noticed, while the initial guess matches perfectly the initial parking orbit constraints, the propagation leads to a final state that is far from the required final parking orbit. Hence, the initial guess obtained can only be used as a starting point for the following exercises.

2.2 Simple Shooting

The aim of this section is to solve the two-impulse Earth-Moon transfer problem by using Simple Shooting method, a technique commonly employed to solve boundary value problems (BVP) in trajectory optimization. The goal of the optimization process is to determine the optimal initial state \mathbf{x}_i and propagation time (t_i, t_f) , which are the optimization variables, so that the corresponding trajectory meets the initial and final constraints in terms of position and velocity, represented by the departure and arrival orbits, while minimizing the overall ΔV of the transfer. Starting from the initial guess computed at the previous point, and selecting the set of optimization variables \mathbf{x}_i, t_i, t_f , the cost function J to minimize is:

$$J = \Delta V_{tot} = \Delta V_1(\mathbf{x}_0) + \Delta V_2(\varphi(\mathbf{x}_0, t_i; t_f)) \quad (9)$$

where ΔV_1 is the change in velocity required to match the initial spacecraft's velocity with the parking orbit one. Similarly, ΔV_2 is the change in velocity to match the arrival orbit velocity. To ensure the spacecraft's trajectory meets the mission requirements —connecting the defined departure orbit to the final orbit — a set of equality constraints \mathbf{c}_{eq} is introduced. These constraints enforce both position and velocity conditions at the initial and final points of the transfer, and are expressed as:

$$\mathbf{c}_{eq} = \begin{bmatrix} (x_i + \mu)^2 + y_i^2 - r_i^2 \\ (x_i + \mu)(v_{x_i} - y_i) + y_i(v_{y_i} + x_i + \mu) \\ (x_f + \mu - 1)^2 + y_f^2 - r_f^2 \\ (x_f + \mu - 1)(v_{x_f} - y_f) + y_f(v_{y_f} + x_f + \mu - 1) \end{bmatrix} = 0, \quad (10)$$

where the first two equations define the constraints at departure (ψ_i), ensuring the spacecraft starts on the initial circular orbit, while the latter two define the constraints at the arrival (ψ_f), ensuring the trajectory meets the final circular orbit conditions around the Moon. Moreover, an inequality constraint on the time of flight is introduced, to check that the solution is not unfeasible ($t_i < t_f$). To retrieve the final position \mathbf{x}_f , the initial state has been propagated from t_i to t_f by numerically integrating the PBRFBP dynamics. The problem described can be expressed as a Non-Linear Problem (NLP), stated as:

$$\min_{\mathbf{x}_i, t_i, t_f} J(\mathbf{x}_i, t_i, t_f), \quad \text{subject to: } \begin{cases} \mathbf{c}(\mathbf{x}_i, \varphi(\mathbf{x}_i, t_i; t_f)) = 0, & (\text{equality constraints}) \\ \mathbf{g}(t_i, t_f) \leq 0, & (\text{inequality constraints}) \end{cases} \quad (11)$$

To solve this constrained optimization problem MATLAB's function `fmincon` was employed. This method is the best choice to handle non-linear cost functions and constraints effectively, due to its high precision and robustness. The algorithm selected is `active-set`, as it is well-suited to manage equality constraints, which are crucial in trajectory optimization problems with strict boundaries. Both the `ConstraintTolerance` and the `OptimalityTolerance` have been set to 10^{-10} to ensure highly accurate solutions.

The problem described has been solved following two different approaches: one without giving any derivative to the solver, and one by providing the derivatives.

In the first case, the solver approximates the derivatives of both the cost function and the constraints at each iteration through finite differences. In this way the convergence is slower and slightly less accurate, but still feasible. In Fig. 5 the trajectory obtained from the propagation of the initial state computed in the first case is presented both in Earth-Moon rotating frame and ECI frame.

In the second case, the gradients of both the cost function and the constraints with respect to the optimization variables are provided to the solver. To compute the gradients, the State Transition Matrix has been estimated through the variational equations presented Eq. (4). The full procedure is detailed in Appendix A.

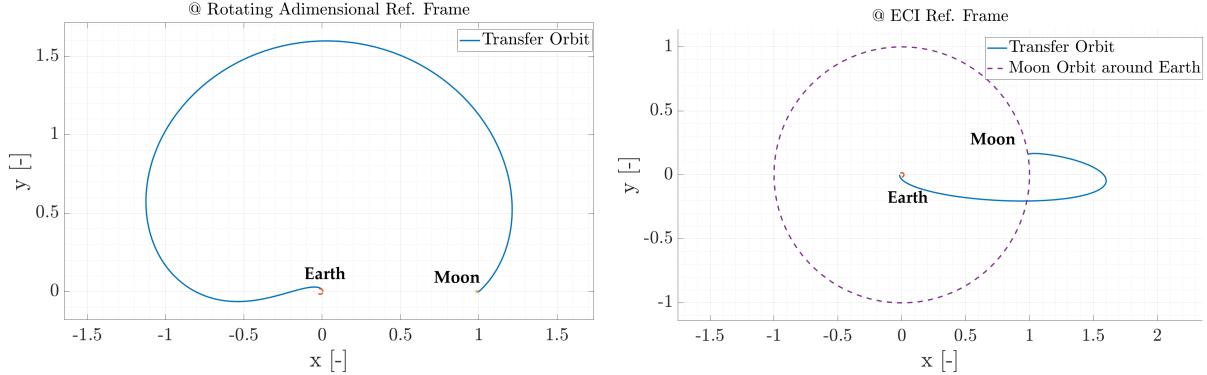


Figure 5: Trajectory computed through simple shooting without the gradients in @EMB EMRF (left) and in @Earth ECI frame (right)

The inclusion of the gradients significantly improved the performances, fastening the convergence, reducing the iterations required and the function evaluation done by the solver at each iteration. Regarding the optimal solution found, the inclusion of the gradients improved the overall velocity change required for the transfer from $\Delta V_{tot} = 4.007274 [VU]$ to $\Delta V_{tot} = 4.006830 [VU]$. The optimal variables obtained in both cases are:

Gradients	$r_{x,0} [DU]$	$r_{y,0} [DU]$	$v_{x,0} [VU]$	$v_{y,0} [VU]$	$t_i [TU]$	$t_f [TU]$
False	0.001586	0.010060	-6.320268	8.630334	2.063	6.072
True	0.001620	0.010013	-6.290741	8.651415	2.198	6.203

Table 6: Simple shooting solutions in @EMB Earth-Moon rotating frame.

To evaluate the performances of the two optimization processes adopted, a comparison of the norm of the residual errors on position and velocity between the two methods is presented in Table 7. These residuals represent the deviations from the equality constraints c_{eq} at the end of the optimization.

Gradients	$\ err_{pos}\ [m]$	$\ err_{vel}\ [m/s]$
False	4.11310e-05	4.66138e-08
True	3.25063e-06	4.93139e-09

Table 7: Comparison of the norm of the residual errors computed at the end of the optimization obtained from the two methods implemented.

These results underline the benefits obtained by the introduction of the gradients not only in terms of computational accuracy, but either in terms of enhanced solution's accuracy. However, since the tolerances adopted are very small, the observed improvement is minimal.

In Fig. 6 the trajectory obtained from the propagation of the initial guess found with gradient-based simple shooting is shown both in the Earth-Moon rotating frame and in the ECI frame. To highlight the precision reached, zoomed views of the departure and arrival positions in the Earth-Moon rotating frame are provided. As can be noticed, the trajectory closely satisfies the constraints.

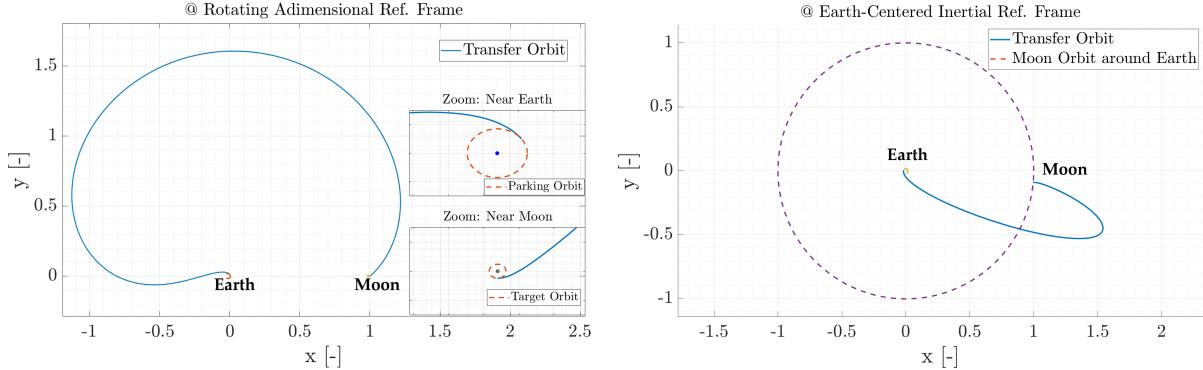


Figure 6: Trajectory computed through simple shooting with the gradients in @EMB EMRF (left) and in @(Earth) ECI frame (right). In the left plot is reported also a zoomed visualization of departure and arrival point.

2.3 Multiple Shooting

The aim of this section is to present the implementation of the multiple shooting method to solve the optimization problem described in Section 2.2.

The multiple shooting method is an extension of the simple shooting. Instead of treating the trajectory as a single continuous path, the trajectory is divided in $N - 1$ sections (where N is the number of nodes), each one propagated independently. At the end of each section, continuity constraints must be enforced in order to ensure the trajectory remains continuous. This segmentation reduces the influence of initial guess errors and improves numerical stability by limiting the propagation of errors across the trajectory, making it particularly effective for highly non-linear dynamics like the PCRFBP.

Focusing on the implementation of the method, the initial guess considered is the first guess solution, and the trajectory is divided in $N = 4$ nodes. The augmented state dimension results to be $4N + 2$, with the set of optimization variables which is now defined as: $\mathbf{y} = \{\mathbf{x}_j, t_1, t_N\}$, $\forall j = 1, \dots, N$, where \mathbf{x}_j is the state of the j -th node $(x_j, y_j, v_{x,j}, v_{y,j})$, while t_1 and t_N are the initial and the final time. The overall time interval is divided in $N - 1$ equally spaced intervals $[t_j, t_{j+1}]$, where the dynamics of PBRFBP is numerically integrated independently within each interval. For the first node, the initial guess used is the first guess solution given. For the next nodes, the initial guess is the final state of the previous interval.

For the definition of the cost function and constraints, the author referred to (Topputo et al., 2019)[†]. The objective cost function is given by:

$$J(\mathbf{y}) = \Delta V_1 + \Delta V_N \quad (12)$$

where ΔV_1 and ΔV_N are the initial and final maneuvers, respectively. The set of equality constraints has been augmented, by adding the continuity constraints ζ_j for the internal nodes to the boundary conditions ψ_1, ψ_N , defined as in Section 2.2. The continuity equality constraints are defined as:

$$\zeta_j = \varphi(\mathbf{x}_j, t_j; t_{j+1}) - \mathbf{x}_{j+1} = 0 \quad \forall j = 1, \dots, N - 1 \quad (13)$$

They need to be satisfied in order to ensure the continuity of the trajectory.

Moreover, for each node to avoid impacts with Earth and Moon respectively, the following two inequality constraints have been defined:

$$\eta_j = \begin{cases} R_E^2 - (x_j + \mu)^2 - y_j^2 \\ R_M^2 - (x_j + \mu - 1)^2 - y_j^2 \end{cases} < 0, \quad \forall j = 1, \dots, N \quad (14)$$

[†]Oshima, K., Topputo, F. & Yanao, T. “Low-energy transfers to the Moon with long transfer time”. Celestial Mechanics and Dynamical Astronomy Vol. 131, 2019, DOI: [10.1007/s10569-019-9883-7](https://doi.org/10.1007/s10569-019-9883-7)

along with the constraints on unfeasibility of negative time of flight ($\tau = t_1 - t_N < 0$). Once defined the cost function and the boundary conditions of the problem, the NLP formulation of the multiple shooting with $4N + 2$ variables and $4N$ constraints is:

$$\min_{\mathbf{y}} \mathbf{J}(\mathbf{y}), \quad \text{subject to: } \begin{cases} \mathbf{c}(\mathbf{y}) = 0, & (\text{equality constraints}) \\ \mathbf{g}(\mathbf{y}) \leq 0, & (\text{inequality constraints}) \end{cases} \quad (15)$$

where $\mathbf{c}(\mathbf{y}) = \{\zeta_j, \psi_1, \psi_N\} \forall j = 1, \dots, N-1$ is the matrix of equality constraints and $\mathbf{g}(\mathbf{y}) = \{\eta_j, \tau\} \forall j = 1, \dots, N$ is the matrix of inequality constraints.

Again, the solver adopted is MATLAB's function `fmincon` due to its high precision and robustness in handling non-linear optimization problems. For the same reasons described in Section 2.2, the active-set algorithm has been selected. Regarding the tolerances, `ConstraintTolerance` has been increased to 10^{-7} , to reduce the computational time, which would otherwise have been excessively high. Despite this adjustment, as discussed in the results section, overall accuracy remains largely unaffected.

To accelerate convergence, the analytical gradients of the cost function and the constraints were provided to `fmincon`. The detailed derivation of these gradients is presented in Appendix B, (Topputto et al., 2019)[‡].

The optimal solution found is $\Delta V_{tot} = 3.997327$ [VU], which is significantly lower with respect to the outcome of simple shooting. The optimal solution in terms of initial state and initial and final times is reported in Table 8.

$r_{x,0}$ [DU]	$r_{y,0}$ [DU]	$v_{x,0}$ [VU]	$v_{y,0}$ [VU]	t_i [TU]	t_f [TU]
0.004074	0.005164	-3.243707	10.190662	2.459	6.124

Table 8: Multiple shooting solution in the Earth-Moon rotating frame.

To assess the validity of the computed solution, the residuals of the equality constraints were evaluated at the final iteration. These errors, which quantify how well the solution satisfies the imposed constraints, are reported in Table 9.

$\ err_{pos}\ $ [m]	$\ err_{vel}\ $ [m/s]
3.06120e-08	-4.76842e-12

Table 9: Norm of the residual errors obtained from the multiple shooting method at the end of the optimization.

The multiple shooting method demonstrates clear advantages over simple shooting, as evidenced by the significantly lower residuals in the equality constraints at the end of the optimization process. Although the computational time is higher due to the increased dimensionality of the problem, this method proves to be more accurate.

In Fig. 7 the trajectory obtained from the propagation of the initial guess computed with the multiple shooting method is shown both in the EMRF and in the ECI frame. To highlight the precision reached, zoomed views of the departure and arrival positions in the EMRF are provided. As can be noticed, the trajectory closely satisfies the constraints. Moreover, along the trajectory in EMRF the nodes used to implement the method have been reported.

[‡]K. Oshima, F. Topputto, T. Yanao, “Low-energy transfers to the Moon with long transfer time”, Celestial Mechanics and Dynamical Astronomy, Vol. 131, Article 4, 2019, DOI: [10.1007/s10569-019-9883-7](https://doi.org/10.1007/s10569-019-9883-7)

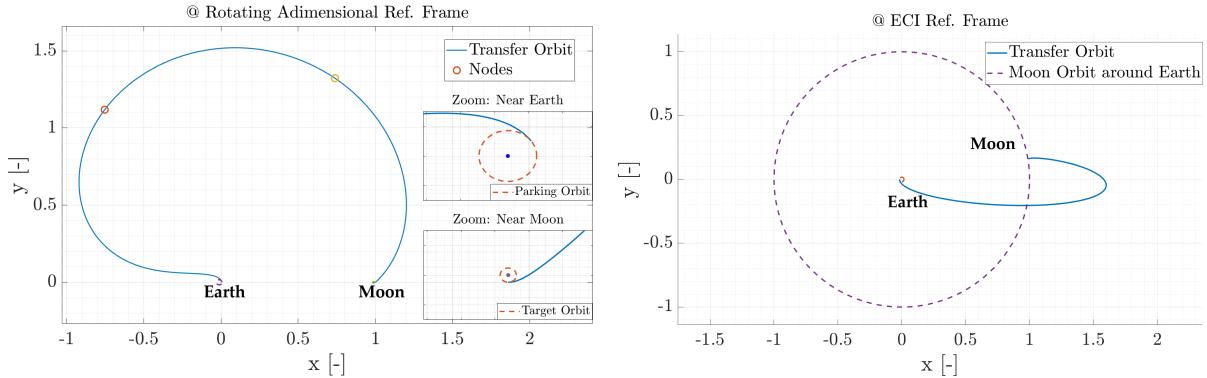


Figure 7: Trajectory computed through simple shooting without the gradients in @EMB Earth Moon Rotating frame (left) and in @Earth ECI frame (right)

2.4 N-Body Propagation

In this section, the optimal trajectory computed in Section 2.2 is validated using an N-body model. This model considers the gravitational influence of Earth, Moon, Sun and other celestial bodies, and provides a more realistic representation of the spacecraft’s dynamics compared to the simplified one of PCRFBP, where the orbits of celestial bodies are assumed planar and circular. Since the N-body propagator relies on the Ephemeris Time (ET) to determine the relative positions of the celestial bodies, first the initial epoch needs to be identified. This means finding the ephemeris time at which the relative positions of the Sun and the Moon are the same as those assumed in the optimization problem.

Considering the definition of the angle $\theta(t)$ as the angle between the rotating frame x axis, aligned to the position vector from the Earth-Moon System Barycenter (EMB) to the Moon, and the Sun, the target value θ_{target} at the epoch t_i can be computed as: $\theta_{target} = \omega_s t_i$, where ω_s is the scaled angular velocity of the Sun.

Then a zero-finding problem can be formulated. The time interval in which the solution can be found is a Moon revolution period around Earth, starting from the provided epoch t_0 (2024 Sep 28 00:00:00.000 TDB). Using MATLAB’s solver `fzero`, which is capable of solving one dimensional zero-finding problems, the epoch at which $\theta(t) = \theta_{target}$ is computed. By adding the time of flight obtained from Table 6 (previously dimensioned), the final epoch is also computed. Once defined the initial epoch, next step is to implement the dynamics of N-body model.

Symbol	Calendar epoch (UTC)
t_i	2024-10-12T11:40:24.354
t_f	2024-10-29T21:05:10.520
<hr/>	
$r_{x,0}$ [km]	$r_{y,0}$ [km]
-6223.62703080	2026.14898124
$r_{z,0}$ [km]	0.0
<hr/>	
$v_{x,0}$ [km/s]	$v_{y,0}$ [km/s]
-3.39800591	-10.437495777
$v_{z,0}$ [km/s]	0.0

Table 10: Initial epoch, final epoch, and initial state in Earth-centered inertial frame.

Defining \mathbf{a}_0 as the gravitational acceleration due to the central body (Earth in this case), the equations of motion for N-body model are:

$$\frac{d}{dt} \begin{bmatrix} \mathbf{r} \\ \mathbf{v} \end{bmatrix} = \begin{bmatrix} \mathbf{v} \\ \mathbf{a} \end{bmatrix}, \quad \mathbf{a} = \mathbf{a}_0 + \sum_{i \neq 0} \mathbf{a}_i \quad (16)$$

where \mathbf{a}_i is the gravitational acceleration due to the $i - th$ celestial body, defined as:

$$\mathbf{a}_i = -\frac{\mu_i}{|\mathbf{r} - \boldsymbol{\rho}_i|^3} (\mathbf{r} + \boldsymbol{\rho}_i f(q)) \quad (17)$$

where μ_i is the gravitational parameter of the $i - th$ body and ρ_i is the distance of the $i - th$ body from the central body. $f(q)$ is the perturbation function defined as:

$$f(q) = \frac{q(3 + 3q + q^2)}{1 + (1 + q)^{3/2}} \quad \text{with} \quad q = \frac{\mathbf{r} \cdot (\mathbf{r} - 2\boldsymbol{\rho}_i)}{|\boldsymbol{\rho}_i|^2} \quad (18)$$

Once propagated the initial state by numerically integrating over the time interval defined the equations of motion detailed above, the trajectory obtained has been compared with the one retrieved in Section 2.2 in Fig. 8

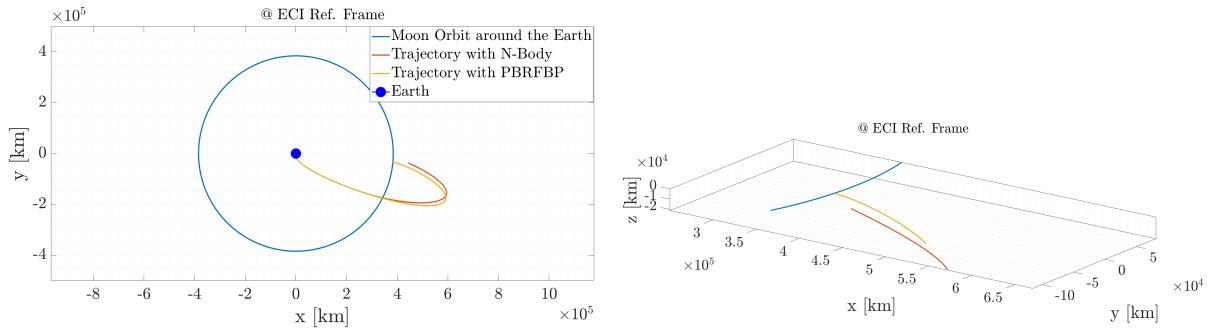


Figure 8: Comparison between the trajectory obtained with N-Body Propagator and the trajectory obtained with PBRFBP Propagator. On the right a zoomed view at the arrival point is reported. @Earth) ECI frame

The comparison shows that the two trajectories are similar but not identical. In particular, the N-Body trajectory does not intersect the Moon's orbit around Earth, while the PBRFBP trajectory perfectly matches the Moon's orbit. This difference arises because the initial state used for the propagation was optimized for the PCRFBP trajectory. Hence, this outcome was predictable. Another difference is that the PBRFBP trajectory has no components in the out-of-plane direction, while the N-Body trajectory presents position and velocity components in the out-of-plane direction. This is due to the 3D nature of N-Body dynamics, whereas the PBRFBP is restricted to the Moon's orbital plane.

3 Continuous guidance

Exercise 3

A low-thrust option is being considered to perform an orbit raising maneuver using a low-thrust propulsion system in Earth orbit. The spacecraft is released on a circular orbit on the equatorial plane at an altitude of 800 km and has to reach an orbit inclined by 0.75 deg on the equatorial plane at 1000 km. This orbital regime is characterized by a large population of resident space objects and debris, whose spatial density q can be expressed as:

$$q(\rho) = \frac{k_1}{k_2 + \left(\frac{\rho - \rho_0}{DU} \right)^2}$$

where ρ is the distance from the Earth center. The objective is to design an optimal orbit raising that minimizes the risk of impact, that is to minimize the following objective function

$$F(t) = \int_{t_0}^{t_f} q(\rho(t)) dt.$$

The parameters and reference Distance Unit to be considered are provided in Table 11.

Symbol	Value	Units	Meaning
h_i	800	km	Altitude of departure orbit
h_f	1000	km	Altitude of arrival orbit
Δi	0.75	deg	Inclination change
R_e	6378.1366	km	Earth radius
μ	398600.435	km ³ /s ²	Earth gravitational parameter
ρ_0	$750 + R_e$	km	Reference radius for debris flux
k_1	1×10^{-5}	DU ⁻¹	Debris spatial density constant 1
k_2	1×10^{-4}	DU ²	Debris spatial density constant 2
m_0	1000	kg	Initial mass
T_{\max}	3.000	N	Maximum thrust
I_{sp}	3120	s	Specific impulse
DU	7178.1366	km	Distance Unit
MU	m_0	kg	Mass Unit

Table 11: Problem parameters and constants. The units of time TU and velocity VU can be computed imposing that the scaled gravitational parameter $\bar{\mu} = 1$.

- 1) Plot the debris spatial density $q(\rho) \in [h_i - 100; h_f + 100]$ km and compute the initial state and target orbital state, knowing that: i) the initial and final state are located on the x -axis of the equatorial J2000 reference frame; ii) the rotation of the angle Δi is performed around the x -axis of the equatorial J2000 reference frame (RAAN = 0).
- 2) Adimensionalize the problem using as reference length $DU = \rho_i = h_i + R_e$ and reference mass $MU = m_0$, imposing that $\mu = 1$. Report all the adimensionalized parameters.
- 3) Using the PMP, write down the spacecraft equations of motion, the costate dynamics, and the zero-finding problem for the unknowns $\{\lambda_0, t_f\}$ with the appropriate transversality condition. **Hint:** the spacecraft has to reach the target state computed in point 1).
- 4) Solve the problem considering the data provided in Table 11. To obtain an initial guess for the costate, generate random numbers such that $\lambda_{0,i} \in [-250; +250]$ while $t_f \approx 20\pi$.

Report the obtained solution in terms of $\{\lambda_0, t_f\}$ and the error with respect to the target. Assess your results exploiting the properties of the Hamiltonian in problems that are not time-dependent and time-optimal solution. Plot the evolution of the components of the primer vector α in a NTW reference frame^s.

- 5) Solve the problem for a lower thrust level $T_{\max} = 2.860$ N. Compare the new solution with the one obtained in the previous point. **Hint:** exploit numerical continuation.
(11 points)

3.1 Debris Spatial Density

The evolution of the debris spatial density $q(\rho)$, where ρ is the distance from the center of the Earth, as function of the orbital altitude is shown in Fig. 9.

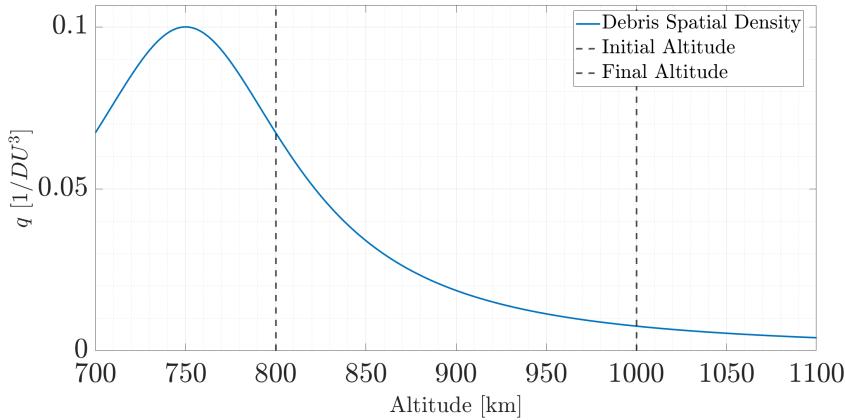


Figure 9: Debris spatial density evolution over orbital altitude

As can be notice, in the altitude range of interest for the mission considered there is a decrement of the debris population.

Since the initial and final state are located on the x -axis of the equatorial J2000 frame, the position vectors in this frame have only the x -component as non-zero. For the initial state, the spacecraft is assumed to lie on the equatorial plane and to follow a circular orbit. Hence, its velocity vector has only a tangential component, with $v_y \neq 0$ and $v_x = v_z = 0$.

For the target state, the inclination change Δi is achieved through a rotation about the x -axis. As a result, the v_y component of the velocity represents the tangential direction in the new orbital plane, while the v_z component comes from the inclination variation. The position remains aligned with the x -axis, as the change of inclination does not affect the radial position. Based on these considerations, the initial and final state have been computed through the following formulations:

$$\mathbf{x}_i = \begin{pmatrix} R_e + h_i \\ 0 \\ 0 \\ 0 \\ \sqrt{\frac{\mu}{R_e + h_i}} \\ 0 \end{pmatrix}, \quad \mathbf{x}_f = \begin{pmatrix} R_e + h_f \\ 0 \\ 0 \\ 0 \\ \sqrt{\frac{\mu}{R_e + h_f}} \cos \Delta i \\ \sqrt{\frac{\mu}{R_e + h_f}} \sin \Delta i \end{pmatrix} \quad (19)$$

In Table 12 the initial and final state are reported.

^sThe T-axis is aligned with the velocity, the N-axis is aligned with the angular momentum, while the W-axis is pointing inwards, i.e., towards the Earth.

$r_{x,i}$ [km]	$r_{y,i}$ [km]	$r_{z,i}$ [km]	$v_{x,i}$ [km/s]	$v_{y,i}$ [km/s]	$v_{z,i}$ [km/s]
7178.136600	0.000000	0.000000	0.00000000	7.45183148	0.00000000
$r_{x,f}$ [km]	$r_{y,f}$ [km]	$r_{z,f}$ [km]	$v_{x,f}$ [km/s]	$v_{y,f}$ [km/s]	$v_{z,f}$ [km/s]
7378.136600	0.000000	0.000000	0.00000000	7.34950906	0.09621034

Table 12: Initial and target state in Earth-centered equatorial J2000 inertial frame.

3.2 Adimensionalized Parameters

Given the reference mass MU and the reference length DU , the other reference units needed are:

$$TU = \sqrt{\frac{DU^3}{\mu}} = 963.2715 \text{ [s]}, \quad VU = \frac{DU}{TU} = 7.4518 \text{ [km/s]} \quad (20)$$

Knowing these four reference units, all the other parameters can be adimensionalized. Their values are reported in Table 13. In Table 14 also the initial and final state adimensionalized are reported.

$\mu \left[\frac{DU^3}{TU^2} \right]$	ρ_0 [DU]	m_0 [MU]	T_{max} $\left[\frac{MU \ DU}{TU^2 \ 10^3} \right]$	I_{sp} [TU]	$g_0 \left[\frac{MU \ DU}{TU^2 \ 10^3} \right]$
1	0.99303	1	3.87799e - 04	3.23896	1.26810

Table 13: Adimensionalized parameters.

$r_{x,i}$ [DU]	$r_{y,i}$ [DU]	$r_{z,i}$ [DU]	$v_{x,i}$ $\left[\frac{DU}{TU} \right]$	$v_{y,i}$ $\left[\frac{DU}{TU} \right]$	$v_{z,i}$ $\left[\frac{DU}{TU} \right]$
1	0	0	0	1	0
$r_{x,f}$ [DU]	$r_{y,f}$ [DU]	$r_{z,f}$ [DU]	$v_{x,f}$ $\left[\frac{DU}{TU} \right]$	$v_{y,f}$ $\left[\frac{DU}{TU} \right]$	$v_{z,f}$ $\left[\frac{DU}{TU} \right]$
1.027862	0	0	0	0.986269	0.012911

Table 14: Initial and target state adimensionalized

3.3 Statement of the Problem

The aim of this section is to present the statement of the problem. The goal of the exercise is to compute through the Pontryagin Maximum Principle the optimal trajectory that minimizes the risk of collision with debris. To describe the theoretical framework, the author referred to the lecture notes provided during the course.

By defining the state $\mathbf{x} = (\mathbf{r}, \mathbf{v})$, where \mathbf{r} is the position vector, \mathbf{v} is the velocity vector, the spacecraft's dynamics is expressed by the following set of equations:

$$\mathbf{f}(\mathbf{x}, \mathbf{u}, t) = \begin{cases} \dot{\mathbf{r}} &= \mathbf{v}, \\ \dot{\mathbf{v}} &= -\frac{\mu}{r^3} \mathbf{r} + u \frac{T}{m} \hat{\boldsymbol{\alpha}}, \\ \dot{m} &= -u \frac{T}{I_{sp} g_0}, \end{cases} \quad (21)$$

where T is the thrust magnitude, $\hat{\boldsymbol{\alpha}}$ is the thrust pointing direction and u is the throttle factor. $u \in [0, 1]$, and when $u = 0$, the engines are off, while for $u = 1$ engines are in full-throttle condition. \mathbf{u} is the control policy adopted and is composed by $(\hat{\boldsymbol{\alpha}}, u)$.

Since the goal is to minimize the risk of collision, the lagrangian function $l(\mathbf{x}, \mathbf{u}, t)$ of the problem is exactly the debris spatial density function $q(\rho)$ given. In addition, the problem is formulated as a time-optimal problem, hence the throttle factor is $u = 1$, since minimizing the risk of collision involves attempting to spend the least possible time within the danger zone.

Once defined the $[7 \times 1]$ vector of costate multipliers of the dynamics $\boldsymbol{\lambda} = \{\boldsymbol{\lambda}_r; \boldsymbol{\lambda}_v; \lambda_m\}$, where the first 3 components refer to the position, the next three to the velocity, and the latter to the mass, the rigorous formulation for the application of PMP to a low-thrust guidance problem can be expressed as:

$$\min_{\mathbf{u} \in \mathbf{U}} \int_{t_0}^{t_f} l(\mathbf{x}, \mathbf{u}, t) dt \quad \text{subject to} \quad \begin{cases} \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, t), \\ \mathbf{x}(t_0) = \mathbf{x}_0, \\ \mathbf{r}(t_f) = \mathbf{r}_f, \\ \mathbf{v}(t_f) = \mathbf{v}_f, \\ \lambda_m(t_f) = 0. \end{cases} \quad (22)$$

where \mathbf{U} represents the set of admissible control actions defined as: $\mathbf{U} := \{(u, \hat{\alpha}) \text{ s.t. } u \in [0, 1]; \|\hat{\alpha}\| = 1\}$.

Next step is the definition of the Hamiltonian of the problem:

$$H = l + \boldsymbol{\lambda}^T \mathbf{f} = 1 + \begin{pmatrix} \boldsymbol{\lambda}_r \\ \boldsymbol{\lambda}_v \\ \lambda_m \end{pmatrix} \begin{pmatrix} \mathbf{v} \\ -\frac{\mu}{r^3} \mathbf{r} + u \frac{T_{max}}{m} \hat{\alpha} \\ -u \frac{T_{max}}{I_{sp} g_0} \end{pmatrix} \quad (23)$$

Applying the PMP means finding the optimal control policy $\mathbf{u}^* = (\hat{\alpha}^*, u^*)$, among those admissible, which minimizes the Hamiltonian H of the problem. As stated above, since it is a time-optimal problem, the control is always on, hence u^* is 1. For the primer vector $\hat{\alpha}^*$, the definition is:

$$\hat{\alpha}^* = -\frac{\boldsymbol{\lambda}_v}{\lambda_v} \quad (24)$$

Through this definition, the problem ensures that the thrust pointing direction is always opposite to the gradient of the velocity costate. Now that all the components required have been defined, the Euler-Lagrange, embedding $\hat{\alpha}^*$ and $u^* = 1$, are defined as:

$$\begin{cases} \dot{\mathbf{x}} = \frac{\partial H}{\partial \boldsymbol{\lambda}} \\ \dot{\boldsymbol{\lambda}} = -\frac{\partial H}{\partial \mathbf{x}} \end{cases} \xrightarrow{\text{from which}} \begin{cases} \dot{\mathbf{r}} = \mathbf{v} \\ \dot{\mathbf{v}} = -\frac{\mu}{r^3} \mathbf{r} - \frac{T_{max}}{m} \frac{\boldsymbol{\lambda}_v}{\|\boldsymbol{\lambda}_v\|} \\ \dot{m} = -\frac{T_{max}}{I_{sp} g_0} \\ \dot{\boldsymbol{\lambda}}_r = -\frac{\partial q(\rho)}{\partial \mathbf{r}} - \frac{3\mu}{r^5} (\mathbf{r} \cdot \boldsymbol{\lambda}_v) \mathbf{r} + \frac{\mu}{r^3} \boldsymbol{\lambda}_v \\ \dot{\boldsymbol{\lambda}}_v = -\boldsymbol{\lambda}_r \\ \dot{\lambda}_m = -\frac{\|\boldsymbol{\lambda}_v\| T_{max}}{m^2} \end{cases} \quad (25)$$

To find t_f the transversality condition reported in Eq. (26) need to be imposed. Since the target is fixed, $\dot{\psi} = 0$.

$$H(t_f) - \boldsymbol{\lambda}(t_f) \cdot \dot{\psi}(t_f) = 0 \xrightarrow{\dot{\psi}=0} H(t_f) = 0 \quad (26)$$

The solution of the Two Boundary Value Problem (TBVP) detailed above consists in finding the $\boldsymbol{\lambda}_0$ such that the integration of the system $\boldsymbol{\varphi}(\begin{bmatrix} \mathbf{x}_0 \\ \boldsymbol{\lambda}_0 \end{bmatrix}, t_0, t_f)$ produces $\mathbf{x}(t)$ and $\boldsymbol{\lambda}(t)$ that satisfy the boundary condition at final time.

3.4 Continuous Guidance Solution

In this section the process implemented to solve the TBVP described previously and obtain the optimal trajectory that minimizes the risk of collision is presented. The initial state of the problem \mathbf{xx}_0 includes not only the initial Cartesian state \mathbf{x}_0 adimensionalized in Section 3.2, but also the initial mass m_0 of the spacecraft adimensionalized.

The set of optimization variables is $\mathbf{y} = \{\boldsymbol{\lambda}, t_f\}$. The initial guess $\boldsymbol{\lambda}_0$ for the costates is obtained randomly between $[-250, 250]$ for $\boldsymbol{\lambda}_{0,r}$ and $\boldsymbol{\lambda}_{0,v}$, and $[0, 250]$ for $\lambda_{0,m}$, as mass can't be negative. The initial guess selected for the final time was $t_{f,0} = 20\pi$.

The zero-finding problem implemented is detailed below.

Algorithm 1 Iterative Optimization for PMP Problem

- 1: **Input:** Augmented initial state \mathbf{xx}_0 , target Cartesian state \mathbf{x}_f .
 - 2: **repeat**
 - 3: **Initialize:** Random guess for costates $\boldsymbol{\lambda}_0$ and initial guess for the final time t_f .
 - 4: **Propagation:** Propagate state and costate equations from t_0 to t_f by integrating the Euler-Lagrange Equations Eq. (25) and STM.
 - 5: **Hamiltonian:** Extract final state vector $[\mathbf{x}_f, \boldsymbol{\lambda}_f]$ and compute the Hamiltonian $H(t_f)$ at the final time through Eq. (23).
 - 6: **Errors:** Evaluate terminal errors: $\mathbf{e}_r = \mathbf{r}(t_f) - \mathbf{r}_f, \mathbf{e}_v = \mathbf{v}(t_f) - \mathbf{v}_f$
 - 7: **Formulation:** Assemble the optimization function \mathbf{F} : $\mathbf{F} = [\mathbf{e}_r; \mathbf{e}_v; \lambda_m(t_f); H(t_f)]$
 - 8: **Optimization:** Update $\boldsymbol{\lambda}_0$ and t_f using a nonlinear solver (`fssolve`).
 - 9: **until** All errors \mathbf{F} are below a predefined tolerance.
 - 10: **Output:** Optimal costates $\boldsymbol{\lambda}_0$ at initial time t_0 and final time t_f .
-

The algorithm selected was Levenberg-Marquardt, as it leverages the strengths of both gradient descent and Gauss-Newton methods, making it effective for solving nonlinear problems when the initial guess is far from the solution. The FunctionTolerance is set to 10^{-10} .

To fasten the convergence and improve the accuracy, the analytical derivatives of both the boundary conditions and the Hamiltonian were provided to the solver. In this way, the number of iterations and the number of function evaluations for each iteration were reduced. Their analytical derivation is reported in Appendix C.

The optimal solution obtained is reported in Table 15. Once a feasible solution was obtained in terms of optimal costates $\boldsymbol{\lambda}_0$ and final time t_f , the initial state was propagated by numerically integrating the Euler-Lagrange equations over the time interval found. In Fig. 10 the orbit raising maneuver is presented in ECI reference frame.

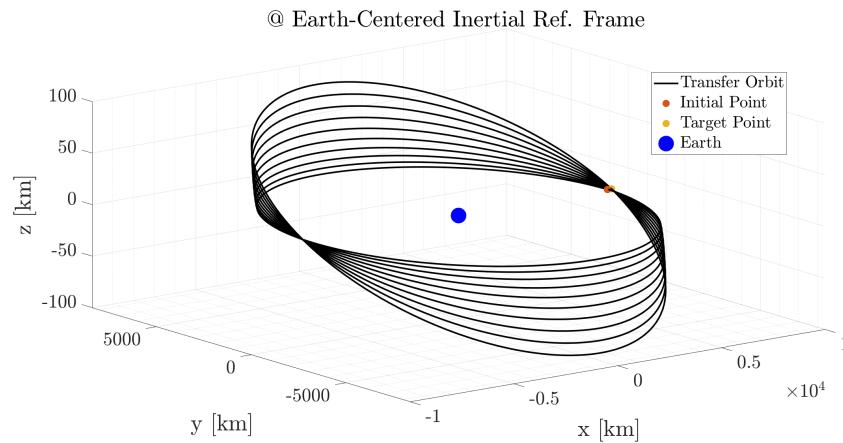


Figure 10: Orbit Raising Maneuver in dimensional @Earth) Earth-Centered Inertial frame

t_f [mins]	m_f [kg]
1035.1974	993.9120
λ_{0,r_x}	λ_{0,r_y}
-214.9812	-10.3659
λ_{0,r_z}	λ_{0,v_x}
0.8856	-10.3929
λ_{0,v_y}	λ_{0,v_z}
-214.6105	-112.9454
$\lambda_{0,m}$	2.5964

Table 15: Optimal orbit raising transfer solution ($T_{\max} = 3.000$ N).

Another sub-optimal solution was found by the solver, with a final time around 22π . Since the aim of the problem is to find the trajectory that minimizes the risk of collisions while also optimizing the time of flight, this solution was discarded. However, further analysis could be done. For instance, comparing the collision risk associated with the two trajectories could help determine which one offers the safest path.

To validate the solution obtained, the norm of the errors in the final state with respect to the target state are reported in Table 16. They represent the deviations at the end of the propagation of the final state from the target value. The values obtained assess the validity of the procedure implemented to solve the problem.

Error	Value	Units
$\ \mathbf{r}(t_f) - \mathbf{r}_f\ $	1.5002e-08	km
$\ \mathbf{v}(t_f) - \mathbf{v}_f\ $	1.5029e-08	m/s

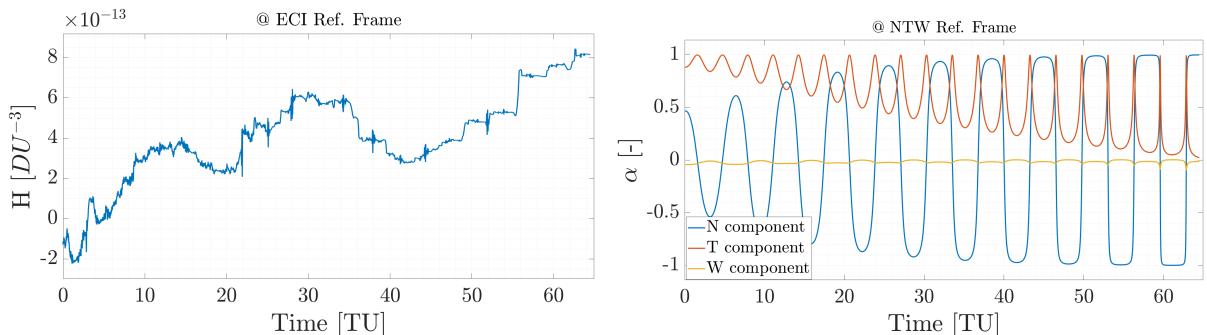
Table 16: Final state error with respect to target position and velocity ($T_{\max} = 3.000$ N).

Furthermore, the properties of the Hamiltonian were assessed by calculating its value at each time step and visualizing its evolution over the entire time interval. As stated in the previous section, since the target is fixed, from the transversality condition (Eq. (26)) it follows that the Hamiltonian should always be zero. In Fig. 11 it can be observed that the Hamiltonian remains in the order of 10^{-13} throughout the trajectory. This indicates that the numerical integration is accurate and that the solution guarantees the expected conservation properties of the Hamiltonian, as required for time-independent and optimal control problems. This validation confirms the correctness of the computed trajectory and the associated costates.

Additionally, the evolution over the time interval of the components of the primer vector $\hat{\alpha}$ in the NTW reference frame is reported in Fig. 12. In the ECI frame $\hat{\alpha}$ is defined using Eq. (24). Then the vector is rotated into the NTW frame:

$$\hat{\alpha}_{NTW} = \mathbf{R} \hat{\alpha}_{ECI}, \quad \mathbf{R} = [\hat{\mathbf{N}}, \hat{\mathbf{T}}, \hat{\mathbf{W}}] \quad (27)$$

where the unit vectors are defined as: $\hat{\mathbf{N}} = \frac{\mathbf{r}_{ECI} \times \mathbf{v}_{ECI}}{\|\mathbf{r}_{ECI} \times \mathbf{v}_{ECI}\|}$; $\hat{\mathbf{T}} = \frac{\mathbf{v}_{ECI}}{\|\mathbf{v}_{ECI}\|}$; $\hat{\mathbf{W}} = \hat{\mathbf{N}} \times \hat{\mathbf{T}}$.

**Figure 11:** Evolution of the Hamiltonian H over the time interval in @ (Earth) ECI frame.**Figure 12:** Evolution of the primer vector $\hat{\alpha}$ components in NTW frame over the time interval.

The oscillatory behavior of the primer vector components is typical in low-thrust transfers, where the thrust is applied in small amounts to gradually adjust the orbit. Among the components, the tangential component $\hat{\alpha}_t$ is observed to be consistently positive. This indicates that thrust is primarily applied along the tangential direction to modify the semi-major axis and achieve the target orbit. Moreover, as the mean value decreases over the time interval, the majority of the maneuver is likely performed in the early stage of the mission due to the higher debris density. The peaks towards the end represent small adjustments required to achieve the target altitude.

The normal component $\hat{\alpha}_n$, on the contrary, oscillates between -1 and 1. This trend reflects that thrust in the normal direction, necessary for changing the orbital inclination, is required only at the ascending and descending nodes of the orbit. This occurs because of the specific maneuver required, as the RAAN must remain the same while providing only a small inclination change. Furthermore, the amplitude of the oscillation increases over time, with the peaks becoming flatter towards the end of the maneuver. This suggests that the majority of the thrust in the normal direction is applied in the last phase of the mission.

Finally, the w component $\hat{\alpha}_w$ remains near zero throughout the maneuver. This is consistent with the lack of thrust along the w -axis, as no significant maneuvers are required along this component for this transfer.

3.5 Lower Thrust Level Solution

In this section a solution with a lower thrust level $T_{max} = 2.860 \text{ N}$ is analyzed. To perform this analysis a numerical continuation methods has been exploited. Starting from $T_{max} = 3.000 \text{ N}$, the thrust applied is gradually reduced in constant steps until the target T_{max} is reached. For each value of T_{max} , the same algorithm detailed in the previous section has been applied, with the solution obtained for one thrust level becoming the initial guess for the next iteration. This iterative approach ensures convergence while maintaining the numerical stability of the optimization process. The only difference is the algorithm used for `fsolve`. In this case, the trust-region-dogleg algorithm was employed, as it is more effective than levenberg-marquardt algorithm, when the initial guess is close to the solution.

The solution obtained for the lower thrust level is reported in Table 17.

	t_f [mins]	m_f [kg]
	1030.9760	994.2198
λ_{0,r_x}		
-593.1529	-11.5904	2.2973
λ_{0,r_y}		
	-11.9293	-592.7890
λ_{0,r_z}		
	-920.3698	17.3814
λ_{0,v_x}		
λ_{0,v_y}		
λ_{0,v_z}		
$\lambda_{0,m}$		

Table 17: Optimal orbit raising transfer solution ($T_{max} = 2.860 \text{ N}$).

As done in the previous section, to validate the solution the norm of the final state errors and the evolution of the Hamiltonian over the time interval have been computed.

Error	Value	Units
$\ \mathbf{r}(t_f) - \mathbf{r}_f\ $	2.7677e-08	km
$\ \mathbf{v}(t_f) - \mathbf{v}_f\ $	2.7456e-08	m/s

Table 18: Final state error with respect to target position and velocity ($T_{max} = 2.860 \text{ N}$).

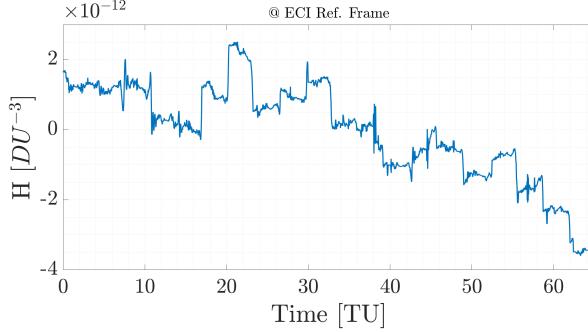


Figure 13: Evolution of the Hamiltonian H in @Earth ECI Ref. Frame over the time interval.

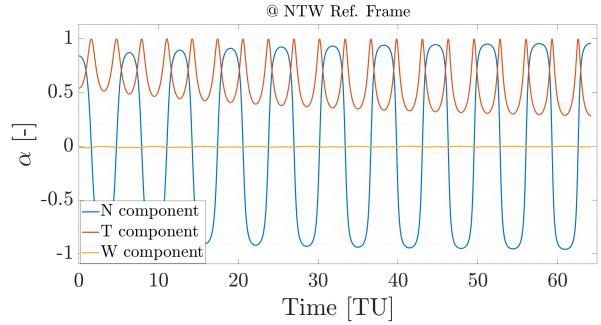


Figure 14: Evolution of the primer vector $\hat{\alpha}$ components in NTW frame over the time interval.

As can be observed in Fig. 13, the Hamiltonian remains in the order of 10^{-12} , which validates the solution obtained. According to the theory, the Hamiltonian should be zero throughout the entire time interval, as required by the transversality condition, and the low values observed confirm the numerical accuracy of the computed trajectory.

The reduction in thrust level to $T_{\max} = 2.86$ N results in a decrease in the time of flight compared to the higher thrust case ($T_{\max} = 3.00$ N), while the increment of the final mass was expected due to the lower propellant consumption, as both thrust and time of flight are lower. This behavior can be explained by analyzing the evolution of the primer vector components ($\alpha_t, \alpha_n, \alpha_w$) in the NTW frame.

For the reduced thrust case, the tangential component (α_t) exhibits more or less the same trend as in the previous case in the first phase of the maneuver. The difference arises in the final part, where the tangential component remains consistently higher compared to the higher thrust case, with less sharp peaks. These two considerations indicate that the thrust is applied for a longer time and more uniformly in the tangential direction to modify the semi-major axis. As a result, the energy change required for the transfer is achieved earlier and more effectively.

Similarly, the normal component (α_n) shows a higher average value in the first phase, with a trend more similar to a sinusoidal curve. Again, this indicates that thrust in the normal direction is applied more smoothly at the ascending and descending nodes, ensuring a consistent inclination change with minimal deviations. This efficient distribution of thrust likely reduces the time required to achieve the desired orbital plane change, contributing to the overall decrease in the time of flight.

The w component (α_w) remains negligible in both cases, confirming that the control effort is predominantly within the tangential and normal planes.

In conclusion, the reduced thrust level appears to promote a more efficient control strategy, with thrust directed more consistently in the optimal directions. This results in a shorter time of flight, despite the lower available thrust, highlighting the adaptability of the numerical continuation method and the robustness of the optimal control solution.

A Analytical Gradients for Simple Shooting

The explicit expression of the gradient of the objective function, $\nabla J(\mathbf{x}) = \left[\frac{\partial J}{\partial \mathbf{x}_0}, \frac{\partial J}{\partial t_i}, \frac{\partial J}{\partial t_f} \right]^T$, and of the constraints, $\nabla \mathbf{c}(\mathbf{x})$, are provided here.

The spatial derivatives of the objective function $J(\mathbf{x})$ are expressed as:

$$\begin{aligned} \frac{\partial J}{\partial \mathbf{x}_i} &= \frac{1}{\sqrt{(v_{x,i} - y_i)^2 + (v_{y,i} + x_i + \mu_{EM})^2}} \begin{bmatrix} v_{y,i} + x_i + \mu_{EM} \\ y_i - v_{x,i} \\ v_{x,i} - y_i \\ v_{y,i} + x_i + \mu_{EM} \end{bmatrix} \\ &+ \Phi_f^T \frac{1}{\sqrt{(v_{x,f} - y_f)^2 + (v_{y,f} + x_f + \mu_{EM} - 1)^2}} \begin{bmatrix} v_{y,f} + x_f + \mu_{EM} - 1 \\ y_f - v_{x,f} \\ v_{x,f} - y_f \\ v_{y,f} + x_f + \mu_{EM} - 1 \end{bmatrix} \end{aligned}$$

The time derivatives are calculated as

$$\frac{dJ}{dt_i} = -\frac{\partial \Delta V_2}{\partial \mathbf{x}_f}^T \Phi(t_0, t_f) \cdot \mathbf{f}(\mathbf{x}_0, t_i) \quad ; \quad \frac{dJ}{dt_f} = \frac{\partial \Delta V_2}{\partial \mathbf{x}_f}^T \cdot \mathbf{f}(\mathbf{x}_f, t_f)$$

where

$$\frac{\partial \Delta V_2}{\partial \mathbf{x}_f} = \frac{1}{\sqrt{(v_{x,f} - y_f)^2 + (v_{y,f} + x_f + \mu_{EM} - 1)^2}} \begin{bmatrix} v_{y,f} + x_f + \mu_{EM} - 1 \\ y_f - v_{x,f} \\ v_{x,f} - y_f \\ v_{y,f} + x_f + \mu_{EM} - 1 \end{bmatrix}$$

As for the constraints:

$$\begin{aligned} \frac{\partial \psi_i}{\partial x_i} &= \begin{bmatrix} 2(x_i + \mu_{EM}) & 2y_i & 0 & 0 \\ v_{x,i} & v_{y,i} & (x_i + \mu_{EM}) & y_i \end{bmatrix} \\ \frac{\partial \psi_f}{\partial x_f} &= \begin{bmatrix} 2(x_f + \mu_{EM} - 1) & 2y_f & 0 & 0 \\ v_{x,f} & v_{y,f} & (x_f + \mu_{EM} - 1) & y_f \end{bmatrix} \end{aligned}$$

Given the right-hand side of the equations of motion \mathbf{f} , $\nabla \mathbf{c}(\mathbf{x})$ is computed as:

$$\nabla \mathbf{c}(\mathbf{x}) = \begin{bmatrix} \frac{\partial \psi_i}{\partial x_i} & 0 & 0 \\ \frac{\partial \psi_f}{\partial x_f} \Phi_f & \frac{\partial \psi_f}{\partial x_f} (-\Phi(t_0, t_f) \mathbf{f}(t_i)) & \frac{\partial \psi_f}{\partial x_f} \cdot \mathbf{f}(t_f) \end{bmatrix}^T$$

The Jacobian matrix \mathbf{A} of the equations of motion consists of partial derivatives of the equations of motion with respect to the state variables. The matrix \mathbf{A} is defined as follows:

$$\mathbf{A} = \begin{bmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} & \frac{\partial f_1}{\partial \dot{x}} & \frac{\partial f_1}{\partial \dot{y}} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} & \frac{\partial f_2}{\partial \dot{x}} & \frac{\partial f_2}{\partial \dot{y}} \\ \frac{\partial f_3}{\partial x} & \frac{\partial f_3}{\partial y} & \frac{\partial f_3}{\partial \dot{x}} & \frac{\partial f_3}{\partial \dot{y}} \\ \frac{\partial f_4}{\partial x} & \frac{\partial f_4}{\partial y} & \frac{\partial f_4}{\partial \dot{x}} & \frac{\partial f_4}{\partial \dot{y}} \end{bmatrix}$$

B Analytical Gradients for Multiple Shooting

This appendix provides the analytical expressions for the gradients of the objective function and the constraints used in the Multiple Shooting optimization method. The author referred to (Topputto et. al., 2019) ^[1] for their implementation.

Gradient of the Objective Function

The derivative of the objective function J with respect to the optimization variables \mathbf{y} is expressed as:

$$\frac{\partial J}{\partial \mathbf{y}} = [P_1 \ O \ P_N \ O],$$

where:

$$P_1 = \frac{1}{\sqrt{(v_{x1} - y_1)^2 + (v_{y1} + x_1 + \mu)^2}} [v_{y1} + x_1 + \mu \ y_1 - v_{x1} \ v_{x1} - y_1 \ v_{y1} + x_1 + \mu],$$

$$P_N = \frac{1}{\sqrt{(v_{xN} - y_N)^2 + (v_{yN} + x_N + \mu - 1)^2}} [v_{yN} + x_N + \mu - 1 \ y_N - v_{xN} \ v_{xN} - y_N \ v_{yN} + x_N + \mu - 1].$$

Jacobian of the Equality Constraints

The derivative of the equality constraints $\mathbf{c} = \{\zeta_j, \psi_1, \psi_N\}$ with respect to \mathbf{y} is expressed as:

$$\frac{\partial \mathbf{c}}{\partial \mathbf{y}} = \begin{bmatrix} \Phi(t_1, t_2) & -I_4 & O & O & Q_1^1 & Q_1^N \\ O & \Phi(t_2, t_3) & -I_4 & O & Q_2^1 & Q_2^N \\ O & O & \Phi(t_3, t_4) & -I_4 & Q_3^1 & Q_3^N \\ R_1 & O & O & O & O & O \\ O & O & O & R_N & O & O \end{bmatrix},$$

where:

$$Q_j^1 = -\frac{N-j}{N-1} \Phi(t_j, t_{j+1}) \mathbf{f}(\mathbf{x}_j, t_j) + \frac{N-j-1}{N-1} \mathbf{f}(\varphi(\mathbf{x}_j, t_j, t_{j+1}), t_{j+1}),$$

$$Q_j^N = -\frac{j-1}{N-1} \Phi(t_j, t_{j+1}) \mathbf{f}(\mathbf{x}_j, t_j) + \frac{j}{N-1} \mathbf{f}(\varphi(\mathbf{x}_j, t_j, t_{j+1}), t_{j+1}),$$

$$R_1 = \frac{\partial \psi_1}{\partial \mathbf{x}_1} = \begin{bmatrix} 2(x_1 + \mu) & 2y_1 & 0 & 0 \\ v_{x1} & v_{y1} & x_1 + \mu & y_1 \end{bmatrix},$$

$$R_N = \frac{\partial \psi_N}{\partial \mathbf{x}_N} = \begin{bmatrix} 2(x_N + \mu - 1) & 2y_N & 0 & 0 \\ v_{xN} & v_{yN} & x_N + \mu - 1 & y_N \end{bmatrix}.$$

Inequality Constraints

The derivative of the inequality constraints $\mathbf{g}(\mathbf{y}) := \{\eta_j, \tau\} < 0$ with respect to the NLP variables \mathbf{y} can be expressed as:

$$\frac{\partial \mathbf{g}}{\partial \mathbf{y}} = \begin{bmatrix} S_1 & O & O & O & O \\ O & S_2 & O & O & O \\ O & O & S_3 & O & O \\ O & O & O & \ddots & O \\ O & O & O & O & S_l \end{bmatrix},$$

where:

$$S_j := \frac{\partial \eta_j}{\partial \mathbf{x}_j} = \begin{bmatrix} -2(x_j + \mu) & -2y_j & 0 & 0 \\ -2(x_j + \mu - 1) & -2y_j & 0 & 0 \end{bmatrix}, \quad j = 1, \dots, N,$$

$$S_t := \begin{bmatrix} \frac{\partial \tau}{\partial t_1} & \dots & \frac{\partial \tau}{\partial t_N} \end{bmatrix} = [1 \ -1].$$

^[1]K. Oshima, F. Topputto, T. Yano, “Low-energy transfers to the Moon with long transfer time”, Celestial Mechanics and Dynamical Astronomy, Vol. 131, Article 4, 2019, DOI: [10.1007/s10569-019-9883-7](https://doi.org/10.1007/s10569-019-9883-7)

C Jacobian Computation for the Shooting Method

The analytic derivatives of the zero-finding problem provided to the solver are detailed in this section. Let $\mathbf{y} = [\boldsymbol{\lambda}_0; t_f]$ be the set of optimization variables, where $\boldsymbol{\lambda}_0$ are the initial costates and t_f is the final time, and let \mathbf{F} represent the optimization function defined by the boundary conditions. The Jacobian of \mathbf{F} with respect to \mathbf{y} is defined as:

$$\mathbf{F} = \begin{bmatrix} \mathbf{e}_r \\ \mathbf{e}_v \\ \lambda_{m,f} \\ H_f \end{bmatrix}_{8 \times 1}, \quad \frac{\partial \mathbf{F}}{\partial \mathbf{y}} = \begin{bmatrix} \frac{\partial \mathbf{r}_f}{\partial \boldsymbol{\lambda}_0} & \frac{\partial \mathbf{r}_f}{\partial t_f} \\ \frac{\partial \mathbf{v}_f}{\partial \boldsymbol{\lambda}_0} & \frac{\partial \mathbf{v}_f}{\partial t_f} \\ \frac{\partial \lambda_{m,f}}{\partial \boldsymbol{\lambda}_0} & \frac{\partial \lambda_{m,f}}{\partial t_f} \\ \frac{\partial H_f}{\partial \boldsymbol{\lambda}_0} & 0 \end{bmatrix}_{8 \times 8} \quad (28)$$

where \mathbf{r}_f , \mathbf{v}_f , $\lambda_{m,f}$, and H_f represent, respectively, the position, velocity, mass costate, and Hamiltonian at the final time t_f , and \mathbf{e}_r and \mathbf{e}_v are the terminal errors defined as:

$$\mathbf{e}_r = \mathbf{r}(t_f) - \mathbf{r}_f, \quad \mathbf{e}_v = \mathbf{v}(t_f) - \mathbf{v}_f \quad (29)$$

Reformulation of the Jacobian Terms

By applying the chain rule of derivatives and leveraging the structure of the Euler-Lagrange equations, the derivative of the Hamiltonian H_f with respect to the initial costates $\boldsymbol{\lambda}_0$ can be expressed as:

$$\frac{\partial H_f}{\partial \boldsymbol{\lambda}_0} = \frac{\partial H_f}{\partial \mathbf{x}_f} \frac{\partial \mathbf{x}_f}{\partial \boldsymbol{\lambda}_0} + \frac{\partial H_f}{\partial \boldsymbol{\lambda}_f} \frac{\partial \boldsymbol{\lambda}_f}{\partial \boldsymbol{\lambda}_0}. \quad (30)$$

Substituting the time derivatives from the Euler-Lagrange equations, the expression becomes:

$$\frac{\partial H_f}{\partial \boldsymbol{\lambda}_0} = -\dot{\boldsymbol{\lambda}}_f \frac{\partial \mathbf{x}_f}{\partial \boldsymbol{\lambda}_0} + \dot{\mathbf{x}}_f \frac{\partial \boldsymbol{\lambda}_f}{\partial \boldsymbol{\lambda}_0}. \quad (31)$$

Here, \mathbf{f} represents the right-hand side of the state and costate equations, while $\Phi(t_0, t_f)$ is the state transition matrix (STM) computed through the variational approach, which maps the variations of the states and costates from the initial to the final time.

Final Jacobian Formulation

The Jacobian matrix can be compactly expressed using the STM as follows:

$$\frac{\partial \mathbf{F}}{\partial \mathbf{y}} = \begin{bmatrix} \Phi_{\mathbf{r}\boldsymbol{\lambda}_r 3 \times 3} & \Phi_{\mathbf{r}\boldsymbol{\lambda}_v 3 \times 3} & \Phi_{\mathbf{r}\lambda_m 3 \times 1} & \dot{\mathbf{r}}_{3 \times 1} \\ \Phi_{\mathbf{v}\boldsymbol{\lambda}_r 3 \times 3} & \Phi_{\mathbf{v}\boldsymbol{\lambda}_v 3 \times 3} & \Phi_{\mathbf{v}\lambda_m 3 \times 1} & \dot{\mathbf{v}}_{3 \times 1} \\ \Phi_{\lambda_m \boldsymbol{\lambda}_r 1 \times 3} & \Phi_{\lambda_m \boldsymbol{\lambda}_v 1 \times 3} & \Phi_{\lambda_m \lambda_m 1 \times 3} & \dot{\lambda}_m \\ -\dot{\boldsymbol{\lambda}}_r^T \Phi_{\mathbf{r}\boldsymbol{\lambda}_r} + \dot{\mathbf{r}}^T \Phi_{\boldsymbol{\lambda}_r \boldsymbol{\lambda}_r} & -\dot{\boldsymbol{\lambda}}_v^T \Phi_{\mathbf{v}\boldsymbol{\lambda}_v} + \dot{\mathbf{v}}^T \Phi_{\boldsymbol{\lambda}_v \boldsymbol{\lambda}_v} & -\dot{\lambda}_m \Phi_{\mathbf{m}\lambda_m} + \dot{m} \Phi_{\lambda_m \lambda_m} & 0 \end{bmatrix}$$

where the subscripts in $\Phi_{a,b}$ indicate the rows and columns of the STM used to compute the submatrices.

State Transition Matrix and Submatrices

The STM, $\Phi(t_0, t_f)$, is defined as:

$$\Phi(t_0, t_f) = \begin{bmatrix} \frac{\partial \mathbf{x}_f}{\partial \mathbf{x}_0} & \frac{\partial \mathbf{x}_f}{\partial \boldsymbol{\lambda}_0} \\ \frac{\partial \boldsymbol{\lambda}_f}{\partial \mathbf{x}_0} & \frac{\partial \boldsymbol{\lambda}_f}{\partial \boldsymbol{\lambda}_0} \end{bmatrix}_{14 \times 14}.$$

Each block of the STM contributes to the submatrices of the Jacobian: - $\Phi_{\mathbf{r}_f, \boldsymbol{\lambda}_0}$: Position sensitivity to initial costates. - $\Phi_{\mathbf{v}_f, \boldsymbol{\lambda}_0}$: Velocity sensitivity to initial costates. - $\Phi_{\boldsymbol{\lambda}_m, \boldsymbol{\lambda}_0}$: Mass costate sensitivity to initial costates. - $\mathbf{f}_{\mathbf{r}_f}, \mathbf{f}_{\mathbf{v}_f}, \mathbf{f}_{\boldsymbol{\lambda}_m}$: Contributions from the dynamics at the final state.

This formulation ensures that the Jacobian is computed analytically, improving the performance and accuracy of the numerical shooting method.

SGN – Assignment #2

Edoardo Mensi Weingrill, 244706

Exercise 1: Uncertainty propagation

You are asked to analyze the state uncertainty evolution along a transfer trajectory in the Planar Bicircular Restricted Four-Body Problem, obtained as optimal solution of the problem stated in Section 3.1 (Topputto, 2013)^{*}. The mean initial state \mathbf{x}_i at initial time t_i with its associated covariance \mathbf{P}_0 and final time t_f for this optimal transfer are provided in Table I.

1. Propagate the initial mean and covariance within a time grid of 5 equally spaced elements going from t_i to t_f , using both a Linearized Approach (LinCov) and the Unscented Transform (UT). We suggest to use $\alpha = 1$ and $\beta = 2$ for tuning the UT in this case. Plot the mean and the ellipses associated with the position elements of the covariances obtained with the two methods at the final time.
2. Perform the same uncertainty propagation process on the same time grid using a Monte Carlo (MC) simulation I. Compute the sample mean and sample covariance and compare them with the estimates obtained at Point 1). Provide the following outputs.
 - Plot of the propagated samples of the MC simulation, together with the mean and the covariance obtained with all methods in terms of ellipses associated with the position elements at the final time.
 - Plot of the time evolution (for the time grid previously defined) for all three approaches (MC, LinCov, and UT) of $3\sqrt{\max(\lambda_i(P_r))}$ and $3\sqrt{\max(\lambda_i(P_v))}$, where P_r and P_v are the 2x2 position and velocity covariance submatrices.
 - Plot resulting from the use of the MATLAB function `qqplot`, for each component of the previously generated MC samples at the final time.

Compare the results, in terms of accuracy and precision, and discuss on the validity of the linear and Gaussian assumption for uncertainty propagation.

Table 1: Solution for an Earth-Moon transfer in the rotating frame.

Parameter	Value
Initial state \mathbf{x}_i	$\mathbf{r}_i = [-0.011965533749906, -0.017025663128129]$ $\mathbf{v}_i = [10.718855256727338, 0.116502348513671]$
Initial time t_i	1.282800225339865
Final time t_f	9.595124551366348
Covariance \mathbf{P}_0	$\begin{bmatrix} +1.041e-15 & +6.026e-17 & +5.647e-16 & +4.577e-15 \\ +6.026e-17 & +4.287e-18 & +4.312e-17 & +1.855e-16 \\ +5.647e-16 & +4.312e-17 & +4.432e-16 & +1.455e-15 \\ +4.577e-15 & +1.855e-16 & +1.455e-15 & +2.822e-14 \end{bmatrix}$

^{*}F. Topputto, “On optimal two-impulse Earth–Moon transfers in a four-body model”, Celestial Mechanics and Dynamical Astronomy, Vol. 117, pp. 279–313, 2013, DOI: [10.1007/s10569-013-9513-8](https://doi.org/10.1007/s10569-013-9513-8)

[†]Use at least 1000 samples drawn from the initial covariance

1. In the first point, initial mean and covariance have been propagated along a reference trajectory using two different approaches: Linearized Covariance (LinCov) and Unscented Transform (UT).

LinCov: Through numerical integration of the Planar Bicircular Restricted Four-Body Problem (PBR4BP) dynamics, the Linear Covariance (LinCov) approach has been used to propagate the initial state \mathbf{x}_0 and the corresponding covariance matrix \mathbf{P}_0 within a time grid from t_0 to t_f . The variational approach has been used to calculate the State Transition Matrix (STM or $\Phi(t_0, t_i)$) at each time-step. Next, the covariance matrix and state vector are updated through:

$$\hat{\mathbf{x}}_{LinCov}(t_i) = \varphi(\mathbf{x}_0, t_0, t_i) \quad \mathbf{P}(t_i) = \Phi(t_0, t_i) \mathbf{P}_0 \Phi(t_0, t_i)^T$$

Therefore, this method propagates uncertainties using a first-order linearization of the dynamics, providing a good approximation for low non-linear systems.

UT: Using the Unscented Transform (UT) technique, the initial state vector \mathbf{x}_0 and its associated covariance matrix \mathbf{P}_0 are propagated forward in time. A set of sigma points (χ_i) is created around the initial state assuming a Gaussian distribution. These sigma points are then propagated by numerically integrating the Planar Bicircular Restricted Four-Body Problem (PBR4BP) dynamics over the time grid. Afterwards, weighted averages of the propagated sigma points ($\gamma = f(\chi)$) are used to recalculate the mean and covariance:

$$\hat{\mathbf{x}}_{UT} = \sum_{j=0}^{2n} W_j \gamma_j \quad \mathbf{P} = \sum_{j=0}^{2n} W_j (\gamma_j - \hat{\mathbf{x}}_{UT}) (\gamma_j - \hat{\mathbf{x}}_{UT})^T$$

Thus, the UT method uses an unscented transformation to propagate the covariance of the state through the non-linear dynamics.

A comparison of the outputs from the two methods, in terms of Mean Position and Covariance, is presented in Fig. 1.

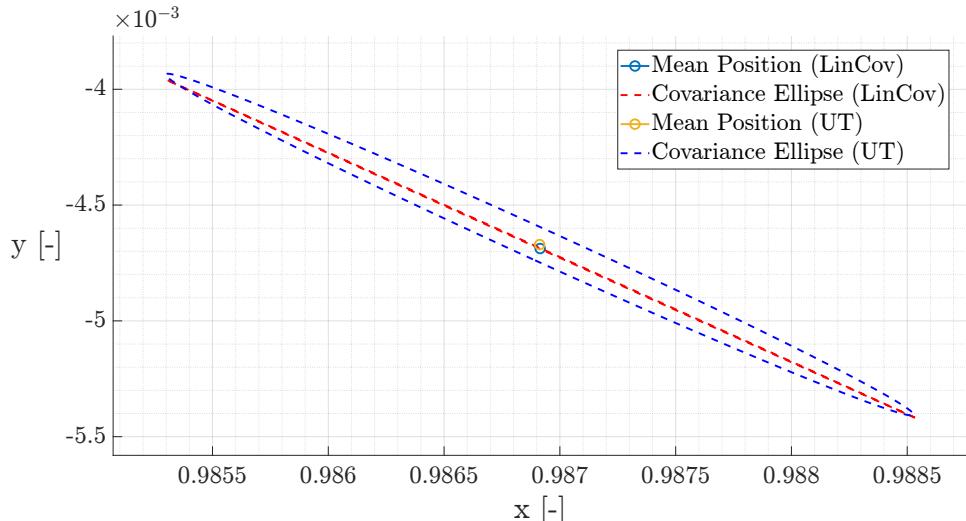


Figure 1: Comparison between LinCov and UT methods @ (EMB) Earth-Moon Rotating Frame

Both methods provided comparable mean trajectories, with UT showing slight deviations in regions of higher nonlinearity, as it accounts for such effects in the dynamics. The similarity in results comes from the system's low non-linearity, allowing the LinCov method to

approximate uncertainties effectively, as it is capable of accounting them in the primary direction of propagation. However, UT remains the more accurate method for uncertainty propagation.

The flattened covariance ellipse in the LinCov method may result from the linear approximation of the dynamics, which does not fully capture the system's non-linearities, leading to a distorted, more elongated or near-linear covariance shape.

2. In the second point of the exercise, the same uncertainty propagation has been performed using a Monte Carlo (MC) simulation.

MC: A Monte Carlo simulation for uncertainty propagation was performed assuming a multivariate Gaussian random distribution of samples, generated using the `mvnrnd` function in MATLAB. These samples, accounting for correlations between the input variables, were propagated through the same model as Ex. 1.1. Due to its high computational cost, a sample size n of 1000 was used, which is the minimum required. Once propagated the samples, the final mean and covariance are estimated using samples mean and covariance:

$$\hat{\mathbf{x}}_{MC} = \frac{1}{n} \sum_{j=1}^n \mathbf{x}_j(t_j) \quad \mathbf{P} = \frac{1}{n-1} \sum_{j=1}^n (\mathbf{x}_j - \hat{\mathbf{x}}_{MC})(\mathbf{x}_j - \hat{\mathbf{x}}_{MC})^T$$

In Fig. 2 the mean position and covariance ellipses obtained using Monte Carlo (MC), Linearized Covariance (LinCov), and Unscented Transform (UT) methods are compared.

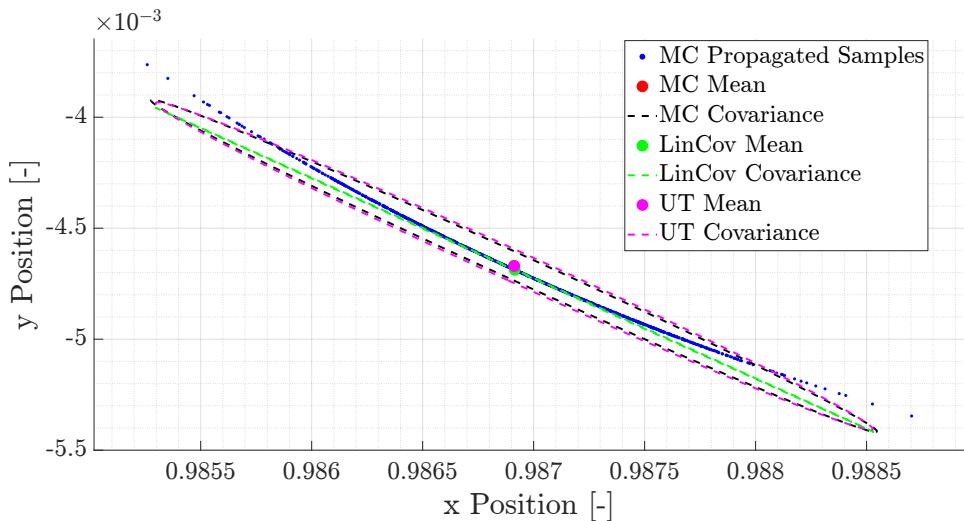


Figure 2: Comparison between the methods @ (EMB) Earth-Moon Rotating Frame

The LinCov method underestimates the uncertainty, as indicated by the smaller covariance ellipse compared to MC and UT. Indeed, the magenta dot and magenta dashed ellipse, which correspond to the UT method, closely matches the MC results. This indicates that the UT captures nonlinear effects more accurately than LinCov.

Overall, the comparison shows that LinCov deviates from the MC benchmark due to its reliance on linearization, while UT provides a more reliable representation of uncertainty, particularly in nonlinear dynamics.

The LinCov method performs similarly to the other methods (MC and UT), which are more accurate overall, in the primary direction of propagation, according to the time evolution of the maximum standard deviation σ_{\max} for both position and velocity covariance matrices, as displayed in Fig. 3.

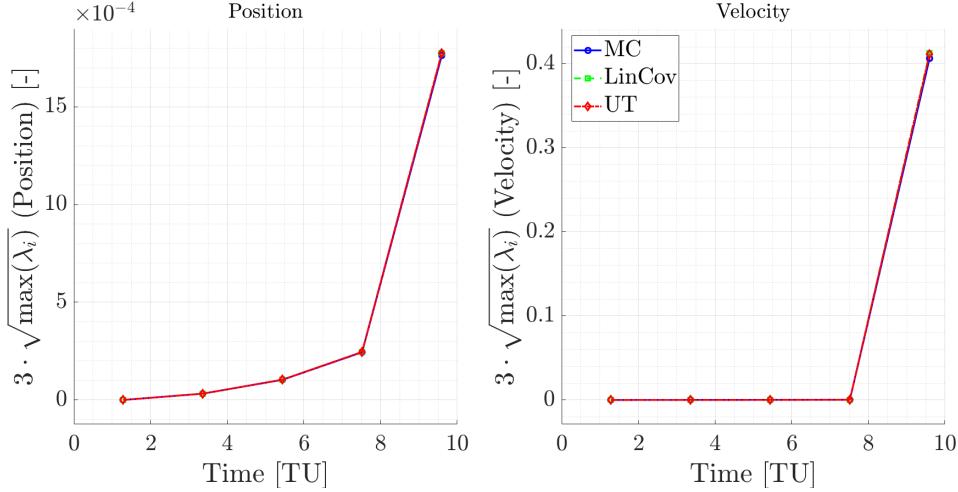


Figure 3: Maximum square root of the eigenvalues of the position and velocity covariance matrix over time, calculated using three different methods: LinCov, UT, and Monte Carlo.

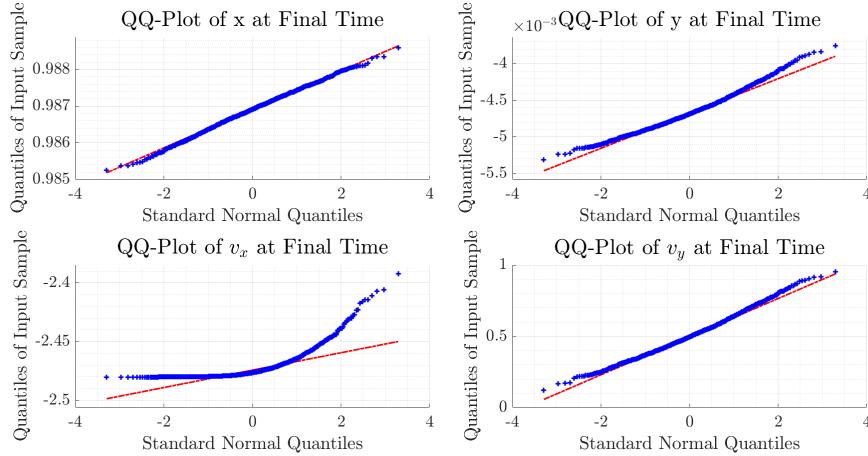


Figure 4: QQ-Plots for each state component (x, y, v_x, v_y) at the final time. These plots compare the theoretical quantiles with the sample quantiles to assess the normality of the propagated distributions.

From QQ-plots shown in Fig. 4 the validation of Gaussianity assumption can be performed. The results indicate that the position components generally follow a Gaussian distribution, as demonstrated by the alignment of points along the reference line in the QQ-plots, whereas noticeable deviations are observed in the velocity components. These deviations are particularly evident in the tails, indicating that the propagated distributions deviate from purely Gaussian condition. This validation suggests that, under the given initial conditions and moderate uncertainties, the Gaussianity assumption underlying both LinCov and UT methods remains mostly valid for position components, while begins to break down for velocity components, likely due to the influence of non-linear dynamics.

In conclusion, MC simulation provided the most accurate results but at a high computational cost. The UT method emerged as the optimal method for balancing accuracy and efficiency, while LinCov, even if computationally efficient, is less reliable in nonlinear regimes and should be applied with caution. Overall, the analysis demonstrated the limitations of linearization and highlighted the importance of nonlinear methods for robust uncertainty propagation.

Exercise 2: Batch filters

The Soil Moisture and Ocean Salinity (SMOS) mission, launched on 2 November 2009, is one of the European Space Agency's Earth Explorer missions, which form the science and research element of the Living Planet Programme.

You have been asked to track SMOS to improve the accuracy of its state estimate. To this aim, you shall schedule the observations from the three ground stations reported in Table 2

1. *Compute visibility windows.* The Two-Line Elements (TLE) set of SMOS are reported in Table 3 (and in WeBeep as 36036.3le). Compute the osculating state from the TLE at the reference epoch t_{ref} , then propagate this state assuming Keplerian motion to predict the trajectory of the satellite and compute all the visibility time windows from the available stations in the time interval from $t_0 = 2024-11-18T20:30:00.000$ (UTC) to $t_f = 2024-11-18T22:15:00.000$ (UTC). Consider the different time grid for each station depending on the frequency of measurement acquisition. Report the resulting visibility windows and plot the predicted Azimuth and Elevation profiles within these time intervals.
2. *Simulate measurements.* Use SGP4 and the provided TLE to simulate the measurements acquired by the sensor network in Table 2 by:
 - (a) Computing the spacecraft position over the visibility windows identified in Point 1 and deriving the associated expected measurements.
 - (b) Simulating the measurements by adding a random error to the expected measurements (assume a Gaussian model to generate the random error, with noise provided in Table 2). Discard any measurements (i.e., after applying the noise) that does not fulfill the visibility condition for the considered station.
3. *Solve the navigation problem.* Using the measurements simulated at the previous point:
 - (a) Find the least squares (minimum variance) solution to the navigation problem without a priori information using
 - the epoch t_0 as reference epoch;
 - the reference state as the state derived from the TLE set in Table 3 at the reference epoch;
 - the simulated measurements obtained for the KOROU ground station only;
 - pure Keplerian motion to model the spacecraft dynamics.
 - (b) Repeat step 3a by using all simulated measurements from the three ground stations.
 - (c) Repeat step 3b by using a J2-perturbed motion to model the spacecraft dynamics.

Provide the results in terms of navigation solution[†], square root of the trace of the estimated covariance submatrix of the position elements, square root of the trace of the estimated covariance submatrix of the velocity elements. Finally, considering a linear mapping of the estimated covariance from Cartesian state to Keplerian elements, provide the standard deviation associated to the semimajor axis, and the standard deviation associated to the inclination. Elaborate on the results, comparing the different solutions.

4. *Trade-off analysis.* For specific mission requirements, you are constrained to get a navigation solution within the time interval reported in Point 1. Since the allocation of antenna time has a cost, you are asked to select the passes relying on a budget of 70.000 €. The cost per pass of each ground station is reported in Table 2. Considering this constraint,

[†]Not just estimated state or covariance

and by using a J2-perturbed motion for your estimation operations, select the best combination of ground stations and passes to track SMOS in terms of resulting standard deviation on semimajor axis and inclination, and elaborate on the results.

5. *Long-term analysis.* Consider a nominal operations scenario (i.e., you are not constrained to provide a navigation solution within a limited amount of time). In this context, or for long-term planning in general, you could still acquire measurements from multiple locations but you are tasked to select a set of prime and backup ground stations. For planning purposes, it is important to have regular passes as this simplifies passes scheduling activities. Considering the need to have *reliable* orbit determination and *repeatable* passes, discuss your choices and compare them with the results of Point 4.

Table 2: Sensor network to track SMOS: list of stations, including their features.

Station name	KOUROU	TROLL	SVALBARD
Coordinates	LAT = 5.25144° LON = -52.80466° ALT = -14.67 m	LAT = -72.011977° LON = 2.536103° ALT = 1298 m	LAT = 78.229772° LON = 15.407786° ALT = 458 m
Type	Radar (monostatic)	Radar (monostatic)	Radar (monostatic)
Measurements type	Az, El [deg] Range (one-way) [km]	Az, El [deg] Range (one-way) [km]	Az, El [deg] Range (one-way) [km]
Measurements noise (diagonal noise matrix R)	$\sigma_{Az,El} = 125$ mdeg $\sigma_{range} = 0.01$ km	$\sigma_{Az,El} = 125$ mdeg $\sigma_{range} = 0.01$ km	$\sigma_{Az,El} = 125$ mdeg $\sigma_{range} = 0.01$ km
Minimum elevation	6 deg	0 deg	8 deg
Measurement frequency	60 s	30 s	60 s
Cost per pass	30.000 €	35.000 €	35.000 €

Table 3: TLE of SMOS.

```

1 36036U 09059A 24323.76060260+00000600+00000-0 20543-3 0 9995
2 36036 98.4396 148.4689 0001262 95.1025 265.0307 14.39727995790658

```

- The visibility window of the SMOS satellite with respect to the assigned ground stations was computed using the TLE set provided in Table 3. The satellite's position in ECI coordinates at the initial time t_0 was derived by numerically integrating the unperturbed 2BP dynamics starting from the reference time $t_{ref} = 2024 - 11 - 18T18 : 15 : 16.064640$. SMOS' position at t_{ref} is reported in Table 4.

Position in ECI [x, y, z] [km]	[-6065.4137948, 3768.0468907, 14.5009335]
Velocity in ECI [v_x, v_y, v_z] [km/s]	[0.6036245, 0.9267430, 7.3907681]

Table 4: SMOS' Cartesian State at t_{ref} @ (Earth) Earth-Centered Inertial reference frame

The satellite's trajectory was then propagated from t_0 to t_f , with the time interval divided according to the measurement frequencies of each ground station (Kourou: 60 s, Troll: 30 s, Svalbard: 60 s). The ground stations' positions were expressed in ECI coordinates using the `cspice_spkezr` function. These positions, along with the satellite's propagated trajectory, were rotated into the Topocentric frame of each ground station. Finally, the range, azimuth, and elevation were calculated using the `cspice_xmfsta` function. The visibility windows were determined by considering only the intervals when the SMOS satellite had an elevation exceeding the minimum detectable elevation of the respective ground station. In Table 5 the resulting visibility windows are reported.

Table 5: Visibility windows for SMOS tracking.

Station	Start Time (UTC)	End Time (UTC)	Total Duration (min)	Min Elevation (deg)
Kourou	2024-11-18T20:40:00.000	2024-11-18T20:49:00.000	9	6
Troll	2024-11-18T21:02:30.000	2024-11-18T21:11:30.000	9	0
Svalbard	2024-11-18T21:56:00.000	2024-11-18T22:06:00.000	10	8

Concurrently, for each ground station the Visibility Window plot is shown on the left, where the time evolution over the total time interval of azimuth and elevation is presented. The highlighted sections in the plot represent the periods during which the satellite is visible, defined by elevation exceeding the station's minimum threshold.

The plot on the right displays the satellite's trajectory in terms of azimuth and elevation. This plot represents the satellite's path as it moves throughout the station's field of view, starting at low elevation, reaching a peak, and descending back towards the horizon. Only the visible portions of the trajectory, where elevation exceeds the threshold, are displayed.

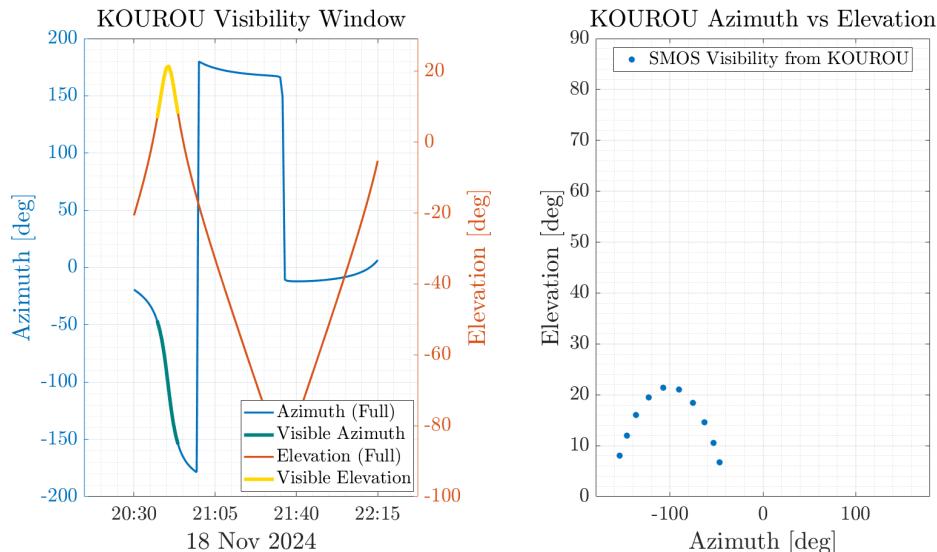
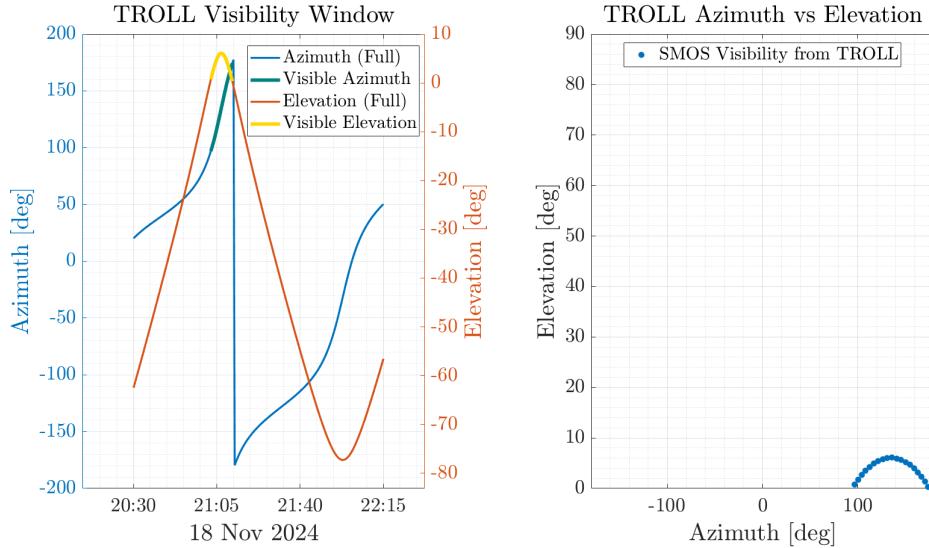
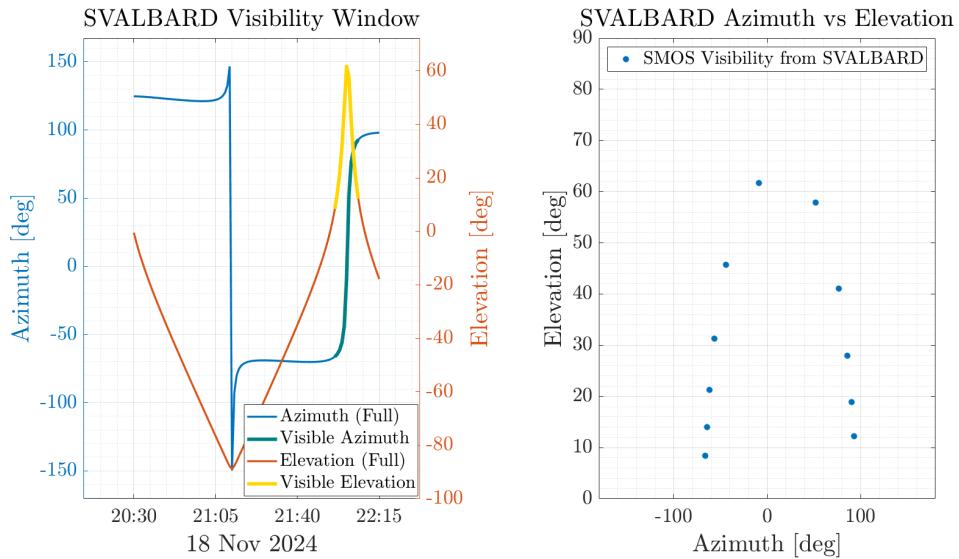


Figure 5: Combined plots for KOUROU station.

**Figure 6:** Combined plots for TROLL station.**Figure 7:** Combined plots for SVALBARD station.

2. The second part of the exercise follows the same rationale but employs the SGP4 propagator, which models major perturbations, including the J_2 effect, atmospheric drag, and third-body perturbations. The TLE set provides the initial conditions for this propagator, resulting in slight differences compared to the Keplerian propagator. These differences between the two methods arise because SGP4 reflects satellite's trajectory more accurately. Afterwards, simulated measurements (range, azimuth, elevation) were generated by perturbing the expected values with multivariate Gaussian random noise, using the parameters provided in Table 2. After filtering by the visibility conditions, Fig. 8 presents a comparison between the expected measurements from the Keplerian propagator and the noisy simulated measurements propagated through SGP4. The noise introduces additional deviations, making the measurements more representative of real-world conditions.

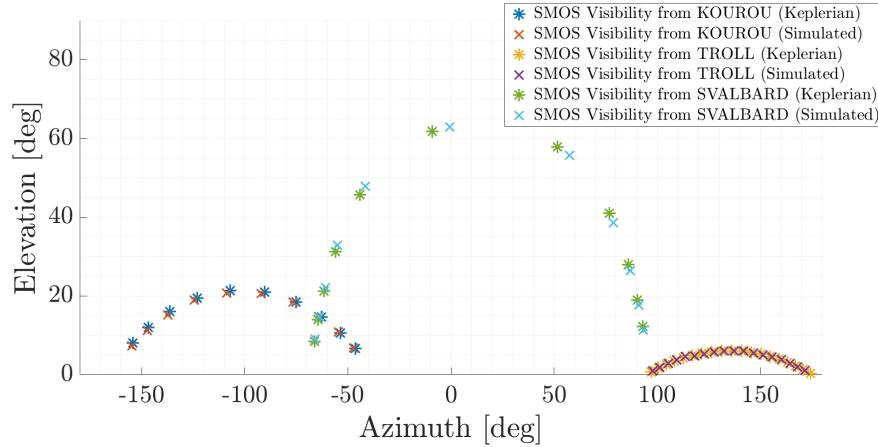


Figure 8: Comparison of visibility windows computed using the Keplerian propagator and noisy simulated measurements propagated with SGP4.

3. In point three the navigation problem is solved under three different configurations: (a) using measurements from Kourou only, (b) using measurements from all three ground stations, and (c) incorporating the J_2 -perturbed motion model. The outputs for each point include the navigation solution at time t_0 , comprehending the state in Cartesian coordinates and covariance matrix, the square root of the trace of the covariance submatrices for position and velocity, and the standard deviations for the semi-major axis and inclination, computed via a Jacobian transformation to Keplerian elements. The procedure adopted is detailed in Appendix 3.

In all cases, the navigation problem was solved using least-squares optimization via MATLAB's `lsqnonlin` function, employing the Levenberg–Marquardt algorithm, which is particularly effective when the initial guess is far from the true solution.

(a) Using Kourou Measurements Only

In this configuration, only the measurements from the Kourou station were used. The process is detailed below:

- Predicted measurements (range, azimuth, elevation) were computed following the same approach used for Ex. 2.1.
- The navigation problem was solved using a least-squares optimization:

$$\min_{\mathbf{x}_0} \sum_{i=1}^N \mathbf{W}_i \mathbf{r}_i$$

where \mathbf{r}_i represents the residuals between the predicted and observed measurements, and \mathbf{W} is the weighting matrix based on the noise characteristics of the measurements.

$$\mathbf{P}_0 = \begin{bmatrix} 6.6220e+01 & 2.0189e+01 & -8.7122e+00 & -6.7000e-02 & -3.3100e-02 & -3.7200e-02 \\ 2.0189e+01 & 3.1027e+01 & -1.9742e+01 & -1.2000e-02 & -2.5100e-02 & -1.6000e-03 \\ -8.7122e+00 & -1.9742e+01 & 1.3159e+01 & 3.1000e-03 & 1.4100e-02 & -2.1000e-03 \\ -6.7000e-02 & -1.2000e-02 & 3.1000e-03 & 7.0720e-05 & 2.8383e-05 & 4.0896e-05 \\ -3.3100e-02 & -2.5100e-02 & 1.4100e-02 & 2.8383e-05 & 2.6579e-05 & 1.3190e-05 \\ -3.7200e-02 & -1.6000e-03 & -2.1000e-03 & 4.0896e-05 & 1.3190e-05 & 2.4923e-05 \end{bmatrix}$$

Table 7: Covariance Matrix at t_0 in @Earth ECI frame using Kourou only. Units: [km², km²/s, km²/s²]

Table 6: Navigation Solution at time t_0 in @Earth ECI frame, using measurements from Kourou only

Position in ECI [x, y, z] [km]	[3933.6560345005, -1420.1799211598, 5782.8626707129]
Velocity in ECI [v_x, v_y, v_z] [km/s]	[4.8789423872, -3.7596031155, -4.2406058775]
$\sqrt{\text{tr}(P_{rr})}$ [km]	10.5074493318
$\sqrt{\text{tr}(P_{vv})}$ [km/s]	0.0110554031
Standard deviation of semi-major axis [km]	7.1779496054
Standard deviation of inclination [deg]	0.0834766936

(b) Using Measurements from All Ground Stations

In this configuration, the measurements from all the ground stations (Kourou, Troll, and Svalbard), previously filtered based on each station's visibility condition, were used to improve the accuracy of the estimation. The process is detailed below:

- The same Keplerian propagator was used to model the satellite's trajectory, and predicted measurements were computed for each station.
- A combined least-squares optimization was performed for all stations by iterating through the structures containing the data for each station:

$$\min_{\mathbf{x}_0} \sum_{j=1}^3 \sum_{i=1}^{N_j} \mathbf{W}_j \mathbf{r}_{i,j},$$

where $\mathbf{r}_{i,j}$ are the residuals for the i -th measurement at the j -th station, and \mathbf{W}_j is the weighting matrix for that station.

Table 8: Navigation Solution at time t_0 in @Earth ECI frame, using measurements from all ground stations (Kourou, Troll, Svalbard)

Position in ECI [x, y, z] [km]	[3926.9003363166, -1411.0328854106, 5780.0210524615]
Velocity in ECI [v_x, v_y, v_z] [km/s]	[4.8885476261, -3.7649786016, -4.2333808716]
$\sqrt{\text{tr}(P_{rr})}$ [km]	0.9634856147
$\sqrt{\text{tr}(P_{vv})}$ [km/s]	0.0009892892
Standard deviation of semi-major axis [km]	0.1395701856
Standard deviation of inclination [deg]	0.0019571508

$$\mathbf{P}_0 = \begin{bmatrix} 5.6700e-01 & -3.1070e-01 & -7.0200e-02 & -6.0000e-04 & 2.0000e-04 & -3.0000e-04 \\ -3.1070e-01 & 2.9660e-01 & 4.8900e-02 & 4.0000e-04 & -1.0000e-04 & 2.0000e-04 \\ -7.0200e-02 & 4.8900e-02 & 6.4700e-02 & 3.0000e-05 & -3.0000e-05 & 7.0000e-05 \\ -6.0000e-04 & 4.0000e-04 & 3.0000e-05 & 6.3200e-07 & -1.6600e-07 & 3.1500e-07 \\ 2.0000e-04 & -1.0000e-04 & -3.0000e-05 & -1.6600e-07 & 1.2100e-07 & -1.2400e-07 \\ -3.0000e-04 & 2.0000e-04 & 7.0000e-05 & 3.1500e-07 & -1.2400e-07 & 2.2600e-07 \end{bmatrix}$$

Table 9: Covariance Matrix at t_0 in @Earth ECI frame using all the ground stations. Units: $[km^2, km^2/s, km^2/s^2]$

(c) Using J_2 -Perturbed Motion

Finally, the J_2 -perturbed motion model was incorporated to account for the oblateness of the Earth. The process is as follows:

- The satellite's trajectory was modeled using the equations for perturbed dynamics, and then numerically integrated as before.
- Measurements from all three stations were used, and a least-squares optimization was performed as in part (b), but taking into account J_2 perturbation in the prediction step.

Table 10: Navigation Solution at time t_0 in @Earth ECI frame, using measurements from all the ground stations and J_2 -perturbed dynamics.

Position in ECI $[x, y, z]$ [km]	[3932.7629524493, -1414.9159440573, 5778.4994167095]
Velocity in ECI $[v_x, v_y, v_z]$ [km/s]	[4.8798032142, -3.7632193074, -4.2328731851]
$\sqrt{\text{tr}(P_{rr})}$ [km]	0.0228519671
$\sqrt{\text{tr}(P_{vv})}$ [km/s]	0.0000239638
Standard deviation of semi-major axis [km]	0.0041842183
Standard deviation of inclination [deg]	0.0000503997

$$\mathbf{P}_0 = \begin{bmatrix} 3.2000e-04 & -1.6000e-04 & -3.9300e-05 & -3.0900e-07 & 1.1300e-07 & -1.9300e-07 \\ -1.6000e-04 & 1.7000e-04 & 3.1600e-05 & 1.9600e-07 & -4.6300e-08 & 1.3200e-07 \\ -3.9300e-05 & 3.1600e-05 & 4.1100e-05 & 1.5100e-08 & -1.2100e-08 & 4.5800e-08 \\ -3.0900e-07 & 1.9600e-07 & 1.5100e-08 & 3.5700e-10 & -8.8900e-11 & 1.9300e-10 \\ 1.1300e-07 & -4.6300e-08 & -1.2100e-08 & -8.8900e-11 & 6.8600e-11 & -6.8800e-11 \\ -1.9300e-07 & 1.3200e-07 & 4.5800e-08 & 1.9300e-10 & -6.8800e-11 & 1.4900e-10 \end{bmatrix}$$

Table 11: Covariance Matrix at t_0 in @Earth ECI frame using measurements from all the ground stations and J_2 -perturbed dynamics. Units: $[km^2, km^2/s, km^2/s^2]$

The results obtained demonstrate how different configurations of ground station measurements and dynamic models impact the accuracy of the satellite's navigation solution.

In the first case, using only Kourou station, the uncertainties in position and velocity estimation are significantly higher, as can be noted in the covariance analysis. Since SMOS is in a LEO polar orbit, it passes minimal time over the equatorial region where Kourou is located. The limited visibility and reduced observation time window lead to increased errors in the estimation of the satellite's position and velocity, as they limit the variety and volume of measurements. Progressively introducing observations from Troll and Svalbard, followed by the inclusion of J_2 perturbations in the dynamic model, results in a significant reduction in position and velocity uncertainties for the SMOS satellite.

This improvement is demonstrated in Table 8 and Table 10. The overall observation time frame is increased by combining the observation periods of the three ground stations, resulting in a more comprehensive coverage of the SMOS satellite's trajectory. The inclusion of the J2-perturbed model in the dynamics captures the perturbative effects of Earth's oblateness, providing a more accurate representation of the satellite's real trajectory. The secular nature of the J2 perturbation's effects would make its impact even more important if a longer time window were taken into account.

In this way, the residuals between the predicted measurements and the actual ones (which are the simulated measurements computed in Ex. 2.2b) are further reduced, improving the precision in position and velocity estimation.

4. A trade-off analysis has been performed to select the best combination of ground stations for estimating SMOS' position and velocity. The procedure followed is identical to that in Ex. 2.3c, but with the analysis restricted to two ground stations due to budget limitations. In Table 12 the outcomes for the three possible combinations are presented in terms of standard deviation of semi-major axis and inclination.

Table 12: Trade-off analysis results

	Kourou-Troll	Kourou-Svalbard	Troll-Svalbard
Standard deviation of semi-major axis [km]	0.6561081864	0.0167381819	0.0298248682
Standard deviation of inclination [deg]	0.0010910264	0.0005219771	0.0003969588

Analyzing the results, the second combination provides the highest accuracy in semi-major axis estimation, while the third combination achieves the best accuracy for inclination estimation. However, these results are not absolute. Due to the stochastic nature of the process, influenced by random noise in the measurements, the outcomes may vary with each iteration. By repeatedly running the procedure, it is observed that the second combination consistently offers the best accuracy for semi-major axis estimation, while for inclination estimation, this occurs frequently but not always.

Hence, the optimal configuration coming from this trade-off analysis consists of Kourou and Svalbard stations. Its effectiveness is due to the complementary geographical locations of the two stations. In the time window selected, from t_0 to t_f , SMOS passes directly over Kourou, increasing the accuracy of its measurements. Meanwhile, Svalbard's high-latitude location ensures excellent coverage in polar regions, which is particularly advantageous for a polar orbit like that of SMOS. This combination, through its geometric diversity, enhances the observability of the satellite's state, with Kourou contributing to along-track precision during low-latitude passes and Svalbard to cross-track accuracy in high-latitude passes. Indeed, Kourou-Troll combination performs worse as Troll is located in the southern hemisphere with a lower latitude compared to Svalbard station. In addition, as can be seen in Fig. 8, Svalbard complements Kourou more effectively than Troll, enhancing SMOS' coverage in the given observation campaign. Moreover, this combination is the most cost-effective, with an overall cost of 65.000€, remaining within the budget.

In conclusion, the combination of Kourou and Svalbard stations guarantees the best accuracy for state estimation while minimizing the costs.

5. In the final point, a time interval spanning from five hours before to five hours after the reference epoch t_{ref} was considered to support a nominal operations scenario. The SGP4 function was used to propagate the state at t_{ref} both forward and backward in time. This time period guarantees the validity of the provided TLE, as extending the interval into the following day would require updating the values of $d\varepsilon$ and $d\psi$ (nutation corrections required by SGP4) for the corresponding times on the next day.

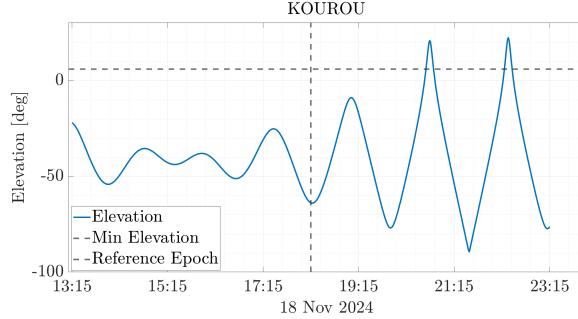


Figure 9: Elevation profile as observed from Kourou over the extended time window.

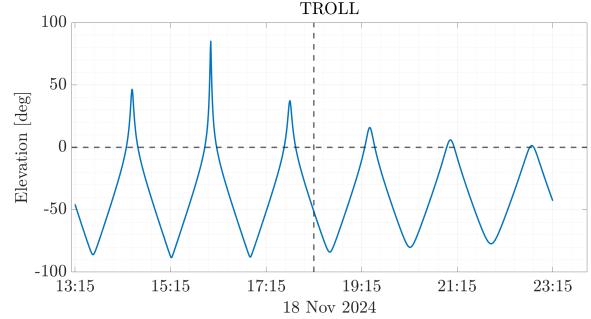


Figure 10: Elevation profile as observed from Troll over the extended time window.

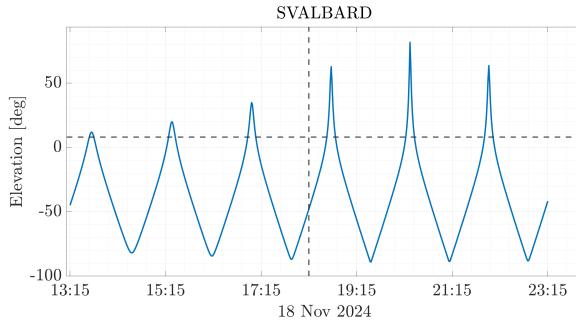


Figure 11: Elevation profile as observed from Svalbard over the extended time window.

In the figures above, the evolution of SMOS' elevation over the extended time interval for each ground station is shown, together with the SMOS' minimum elevation needed to be visible from the ground station. As remarked in Table I3, Svalbard and Troll provide noticeably higher visibility opportunities than Kourou during the extended time window. Kourou serves as a backup during equatorial runs, whereas Troll and Svalbard are selected as the principal ground stations because of the focus on the repeatability of passes and the reliability of orbit determination. The rationale behind this choice is based on SMOS' polar orbit and the geographical position of Troll (72°S) and Svalbard (78°N), which ensures excellent repeatability by providing extensive and frequent coverage in polar regions, where SMOS spends the majority of its orbital period. In this way, these two stations complement each other, allowing a complete coverage of high-latitude regions. Moreover, the high density of measurements achieved enhances the precision of orbital parameters determination, in particular inclination and cross-track components.

Meanwhile using Kourou as backup station, which is located near the Equator, during SMOS equatorial passes it complements the polar data achieved by the primary stations. The observations during equatorial passes provide an increased accuracy in semi-major axis and along-track parameters determination. Hence, the inclusion of Kourou allows a refinement of the orbital measurements provided by Troll and Svalbard.

By combining regular polar observations with specially designed equatorial measurements to improve the precision, this arrangement successfully strikes a compromise between the observations' repeatability and reliability. This ensures a strong tracking network for long-term operations.

Station	Extended Visibility Window
Kourou	18 minutes
Troll	71 minutes
Svalbard	56 minutes

Table 13: Extended visibility window for each ground station

Exercise 3: Sequential filters

An increasing number of lunar exploration missions will take place in the next years, many of them aiming at reaching the Moon's surface with landers. In order to ensure efficient navigation performance for these future missions, space agencies have plans to deploy lunar constellations capable of providing positioning measurements for satellites orbiting around the Moon.

Considering a lander on the surface of the Moon, you have been asked to improve the accuracy of the estimate of its latitude and longitude (considering a fixed zero altitude). To perform such operation you can rely on the use of a lunar orbiter, which uses its Inter-Satellite Link (ISL) to acquire range measurements with the lander while orbiting around the Moon. At the same time, assuming the availability of a Lunar Navigation Service, you are also receiving measurements of the lunar orbiter inertial position vector components, such that you can also estimate the spacecraft state within the same state estimation process.

To perform the requested tasks you can refer to the following points.

1. *Check the visibility window.* Considering the initial state \mathbf{x}_0 and the time interval with a time-step of 30 seconds from t_0 to t_f reported in Table 14, predict the trajectory of the satellite in an inertial Moon-centered reference frame assuming Keplerian motion. Use the estimated coordinates given in Table 15 to predict the state of the lunar lander. Finally, check that the lander and the orbiter are in relative visibility for the entire time interval.
2. *Simulate measurements.* Always assuming Keplerian motion to model the lunar orbiter dynamics around the Moon, compute the time evolution of its position vector in an inertial Moon-centered reference frame and the time evolution of the relative range between the satellite and the lunar lander. Finally, simulate the measurements by adding a random error to the spacecraft position vector and to the relative range. Assume a Gaussian model to generate the random error, with noise provided in Table 14 for both the relative range and the components of the position vector. Verify (graphically) that the applied noise level is within the desired boundary.
3. *Estimate the lunar orbiter absolute state.* As a first step, you are asked to develop a sequential filter to narrow down the uncertainty on the knowledge of the lunar orbiter absolute state vector. To this aim, you can exploit the measurements of the components of its position vector computed at the previous point. Using an Unscented Kalman Filter (UKF), provide an estimate of the spacecraft state (in terms of mean and covariance) by sequentially processing the acquired measurements in chronological order. To initialize the filter in terms of initial covariance, you can refer to the first six elements of the initial covariance \mathbf{P}_0 reported in Table 14. For the initial state, you can perturb the actual initial state \mathbf{x}_0 by exploiting the MATLAB function `mvnrnd` and the previously mentioned initial covariance. We suggest to use $\alpha = 0.01$ and $\beta = 2$ for tuning the UT in this case. Plot the time evolution of the error estimate together with the 3σ of the estimated covariance for both position and velocity.
4. *Estimate the lunar lander coordinates.* To fulfill the goal of your mission, you are asked to develop a sequential filter to narrow down the uncertainty on the knowledge of the lunar lander coordinates (considering a fixed zero altitude). To this aim, you can exploit the measurements of the components of the lunar orbiter position vector together with the measurements of the relative range between the orbiter and the lander computed at the previous point. Using an UKF, provide an estimate of the spacecraft state and the lunar lander coordinates (in terms of mean and covariance) by sequentially processing the acquired measurements in chronological order. To initialize the filter in terms of initial covariance, you can refer to the initial covariance \mathbf{P}_0 reported in Table 14. For the initial state, you can perturb the actual initial state, composed by \mathbf{x}_0 and the latitude

and longitude given in Table 15, by exploiting the MATLAB function `mvnrnd` and the previously mentioned initial covariance. We suggest to use $\alpha = 0.01$ and $\beta = 2$ for tuning the UT in this case. Plot the time evolution of the error estimate together with the 3σ of the estimated covariance for both position and velocity.

Table 14: Initial conditions for the lunar orbiter.

Parameter	Value
Initial state \mathbf{x}_0 [km, km/s]	$\mathbf{r}_0 = [4307.844185282820, -1317.980749248651, 2109.210101634011]$ $\mathbf{v}_0 = [-0.110997301537882, -0.509392750828585, 0.815198807994189]$
Initial time t_0 [UTC]	2024-11-18T16:30:00.000
Final time t_f [UTC]	2024-11-18T20:30:00.000
Measurements noise	$\sigma_p = 100$ m
Covariance \mathbf{P}_0 [km ² , km ² /s ² , rad ²]	<code>diag([10, 1, 1, 0.001, 0.001, 0.001, 0.00001, 0.00001])</code>

Table 15: Lunar lander - initial guess coordinates and horizon mask

Lander name	MOONLANDER
Coordinates	LAT = 78° LON = 15° ALT = 0 m
Minimum elevation	0 deg

1. In the first point, starting from the initial state \mathbf{x}_0 at time t_0 and assuming an unperturbed Keplerian motion, the trajectory of the satellite has been predicted by numerically integrating the unperturbed dynamics of the 2-Body Problem over the assigned time interval from t_0 to t_f . The predicted trajectory in Moon-Centered Inertial (MCI) reference frame is reported in Fig. 12, overlaying the Moon's surface for context.

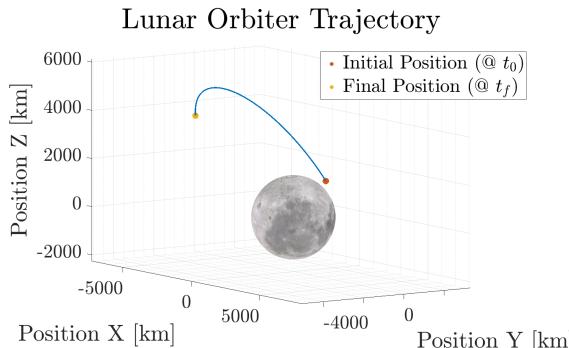


Figure 12: Lunar Orbiter Trajectory @ (Moon) Moon-Centered Inertial frame

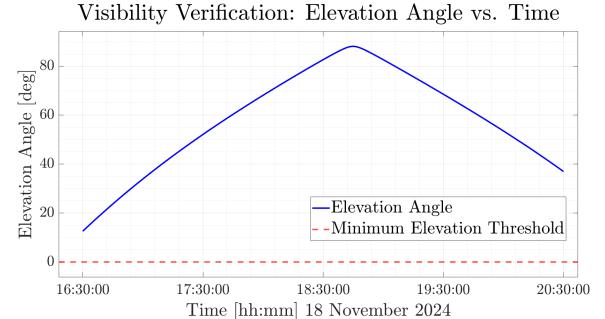


Figure 13: Elevation profile of Lunar Orbiter as observed from Lunar Lander over the time interval.

Table 16: Initial conditions at t_0 for the Lunar Lander @ (Moon) MCI frame

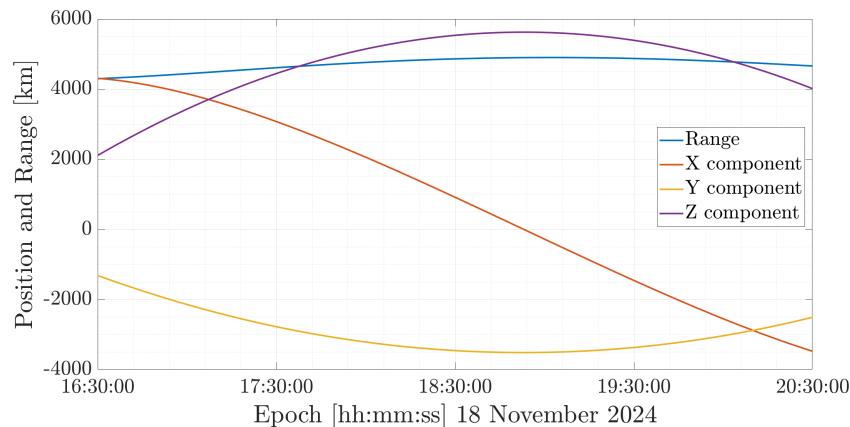
Estimated state \mathbf{x}_0	$\mathbf{r}_0 = [76.1256240239 \ -961.0674725525 \ 1445.3764086108]$ [km, km/s]
	$\mathbf{v}_0 = [0.0009393033 \ 0.0001943476 \ 0.0000797551]$

The state of the Lunar Lander was predicted starting from the given coordinates and then transformed to Cartesian coordinates in the Moon-Centered Moon-Fixed (MCMF) reference frame through the MATLAB function `cspice_latrec`, and subsequently rotated in the MCI reference frame. In Table 16 the predicted state at t_0 is reported.

Knowing the propagated position of the lunar orbiter and the predicted position of the lunar lander, the relative position of the orbiter with respect to the lander in the MCI frame was computed. After rotating the relative position in the Lunar Lander Topocentric reference frame, range, azimuth and elevation were retrieved using `cspice_xmfsta` function.

Continuous orbiter visibility from the lander was ensured by verifying the relative visibility during the entire time interval. As can be seen in Fig. 13, the calculated elevation angles continuously met the criterion, indicating no visibility interruptions.

2. The procedure adopted is the same as in the previous point, except for the determination of Lunar Lander's position. While in the previous step the position was estimated from the given coordinates, here position is retrieved directly from the given Kernels through `cspice_spkezr` function. This approach allows a more accurate evaluation of Lunar Lander's position.

**Figure 14:** Time evolution of lunar orbiter's position and relative range between orbiter and lander over the selected time interval

Next step consists in simulating real sensor data by adding a random error to the measurements. Noise with a standard deviation of $\sigma_p = 100m$ was applied to the components of the orbiter's position vector in the Moon-Centered Inertial (MCI) reference frame and the relative range. The noise was generated using a multivariate Gaussian random distribution through `mvnrnd` MATLAB function to accurately simulate sensor uncertainties.

To validate Gaussian distribution assumption, in Fig. 17 are reported the errors between the noisy and the true measurements compared with the $\pm 3\sigma$ boundaries. Since the simulated measurements consistently remain within these boundaries, the assumption of Gaussian distribution is validated, and the simulated measurements are therefore suitable for use in subsequent filtering and estimation tasks.

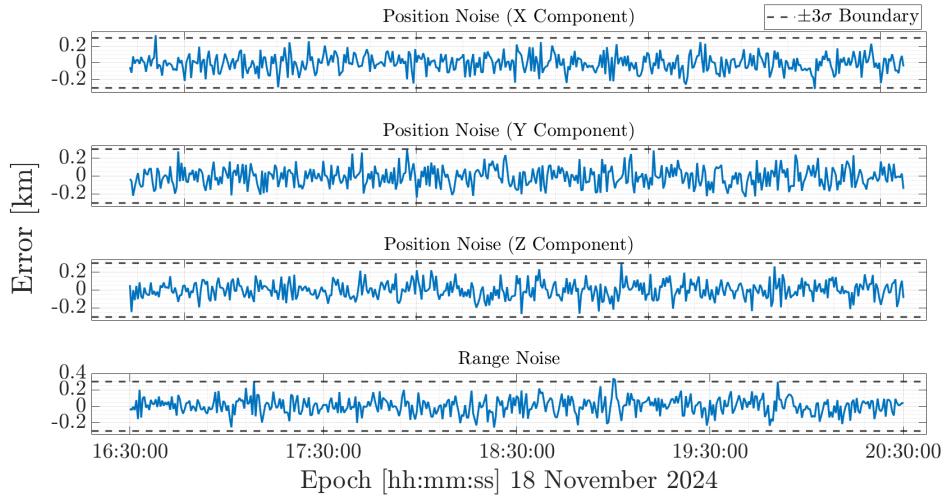


Figure 15: Comparison of the errors between noisy and true measurements for the three position components and relative range with the $\pm 3\sigma$ boundaries.

3. In the third point, an Unscented Kalman Filter (UKF) was implemented to reduce the uncertainties on the knowledge of the lunar orbiter state vector. Given the parameters required to tune up the UT (α, β), λ can be computed as:

$$\lambda = \alpha^2(N + k)$$

where N is the state dimension (6 in this case) and k is zero. The algorithm implemented is detailed hereafter.

Algorithm 1 Unscented Kalman Filter (UKF) for Lunar Orbiter

Input: Initial state \mathbf{x}_0 , Initial covariance \mathbf{P}_0 , Time span t_{span} , Simulated noisy measurements, Measurement noise covariance \mathbf{R} , UKF parameters α, β, λ

Output: Estimated state \mathbf{x}_k^+ , Covariance History \mathbf{P}_k^+

Compute weights $W_i^{(m)}, W_i^{(c)}$ based on UKF parameters

for each time step k from 1 to $\text{length}(t_{\text{span}}) - 1$ **do**

- Prediction Step:**
- Generate sigma points χ_{k-1} from $\hat{\mathbf{x}}_{k-1}^+$ and \mathbf{P}_{k-1}^+
- Propagate sigma points using Keplerian dynamics: χ_k
- Compute predicted state mean: $\hat{\mathbf{x}}_k^-$ using $W_i^{(m)}$
- Compute predicted covariance: \mathbf{P}_k^- using $W_i^{(c)}$
- Measurement Update:**
- Transform sigma points to measurement space: γ_k by extracting position components
- Compute predicted measurement mean: $\hat{\mathbf{y}}_k^-$ using $W_i^{(m)}$
- Compute measurement covariance: $\mathbf{P}_{yy,k} = \mathbf{P}_{ee,k} + \mathbf{R}$
- Compute cross-covariance: $\mathbf{P}_{xy,k}$
- Compute Kalman gain: $\mathbf{K}_k = \mathbf{P}_{xy,k} \mathbf{P}_{yy,k}^{-1}$
- Update state using simulated noisy measurements: $\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{y}_k - \hat{\mathbf{y}}_k^-)$
- Update covariance: $\mathbf{P}_k^+ = \mathbf{P}_k^- - \mathbf{K}_k \mathbf{P}_{yy,k} \mathbf{K}_k^\top$

end for

The output of UKF is the set of estimated states of the lunar orbiter and the associated covariance matrix for each time step. In Table 19 the final estimated state and the last covariance matrix at t_f , the last outcome of UKF for Lunar Orbiter, are reported.

Table 17: Final estimated state and covariance matrix at t_f of UKF for the Lunar Orbiter @ (Moon) MCI frame

Estimated state \mathbf{x}_f [km, km/s]	$\mathbf{r}_f = [-3479.03198, -2506.20610, 4010.74990]$ $\mathbf{v}_f = [-0.45012, 0.34854, -0.55778]$
--	---

$$\mathbf{P}_f = \begin{bmatrix} 7.1889e-05 & 6.7147e-06 & -1.0725e-05 & 5.7098e-09 & 3.5623e-09 & -5.6973e-09 \\ 6.7147e-06 & 8.1518e-05 & -2.8262e-05 & 5.2357e-09 & 8.8739e-09 & -9.4276e-09 \\ -1.0725e-05 & -2.8262e-05 & 1.0906e-04 & -8.3674e-09 & -9.4275e-09 & 1.8059e-08 \\ 5.7098e-09 & 5.2357e-09 & -8.3674e-09 & 1.9460e-12 & 1.2154e-12 & -1.9417e-12 \\ 3.5623e-09 & 8.8739e-09 & -9.4275e-09 & 1.2154e-12 & 1.9570e-12 & -1.8683e-12 \\ -5.6973e-09 & -9.4276e-09 & 1.8059e-08 & -1.9417e-12 & -1.8683e-12 & 3.7757e-12 \end{bmatrix}$$

Table 18: Covariance Matrix at t_f @ (Moon) MCI frame. Units: [km², km²/s, km²/s²]

To assess the performance of the UKF in estimating the lunar orbiter's position and velocity, two different errors have been computed.

The position error was defined as the norm of the difference between the true propagated position and the UKF-estimated position. Similarly, the velocity error was computed as the norm of the difference between the true propagated velocity and the UKF-estimated one. These two errors clearly assess the UKF's true effectiveness in estimating the lunar orbiter's state.

Then, from the diagonal elements of the corresponding submatrix, the 3σ bounds for both position and velocity were derived. These bounds are necessary to validate the filter's effectiveness, as they represent the uncertainty in the state estimate.

In Fig. 16, the time evolution of position and velocity errors along with their corresponding 3σ boundaries is shown. This visualization helps in verifying whether the estimation errors remain within the statistical confidence intervals predicted by the UKF.

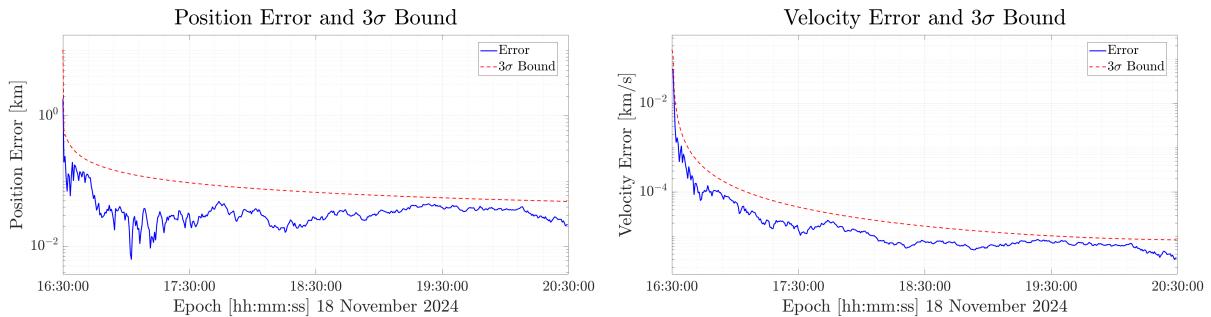


Figure 16: Evolution of the Lunar Orbiter's position and velocity errors over the selected time interval, along with the corresponding 3σ boundaries.

The efficiency of the UKF is evident from the generated plots for position and velocity errors. The perturbed initial state causes the errors to be relatively high at first, but position and velocity errors converge as the filter processes more samples. The errors stay well inside the 3σ boundaries throughout the simulation, confirming the Gaussian assumptions and showing that the filter is correctly tuned.

Because position estimations are more sensitive to measurement noise, the position error decreases more slowly than the velocity error. Since deterministic orbital dynamics dominate velocity propagation, the velocity error exhibits a faster convergence. By balancing the data from the dynamics model and the implemented UKF, these observations confirm its ability to produce accurate state estimations.

4. In the last point, an UKF has been implemented to reduce uncertainties in the knowledge of the lunar lander's coordinates exploiting the measurements of lunar orbiter position and of relative range between the orbiter and the lander. The approach implemented is similar to that used in Ex. 3.3, with some slight differences. The augmented initial state \mathbf{x}_0 now includes both the orbiter's initial state and the coordinates of the lander, resulting in a state vector of dimension $[8 \times 1]$. Analogously, the initial covariance matrix \mathbf{P}_0 has been updated to $[8 \times 8]$. The relative range, which is dynamically calculated at each algorithmic time step, is now also included in the measurements. The complete method is described below.

Algorithm 2 Unscented Kalman Filter (UKF) for Lunar Lander

Input: Initial state \mathbf{x}_0 (orbiter state and lander coordinates), Initial covariance \mathbf{P}_0 , Time span t_{span} , Simulated noisy measurements (orbiter position and relative range), Measurement noise covariance \mathbf{R} , UKF parameters α, β, λ

Output: Estimated state \mathbf{x}_k^+ (orbiter state and lander coordinates), Covariance \mathbf{P}_k^+

Compute weights $W_i^{(m)}, W_i^{(c)}$ based on UKF parameters

for each time step k from 1 to $\text{length}(t_{\text{span}}) - 1$ **do**

Prediction Step:

Same as algorithm 1

Measurement Update:

Transform sigma points to measurement space:

- Extract orbiter position for the first three components
- Compute the relative range by converting the lander's coordinates to Cartesian coordinates in the Moon-Centered Inertial (MCI) frame and calculating the norm of the difference between the orbiter sigma points and the transformed lander position.

Same as algorithm 1

end for

The output of UKF is the set of estimated states of the lunar orbiter and the estimated coordinates of the lunar lander, along with the associated covariance matrix for each time step. In the final estimated state and the last covariance matrix at t_f are reported.

Table 19: Final estimated state and covariance matrix at t_f of UKF for the Lunar Orbiter and Lander @ (Moon) MCI frame.

								$\mathbf{r}_f = [-3479.02670, -2506.20455, 4010.74746]$
Estimated state \mathbf{x}_f								$\mathbf{v}_f = [-0.45012, 0.34854, -0.55778]$
[km, km/s, deg, deg]								$lat_f = 78.2375$
								$lon_f = 15.4205$

$$\mathbf{P}_f = \begin{bmatrix} 6.2618e-05 & 4.0603e-06 & -6.5146e-06 & 3.7214e-09 & 2.6343e-09 & -4.2101e-09 & -2.4898e-04 & -2.5679e-04 \\ 4.0603e-06 & 8.1562e-05 & -2.8390e-05 & 5.1070e-09 & 8.8111e-09 & -9.3340e-09 & -2.8443e-04 & -2.5160e-04 \\ -6.5146e-06 & -2.8390e-05 & 1.0935e-04 & -8.1939e-09 & -9.3431e-09 & 1.7935e-08 & 4.6000e-04 & 4.0533e-04 \\ 3.7214e-09 & 5.1070e-09 & -8.1939e-09 & 1.7440e-12 & 1.1257e-12 & -1.8017e-12 & -5.3990e-11 & -6.0460e-11 \\ 2.6343e-09 & 8.8111e-09 & -9.3431e-09 & 1.1257e-12 & 1.9152e-12 & -1.8033e-12 & -5.2676e-11 & -4.4607e-11 \\ -4.2101e-09 & -9.3340e-09 & 1.7935e-08 & -1.8017e-12 & -1.8033e-12 & 3.6747e-12 & 8.3694e-11 & 7.0375e-11 \\ -2.4898e-04 & -2.8443e-04 & 4.6000e-04 & -5.3990e-11 & -5.2676e-11 & 8.3694e-11 & 1.0199e-08 & 1.1236e-08 \\ -2.5679e-04 & -2.5160e-04 & 4.0533e-04 & -6.0460e-11 & -4.4607e-11 & 7.0375e-11 & 1.1236e-08 & 1.3342e-08 \end{bmatrix}$$

Table 20: Covariance Matrix at t_f @ (Moon) MCI frame Units: [km², km²/s, km²/s², km rad, km/s rad, rad²]

As in the previous point, to assess the performance of the UKF in estimating lunar orbiter's position and velocity and lunar lander coordinates, four different errors have been computed. Along with the position and velocity errors computed as before, latitude and longitude errors were computed as the absolute value of the difference between the true coordinates with the UKF-estimated ones.

In the plots presented hereafter are reported the time evolution of the errors along with their corresponding 3σ boundaries.

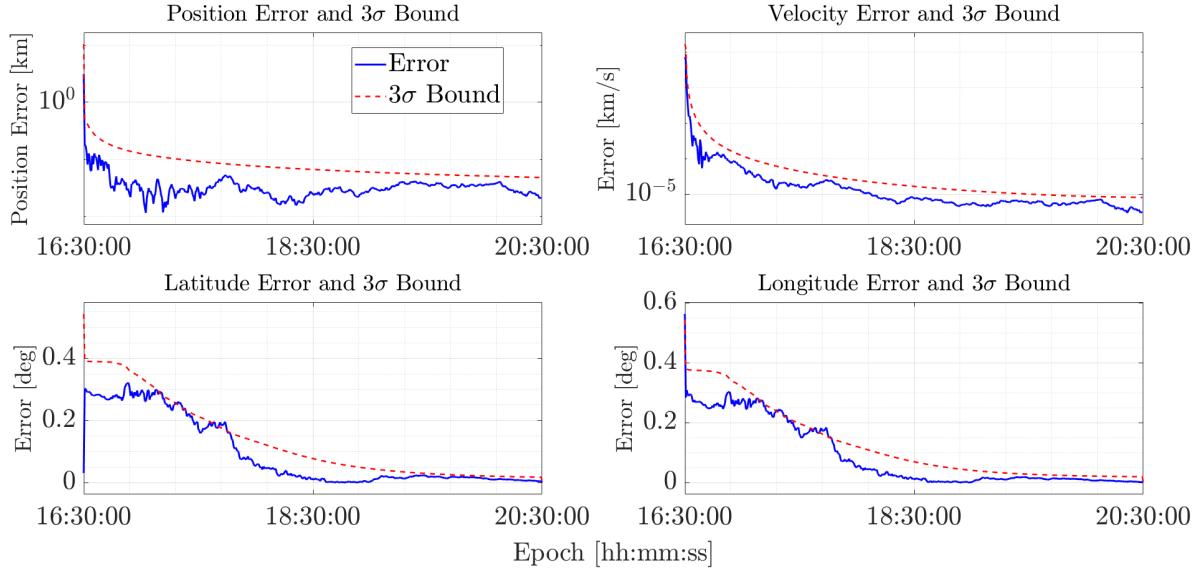


Figure 17: Evolution of the Lunar Orbiter's position and velocity errors and the Lunar Lander's coordinates over the selected time interval, along with the corresponding 3σ boundaries.

The first row presents the errors for the lunar orbiter's position and velocity. The UKF performances are satisfactory as the errors systematically falls below the 3σ boundary. Regarding the different trend, same considerations as Ex. 3.3 can be done.

The bottom row presents the errors in lander's coordinates estimation. It can be noticed they slightly exceed the 3σ bounds, particularly in the early stages of the simulation. This behaviour can be attributed to the non-linearities present in the measurement model adopted to estimate the relative range, along with the random nature of the generated noise. The UKF may not fully represent these non-linearities in the initial phase, where uncertainties are larger.

Despite these occasional violations, as the UKF incorporates additional observations and reduces the uncertainty in the state estimate, the latitude and longitude errors exhibit clear convergence. The filter's overall performance demonstrates its robustness and capability in determining the lander's coordinates based on the orbiter's location and relative range measurements. The efficiency of the UKF in reducing uncertainty within a non-linear system is further validated by the convergence of all errors over time toward their respective 3σ bounds.

Appendix A: Mapping Covariance from Cartesian State to Keplerian Elements

To analyze the uncertainties in the estimated orbital parameters, the covariance matrix expressed in the Cartesian state (\mathbf{r}, \mathbf{v}) is mapped to the Keplerian orbital elements ($a, e, i, \Omega, \omega, \theta$). This is achieved through a linear mapping procedure that employs the Jacobian matrix of the transformation between these states.

1. Jacobian Matrix Computation:

The Jacobian matrix $\mathbf{J}_{\text{Kepler}}$ relates small variations in the Cartesian state ($\Delta\mathbf{r}, \Delta\mathbf{v}$) to corresponding variations in the Keplerian elements ($\Delta a, \Delta e, \Delta i, \Delta \Omega, \Delta \omega, \Delta \theta$):

$$\Delta \mathbf{k} = \mathbf{J}_{\text{Kepler}} \Delta \mathbf{x}$$

where:

$$\mathbf{k} = \begin{bmatrix} a \\ e \\ i \\ \Omega \\ \omega \\ \theta \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} \mathbf{r} \\ \mathbf{v} \end{bmatrix}, \quad \mathbf{J}_{\text{Kepler}} = \frac{\partial \mathbf{k}}{\partial \mathbf{x}}$$

The elements of $\mathbf{J}_{\text{Kepler}}$ are computed numerically using finite differences:

$$J_{ij} = \frac{\partial k_i}{\partial x_j} \approx \frac{k_i(\mathbf{x} + \epsilon \mathbf{e}_j) - k_i(\mathbf{x})}{\epsilon}$$

where ϵ is a small perturbation applied to the j -th Cartesian component, and \mathbf{e}_j is the corresponding unit vector.

2. Covariance Mapping:

Once $\mathbf{J}_{\text{Kepler}}$ is computed, the covariance matrix in Keplerian elements, $\mathbf{P}_{\text{Kepler}}$, is obtained using the following transformation:

$$\mathbf{P}_{\text{Kepler}} = \mathbf{J}_{\text{Kepler}} \mathbf{P}_{\text{Cartesian}} \mathbf{J}_{\text{Kepler}}^T$$

Here, $\mathbf{P}_{\text{Cartesian}}$ is the covariance matrix in the Cartesian state, obtained as part of the least-squares estimation process.

3. Extraction of Orbital Uncertainty:

The diagonal elements of $\mathbf{P}_{\text{Kepler}}$ represent the variances of the Keplerian elements. For example:

$$\sigma_a = \sqrt{P_{\text{Kepler},1,1}}, \quad \sigma_i = \sqrt{P_{\text{Kepler},3,3}}$$

These values directly quantify the uncertainty in the semimajor axis (a) and inclination (i).

This linear approximation is valid under the assumption of small uncertainties in the Cartesian state, where the transformation between states remains locally linear. It provides a computationally efficient way to evaluate the impact of estimated uncertainties on the physical orbital parameters, aiding in both interpretation and mission planning. However, for highly nonlinear scenarios, higher-order techniques or Monte Carlo simulations may be needed to capture the full impact of uncertainties.