SGN – Assignment #2

Emanuele Gallo, 243222

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 (Topputo, 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 1.

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 [†]. 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. 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.

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

1.1 LinCov and UT for uncertainty propagation

1.1.1 Introduction

This paragraph addresses the uncertainty propagation using two methods: the linear covariance (LinCov) and the Unscented Transform (UT). The framework is the Planar Bicircular Restricted Four-Body Problem (PBRFBP) for an optimal Earth-Moon transfer trajectory, as outlined in [3]. This non-linear dynamics model has been implemented in the MATLAB® function `xyPBRFBP_STM_ROT` accordingly to the previous reference and can be found in subsubsection A.1.1 in Appendix A. The solver chosen to propagate the dynamics is `ode78`, a high-order fixed-step solver, which is employed for its robustness and reliability in smooth, non-stiff dynamics such as spacecraft trajectory. This fixed-step approach ensures convergence properties are attributed solely to the optimization methods, isolating algorithmic behavior. Integration tolerances are tailored to balance precision and efficiency: $\text{Tol}_{\mathbf{r}} = 10^{-10}$ for position, $\text{Tol}_{\mathbf{v}} = 10^{-13}$ for velocity, and $\text{Tol}_{\mathbf{\Phi}} = 10^{-13}$ for the state transition matrix, reflecting their respective influences on the dynamics.

1.1.2 Methods

The initial state $\mathbf{x}_i$, covariance $\mathbf{P}_0$, and time interval $[t_i, t_f]$ are provided in Table 1. These are propagated through the dynamics to obtain the mean state and associated covariance at discrete time steps according to the assigned methods: Linearized Covariance (LinCov), and Unscented Transform (UT). Here, only their general method is described, but their treatment is deepened in subsubsection A.1.3 and subsubsection A.1.4, respectively. The methods adopted are:

- **Linearized Covariance Method**: the LinCov method propagates uncertainty in non-linear dynamics by linearizing the system around the nominal trajectory. The mean final state coincides with the propagated one. At the same time, the evolution of the covariance matrix is governed by the following equation: $\mathbf{P}(t) = \mathbf{\Phi}(t, t_i)\mathbf{P}_0\mathbf{\Phi}^T(t, t_i)$. Where $\mathbf{\Phi}(t, t_i)$ is the State Transition Matrix (STM), which is computed by integrating the variational equations of the system dynamics;

- **Unscented Transform Method**: the Unscented Transform (UT) method approximates uncertainty propagation by deterministically sampling sigma points around the initial state mean and propagating them through the nonlinear dynamics. Sigma points and their weights are constructed based on the covariance matrix, and parameters like n, the dimension of the state to be propagated, α and κ, user-defined constant that influence the spread and location of the sigma points, and β. The sigma points are propagated through the dynamics, and the sample mean and covariance of the propagated states are computed using weighted sums.

1.1.3 Results: LinCov and UT comparison

To compare the methods, the mean state at final time $\mathbf{x_f}$ and the related covariance matrix are used. The results are shown in the following Table 2. Moreover, to make the comparison more visually clear, also the 3σ uncertainty ellipse is derived from the position covariance matrix is plotted. In particular, the uncertainty ellipse is retrieved as explained in subsubsection A.1.5 in Appendix A, which is in turn taken from [1]. The related plot is presented in Figure 1.

Method	**Final State** [DU, VU]	**Covariance Matrix** [DU^2, VU^2, DU·VU]
LC	$\begin{bmatrix} 0.98691479 \\ -0.00468759 \\ -2.48005110 \\ 0.49448183 \end{bmatrix}$	$\begin{bmatrix} 2.9098 \cdot 10^{-7} & -1.3134 \cdot 10^{-7} & 6.2352 \cdot 10^{-7} & -7.4218 \cdot 10^{-5} \\ -1.3134 \cdot 10^{-7} & 5.9283 \cdot 10^{-8} & -2.8150 \cdot 10^{-7} & 3.3500 \cdot 10^{-5} \\ 6.2352 \cdot 10^{-7} & -2.8150 \cdot 10^{-7} & 1.3474 \cdot 10^{-6} & -0.0001590 \\ -7.4218 \cdot 10^{-5} & 3.3500 \cdot 10^{-5} & -0.0001590 & 1.8930 \cdot 10^{-2} \end{bmatrix}$
UT	$\begin{bmatrix} 0.98691271 \\ -0.00467069 \\ -2.47201430 \\ 0.49968622 \end{bmatrix}$	$\begin{bmatrix} 2.8995 \cdot 10^{-7} & -1.3175 \cdot 10^{-7} & 9.1704 \cdot 10^{-8} & -7.3727 \cdot 10^{-5} \\ -1.3175 \cdot 10^{-7} & 6.0521 \cdot 10^{-8} & 2.8861 \cdot 10^{-7} & 3.3692 \cdot 10^{-5} \\ 9.1704 \cdot 10^{-8} & 2.8861 \cdot 10^{-7} & 1.6633 \cdot 10^{-4} & 7.3425 \cdot 10^{-5} \\ -7.3727 \cdot 10^{-5} & 3.3692 \cdot 10^{-5} & 7.3425 \cdot 10^{-5} & 1.8803 \cdot 10^{-2} \end{bmatrix}$

Table 2: Final State and Covariance Matrices for LC and UT Methods (@EMB Earth-Moon rotating frame at time t_f)

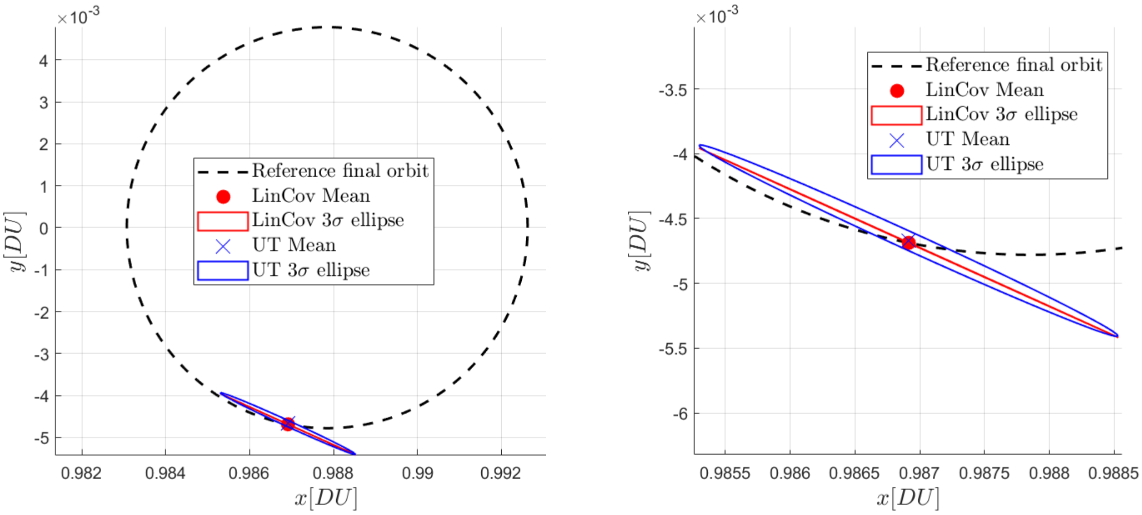

Figure 1: Mean and associated 3σ uncertainty ellipse for LinCov and UT (@EMB Earth-Moon rotating frame at time t_f). General view (left), and zoomed-in (right).

The comparison between the Linear Covariance (LinCov) and Unscented Transform (UT) methods demonstrates strong agreement in the mean state estimation, with both methods yielding nearly identical results, as depicted by the red dot (LinCov) and blue cross (UT) in the figure. This consistency reflects the reliability of both methods in propagating the mean state under the assumption of Gaussian distributions. However, differences are evident in the representation of the uncertainty ellipses. The LinCov 3σ ellipse is noticeably thinner and more elongated along the semi-major axis compared to the UT 3σ ellipse. This discrepancy arises from LinCov's reliance on a first-order Taylor expansion, which underestimates uncertainties, particularly along directions orthogonal to the primary mode of variation, as it neglects higher-order nonlinearities. In contrast, the UT method captures the nonlinear dynamics more accurately by propagating sigma points through the nonlinear system, thus incorporating higher-order moments. This results in a UT ellipse that appears slightly broader, particularly along the semi-minor axis and provides a more faithful representation of the true uncertainty distribution. The slight misalignment of the semi-major axes and the corresponding differences in the eigenvalues of the covariance matrices reflect these fundamental differences between the two methods. On the other hand, the UT method is intrinsically more computationally demanding than LinCov.
Overall, it is suggested to prefer the UT method whenever nonlinearities play a significant role in uncertainty propagation, since, in this case, LinCov, although computationally efficient, might not be capable of modelling the uncertainty distribution accurately.

1.2 Monte Carlo method for uncertainty propagation

1.2.1 Monte Carlo Simulation for Uncertainty Propagation: Method

In the following paragraph, the Monte Carlo (MC) method is briefly explained, the full treatment can be found in subsubsection A.1.6 in Appendix A.
The Monte Carlo (MC) method involves sampling a large number of initial states from a multivariate normal distribution, generated using `mvnrnd`, based on the initial mean $\mathbf{x_i}$ and covariance $\mathbf{P_0}$. Each sample is propagated through the nonlinear dynamics over a specified time sequence. At each time step, the mean and covariance of the propagated states are computed using MATLAB's `mean` and `cov` functions. This approach provides a statistical representation of the uncertainty by directly simulating the dynamics of perturbed states. As a final remark, the Monte Carlo (MC) method assumes an initial Gaussian uncertainty, sampling perturbed states from a multivariate normal distribution using the `mvnrnd` function. While the Gaussian assumption holds initially, it is invalidated as uncertainty propagates through the system's nonlinear dynamics.

1.2.2 Results and final comparison of all methods

The mean state and covariance matrix at final time in rotating reference frame at final time t_f are summarized in the following Table 3:

Final State [DU, VU]	**Covariance Matrix** [DU^2, VU^2, DU·VU]
$\begin{bmatrix} 0.98691569 \\ -0.00467224 \\ -2.47213826 \\ 0.49880240 \end{bmatrix}$	$\begin{bmatrix} 2.8555 \cdot 10^{-7} & -1.3037 \cdot 10^{-7} & -2.7848 \cdot 10^{-7} & -7.2368 \cdot 10^{-5} \\ -1.3037 \cdot 10^{-7} & 6.0023 \cdot 10^{-8} & 3.7707 \cdot 10^{-7} & 3.3179 \cdot 10^{-5} \\ -2.7848 \cdot 10^{-7} & 3.7707 \cdot 10^{-7} & 1.2458 \cdot 10^{-4} & 1.3957 \cdot 10^{-4} \\ -7.2368 \cdot 10^{-5} & 3.3179 \cdot 10^{-5} & 1.3957 \cdot 10^{-4} & 1.8381 \cdot 10^{-2} \end{bmatrix}$

Table 3: Final State and Covariance Matrix for MC Method in @EMB Earth-Moon rotating frame at time t_f

The uncertainty ellipses associated with all the methods exposed, together with the final state for the MC simulation are reported in the following Figure 2:

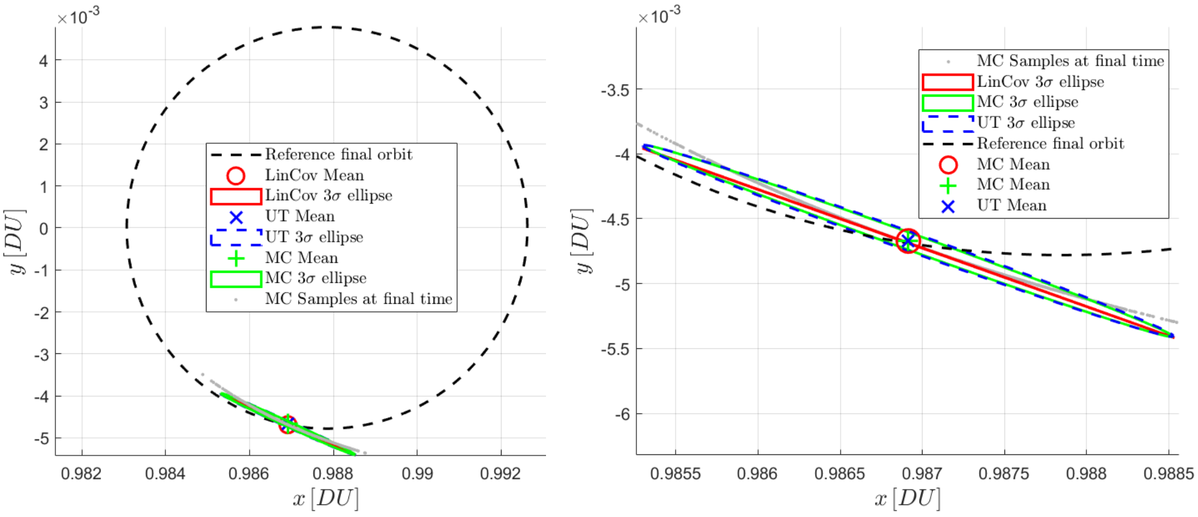

Figure 2: Final states and associated 3σ covariance ellipses of LinCov, UT, and MC (@EMB Earth-Moon rotating frame at time t_f). General view (left) and zoomed-in (right).

MC results, depicted in Figure 2, serve as a benchmark for assessing the performance of the other methods in approximating uncertainty evolution. In fact, as explained in subsubsection 1.2.1, the MC 3σ ellipse and associated state capture the true uncertainty distribution, reflecting distortions and skewness induced by nonlinear dynamics.

Both LinCov and UT methods predict the mean final state accurately, but only the UT is capable of estimating precisely the uncertainty ellipse. This is due to the underlying hypotheses of the two methods. In fact, LinCov assumes linear dynamics and Gaussian uncertainty, resulting in a thinner 3σ ellipse (red) that underestimates the true uncertainty, particularly in directions orthogonal to the main variation mode. This limitation arises from neglecting higher-order nonlinear effects, which stretch and distort the uncertainty distribution. While the UT method propagates sigma points through the nonlinear dynamics, capturing higher-order effects and yielding a 3σ ellipse (blue) that better approximates the MC results.

This comparison highlights the importance of evaluating the Gaussian assumption in uncertainty propagation:

- For systems with low or mild nonlinearities and Gaussian-like uncertainties, the LinCov method offers a computationally efficient and reasonably accurate, but not precise approximation.
- For systems where nonlinearities significantly affect uncertainty evolution, the UT method provides a more accurate and precise approximation of the mean and covariance, albeit at higher computational cost.
- The MC method delivers the most robust and accurate results, serving as a baseline to validate approximate methods. However, its computational demand makes it impractical unless high precision is essential.

For the analyzed case, the UT method strikes a balance between accuracy, precision and computational efficiency, providing a highly accurate approximation without the exhaustive computational cost of the MC method.

The time evolution of $3\sqrt{\max \lambda_i(\mathbf{P_r})}$ and $3\sqrt{\max \lambda_i(\mathbf{P_v})}$, where $\mathbf{P_r}$ and $\mathbf{P_v}$ are the 2×2 position and velocity covariance submatrices (i.e. extracted from the covariance matrix relative to the state at that time instant), provides insights into the maximum uncertainty along the principal directions of the system. These quantities, derived from the largest eigenvalues of $\mathbf{P_r}$ and $\mathbf{P_v}$, represent the growth of the maximum uncertainty over time but do not capture the overall uncertainty. Figure 3 illustrates these results:

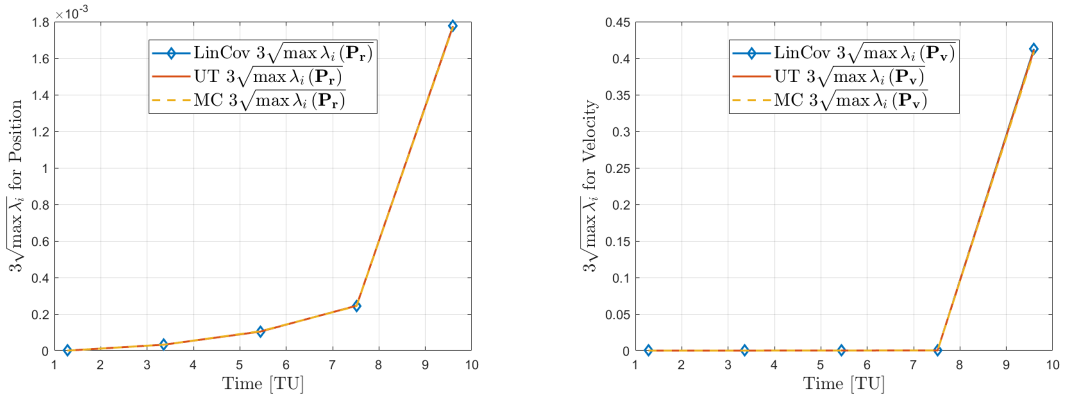

Figure 3: Time evolution of $3\sqrt{\max \lambda_i(P_r)}$ and $3\sqrt{\max \lambda_i(P_v)}$.

Initially, both metrics of uncertainty remain small, but they grow significantly over time. The increase in $3\sqrt{\max \lambda_i(P_r)}$ is particularly rapid, becoming explosive near the final times, reflecting the system's sensitivity to perturbations and the accumulation of nonlinear effects. Both LinCov (blue diamonds) and UT (red) methods align closely with MC (yellow dashed line)

results, validating their accuracy in modeling the maximum uncertainty along the principal direction. This conclusion is consistent with Figure 2 results, where both the method where capable of modelling the uncertainty ellipse along the maximum uncertainty direction.
The final plot to compare the data is the Quantile-Quantile plots for each component, which is represented in the following Figure 4

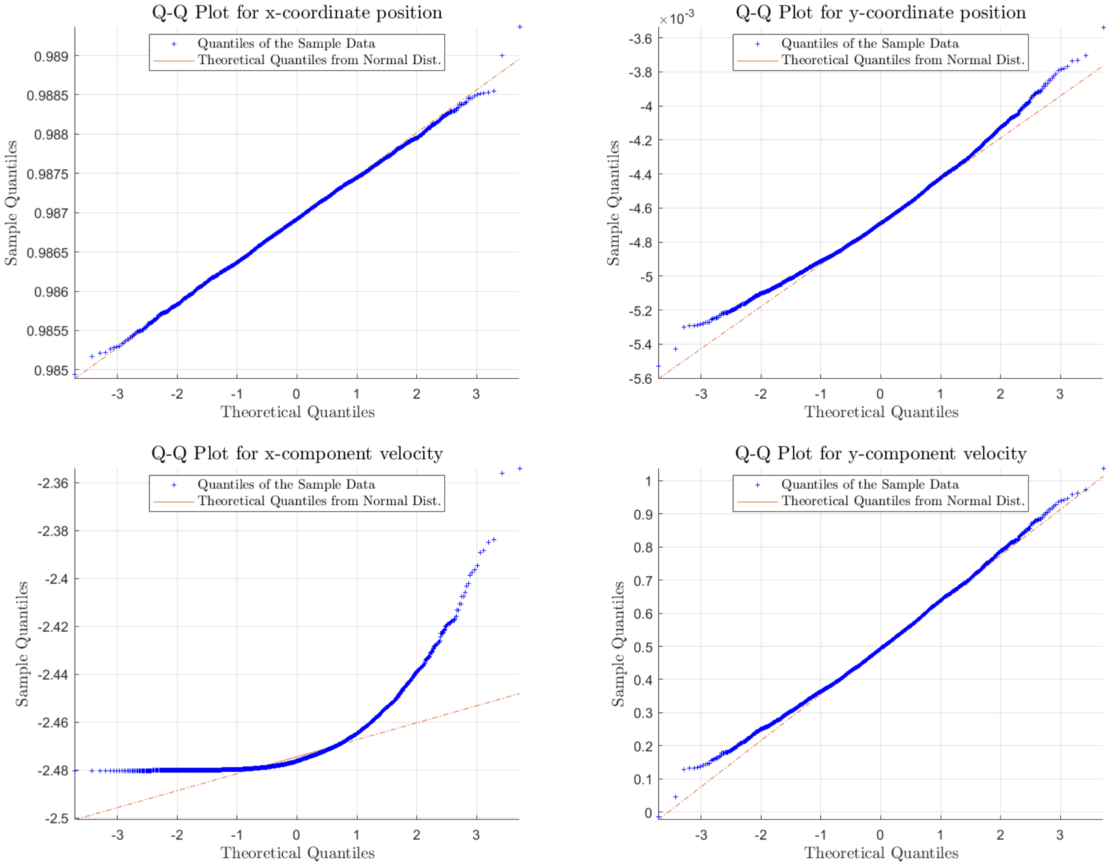

Figure 4: QQ plots for the components of the state at time t_f @EMB Earth-Moon rotating frame

The Q-Q plots provide a fundamental tool for assessing the validity of the Gaussian assumption at the final time. The red reference line represents the theoretical quantiles of the normal distribution, which serve as a benchmark. If the Monte Carlo (MC) samples were perfectly Gaussian, their empirical quantiles would align closely with this line.
At the initial time, the MC samples were generated under the assumption of Gaussianity; hence, deviations from this line at the final time arise primarily due to the effects of nonlinear dynamics. These discrepancies highlight regions where higher-order nonlinear effects influence the evolution of uncertainties. Specifically, the most significant departures are observed in the x-component of the velocity and the y-component of the position. The right-skewed nature of these curves indicates the presence of asymmetric dynamics, which are particularly pronounced in the Planar Bicircular Restricted Four-Body Problem (PBRFBP).
Overall, the observed skewness and deviations underscore the limitations of the Gaussian assumption in capturing the full behavior of the propagated uncertainties under nonlinear dynamics. This analysis reinforces the necessity of considering higher-fidelity and non-Gaussian-like methods whenever high accuracy is of concern.

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 4.

1. *Compute visibility windows.* The Two-Line Elements (TLE) set of SMOS are reported in Table 5 (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 4 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 4). 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 5 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 4. 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 4: 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 σ_{range} = 0.01 km	$\sigma_{Az,El}$ = 125 mdeg σ_{range} = 0.01 km	$\sigma_{Az,El}$ = 125 mdeg σ_{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 5: 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
```

2.1 Visibility Window

This section describes the method used to compute the visibility windows for a network of ground stations comprising KOUROU, TROLL, and SVALBARD, as presented in Table 7. The analysis considers the SMOS (Soil Moisture and Ocean Salinity) satellite, launched on November 2, 2009, by ESA. The satellite's orbital information is derived from its Two-Line Elements (TLE), as provided in Table 6. Based on this information, the satellite's Keplerian orbital parameters are extracted, and the results are presented in Table 6:

Parameter	Value
Satellite ID	36036
TLE Reference Epoch t_{ref}	UTC 2024-11-18T18:15:16.064640
Semi-major Axis, a	7143.713 km
Eccentricity, e	0.001244
Inclination, i	1.718001 rad (98.434 deg)
RAAN, Ω	2.591271 rad (148.469 deg)
Argument of Periapsis, ω	1.209390 rad (69.293 deg)
True Anomaly, θ	5.076121 rad (290.840 deg)

Table 6: Satellite Keplerian Elements @Earth J2000 inertial

2.1.1 Method and results

The computation of visibility windows for the time interval $[t_0, t_f]$ assumes Keplerian orbital motion for the satellite. The key steps of the method are as follows:

- The spacecraft's state vector is propagated from t_0 to t_f using Keplerian orbital dynamics, as described in subsubsection A.2.1 in Appendix A.
- The ground station positions and velocities are obtained in the Earth-Centered Inertial (ECI) reference frame from SPICE kernel data.
- The relative position vector of the spacecraft with respect to each station, initially expressed in the ECI frame, is transformed into the corresponding topocentric (local horizon) reference frame of the station.
- Azimuth, elevation, and range are computed for each time step. The elevation angle is compared against the station-specific visibility mask angle to determine if the spacecraft is visible.
- If visibility conditions are met, the corresponding values of azimuth, elevation, range, and time are stored for further analysis.

The results of this analysis are reported in the following Table 7:

Table 7: Visibility Windows for Ground Stations

Ground Station	Start (UTC)	End (UTC)
KOUROU	2024-11-18 20:40:00.000	2024-11-18 20:49:00.000
TROLL	2024-11-18 21:02:30.000	2024-11-18 21:11:30.000
SVALBARD	2024-11-18 21:56:00.000	2024-11-18 22:06:00.000

The station with bigger visibility is clearly SVALBARD. This is due to the trajectory arc (from t_0 to t_f) done by the satellite, which clearly advantages SVALBARD station over the other

two. Additionally, also if it does not have the spacecraft in visibility for the biggest amount of time, the station which acquires more measurement is TROLL, due to its higher frequency of measurement acquisition.
The predicted Azimuth and Elevation profiles within the proper time interval for each station are reported in the following Figure 5:

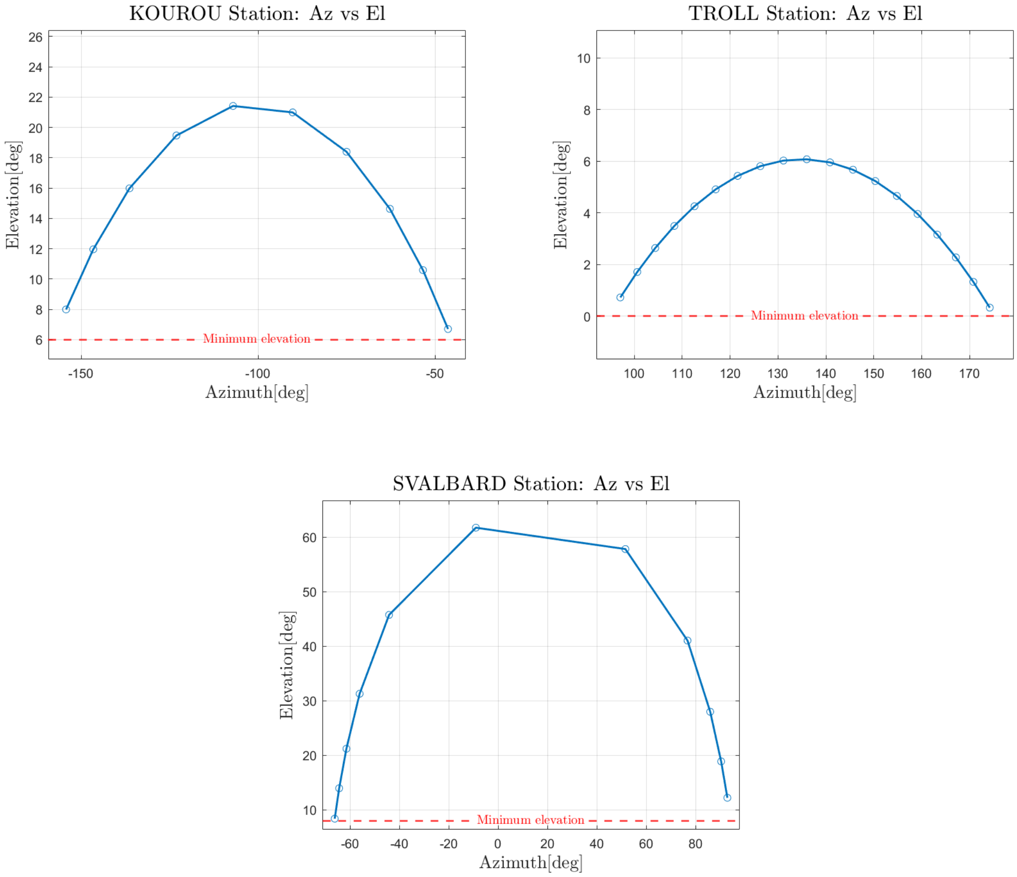

Figure 5: Az-El time evolution profile for different stations within the proper visibility window (in station centred topocentric reference frame)

2.2 Measurement Simulation

This section details the simulation of measurements acquired by the sensor network. The process is implemented in the function `measurements` and follows an analogous procedure to that described in subsubsection 2.1.1. However, the propagation of the satellite's state is performed using the `sgp4` model instead of Keplerian dynamics, and random measurement errors are introduced. The introduction of noise is achieved using MATLAB® built-in function `mvnrnd`, which generates Gaussian noise based on the diagonal covariance matrix **R**. The covariance matrix is defined as:

$$\mathbf{R} = \begin{bmatrix} \sigma_{Az}^2 & 0 & 0 \\ 0 & \sigma_{El}^2 & 0 \\ 0 & 0 & \sigma_{\text{range}}^2 \end{bmatrix} \tag{1}$$

where σ_{Az}, σ_{El}, and σ_{range} represent the standard deviations of the azimuth, elevation, and range noise components, respectively. These values are determined based on the measurement noise characteristics provided in Table 4.
To enhance computational efficiency, the `measurements` function is only executed within the visibility intervals determined from the visibility window analysis. Additionally, the visibility

condition is re-evaluated after the application of noise. This ensures that any inconsistent or spurious measurements are identified and discarded, preserving the reliability of the simulated data.

2.2.1 Results

The results of the measurement simulation are presented in Figure 6, where the measured azimuth and elevation profiles are compared with the predicted profiles shown in Figure 5. This comparison allows for a detailed examination of the small discrepancies between the simulated and predicted values.

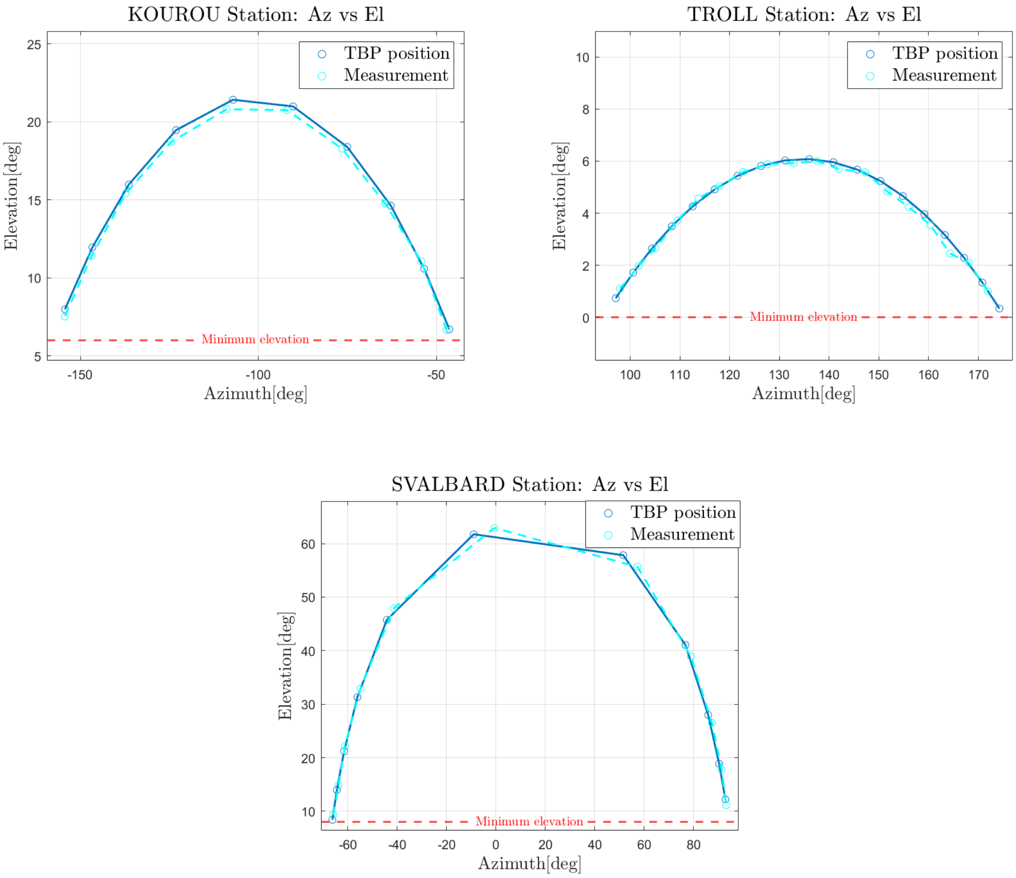

Figure 6: Az-El measurements time evolution profile for different stations within the proper visibility window, i.e. over the mask angle (red dotted line). The reference frame is station centred topocentric

From the results, it is observed that the visibility windows for the measurements align closely with those obtained under the Two-Body Problem (TBP) assumption, with minimal variations. Notably, the visibility window for the TROLL station is slightly reduced by 30 seconds, which corresponds to the loss of a single measurement. Despite this, TROLL station maintains the largest number of measurements, while SVALBARD maintains the largest visibility window across all stations.

2.3 Solution to the Navigation Problem

The navigation problem is addressed by estimating the spacecraft's state using simulated measurements, applying a least squares (minimum variance) method. The solution is computed using MATLAB's `lsqnonlin` function with the following inputs:

- **Cost function**: Defined through the `costfunction` MATLAB® function, which follows these steps: i) The initial guess is propagated to the final time using the appropriate model; ii) The spacecraft's relative position with respect to the selected ground stations is computed in the topocentric reference frame; iii) The relative state is converted into range, azimuth, and elevation; iv) The residuals between simulated and predicted measurements are computed, with `angdiff` used to calculate angular differences, ensuring proper handling of periodicity.

- **Initial guess**: The initial guess is obtained by propagating the state from the reference time t_ref to the initial time t_0, assuming Keplerian dynamics.

- **Weight matrix**: Defined as the inverse of the square root of the covariance matrix, this normalization prioritizes measurements with lower uncertainty (smaller variances) and reduces the weight of those with higher uncertainty.

- **Time span**: The time span is consistent with the measurement intervals.

- **Measurements**: The measurements used are those discussed in subsection 2.2.

- **Options**: The 'Levenberg-Marquardt' algorithm is employed to minimize the residuals between simulated and predicted measurements. This algorithm is well-suited for the navigation problem due to its efficiency in nonlinear least-squares optimization. By combining Gauss-Newton and gradient descent methods, it ensures robust convergence for both simple (TBP) and perturbed (J2 effects) orbital dynamics. The Levenberg-Marquardt algorithm handles nonlinearities and ill-conditioning effectively, providing rapid convergence when the initial guess is sufficiently close to the true solution.

2.3.1 Solution to Navigation Problem: Results

The results for each case is reported in terms of mean state at initial time $\mathbf{S_0}$, related covariance matrix $\mathbf{P_0}$ at precise reference time t_0 and reference frame, norm of the position error err_p, norm of the velocity error err_v, $\lambda_{t,p} = \sqrt{tr(\mathbf{P_0}(1:3,1:3))}$, $\lambda_{t,v} = \sqrt{tr(\mathbf{P_0}(4:6,4:6))}$, the standard deviation of semi-major axis σ_a, and standard deviation of the inclination σ_i. In every case the assumed exact solution is the one deriving from sgp4 propagation, which is reasonably accurate with respect to the real case in the scenario analysed. While the standard deviation of semi-major axis and inclination are obtained as square root of the diagonal elements of the covariance matrix related to semi-major axis and inclination $\mathbf{P_{ai}}$. This is obtained by linearly transforming the covariance matrix associated with the mean state at instant t_0. The transformation is done using the following formula:

$$\mathbf{P_{ai}} = \mathbf{J_{ai}} \mathbf{P_0} \mathbf{J_{ai}}^T \tag{2}$$

Where the $\mathbf{J_{ai}}$ matrix is the jacobian of the transformation from state to the reduce keplerian state, formed by semi-major axis and inclination. This is retrieved by symbolic computations, and it is fully reported in subsubsection A.2.4 in Appendix A.

The results are all referred to time t_0 and expressed in ECI reference frame. The following tables resumes the results:

Parameter	KOUROU (TBP)	All (TBP)	All (PTBP)	Exact (sgp4)
$\mathbf{S_0}$(mean) [km, km/s]	$\begin{bmatrix}+3934.8332\\-1416.9513\\+5781.0875\\+4.878691\\-3.762239\\-4.239432\end{bmatrix}$	$\begin{bmatrix}+3926.9615\\-1411.0890\\+5780.0069\\+4.888482\\-3.764957\\-4.233425\end{bmatrix}$	$\begin{bmatrix}+3932.7735\\-1414.9160\\+5778.5011\\+4.879798\\-3.763210\\-4.232876\end{bmatrix}$	$\begin{bmatrix}+3932.7688\\-1414.9188\\+5778.5030\\+4.8798\\-3.763216\\-4.232871\end{bmatrix}$
err_p [km]	3.8824	7.1172	0.0057	0
err_v [km/s]	0.0067	0.0089	0.000011	0
$\lambda_{t,p}$ [km]	9.9991	1.0711	0.0341	0
$\lambda_{t,v}$ [km/s]	0.0098	0.0011	0.000035	0
σ_a [km]	$1.0715\,10^{1}$	$1.4696\,10^{-1}$	$4.7312\,10^{-3}$	0
σ_i [deg]	$6.2935\,10^{-2}$	$2.0795\,10^{-3}$	$6.7008\,10^{-5}$	0

$\mathbf{P_0}$[km^2, (km/s)^2, km^2/s]	**Case**
$\begin{bmatrix}1.7570\,10^{1} & -7.7117 & 7.0253 & -2.1595\,10^{-2} & -3.3272\,10^{-3} & -1.7769\,10^{-2}\\-7.7117 & 5.9176\,10^{1} & -3.6562\,10^{1} & 2.7116\,10^{-2} & -3.4727\,10^{-2} & 4.1559\,10^{-2}\\7.0253 & -3.6562\,10^{1} & 2.3235\,10^{1} & -1.9120\,10^{-2} & 1.9653\,10^{-2} & -2.7660\,10^{-2}\\-2.1595\,10^{-2} & 2.7116\,10^{-2} & -1.9120\,10^{-2} & 3.2153\,10^{-5} & -7.5849\,10^{-6} & 3.2428\,10^{-5}\\-3.3272\,10^{-3} & -3.4727\,10^{-2} & 1.9653\,10^{-2} & -7.5849\,10^{-6} & 2.5750\,10^{-5} & -1.7893\,10^{-5}\\-1.7769\,10^{-2} & 4.1559\,10^{-2} & -2.7660\,10^{-2} & 3.2428\,10^{-5} & -1.7893\,10^{-5} & 3.8662\,10^{-5}\end{bmatrix}$	**KOUROU (TBP)**
$\begin{bmatrix}6.5463\,10^{-1} & -4.1029\,10^{-1} & -1.1301\,10^{-1} & -6.5404\,10^{-4} & 2.2102\,10^{-4} & -4.0762\,10^{-4}\\-4.1029\,10^{-1} & 4.0672\,10^{-1} & 1.0711\,10^{-1} & 4.6105\,10^{-4} & -1.1661\,10^{-4} & 3.0407\,10^{-4}\\-1.1301\,10^{-1} & 1.0711\,10^{-1} & 8.5842\,10^{-2} & 7.9029\,10^{-5} & -3.1067\,10^{-5} & 1.1049\,10^{-4}\\-6.5404\,10^{-4} & 4.6105\,10^{-4} & 7.9029\,10^{-5} & 7.3961\,10^{-7} & -1.8509\,10^{-7} & 4.1127\,10^{-7}\\2.2102\,10^{-4} & -1.1661\,10^{-4} & -3.1067\,10^{-5} & -1.8509\,10^{-7} & 1.2144\,10^{-7} & -1.3410\,10^{-7}\\-4.0762\,10^{-4} & 3.0407\,10^{-4} & 1.1049\,10^{-4} & 4.1127\,10^{-7} & -1.3104\,10^{-7} & 3.0056\,10^{-7}\end{bmatrix}$	**All (TBP)**
$\begin{bmatrix}6.8087\,10^{-4} & -4.0893\,10^{-4} & -1.0814\,10^{-4} & -6.8643\,10^{-7} & 2.3125\,10^{-7} & -4.2297\,10^{-7}\\-4.0893\,10^{-4} & 3.9500\,10^{-4} & 9.9200\,10^{-5} & 4.6985\,10^{-7} & -1.1607\,10^{-7} & 3.0495\,10^{-7}\\--1.0814\,10^{-4} & 9.9200\,10^{-5} & 8.6891\,10^{-5} & 7.2632\,10^{-8} & -2.9430\,10^{-8} & 1.0965\,10^{-7}\\-6.8643\,10^{-7} & 4.6985\,10^{-7} & 7.2632\,10^{-8} & 7.8798\,10^{-10} & -1.9395\,10^{-10} & 4.3226\,10^{-10}\\2.3125\,10^{-7} & -1.1607\,10^{-7} & -2.9430\,10^{-8} & -1.9395\,10^{-10} & 1.2949\,10^{-10} & -1.4017\,10^{-10}\\-4.2297\,10^{-7} & 3.0495\,10^{-7} & 1.0965\,10^{-7} & 4.3226\,10^{-10} & -1.4017\,10^{-10} & 3.1413\,10^{-10}\end{bmatrix}$	**All (PTBP)**

Table 8: Navigation solutions and related errors for all cases and relative errors (@ t_0 in Earth J2000 frame)

These results show that the most precise solutions are those obtained with J2 perturbed Keplerian dynamics and using the measurements from all the stations. Within the TBP framework, obviously, the more accurate navigation solution processes the measurements from all the stations, while the case with KOUROU only is the least accurate.

To demonstrate this further, the solution is propagated up to $t_{f,\mathrm{prop}}$ = 2024-11-19T08:30:00.000, i.e. for the maximum time interval of TLEs validity, and the residual evolution over time is analyzed. The residual is calculated by subtracting the propagated solution from the reference one (assumed to be derived from sgp4). Figure 7 and Figure 8 illustrate the position and velocity residuals.

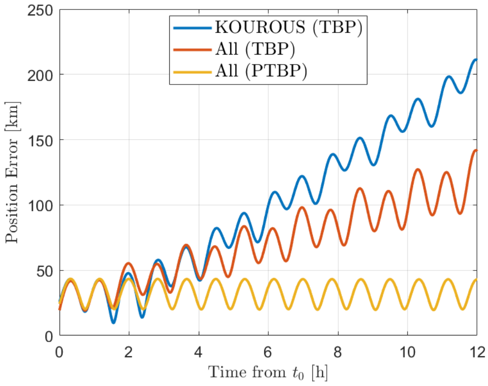

Figure 7: Position error evolution in time in the ECI frame.

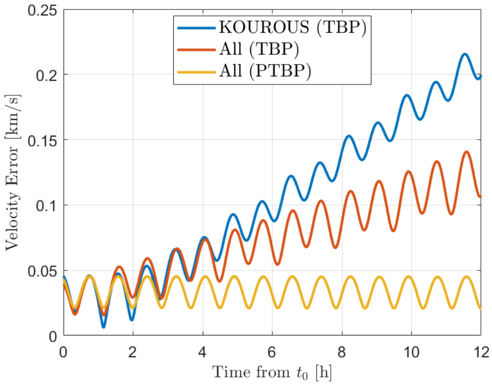

Figure 8: Velocity error evolution in time in the ECI frame.

It is observed that all solutions start with comparable and negligible residuals, but the error grows over time. Notably, the error for the solution based on the TBP model grows to unacceptable levels within 6 hours and continues increasing. In contrast, the J2-perturbed solution, beginning with a more precise estimate, keeps the residuals negligible. This behavior reflects the better physical adherence of the J2-perturbed TBP to the dynamics modeled by `sgp4`, where the J2 effect dominates.

2.4 Trade-off Analysis: Method

A trade-off analysis is performed under a budget constraint of $70,000$ €, where the cost per pass is defined in Table 4. As a result, only two ground stations can be selected. During the analysis period $[t_0, t_f]$, the satellite overpasses each station once, reducing the possible station pairs to [KOUROU+TROLL, KOUROU+SVALBARD, TROLL+SVALBARD]. The navigation solution approach outlined in subsection 2.3 is applied, considering only measurements from the selected stations. The J2-perturbed Keplerian model is used for the orbital dynamics.
The selection criterion is the *generalized variance* Σ, defined as $\Sigma = \det(\mathbf{P_{ai}})$,where $\mathbf{P_{ai}}$ is the covariance matrix of the semi-major axis and inclination. The generalized variance Σ quantifies the "volume" of the uncertainty ellipsoid. A smaller determinant corresponds to a reduced uncertainty region, yielding a more precise solution. For each station pair, a unique value of Σ is computed. The pair yielding the minimum Σ is selected, as it provides the most accurate estimation of the semi-major axis and inclination.

2.4.1 Trade-off Analysis: Results

The results are summarized in Table 9.

Station Pair	σ_a [km]	σ_i [deg]	Σ [-]	Cost [k€]
KOUROU-TROLL	2.7532	$6.0812 \cdot 10^{-3}$	$9.1722 \cdot 10^{-11}$	65
KOUROU-SVALBARD	$1.7530 \cdot 10^{-2}$	$7.2191 \cdot 10^{-4}$	$2.4015 \cdot 10^{-14}$	65
TROLL-SVALBARD	$5.6775 \cdot 10^{-2}$	$6.0746 \cdot 10^{-4}$	$1.5937 \cdot 10^{-13}$	70

Table 9: Visibility Windows for Ground Stations (@t_0 in Earth J2000)

The results indicate that the KOUROU-SVALBARD pair yields the smallest generalized variance Σ, achieving the lowest uncertainty in the semi-major axis, and also the lowest cost. Although the TROLL-SVALBARD pair provides the smallest standard deviation for the inclination, its overall uncertainty volume is larger, and the cost is higher. Therefore, the KOUROU-SVALBARD combination is the optimal choice.

2.5 Long-term analysis

In the context of long-term orbit determination planning, the objective is to ensure both reliable orbit determination and repeatable passes. This analysis assumes a nominal scenario with no strict time or budget constraints for navigation solutions, allowing flexibility in acquiring measurements from multiple ground stations. Key factors in selecting ground stations for long-term operations include orbit coverage, pass repeatability, and measurement reliability. The available ground stations are KOUROU, TROLL, and SVALBARD, each with distinct geographical locations and measurement characteristics, as detailed in Table 4.
Given that the SMOS satellite is almost in a polar orbit (see Table 6), the stations positioned at higher altitudes, TROLL and SVALBARD, are more likely to have longer visibility windows and, consequently, greater measurement opportunities. This observation is supported by propagating the SMOS orbit using J2 perturbed Two-Body dynamics from $t_0 =$ `2024-11-18T20:30:00.000` (UTC) to $t_f =$ `2024-11-21T20:30:00.000` (UTC), covering three days. The results, showing the measurements for the elevation along this period, are presented in Figure 9.

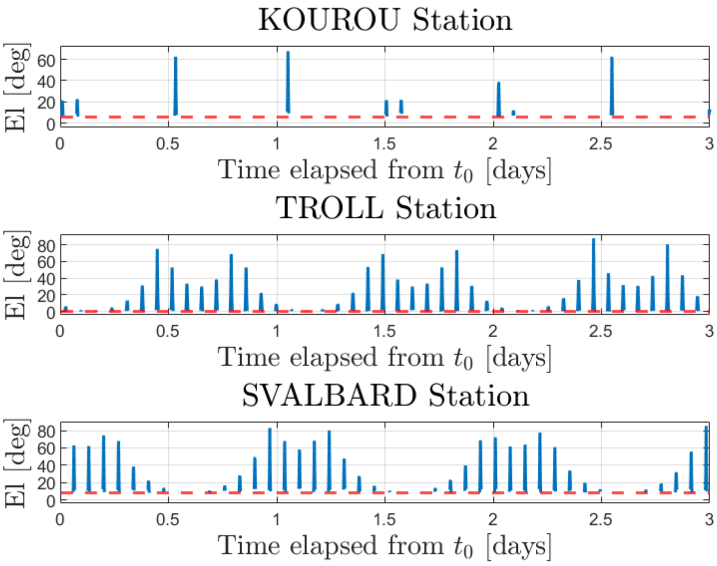

Figure 9: Time evolution of the elevation profile (blue lines) for the ground stations, which is visible, i.e. over the station mask angle (red line).

As shown in the plot, TROLL and SVALBARD maintain the satellite's visibility for longer than KOUROU, confirming the qualitative expectation. This extended visibility allows to fulfill the repeatability driver, a critical factor in selecting prime stations for long-term planning. The extended visibility provided by TROLL and SVALBARD ensures a higher number of measurements, enhancing reliability. Conversely, combinations involving KOUROU yield fewer measurements and lower reliability.
This results are different from the subsection 2.4 ones, where KOUROU-SVALBARD was selected as the best solution. This difference arises from the difference in time span considered. In fact, the trade-off was performed for 1 hour and 45 minutes, while the long-term analysis spans 3 days. During the trade-off analysis SMOS satellite occupies positions such that to allow KOOUROU-SVALBARD pair to perform better measurements. The initial advantage of this pair is also highlighted by the initial time instants of the Figure 9, where a more consistent number of measurements are made. In this limited time span TROLL is not in the perfect relative position for visibility. Obviously, as already analysed in this section, the same is not true for the long-term analysis, in which the TROLL-SVALBARD combinations must be chosen.

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 10, predict the trajectory of the satellite in an inertial Moon-centered reference frame assuming Keplerian motion. Use the estimated coordinates given in Table 11 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 10 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 10. 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 10. For the initial state, you can perturb the actual initial state, composed by $\mathbf{x}_0$ and the latitude

and longitude given in Table 11, 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 10: 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]	`diag([10,1,1,0.001,0.001,0.001,0.00001,0.00001])`

Table 11: Lunar lander - initial guess coordinates and horizon mask

Lander name	MOONLANDER

Coordinates	LAT = 78°
	LON = 15°
	ALT = 0 m

Minimum elevation	0 deg

3.1 Visibility Window Check

The determination of visibility windows is performed through the following steps:

- The lander's latitudinal coordinates, provided in Table 11, are converted into a Cartesian position vector in the IAU_MOON reference frame. This vector is then transformed into the MCI[§] using SPICE routines. The lander's position in the MCI frame is subtracted from the orbiter's propagated trajectory, which is computed by numerically integrating the two-body problem with appropriate tolerances (see subsubsection A.2.1), starting from initial conditions in the MCI frame, as specified in Table 10. The propagation covers the time interval from t_0 to t_f, expressed in Ephemeris Time (ET).
- The relative position vector between the orbiter and the lander in MCI is then rotated into IAU_MOON reference frame (extracting the rotation matrix from SPICE kernels) and then in lander's topocentric reference frame using the rotation matrices computed thanks to SPICE routines.
- The elevation angle, as observed from the lander, is computed from the relative position vector in the topocentric frame. Elevation values exceeding the defined mask angle are considered valid visibility measurements.

Finally, the size of the valid visibility vector is compared to the total time interval. If they match, it is concluded that the lander and orbiter remain in continuous relative visibility throughout the analyzed period.
The function is optimized for computational efficiency in MATLAB®, utilizing vectorized operations to minimize processing time.
The results of this analysis are presented in Figure 10:

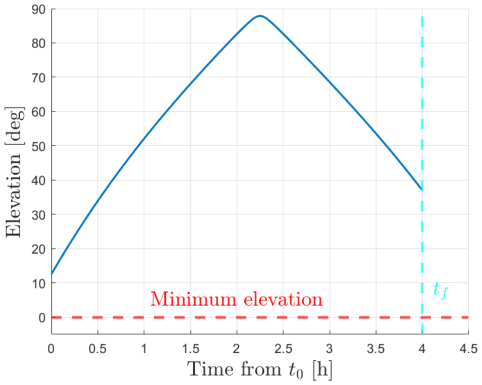

Figure 10: Elevation profile of lander-orbiter

As seen, visibility (positive elevation values) is maintained throughout the entire time interval, from t_0 (marked by the y-axis) to t_f.

3.2 Measurements Simulation

The initial state provided in Table 10 is propagated using Keplerian dynamics with `ode78`, employing position and velocity tolerances of 10^{-10} and 10^{-13}, respectively. The rationale for

[§] In this paper, MCI (Moon-Centered Inertial) refers to an inertial reference frame centered at the Moon, with axes aligned to the J2000 frame these choices is discussed in subsubsection A.1.2. The exact position of the lander is then extracted from the SPICE kernels in the MCI frame. By subtracting the lander and orbiter positions and computing the norm of the resulting vector, the exact range values are obtained. The time evolution of the orbiter's position in MCI and the corresponding range are considered exact solutions.

Starting from these exact solutions, measurements are generated by adding Gaussian noise using MATLAB's built-in `mvnrnd` function. The noise is shaped by the measurement noise matrix $\mathbf{R_k}$, a 4 × 4 diagonal matrix with σ_p^2 on the diagonal and zeros on the off-diagonal elements. To ensure the applied noise level remains within the desired boundaries, the error between the exact solutions and the measurements is computed and graphically verified to stay within the $\pm 3\sigma_p$ limits. The verification is presented in Figure 11:

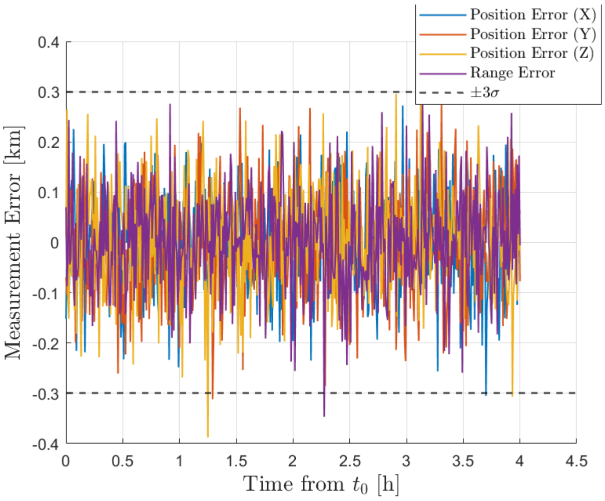

Figure 11: Visual verification that the time evolution of the position and range error remains within the $\pm 3\sigma_p$ boundaries. The errors are expressed in @Moon J2000 reference frame

As shown, the noise level remains within acceptable limits, with only a few outliers, thus validating the measurements for subsequent steps.

3.3 Unscented Kalman Filter for Lunar Orbiter Absolute State Estimation

The lunar orbiter's absolute state is estimated using the Unscented Kalman Filter (UKF), implemented in the MATLAB® function `UKF`. The implementation follows the scheme shown in Figure 12, adapted from [2].

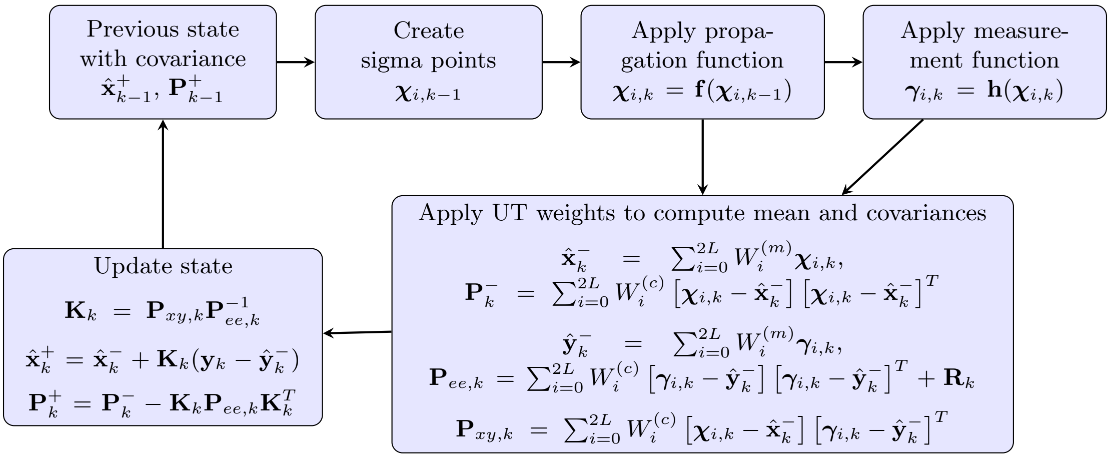
Previous state with covariance $\hat{\mathbf{x}}_{k-1}^+, \mathbf{P}_{k-1}^+$
Create sigma points $\boldsymbol{\chi}_{i,k-1}$
Apply propagation function $\boldsymbol{\chi}_{i,k} = \mathbf{f}(\boldsymbol{\chi}_{i,k-1})$
Apply measurement function $\boldsymbol{\gamma}_{i,k} = \mathbf{h}(\boldsymbol{\chi}_{i,k})$
Apply UT weights to compute mean and covariances
$\hat{\mathbf{x}}_k^- = \sum_{i=0}^{2L} W_i^{(m)} \boldsymbol{\chi}_{i,k},$
$\mathbf{P}_k^- = \sum_{i=0}^{2L} W_i^{(c)} \left[\boldsymbol{\chi}_{i,k} - \hat{\mathbf{x}}_k^-\right] \left[\boldsymbol{\chi}_{i,k} - \hat{\mathbf{x}}_k^-\right]^T$
$\hat{\mathbf{y}}_k^- = \sum_{i=0}^{2L} W_i^{(m)} \boldsymbol{\gamma}_{i,k},$
$\mathbf{P}_{ee,k} = \sum_{i=0}^{2L} W_i^{(c)} \left[\boldsymbol{\gamma}_{i,k} - \hat{\mathbf{y}}_k^-\right] \left[\boldsymbol{\gamma}_{i,k} - \hat{\mathbf{y}}_k^-\right]^T + \mathbf{R}_k$
$\mathbf{P}_{xy,k} = \sum_{i=0}^{2L} W_i^{(c)} \left[\boldsymbol{\chi}_{i,k} - \hat{\mathbf{x}}_k^-\right] \left[\boldsymbol{\gamma}_{i,k} - \hat{\mathbf{y}}_k^-\right]^T$
Update state
$\mathbf{K}_k = \mathbf{P}_{xy,k} \mathbf{P}_{ee,k}^{-1}$
$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{y}_k - \hat{\mathbf{y}}_k^-)$
$\mathbf{P}_k^+ = \mathbf{P}_k^- - \mathbf{K}_k \mathbf{P}_{ee,k} \mathbf{K}_k^T$

Figure 12: Unscented Kalman Filter (UKF) scheme.

The UKF begins by generating sigma points $\boldsymbol{\chi}_{\boldsymbol{i,k-1}}$ based on the prior state estimate $\hat{\mathbf{x}}^{+}_{\mathbf{k-1}}$ and its associated covariance $\mathbf{P}^{+}_{\mathbf{k-1}}$. These sigma points are propagated using Keplerian dynamics, as discussed previously, to produce the predicted sigma points $\hat{\boldsymbol{\chi}}^{+}_{\boldsymbol{k}}$.
Next, the propagated sigma points are transformed into the measurement space via the measurement function $\mathbf{h}$, yielding simulated measurements $\boldsymbol{\gamma}_{\boldsymbol{i,k}}$. In this context, the measurement function simplifies to selecting the position coordinates (i.e., the first three components) of the propagated sigma points. Using these, the following quantities are computed: the predicted state mean $\hat{\mathbf{x}}^{-}_{\mathbf{k}}$ and covariance $\mathbf{P}^{-}_{\mathbf{k}}$, the predicted measurement mean $\hat{\mathbf{y}}^{-}_{\mathbf{k}}$ and covariance $\mathbf{P}_{\mathbf{ee,k}}$, and the cross-covariance between state and measurements $\mathbf{P}_{\mathbf{xy,k}}$. In the definition of $\mathbf{P}_{\mathbf{ee,k}}$, the noise covariance matrix $\mathbf{R}_{\mathbf{k}}$ is derived by extracting the first 3×3 submatrix from subsection 3.2. Finally, the Kalman gain $\mathbf{K}_{\mathbf{k}}$ is computed, followed by the posterior state estimate $\hat{\mathbf{x}}^{+}_{\mathbf{k}}$ and covariance $\mathbf{P}^{+}_{\mathbf{k}}$.
This filter sequentially processes the information in chronological order. This means that the posterior state is fed as prior state of the following operation. At this point, the operations are restarted.
The algorithm is initialized by perturbing the actual initial state $\mathbf{x}_0$ using the MATLAB® function `mvnrnd`, with the specified initial covariance. For tuning the Unscented Transform (UT), the parameters $\alpha = 0.01$, $\beta = 2$, and $\kappa = 0$ are used. The measurements used in this case are only the ones relative to the position vector of the lander.

3.3.1 UKF for orbiter position: results

The output of the UKF function includes the time evolution of the orbiter's state mean and covariance matrix. In Table 12 the results in Moon centered J2000 inertial reference frame at time t_0 and t_f in terms of state are shown. While, in Table 13 the covariance matrix at the final state t_f is reported.

	Time t_0		**Time t_f**	
State Component	**Estimated**	**True**	**Estimated**	**True**
x [km]	4307.3194	4307.8442	-3479.0255	-3479.0233
y [km]	-1316.8876	-1317.9807	-2506.1834	-2506.1869
z [km]	2108.6003	2109.2101	4010.7491	4010.7374
v_x [km/s]	-0.107632	-0.110997	-0.450123	0.450123
v_y [km/s]	-0.513435	-0.509393	0.348542	0.348542
v_z [km/s]	0.789488	0.815199	-0.557782	-0.557783

Table 12: Estimated State at time t_0 and t_f (@Moon J2000 inertial frame).

$\mathbf{P}(t_f)$ $[km^2,\ km^2/s^2,\ km^2/s]$

6.7883e-05	7.4605e-06	-1.1947e-05	5.7759e-09	3.5504e-09	-5.6824e-09
7.4605e-06	7.9669e-05	-2.5324e-05	4.3644e-09	8.3985e-09	-8.6726e-09
-1.1947e-05	-2.5324e-05	1.0439e-04	-6.9921e-09	-8.6751e-09	1.6864e-08
5.7759e-09	4.3644e-09	-6.9921e-09	1.4581e-12	9.6744e-13	-1.5488e-12
3.5504e-09	8.3985e-09	-8.6751e-09	9.6744e-13	1.8223e-12	-1.6546e-12
-5.6824e-09	-8.6726e-09	1.6864e-08	-1.5488e-12	-1.6546e-12	3.4368e-12

Table 13: Estimated covariance matrix at time t_f (@Moon J2000 inertial frame).

The orbiter's state mean and covariance matrix are used to compute the norm of the error relative to the exact solution, as defined in subsection 3.2, where the exact solution corresponds to the Keplerian dynamics propagation of the initial state in the MCI reference frame.
To visually assess whether the errors remain within acceptable thresholds, the $3\sigma_{pc}$ and $3\sigma_{vc}$ bounds of the position and velocity covariance submatrices are computed. Specifically, for each

time step, the following expressions are used:

$$\sigma_{pc} = \sqrt{\mathrm{tr}(\mathbf{P}_\mathbf{k}^+(1:3,1:3))}; \quad \sigma_{vc} = \sqrt{\mathrm{tr}(\mathbf{P}_\mathbf{k}^+(4:6,4:6))}$$

In this case, the trace of the covariance matrix is used, instead of the maximum eigenvalue, to have an indication of the overall uncertainty, rather than the maximum uncertainty along one particular direction. The results of these calculations are shown in Figure 13:

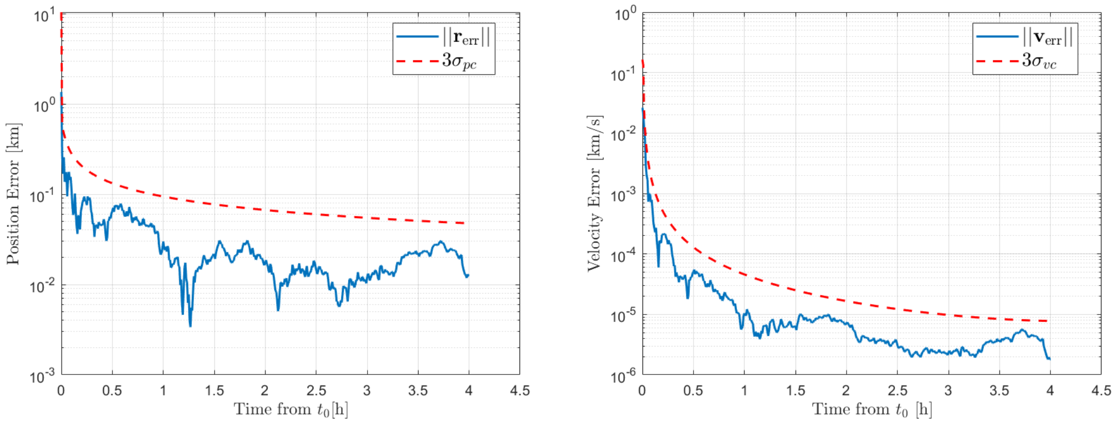

Figure 13: Time evolution of position and velocity error estimates, with 3σ covariance bounds. Quantities expressed @Moon J2000 inertial frame

The errors are presented in logarithmic scale to enhance visibility. As observed, both position and velocity errors exhibit a generally decreasing trend. This confirms the efficacy of the implemented filter in refining the lunar orbiter's absolute state. Additionally, the decreasing trend of the $3\sigma_{pc}$ and $3\sigma_{vc}$ bounds indicates that the covariance matrix trace is also decreasing, suggesting convergence toward the assumed true state.

3.4 Unscented Kalman Filter for Lunar Lander Coordinates

The implementation of the Unscented Kalman Filter (UKF) for determining the lunar lander coordinates follows the same structure as described in subsection 3.3, with key modifications. The extended state vector now includes the lunar orbiter state vector and the lander's latitude and longitude. The measurement model incorporates the orbiter's position vector and the relative range between the orbiter and the lander. Thus, the noise covariance matrix $\mathbf{R_k}$ is 4 in this case.

Given that the lander remains at a fixed latitude, longitude, and altitude during the propagation time span, the sigma points corresponding to the latitude and longitude remain unchanged. Additionally, the measurement function is updated to include the relative range, which is computed from the propagated sigma points of the orbiter's position and the lander's position vector. The lander position vector is derived assuming zero altitude, using the fixed latitude and longitude sigma points. Apart from these adjustments, the UKF logic remains unaltered.

3.4.1 Results: UKF for Lunar Lander Coordinates

The errors for the position and velocity components, as well as the relative range, are computed consistently with the methodology outlined in subsubsection 3.3.1. The errors on latitude and longitude are determined by comparing the lander's position extracted from SPICE kernels and converted to geodetic coordinates using SPICE routines.

In Table 14 the results in Moon centered J2000 inertial reference frame at time t_0 and t_f in terms of state, longitude and latitude are shown. While, in Table 15 the covariance matrix at the final state t_f is reported.

	Time t_0		Time t_f	
State Component	**Estimated**	**True**	**Estimated**	**True**
x [km]	4306.0194	4307.8442	-3479.0278	-3479.0233
y [km]	-1317.8622	-1317.9807	-2506.1832	-2506.1869
z [km]	2108.2999	2109.2101	4010.7489	4010.7374
v_x [km/s]	-0.072131	-0.110997	-0.450124	0.450123
v_y [km/s]	-0.528345	-0.509393	0.348542	0.348542
v_z [km/s]	0.807329	0.815199	-0.557782	-0.557783
Lat [rad]	1.364449	1.365367	1.365430	1.365367
Lon [rad]	0.264890	0.268917	0.269012	0.268917

Table 14: Estimated state, including longitude and latitude, at times t_0 and t_f (@Moon J2000 inertial frame).

$\mathbf{P}(t_f)$ $[km^2,\ km^2/s^2,\ rad^2,\ km^2/s,\ rad\,km,\ rad\,km/s]$							
5.7788e-05	6.2527e-06	-1.0062e-05	4.4743e-09	2.9849e-09	-4.7741e-09	-2.5118e-07	-2.6708e-07
6.2527e-06	7.9470e-05	-2.5007e-05	4.2040e-09	8.3207e-09	-8.5471e-09	-3.0392e-07	-2.7722e-07
-1.0062e-05	-2.5007e-05	1.0389e-04	-6.7380e-09	-8.5518e-09	1.6665e-08	4.9145e-07	4.4672e-07
4.4743e-09	4.2040e-09	-6.7380e-09	1.2872e-12	8.9318e-13	-1.4289e-12	-6.1404e-11	-6.9408e-11
2.9849e-09	8.3207e-09	-8.5518e-09	8.9318e-13	1.7886e-12	-1.6003e-12	-5.7303e-11	-5.0681e-11
-4.7741e-09	-8.5471e-09	1.6665e-08	-1.4289e-12	-1.6003e-12	3.3491e-12	9.1103e-11	8.0104e-11
-2.5118e-07	-3.0392e-07	4.9145e-07	-6.1404e-11	-5.7303e-11	9.1103e-11	1.0735e-08	1.1984e-08
-2.6708e-07	-2.7722e-07	4.4672e-07	-6.9408e-11	-5.0681e-11	8.0104e-11	1.1984e-08	1.4337e-08

Table 15: Estimated (extended) covariance matrix at time t_f (@Moon J2000 inertial frame).

The uncertainties on latitude and longitude are evaluated as $3\sigma_{\text{lat}} = 3\sqrt{P_{\text{lat}}}$ and $3\sigma_{\text{lon}} = 3\sqrt{P_{\text{lon}}}$, respectively, where P_{lat} and P_{lon} are the diagonal elements of the covariance matrix corresponding to latitude and longitude, respectively. The graphical results are presented in Figure 14.

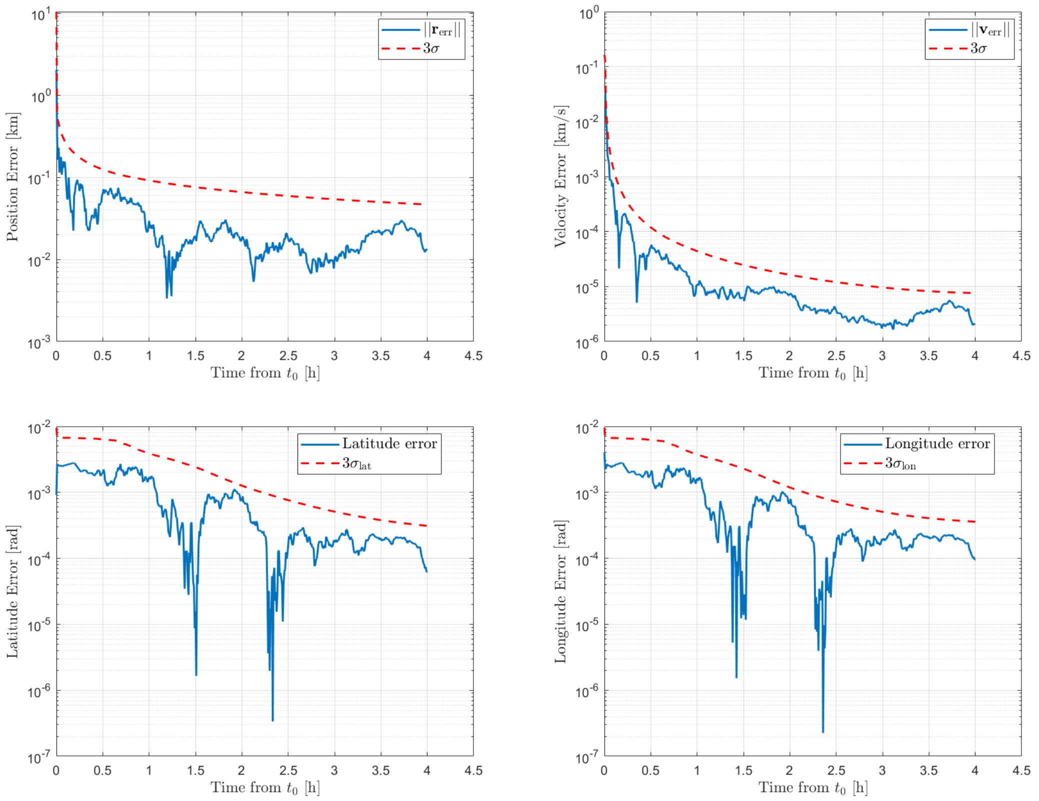

Figure 14: Time evolution of position, velocity, latitude and longitude error estimates, with proper 3σ covariance bounds. The errors are expressed in @Moon J2000 inertial frame

The errors are presented on a logarithmic scale. For all analyzed quantities, the errors remain well within the acceptable boundaries. The position and velocity errors exhibit trends similar to the previous case, while the latitude and longitude errors converge to negligible values over time. It must be noted that the error on latitude and longitude already starts from low values, and their reduction is significantly smaller than the one for the position and velocity. This can be attributed to the fact that longitude and latitude of the lunar lander are constant, while the position and velocity of the lunar orbiter are time-varying.
As a last remark, the achieved errors are negligible, thus, the mission of improving the estimate of lander latitude and longitude can be considered accomplished. The more refined estimate of those latitudinal coordinates (considering a fixed zero altitude) are:

$$\text{LAT} = 78.2334^\circ, \quad \text{LON} = 15.4133^\circ, \quad \text{ALT} = 0\,\text{m}.$$

Exercise A: Appendix

A.1 Appendix: Exercise 1

A.1.1 2D PBRFBP dynamics in rotating frame

This paragraph deepens the two-dimensional Planar Bicircular Restricted Four Body Problem (2D PBRFBP) in the rotating frame for the problem exposed in section 1.
The 2D equations of motion for the negligible mass in the rotating frame are:

$$\ddot{x} = \frac{\partial \Omega_4}{\partial x} + 2\dot{y}, \quad \ddot{y} = \frac{\partial \Omega_4}{\partial y} - 2\dot{x},$$

where x and y denote the particle's coordinates in the rotating frame. The effective potential of the four-body system, $\Omega_4(x, y, t)$, is defined as:

$$\Omega_4(x, y, t) = \Omega_3(x, y) + \frac{m_s}{r_3(t)} - \frac{m_s}{\rho^2}\left(x\cos(\omega_s t) + y\sin(\omega_s t)\right), \tag{3}$$

where Ω_3 represents the potential of the CR3BP.
In this model, t is the time, m_s is the Sun's mass, ρ is the distance between the Sun and the barycenter of the Earth–Moon system, and ω_s is the Sun's angular velocity in the rotating reference frame. The Sun's phase angle is given by $\theta(t) = \omega_s t$, and its location at time t is $(\rho\cos(\omega_s t), \rho\sin(\omega_s t))$. Using this position, the distance between the Sun and the particle is:

$$r_3(t) = \sqrt{(x - \rho\cos(\omega_s t))^2 + (y - \rho\sin(\omega_s t))^2}. \tag{4}$$

It is important to highlight that in this framework, the Sun is not stationary within the chosen reference frame. As a result, the equations of motion depend explicitly on time, rendering the Planar Bicircular Restricted Four-Body Problem (PBRFBP) a non-autonomous system. A table summarizing the system's constants and their units (e.g., distance, time, and velocity) can be found in [3].
The full treatment is in [3]. Here, only the final expression of the dynamics of the 2D PBRFBP is reported:

$$\begin{cases} \dot{x} = v_x \\ \dot{y} = v_y \\ \dot{v_x} = \frac{\partial \Omega_4}{\partial x} + 2v_y \\ \dot{v_y} = \frac{\partial \Omega_4}{\partial y} - 2v_x \end{cases} \quad \text{with initial conditions:} \quad \begin{cases} x(t_0) = x_0 \\ y(t_0) = y_0 \\ v_x(t_0) = v_{x0} \\ v_y(t_0) = v_{y0} \end{cases} \tag{5}$$

Rewriting the system in matrix form:

$$\dot{\mathbf{S}} = \left[v_x, v_y, \frac{\partial \Omega_4(x, y, t)}{\partial x} + 2v_y, \frac{\partial \Omega_4(x, y, t)}{\partial y} - 2v_x\right]^T = \mathbf{f_4}(\mathbf{S}, t), \tag{6}$$

Where $\mathbf{S} = [x, y, v_x, v_y]^T$ is the state vector, and $\mathbf{f_4}(\mathbf{S}, t)$ is the vector of the dynamics of the 2D PBRFBP.
The STM from initial time t_0 to final time t can be found by solving the following 16 scalar ODEs:

$$\begin{cases} \text{Variational equation :} & \dot{\mathbf{\Phi}}(t_0, t) = \mathbf{A_4}(t)\mathbf{\Phi}(t_0, t) \\ \text{Initial conditions :} & \mathbf{\Phi}(t_0, t_0) = \mathbf{I} \end{cases} \tag{7}$$

Where $\mathbf{I} \in \Re^{4\times 4}$ is the identity matrix, and the matrix $\mathbf{A}(t)$ can be retrieved symbolically by computing the Jacobian matrix of the dynamics over the state, that is $\mathbf{A}(t) = \frac{d\mathbf{f}}{d\mathbf{S}}$. The result is reported here:

$$\mathbf{A_4}(t) = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \Omega_{4,xx} & \Omega_{4,xy} & 0 & 2 \\ \Omega_{4,yx} & \Omega_{4,yy} & -2 & 0 \end{bmatrix},$$

where:

$$\Omega_{4,xx} = \frac{\mu-1}{r_1^3} - \frac{m_s}{r_3(t)^3} - \frac{\mu}{r_2^3} - \frac{3(\mu+x)^2(\mu-1)}{r_1^5} + \frac{3\mu(\mu+x-1)^2}{r_2^5} + \frac{3m_s(x-\rho\cos(\omega_s t))^2}{r_3(t)^5} + 1;$$

$$\Omega_{4,xy} = \frac{3\mu y(\mu+x-1)}{r_2^5} + \frac{3m_s(x-\rho\cos(\omega_s t))(y-\rho\sin(\omega_s t))}{r_3(t)^5} - \frac{3y(\mu+x)(\mu-1)}{r_1^5} = \Omega_{4,yx};$$

$$\Omega_{4,yy} = \frac{\mu-1}{r_1^3} - \frac{m_s}{r_3(t)^3} - \frac{\mu}{r_2^3} - \frac{3y^2(\mu-1)}{r_1^5} + \frac{3m_s(y-\rho\sin(\omega_s t))^2}{r_3(t)^5} + \frac{3\mu y^2}{r_2^5} + 1.$$

Where $r_1 = \sqrt{(x+\mu)^2+y^2}$ and $r_2 = \sqrt{(x+\mu-1)^2+y^2}$.

A.1.2 `ode78` and relative tolerances

The integrator employed for this task is `ode78`, a high-order solver with fixed step sizes. This solver is particularly well-suited for long-duration integrations due to its robustness, stability, and numerical reliability in scenarios characterized by smooth and predictable dynamics. Designed for non-stiff problems, `ode78` is ideal for applications such as spacecraft trajectory guidance, where the dynamics are not subject to rapid or abrupt changes. The choice of a fixed-step algorithm ensures that the convergence behavior is solely attributed to the optimization methods under investigation, not enhanced by the ode solver used. This choice, thereby, isolates and validates the inherent properties of the algorithms being analyzed. The integration is performed with specific absolute tolerances for different quantities to balance computational precision and efficiency. For the position $\mathbf{r}$, the tolerance is set to $\text{Tol}_\mathbf{r} = 10^{-10}$, providing sufficient accuracy without imposing excessive computational demands. The velocity $\mathbf{v}$ requires higher precision due to its direct influence on the dynamics, and thus its tolerance is set to $\text{Tol}_\mathbf{v} = 10^{-13}$. For the components of the state transition matrix (STM) $\boldsymbol{\Phi}$, the tolerance is chosen as $\text{Tol}_{\boldsymbol{\Phi}} = 10^{-13}$. While the STM is crucial for understanding the trajectory's evolution, extreme precision is deemed unnecessary because the perturbations it experiences over time are relatively small. These tolerances are carefully selected to ensure computational resources are allocated effectively and efficiently.

A.1.3 Linearized Covariance Method

The LinCov method approximates the propagation of uncertainty in nonlinear dynamics by linearizing the system around the nominal trajectory. Let $\hat{\mathbf{x}}_i$ represent the mean of the initial state at time t_i, with an associated covariance matrix $\mathbf{P}_0$, and let $\mathbf{x}(t) = \varphi(\hat{\mathbf{x}}_i, t_i; t)$ denote the nominal solution of the 2D PBRFBP at time t. In this framework, the evolution of the state mean $\hat{\mathbf{x}}(t)$ and the covariance matrix $\mathbf{P}(t)$ is given by:

$$\hat{\mathbf{x}}(t) = \mathbf{x}(t), \qquad \mathbf{P}(t) = \boldsymbol{\Phi}(t,t_i)\mathbf{P}_0\boldsymbol{\Phi}^T(t,t_i), \tag{8}$$

where $\boldsymbol{\Phi}(t,t_i)$ is the State Transition Matrix (STM), obtained by integrating the variational equations of the system dynamics as detailed in subsubsection A.1.1.

A.1.4 Unscented Transform Method

The Unscented Transform (UT) method provides a higher-fidelity approximation of uncertainty propagation through nonlinear dynamics compared to linearization-based methods. It achieves this by deterministically sampling a set of sigma points around the initial state mean and propagating them through the nonlinear system. These sigma points are designed to capture the first two statistical moments (mean and covariance) of the state distribution, ensuring an accurate representation of the uncertainty. The UT analyzed here is the Julier-Uhlmann formulation.

Let $\hat{\mathbf{x}}_i$ represent the mean of the initial state at time t_i, with an associated covariance matrix $\mathbf{P}_0$, and let $\mathbf{x}(t) = \varphi(\hat{\mathbf{x}}_i, t_i; t)$ denote the nominal solution of the 2D PBRFBP at time t, where φ is the nonlinear flow function linking the initial state to the propagated one. For an n-dimensional state, $2n+1$ sigma points are generated at time t_i as:

$$\chi_0 = \hat{\mathbf{x}}_i,$$

$$\chi_j = \hat{\mathbf{x}}_i + \sqrt{(n+\lambda)\mathbf{P}_0}, \quad j = 1, \ldots, n,$$

$$\chi_{j+n} = \hat{\mathbf{x}}_i - \sqrt{(n+\lambda)\mathbf{P}_0}, \quad j = 1, \ldots, n,$$

where $\lambda = \alpha^2(n+\kappa) - n$ is a scaling parameter. The parameters α and κ are user-defined constants that influence the spread and location of the sigma points, with typical values of $\alpha \simeq 10^{-3}$ and $\kappa = 0$. The matrix square root of $\mathbf{P}_i$ is usually computed using a Cholesky decomposition.

The weights for the sigma points are computed as:

$$W_0^{(m)} = \frac{\lambda}{n+\lambda}, \quad W_0^{(c)} = \frac{\lambda}{n+\lambda} + (1 - \alpha^2 + \beta),$$

$$W_j^{(m)} = W_j^{(c)} = \frac{1}{2(n+\lambda)}, \quad j = 1, \ldots, 2n,$$

where $W_j^{(m)}$ and $W_j^{(c)}$ are the weights for the mean and covariance, respectively. The parameter β is used to account for prior knowledge about the distribution, with a typical value of $\beta = 2$ for Gaussian distributions.

The sigma points are propagated through the nonlinear dynamics, represented here by the 2D PBRFBP (subsubsection A.1.1), resulting in their transformed counterparts:

$$\mathbf{\Upsilon}_j(t) = \varphi(\chi_j, t_i; t), \quad j = 0, \ldots, 2n.$$

The weighted sample mean $\hat{\mathbf{y}}$ and covariance $\mathbf{P}_y$ of the propagated state are then calculated as:

$$\hat{\mathbf{y}} = \sum_{j=0}^{2n} W_j^{(m)} \mathbf{\Upsilon}_j, \tag{9}$$

$$\mathbf{P}_y(t) = \sum_{j=0}^{2n} W_j^{(c)} \left(\mathbf{\Upsilon}_j - \hat{\mathbf{y}}\right) \left(\mathbf{\Upsilon}_j - \hat{\mathbf{y}}\right)^\top. \tag{10}$$

A.1.5 Uncertainty Ellipse Representation

To represent the uncertainty in a 2D Gaussian distribution, an uncertainty ellipse was plotted using a MATLAB® function, which is taken from [1]. The ellipse is centred at the mean and defined by the covariance matrix, scaled by a specified number of standard deviations, in this case 3σ was chosen. The methodology for plotting the ellipse is as follows:

1. **Eigenvalue Decomposition**: The covariance matrix is decomposed into its eigenvalues and eigenvectors using the MATLAB® built-in function `eig`. The eigenvectors define the ellipse's orientation, while the eigenvalues correspond to the squared lengths of the semi-major and semi-minor axes.

2. **Sorting of Eigenvalues and Eigenvectors**: The eigenvalues are sorted in descending order, and the corresponding eigenvectors are reordered to match. This ensures the proper alignment of the ellipse's principal axes.

3. **Ellipse Construction**: It followed the subsequent procedure:

   - A unit circle was generated in the parametric form $(\cos\theta, \sin\theta)$ with $\theta \in [0, 2\pi]$.
   - The unit circle was scaled by the square root of the eigenvalues and rotated by the eigenvectors, effectively transforming it into the desired ellipse.
   - The ellipse was further scaled by the desired number of standard deviations, σ, and translated to the mean.

This approach provides a scientifically rigorous visualization of the uncertainty, capturing both the orientation and the spread of the Gaussian distribution in the state space.

A.1.6 Monte Carlo Simulation for Uncertainty Propagation

In this context, the MC method is used to propagate the initial state and its associated covariance matrix through a nonlinear system over a sequence of time steps. For each time step, a large number of state samples are generated, perturbing the initial state according to the specified covariance matrix, and then propagated through the system. The resulting states are then used to compute the mean and covariance, which provide a statistical representation of the system's uncertainty.
Given an initial state vector $\mathbf{x}_i$ of dimension n, with associated covariance matrix $\mathbf{P}_0$, and a time vector $\mathbf{t} = [t_1, t_2, \ldots, t_m]$, the Monte Carlo simulation proceeds as follows:

1. **State Perturbation**: At each time step t_j, n_{samples} independent realizations of the initial state are generated by perturbing $\mathbf{S}_0$ according to a multivariate normal distribution with mean $\mathbf{S}_0$ and covariance $\mathbf{P}_0$. Specifically, each perturbed state $\mathbf{S}_{\text{perturbed}}$ is sampled from the distribution:

$$\mathbf{S}_{\text{perturbed}} \sim \mathcal{N}(\mathbf{S}_0, \mathbf{P}_0),$$

   where $\mathcal{N}(\mathbf{S}_0, \mathbf{P}_0)$ denotes the multivariate normal distribution with mean $\mathbf{S}_0$ and covariance $\mathbf{P}_0$, obtained thanks to the MATLAB function `mvnrnd`.

2. **State Propagation**: Each of the perturbed states is propagated through the nonlinear PBRFBP dynamics, already deepened in subsubsection A.1.1, over the time interval $[t_0, t_j]$, where t_0 is the initial time, and t_j is one of the elements of the vector $\mathbf{t}$, created as specified in the assignments.

3. **Statistical Computation**: Once all samples have been propagated to the current time step t_j, the mean state $\mathbf{E}_j$ and covariance matrix $\mathbf{P}_j$ at that time step are computed as the sample mean and sample covariance of the propagated states using the MATLAB® functions `mean` and `cov`.

4. **Final State Sampling**: At the final time step t_m, the final states $\mathbf{S}_{\text{f},i}(t_m)$ of all n_{samples} realizations are stored for further analysis. These sampled states are used to assess the uncertainty in the system's final state, providing insight into the possible variability and range of outcomes.

A.2 Appendix: Exercise 2

A.2.1 Two-Body Problem (TBP function)

The TBP function computes the time derivative of the state vector for a body governed solely by the gravitational attraction of a central body, following the classical Two-Body Problem (TBP) dynamics. The governing equations, and the subsequent matricial form, are expressed as:

$$\begin{cases} \frac{\mathrm{d}\mathbf{r}}{\mathrm{d}t} &= \mathbf{v} \\ \frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t} &= -\mu \frac{\mathbf{r}}{r^3} \end{cases} \qquad \Rightarrow \qquad \dot{\mathbf{S}} = \begin{bmatrix} \mathbf{v} \\ -\mu \frac{\mathbf{r}}{r^3} \end{bmatrix}$$

where $\mathbf{r} = [x, y, z]^T$ is the position vector, $\mathbf{v} = [v_x, v_y, v_z]^T$ is the velocity vector, $r = \|\mathbf{r}\|$ is the radial distance, and μ is the gravitational parameter, and $\dot{\mathbf{S}}$ is the vector describing the state. This function is autonomous, and the time variable is not explicitly required in the calculations. It must be noted that these equations are valid in an inertial reference frame, so this must be accounted for in case every other contribution is added.

A.2.2 J2-Perturbed Two-Body Problem (J2_TBP function)

The J2_TBP function extends the classical TBP dynamics, developed in subsubsection A.2.1 by including the J2 perturbation, which accounts for the oblateness of the central body. The acceleration due to the J2 perturbation is modeled in ECEF reference frame[¶] as:

$$\mathbf{a}_{J2,ECEF} = \frac{3}{2}\mu\, J_2 \left(\frac{R_e^2}{r^4}\right) \mathbf{r} \left(5\left(\frac{z}{r}\right)^2 - \begin{bmatrix}1\\1\\3\end{bmatrix}\right),$$

where $J_2 = 0.0010826269$ is the oblateness coefficient, R_e is the equatorial radius of the Earth, $\mathbf{r} = [x, y, z]^T$ is the position vector expressed in ECEF, and $r = ||\mathbf{r}||$ is the Euclidean norm of the position vector. The ECEF acceleration is transformed back to the Earth-centered inertial (ECI) frame, so that its effects can be properly summed to the one coming from the gravitational attraction of the main body, and then integrated. The transformation matrices are computed thanks to `cspice_pxform` function.

A.2.3 Cost function

The residuals in the cost function are computed as the weighted differences between predicted and real measurements (range, azimuth, and elevation) obtained from ground stations. The methodology involves the following steps:

1. **Numerical Propagation:** The spacecraft state is propagated from the reference epoch to the time spans of visibility using either the Two-Body Problem (TBP subsubsection A.2.1) or the J2-perturbed Two-Body Problem (J2-TBPsubsubsection A.2.2). The propagation is performed using a high-accuracy numerical integrator with stringent relative and absolute tolerances. The choice of the propagator and the relative tolerance are already treated in subsubsection A.1.2;

2. **Station Positions in ECI:** The positions and velocities of the ground stations in the Earth-Centered Inertial (ECI) frame are computed using SPICE routines. These data ensure accurate alignment with the spacecraft state. Then the relative position between the spacecraft and the ground station is computed;

[¶] The Earth-centered, Earth-fixed (ECEF) frame is a non-inertial frame. The x-axis points towards the North pole, and the z-axis intersects the equator and the prime meridian. The ECEF rotates with the Earth, so the axis always points through the equator and the prime meridian.

3. **Transformation to Topocentric Frame:** The relative position of the spacecraft with respect to each ground station is transformed from the ECI frame to a topocentric frame centred at the ground station. The transformation employs SPICE-provided state transformation matrices;

4. **Predicted Measurements:** In the topocentric frame, the predicted measurements in terms of range, azimuth and elevation are computed;

5. **Residuals:** The residuals are calculated by subtracting the simulated measurements from the predicted measurements. To account for the angular periodicity, the differences in azimuth and elevation are computed using the MATLAB® built-in function `angdiff`. The residuals are multiplied by a weighting matrix W_m to incorporate measurement uncertainties and emphasize the reliability of specific measurement types;

6. **Post-processing:** Any non-numerical values resulting from null measurements are discarded. This ensures a clean input to the optimization algorithm.

This methodology ensures consistency between predicted and simulated measurements, while the inclusion of a weighting matrix allows the least squares solution to account for measurement confidence effectively.

A.2.4 Jacobian matrix from cartesian state to reduced keplerian state

The jacobian matrix from cartesian state $\mathbf{S} = [x, y, z, v_x, v_y, v_z]^T$ to reduced keplerian state $\mathbf{k_r} = [a, i]^T$, is computed symbolically. The final result is reported here:
The Jacobian matrix for the Keplerian elements (a, i) is given by:

$$\mathrm{Jac}_{ai} = \begin{bmatrix} J_{11} & J_{12} & J_{13} & J_{14} & J_{15} & J_{16} \\ J_{21} & J_{22} & J_{23} & J_{24} & J_{25} & J_{26} \end{bmatrix}$$

Where the individual elements are defined as:

$$J_{11} = \frac{2x}{\left(\frac{v_x^2+v_y^2+v_z^2}{\mu_E} - \frac{2}{\sqrt{x^2+y^2+z^2}}\right)^2 (x^2+y^2+z^2)^{3/2}}, \quad J_{12} = \frac{2y}{\left(\frac{v_x^2+v_y^2+v_z^2}{\mu_E} - \frac{2}{\sqrt{x^2+y^2+z^2}}\right)^2 (x^2+y^2+z^2)^{3/2}},$$

$$J_{13} = \frac{2z}{\left(\frac{v_x^2+v_y^2+v_z^2}{\mu_E} - \frac{2}{\sqrt{x^2+y^2+z^2}}\right)^2 (x^2+y^2+z^2)^{3/2}}, \quad J_{14} = \frac{2v_x}{\mu_E\left(\frac{v_x^2+v_y^2+v_z^2}{\mu_E} - \frac{2}{\sqrt{x^2+y^2+z^2}}\right)^2},$$

$$J_{15} = \frac{2v_y}{\mu_E\left(\frac{v_x^2+v_y^2+v_z^2}{\mu_E} - \frac{2}{\sqrt{x^2+y^2+z^2}}\right)^2}, \quad J_{16} = \frac{2v_z}{\mu_E\left(\frac{v_x^2+v_y^2+v_z^2}{\mu_E} - \frac{2}{\sqrt{x^2+y^2+z^2}}\right)^2},$$

$$J_{21} = \frac{-(v_y y^2 + v_x xy + v_y z^2)}{\sqrt{1 - \frac{(v_y x - v_x y)^2}{(v_x^2+v_y^2+v_z^2)(x^2+y^2+z^2)}}} \cdot \left(v_x^2+v_y^2+v_z^2\right)^{1/2} \left(x^2+y^2+z^2\right)^{3/2},$$

$$J_{22} = \frac{v_x x^2 + v_y yx + v_x z^2}{\sqrt{1 - \frac{(v_y x - v_x y)^2}{(v_x^2+v_y^2+v_z^2)(x^2+y^2+z^2)}}} \cdot \left(v_x^2+v_y^2+v_z^2\right)^{1/2} \left(x^2+y^2+z^2\right)^{3/2},$$

$$J_{23} = \frac{z(v_y x - v_x y)}{\sqrt{1 - \frac{(v_y x - v_x y)^2}{(v_x^2+v_y^2+v_z^2)(x^2+y^2+z^2)}}} \cdot \left(v_x^2+v_y^2+v_z^2\right)^{1/2} \left(x^2+y^2+z^2\right)^{3/2},$$

$$J_{24} = \frac{y v_y^2 + v_x x v_y + y v_z^2}{\sqrt{1 - \frac{(v_y x - v_x y)^2}{(v_x^2+v_y^2+v_z^2)(x^2+y^2+z^2)}}} \cdot \left(v_x^2 + v_y^2 + v_z^2\right)^{3/2} \left(x^2 + y^2 + z^2\right)^{1/2},$$

$$J_{25} = \frac{-(x v_x^2 + v_y y v_x + x v_z^2)}{\sqrt{1 - \frac{(v_y x - v_x y)^2}{(v_x^2+v_y^2+v_z^2)(x^2+y^2+z^2)}}} \cdot \left(v_x^2 + v_y^2 + v_z^2\right)^{3/2} \left(x^2 + y^2 + z^2\right)^{1/2},$$

$$J_{26} = \frac{v_z (v_y x - v_x y)}{\sqrt{1 - \frac{(v_y x - v_x y)^2}{(v_x^2+v_y^2+v_z^2)(x^2+y^2+z^2)}}} \cdot \left(v_x^2 + v_y^2 + v_z^2\right)^{3/2} \left(x^2 + y^2 + z^2\right)^{1/2}.$$

References

[1] R. A. (A. Johnson and D. W. Wichern. *Applied multivariate statistical analysis*. eng. 6th ed. Upper Saddle River, N.J: Pearson Prentice Hall, 2007 (cit. on pp. 2, 26).

[2] N. J. Kasdin. "Optimal estimation of dynamical systems, John L. Crassidis and John L. Junkins, by Chapman and Hall/CRC, London, Boca Raton, 2004, ISBN 1-58488-391-X". In: *International Journal of Robust and Nonlinear Control* 16.7 (Jan. 2006), pp. 369–371. DOI: `10.1002/rnc.1050` (cit. on p. 19).

[3] F. Topputo. "On optimal two-impulse Earth-Moon transfers in a four-body model". In: *Celestial Mechanics and Dynamical Astronomy* 117 (2013), pp. 279–313. DOI: `10.1007/s10569-013-9513-8` (cit. on pp. 2, 24).