SGN – Assignment #2

Frassinella Luca, 244643

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 Uncertainty Propagation using LinCov and UT

In this first point, the initial estimates of the mean state vector, $\mathbf{x_i}$, and the covariance matrix, $\mathbf{P_0}$, at the initial time t_i have been propagated up to the final time t_f over a time grid of five equally spaced intervals. The propagation has been performed using both a Linearized Approach (LinCov) and the Unscented Transform (UT), within the framework of the Planar Bicircular Restricted Four-Body Problem (PBRFBP) dynamics model.

LinCov: The Linear Covariance (LinCov) method estimates the evolution of uncertainties by applying a first-order linearization of the dynamics model to propagate the state covariance matrix. At each of the five time nodes t_k, the state transition matrix $\mathbf{\Phi}(t_0, t_f)$ is calculated using a variational approach and then used to update the covariance. Meanwhile, the mean state vector is obtained by propagating the system dynamics through the flow ($\boldsymbol{\varphi}(\mathbf{x_0}, t_0, t_f)$):

$$\hat{\mathbf{x}}_{LinCov}(t_k) = \boldsymbol{\varphi}(\mathbf{x_i}, t_i, t_k) \qquad \mathbf{P}(t_k) = \mathbf{\Phi}(t_i, t_k)\mathbf{P_0}\mathbf{\Phi}(t_i, t_k)^T \tag{1}$$

The LinCov method assumes both linearity of the dynamics and gaussianity of the uncertainties, which allows it to efficiently propagate uncertainties primarily along the principal direction of the trajectory.

UT: Assuming gaussianity of the initial distribution at time t_i, the Unscented Transform (UT) method parametrizes the initial distribution with a set of sigma points ($\boldsymbol{\chi}_i$), representing a set of initial states. The sigma points are then propagated to each of the five nodes t_k, by integrating the PBRFBP dynamics, obtaining the propagated sigma points: $\boldsymbol{\gamma}_j(t_k) = \boldsymbol{\varphi}(\boldsymbol{\chi}_i, t_i, t_k)$. The mean state and covariance are then computed with a weighted average of the propagated samples:

$$\hat{\mathbf{x}}_{UT}(t_k) = \sum_{j=0}^{2n} W_j^{(m)}\boldsymbol{\gamma}_j \qquad \mathbf{P}(t_k) = \sum_{j=0}^{2n} W_j^{(c)}(\boldsymbol{\gamma}_j - \hat{\mathbf{x}}_{UT}(t_k))(\boldsymbol{\gamma}_j - \hat{\mathbf{x}}_{UT}(t_k))^T \tag{2}$$

Since the UT method propagates the sigma points through the nonlinear dynamics model, it provides a more accurate approximation of the evolution of the mean state and covariance, especially in highly non-linear regimes. A comparison of the outputs obtained by the two methods in terms of mean position and covariance ellipse is showcased in Fig. 1.

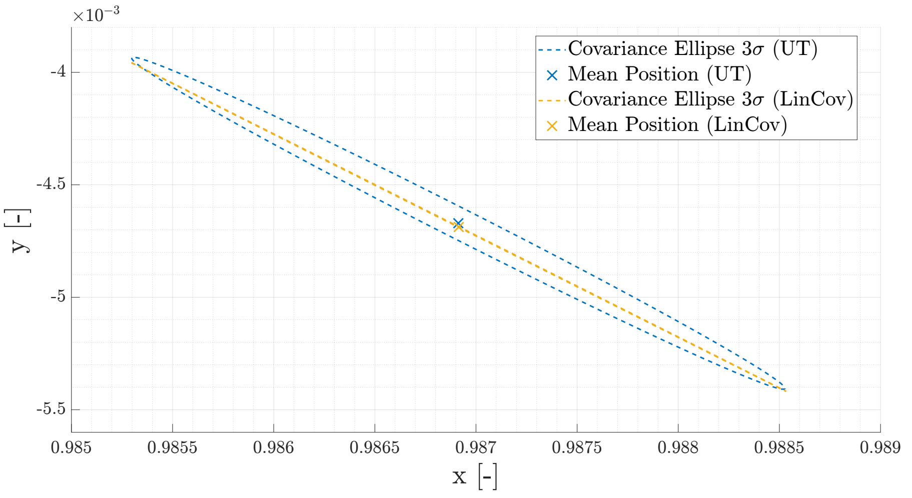


Figure 1: Mean states and covariance ellipses for LinCov and UT methods

As can be seen in Fig. 1, both methods provide similar mean state estimates at the final time t_f but a slight shift is observed. This is because the UT method better represents the non-linearities of the dynamics model, leading to an improved estimate of the mean state. Considering the covariance, the ellipse obtained by the LinCov method is considerably narrower compared to that obtained from UT. Consequently, LinCov may underestimate the distribution of uncertaintes, particularly along the secondary directions of the trajectory, while in the primary direction it provides a comparable distribution, as supported by the theory. The differences between the results obtained by LinCov and UT are mainly due to the models' initial assumptions. LinCov assumes that the uncertainty distribution remains Gaussian during propagation, and is based on a locally linearized dynamics around the mean state. On the other hand, UT assumes gaussianity only for the initial distribution, and then the initial samples are propagated using the full nonlinear dynamics model. For these reasons, the UT is more robust with respect to possible deviations from Gaussianity and it accounts for non-linearity, explaining the differences in the results.

1.2 Monte Carlo (MC) Simulation

In this section, the same uncertainty propagation presented in Section 1.1 is performed using a Monte Carlo simulation with a population size n of 1000, chosen to balance the computational cost and the accuracy of the simulation.

MC: To perform the Monte Carlo simulation, first a set of n samples are generated assuming an initial multivariate Gaussian random distribution at t_i, by using MATLAB's function `mvnrnd`. Then, all the samples are propagated to each of the five nodes t_k by integrating the PBRFBP dynamics. The final mean state and covariance matrix are then computed using MATLAB's `mean` and `cov` functions.

Fig. 2 shows the mean state and covariance ellipse obtained by the Monte Carlo simulation at the final time t_f, as well as the previous results obtained by the UT and LinCov methods. The final propagated samples are also represented.

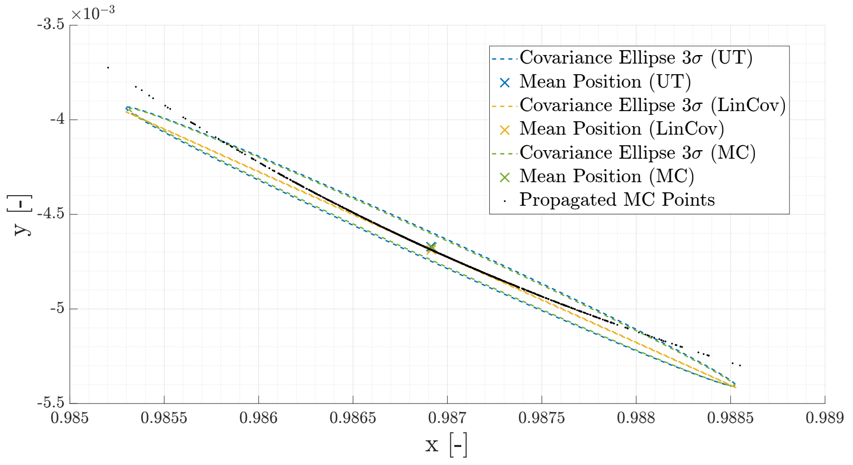


Figure 2: Mean states and covariance ellipses LinCov, UT and MC methods

As can be seen, the plots confirm the considerations made in Section 1.1, as the UT and MC approaches yield similar results both in terms of mean state and covariance ellipse, while LinCov provides a much smaller covariance ellipse, especially in the secondary direction of the trajectory. Therefore, the Monte Carlo simulation validates the assumption that the Linear and Gaussian regimes have slightly dropped during the propagation. This is further backed by the representation of the propagated MC samples, which are starting to align to a banana shape, typical of non-Gaussian regimes.

A further analysis of the results can be performed by evaluating the time evolution of $3\sqrt{\max(\lambda_i(\mathbf{P})}$, both for position and velocity covariance submatrices, as shown in Fig. 3:

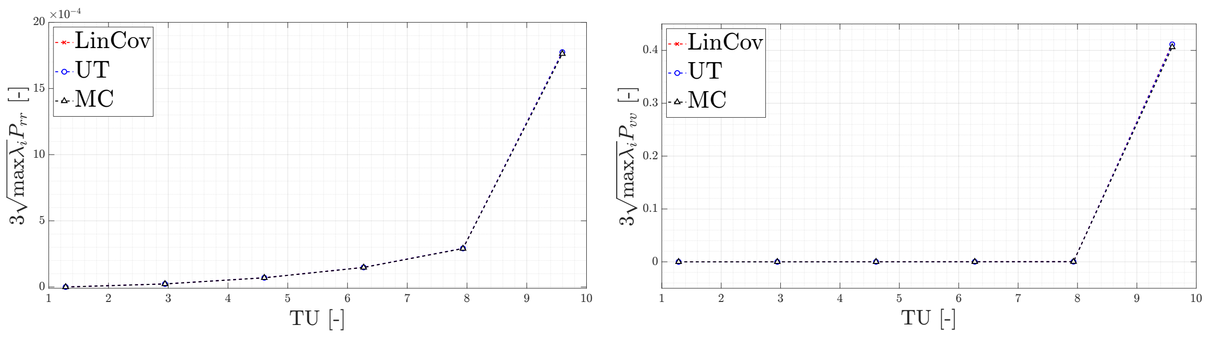


Figure 3: Time evolution of the maximum square root of the eigenvalues of position and velocity covariance matrices (LinCov, UT, MC)

This plot shows a steep increase of the maximum eigenvalues for the position and velocity covariances, highlighting that the uncertainties are rapidly growing towards the end of the propagation. In addition, it can be noted that the time evolution is similar for all three approaches, showing that LinCov accurately captures the uncertainties along the principal direction of the trajectory, associated to the maximum eigenvalue of the covariance matrix (through the corresponding eigenvector).
Furthermore, the absence of significant deviations from the three methods shows that, despite the non-linear dynamics, the system has not diverged from a regime in which the linear approximation holds, at least for the primary direction of the trajectory.

Finally, a more thourough validation of the Gaussianity assumption is performed by evaluating the QQ-plots shown in Fig. 4.

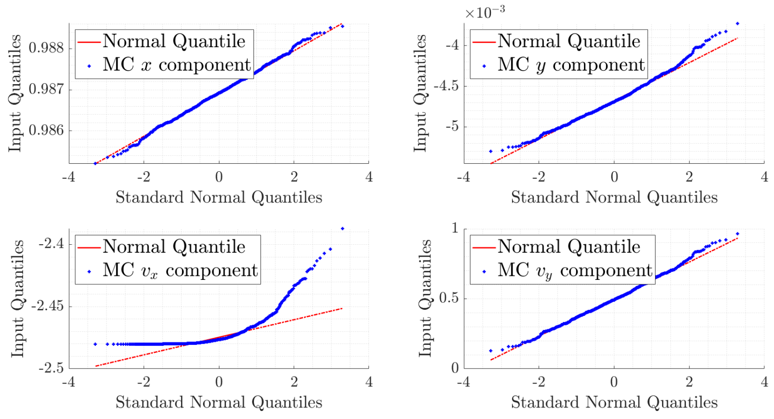


Figure 4: QQ-plots for each state component at final time

These plots show that the position components (x and y) align well with the theoretical Gaussian distribution, with minimum deviations in the tails. On the other hand, the velocity components (in particular v_x) show more evident deviations, especially at the extremes, showing some divergence from the theoretical Gaussian behaviour. This indicates that, while the Gaussian assumption holds reasonably well for the position states, it becomes less accurate for the velocities due to the stronger impact of non-linearities during the propagation.

In conclusion, the analysis has yielded significantly accurate results for both LinCov and UT approaches, despite the non-linear nature of the model. In fact, UT effectively captures the uncertainties in all directions, while LinCov performs well only in the primary direction of the trajectory. The Monte Carlo simulation validates the results obtained with the two methods, providing further insights on the validity of the Gaussian regime through the obtained QQ-plots.

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 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 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
```

2.1 Visibility Windows

The aim of this section is to compute the time windows for the three ground stations in which the SMOS satellite is above the minimum elevation (reported in Table 2), meaning that the spacecraft is inside the field of view and measurements can be performed. The reference satellite's state in ECI coordinates is extracted from the TLE's given in Table 3 at the reference epoch t_{ref}.

Position $[x, y, z]$ [km]	$[-6065.4138, 3768.0469, 14.5009]$
Velocity $[v_x, v_y, v_z]$ [km/s]	$[0.6036, 0.9267, 7.3908]$

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

Afterwards, the satellite's trajectory is propagated from t_0 to t_f using a Keplerian propagator, with the time interval being discretized according to the given measurement frequency for each ground station. Computing the station's position in the ECI frame using `cspice_spkezr`, it is possible to retrieve the relative position vector of the satellite with respect to the ground station. Finally, this vector is rotated into the topocentric frame and the local coordinates of the spacecraft in terms of range, azimuth and elevation are computed using `cspice_xfmsta`. Comparing the propagated elevation to the threshold for each station, it is possible to retrieve the visibility windows, as shown in Table 5:

Station	**Start Time (UTC)**	**End Time (UTC)**	**Number of Acquisitions**
Kourou	2024-11-18T20:40:00.000	2024-11-18T20:49:00.000	10
Troll	2024-11-18T21:02:30.000	2024-11-18T21:11:30.000	19
Svalbard	2024-11-18T21:56:00.000	2024-11-18T22:06:00.000	11

Table 5: Visibility windows for SMOS tracking

Concurrently, Fig. 5, Fig. 6 and Fig. 7 show on the right the time evolution of azimuth and elevation over the total time interval, while on the right the spacecraft's trajectory is displayed in terms of local coordinates, showing the satellite's path through the field of view of each station.

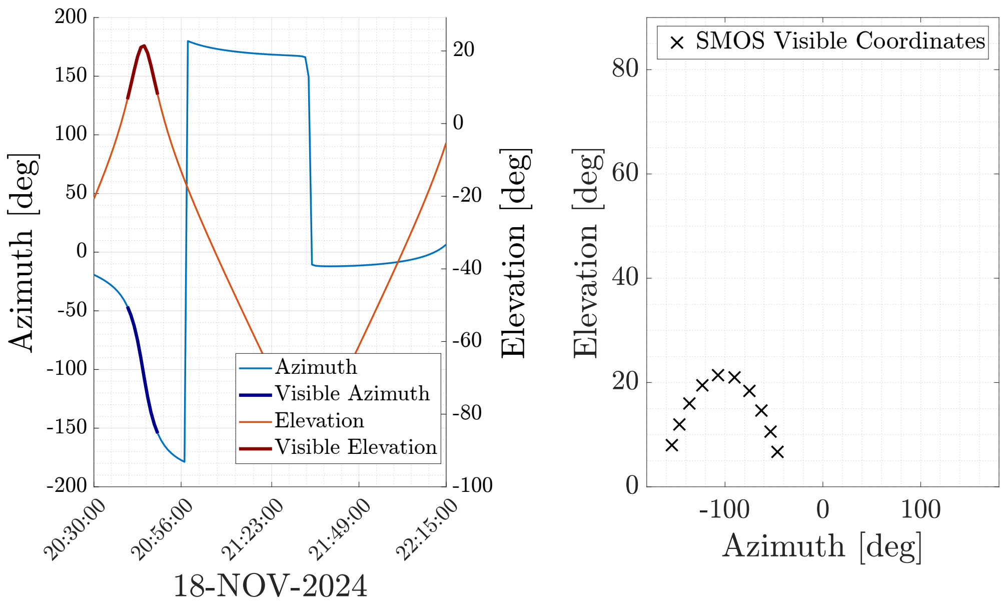


Figure 5: SMOS pass from KOUROU Ground Station

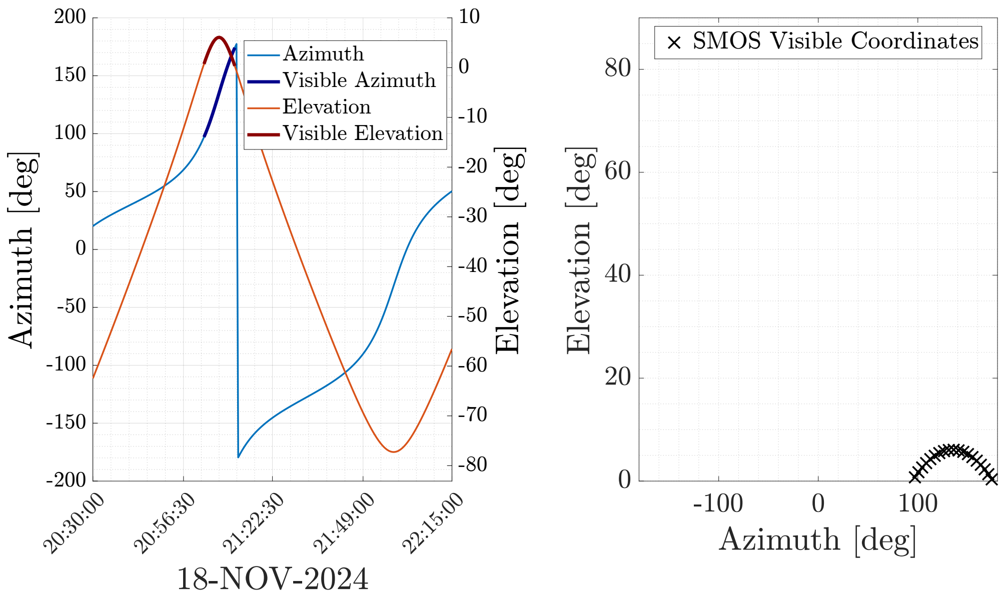


Figure 6: SMOS pass from TROLL Ground Station

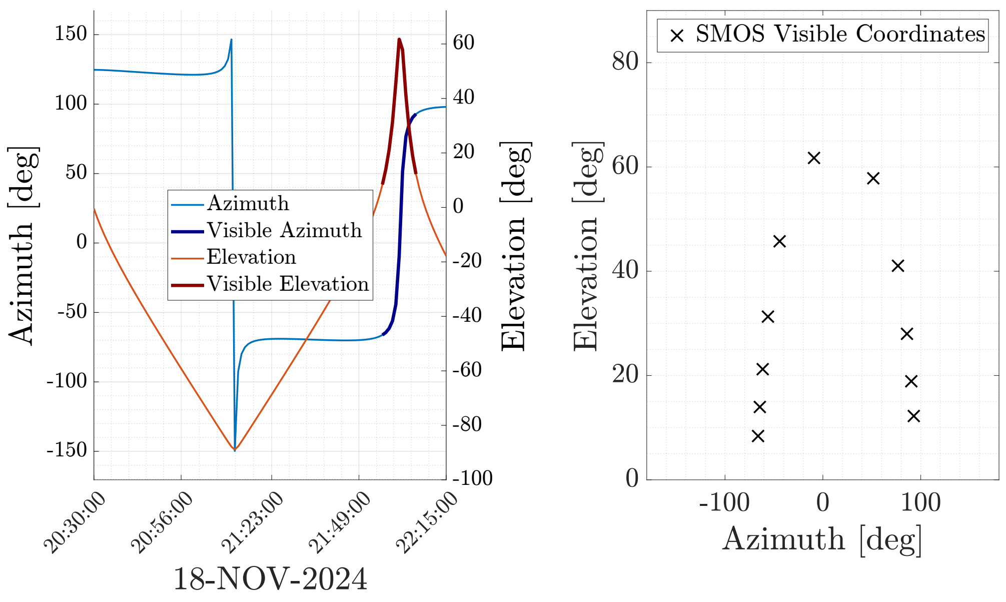


Figure 7: SMOS pass from SVALBARD Ground Station

2.2 Simulated Measurements

The aim of this section is to simulate the measurements provided by each ground station during SMOS's pass. The ideal measurements of range, azimuth and elevation are computed following the same rationale as the one presented in Section 2.1, this time using SGP4 to propagate the satellite's trajectory starting from the TLE data over the given time interval. SGP4 is used in this case as it models major perturbations, yielding more realistic outputs with respect to the Keplerian propagator.

The simulated measurements are generated from the ideal ones by perturbing the expected values with random noise through the use of MATLAB's `mvnrnd`, based on each station's noise covariance matrix. This step adds further deviations, ensuring that the simulated measurements closely resemble real-world scenarios. The difference between the propagated local coordinates obtained in Section 2.1 and the simulated measurements inside each station's visibility window in terms of azimuth and elevation is showcased in Section 2.2.

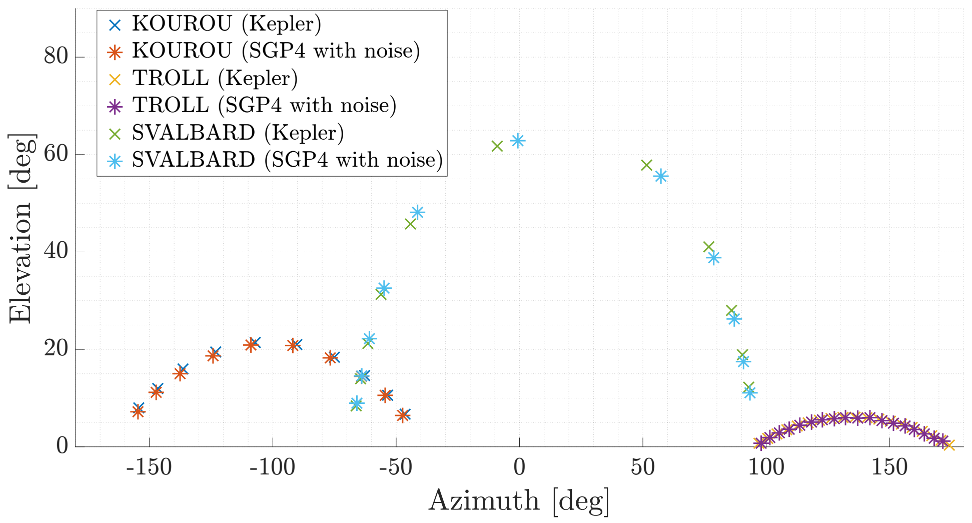


Figure 8: Measurements from GS: ideal (TBP Propagation) vs simulated (`SGP4` with noise)

2.3 Navigation Solutions

In this section, the navigation problem is solved in terms of least squares estimate using a batch filter for three different scenarios:

(a) using the simulated measurements obtained for the KOUROU ground station only

(b) using all simulated measurements from all the ground stations

(c) incorporating a J2-perturbed dynamic model

The navigation problem is solved using least-squares optimization by the `lsqnonlin` function in MATLAB, with `levenberg-marquardt` as algorithm, which solves a problem in the form:

$$\min_{\mathbf{x}_0} \sum_{j=1}^{N_s} \sum_{i=1}^{N_m} \mathbf{W_j} \mathbf{r_{i,j}} \tag{3}$$

Where $\mathbf{r_{i,j}}$ is the vector of residuals for the i-th measurements of the j-th ground station, $\mathbf{W_j}$ is a weight matrix based on the measurement noise of each station. N_s is the number of stations considered for each scenario, while N_m is the number of measurements provided by each station. A schematic overview of the algorithm used to compute the weighted residuals is showcased in Appendix A.

The obtained solutions to the navigation problems are presented for each of the three scenarios in the following Table 6, in terms of optimal mean state and covariance, along with the traces of the position and velocity submatrices.

As requested, the standard deviations for the orbit semi-major axis and inclination are derived by applying a linear transformation from the state covariance to the Keplerian elements covariance matrices. The optimal mean and associated covariance found in the optimization algorithm were used to compute the nominal Keplerian elements, along with the corresponding Jacobian matrix. The Jacobian was computed by perturbing the state components using a Forward Finite Differences method.

Finally, the covariance matrix associated to the nominal Keplerian elements is found as:

$$\mathbf{P}_{\text{kepler}} = \mathbf{J}\mathbf{P}_{\text{state}}\mathbf{J}^T \tag{4}$$

where $\mathbf{P}_{\text{state}}$ is the state covariance matrix. This process was crucial to assess the precision of the orbit determination of SMOS provided by the different scenarios.

Table 6: Navigation Solutions at time t_0 (@Earth ECI Frame)

(a) Kourou Measurements only

Position in ECI $[x, y, z]$ **[km]**	[3933.8923, -1416.6467, 5780.8165]
Velocity in ECI $[v_x, v_y, v_z]$ **[km/s]**	[4.8798, -3.7621, -4.2386]
σ_r **[km]**	9.4745
σ_v **[km/s]**	0.0088
σ_a **[km]**	6.6903
σ_i **[deg]**	0.0740

$$\mathbf{P_0} = \begin{bmatrix} 38.3546 & 14.2179 & -9.7085 & -0.0377 & -0.0185 & -0.0130 \\ 14.2179 & 35.8368 & -23.5899 & -0.0042 & -0.0251 & 0.0147 \\ -9.7085 & -23.5899 & 15.5757 & 0.0032 & 0.0165 & -0.0095 \\ -0.0377 & -0.0042 & 0.0032 & 4.0259e-05 & 1.2221e-05 & 1.9020e-05 \\ -0.0185 & -0.0251 & 0.0165 & 1.2221e-05 & 2.0235e-05 & -5.3320e-06 \\ -0.0130 & 0.0147 & -0.0095 & 1.9020e-05 & -5.3320e-06 & 1.6864e-05 \end{bmatrix}$$

(b) Measurements from all Ground Stations

Position in ECI $[x, y, z]$ **[km]**	[3926.8741, -1411.0056, 5780.0332]
Velocity in ECI $[v_x, v_y, v_z]$ **[km/s]**	[4.8886, -3.7650, -4.2334]
σ_r **[km]**	1.0304
σ_v **[km/s]**	0.0010
σ_a **[km]**	0.1538
σ_i **[deg]**	0.0021

$$\mathbf{P_0} = \begin{bmatrix} 0.6061 & -0.3722 & -0.0912 & -0.0006 & 0.0002 & -0.0004 \\ -0.3722 & 0.3812 & 0.0806 & 0.0004 & -0.0001 & 0.0003 \\ -0.0912 & 0.0806 & 0.0745 & 0.0001 & -0.0000 & 0.0001 \\ -0.0006 & 0.0004 & 0.0001 & 6.8445e-07 & -1.7216e-07 & 3.6412e-07 \\ 0.0002 & -0.0001 & -0.0000 & -1.7216e-07 & 1.1865e-07 & -1.2691e-07 \\ -0.0004 & 0.0003 & 0.0001 & 3.6412e-07 & -1.2691e-07 & 2.6330e-07 \end{bmatrix}$$

(c) Measurements from all Ground Stations and J2-Perturbed Dynamics

Position in ECI $[x, y, z]$ **[km]**	[3932.7817498518 -1414.9278764592 5778.5016919446]
Velocity in ECI $[v_x, v_y, v_z]$ **[km/s]**	[4.8797808751 -3.7632037553 -4.2328815151]
σ_r **[km]**	0.0300
σ_v **[km/s]**	3.0976e-05
σ_a **[km]**	0.0036
σ_i **[deg]**	5.1834e-05

$$\mathbf{P_0} = \begin{bmatrix} 5.1316e-04 & -3.3054e-04 & -1.3574e-04 & -5.1151e-07 & 1.7055e-07 & -3.7232e-07 \\ -3.3054e-04 & 3.0570e-04 & 1.2057e-04 & 3.6109e-07 & -9.5763e-08 & 2.8293e-07 \\ -1.3574e-04 & 1.2057e-04 & 8.3446e-05 & 1.1805e-07 & -3.8964e-08 & 1.3177e-07 \\ -5.1151e-07 & 3.6109e-07 & 1.1805e-07 & 5.6316e-10 & -1.4744e-10 & 3.7766e-10 \\ 1.7055e-07 & -9.5763e-08 & -3.8964e-08 & -1.4744e-10 & 8.6542e-11 & -1.1986e-10 \\ -3.7232e-07 & 2.8293e-07 & 1.3177e-07 & 3.7766e-10 & -1.1986e-10 & 3.0981e-10 \end{bmatrix}$$

The results for the Kourou ground station only can be consulted in Table 6a. As can be seen, this scenario yields significantly high uncertainties in position and velocity estimation, due to the limited observation time and geographical location of Kourou, with SMOS being in a polar orbit and therefore spending minimal time over the equatorial region where the station is located. In addition, the 6° constraint on the minimum elevation further limits the visibility. Combining results from Troll and Svalbard greatly improves the navigation solution, as can be seen in Table 6b: this is due to an increased observation time and a better coverage of the satellite's trajectory. Including different stations enhances the reliabilty and robustness of the solution, adding redundancy and limiting dependancy on a single station's geographical location.
Finally, adding the J2-perturbed dynamics further improves the solution, as showcased in Table 6c. This model, in fact, effectively captures the effects of Earth's oblateness, particularly the advancement of the perigee and the precession of the ascending node, leading to a more accurate estimation of the spacecraft's real trajectory. It has to be noted that the J2-perturbed model would yield even better results for a more extended observation campaign.

In conclusion, the obtained results emphasize the importance of combining precise dynamic models and proper geographic distribution of the stations to achieve precise navigation solutions.

2.4 Trade-Off Analysis

In this section, a trade-off analysis is performed. The procedure to gather the navigation solution is the same outlined in Section 2.3, estimated using only two stations (due to the given budget constraints) for each scenario and adopting a J2-perturbed model of the spacecraft's dynamics. The obtained results for the orbit determination are presented in Table 7, in terms of standard deviation of the semi-major axis and inclination.

	Kourou-Troll	**Kourou-Svalbard**	**Troll-Svalbard**
$\boldsymbol{\sigma}_a$ [km]	0.4834	0.0160	0.0557
$\boldsymbol{\sigma}_i$ [deg]	9.3182e-04	6.3214e-04	6.9387e-04

Table 7: Trade-off analysis results

The results of the analysis highlight some key differences among the possible scenarios.
The main factors affecting the performance are the geometric diversity of the network and the orbital coverage and visibility of the stations involved in the measurements. Ideally, the network should consist of stations located at very different latitudes to ensure complementary measurements and that wide, frequent visibility windows. Additionally, due to SMOS' polar orbit, stations at high latitudes are better suited for estimating the orbit inclination, while observations from equatorial locations like Kourou enable more precise estimation of the semi-major axis.

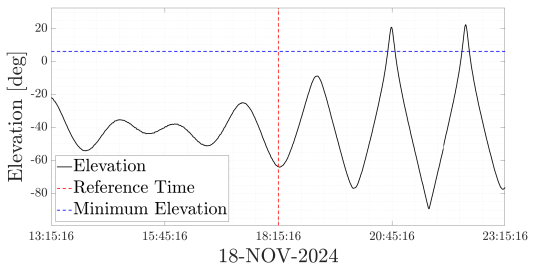
From these considerations, it is evident from Table 7 that the Kourou-Svalbard scenario delivers the best results for estimating both a and i. This is because the two stations have a significant latitude diversity and good orbital coverage within the analyzed time window. The Kourou-Troll combination yields the poorest results for estimating orbital elements due to its less pronounced geometric diversity (Troll is located at approximately $-72°$, while Svalbard is at $78°$) and poorer orbital coverage during the time window. The last combination, Troll-Svalbard, provides accurate estimates of the orbit inclination but performs worse for the semi-major axis, because both stations are located at high latitudes.

Due to the stochastic nature of the analysis, multiple simulations were performed. The Kourou-Svalbard network consistently delivered the best estimates for a, while for i it often performed best but not always, as Troll-Svalbard occasionally achieved better results.
In conclusion, the optimal choice of ground for the analyzed time window is Kourou-Svalbard.

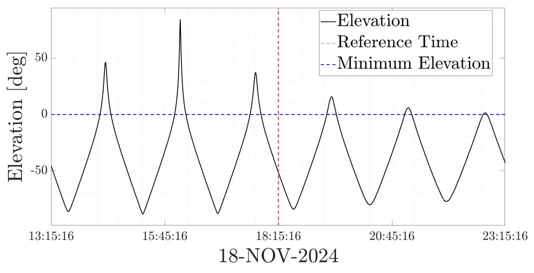
This combination is also cost-effective, amounting to 65.000 €, which is within the budget constraints.

2.5 Long-Term Analysis

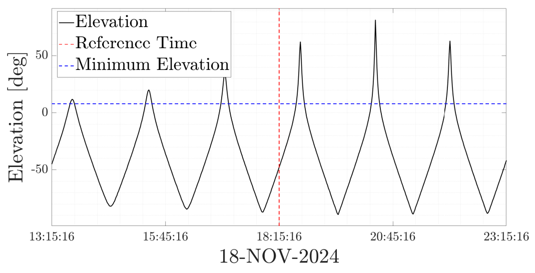
To perform the long-term analysis, the time window of the observation campaign was extended to ±5 hours from the reference epoch of the provided TLEs. This choice was dictated by the need to have a larger observational period while avoiding additional corrections for precession and nutation that would have been required beyond this interval. The time evolution of the satellite's elevation was computed, determining the different visibility windows for the three ground stations, displayed in Fig. 9.



(a) Kourou



(b) Troll



(c) Svalbard

Figure 9: Elevation profile observed from the stations over the extended time window

As can be seen, SMOS passes over Kourou are very infrequent, while over Troll and Svalbard the passes are far more repeated. This indicates that, for long-term tracking, the Kourou-Svalbard configuration - which was optimal for the short-term analysis of Section 2.4 - is no longer the best choice. Indeed, Troll-Svalbard emerges as the optimal combination, because it has better coverage and is more reliable over time. Nevertheless, Kourou can still be utilized as a backup station to be used when SMOS passes directly over it, as was the case of the short-term analysis. During those passes, the station would provide better estimates of the orbital elements, especially the semi-major axis.

In conclusion, for long-term operations, a network of Troll and Svalbard would serve best as a primary configuration, with Kourou as a backup option. This scenario offers an optimal balance to have reliable orbit determination and repeatable measurements.

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 8, predict the trajectory of the satellite in an inertial Moon-centered reference frame assuming Keplerian motion. Use the estimated coordinates given in Table 9 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 8 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 8. 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 8. For the initial state, you can perturb the actual initial state, composed by $\mathbf{x}_0$ and the latitude

and longitude given in Table 9, 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 8: 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 9: 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

The aim of this section is to predict the trajectory of the lunar orbiter using a Keplerian motion model in a MCI reference frame and verify that the lander and orbiter are in mutual visibility throughout the entire time interval from t_0 to t_f.
The initial state $\mathbf{x}_0$ is propagated using Two-Body Problem dynamics over the entire time interval, with the required time step. The plot of the trajectory can be seen in Fig. 10.
To check the relative visibility of the orbiter with respect to the lunar lander, the local coordinates evolution need to be retrieved. From initial guess of the coordinates of the lander, the position vector in the Moon Centered Moon Fixed (MCMF) frame can be retrieved using `cspice_latrec`, while the velocity vector is null as the lander is not moving on the Moon's surface. The lander initial state is presented in Table 10.

x [km]	y [km]	z [km]	v_x [km/s]	v_y [km/s]	v_z [km/s]
348.917	93.492	1699.434	0	0	0

Table 10: Lunar Lander State at t_0 (@Moon MCMF)

The MCMF state of the lander is then rotated to the MCI frame using `cspice_sxform`, which allows to compute the relative position of the orbiter with respect to the lander. The relative state is rotated into the topocentric reference frame, and then by exploting `cspice_xfmsta` the local coordinates in terms of range, azimuth and elevation of the orbiter are computed.
Fig. 11 shows the time evolution of the orbiter elevation over the entire time interval of interest and confirms that the relative visibility is always ensured, as the satellite always stays above the minimum threshold.

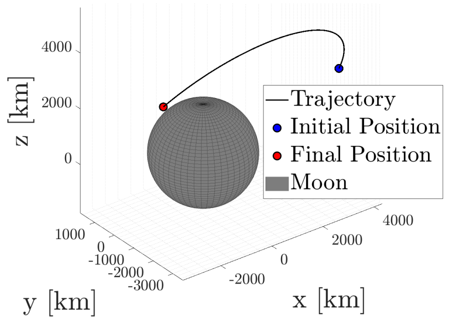


Figure 10: Lunar Orbiter Trajectory (@Moon MCI)

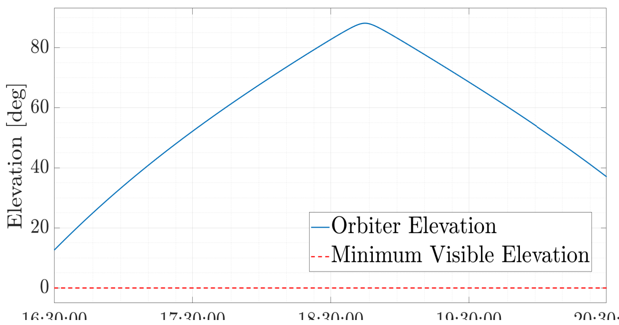


Figure 11: Time evolution of Lunar Orbiter elevation observed from Lunar Lander

3.2 Simulated Measurements

This section aims at simulating the real measurements provided by the lunar lander by introducing Gaussian random noise into the position and relative range data, as well as verifying that the applied noise level falls within the 3σ bounds.
In this case, the lunar lander states in the MCI frame at the different discretized epochs are retrieved by calling `cspice_spkezr`, while the evolution of the orbiter state in the MCI frame is computed following the same procedure outlined in Section 3.1. As done previously, the relative range is computed by rotating to the topocentric frame and using `cspice_xfmsta`. Fig. 12 shows the time evolution of the ideal measurements of position and range over the time interval of interest.

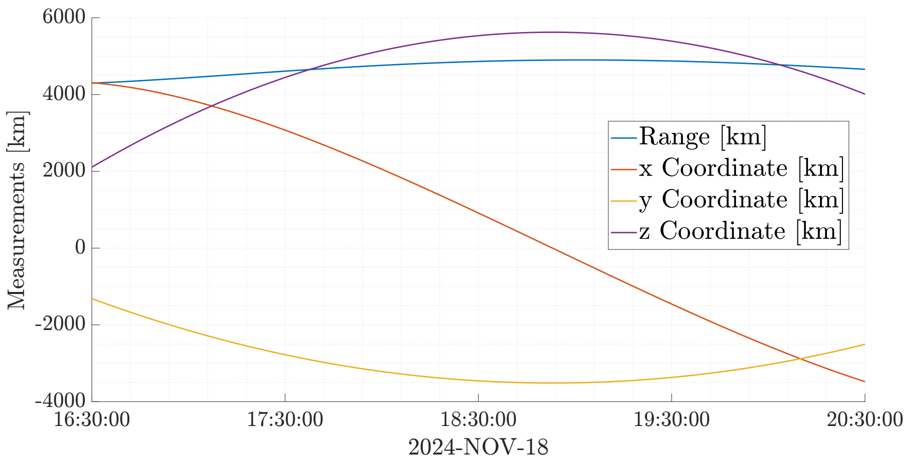


Figure 12: Time evolution of Lunar Orbiter's position and relative range with respect to Lunar Lander

Afterwards, the ideal measurements are perturbed by adding random errors using `mvnrnd` MATLAB's function, obtaining the simulated measurements. Fig. 13 shows the time evolution of the relative errors between simulated and ideal measurements, as well ass the $\pm 3\sigma$ bounds.

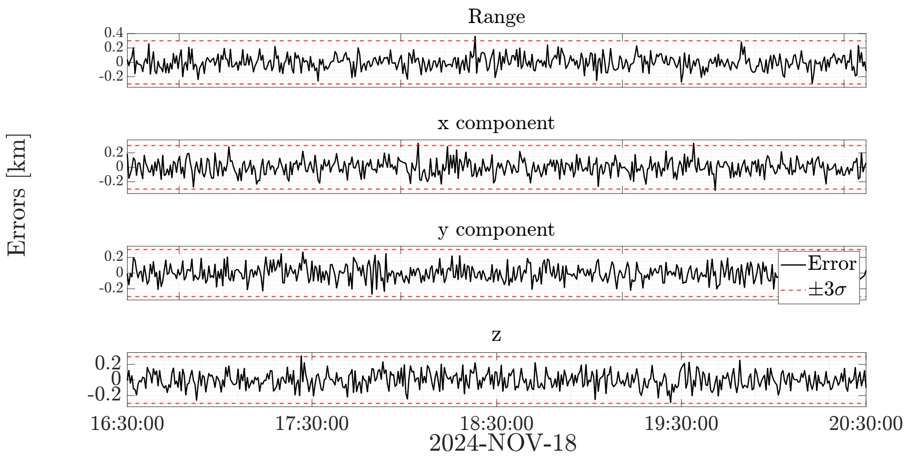


Figure 13: Errors between ideal and simulated measurements with $\pm 3\sigma$ boundaries

The plots validate the assumption of Gaussianity for the noise distribution, as more than 99% of the noise values for position and range fall within the expected bounds.

3.3 Lunar Orbiter State Estimation

In this section, an Unscented Kalman Filter (UKF) is developed to to improve the estimate of the orbiter's state using the simulated measurements of the position vector components.
At each discretized time step, the filter works by propagating a set of sigma points through the dynamics model to predict the mean and covariance. Afterwards, the available measurements are compared to the predicted ones to correct the predicted mean and covariance, narrowing down the uncertainties on the state of the orbiter. A more comprehensive overview of the UKF algorithm is displayed in Appendix B.
The process is iterated sequentially until all measurements in the time window are processed. Fig. 14 show the time evolution of position and velocity errors computed between the reference propagated trajectory and the states estimated by the UKF.

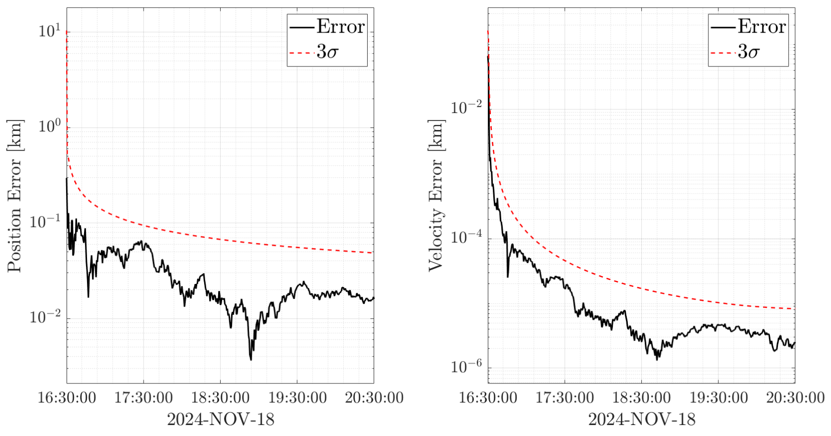


Figure 14: Time evolution of Position and Velocity Errors with 3σ bounds

The exhibited plots show that, as the filter processes more measurements, position and velocity errors - as well as their standard deviation - decrease over time, remaining within the 3σ bounds.

Table 11 shows the estimated mean state and associated covariance matrix by the UKF algorithm for the final time.

Position $[x, y, z]$ **[km]**	[-3479.0354, -2506.1802, 4010.7454]
Velocity $[v_x, v_y, v_z]$ **[km/s]**	[-0.4501, 0.3485, -0.5578]

$$\mathbf{P} = \begin{bmatrix} 7.1743e-05 & 6.9028e-06 & -1.1054e-05 & 5.7777e-09 & 3.6027e-09 & -5.7687e-09 \\ 6.9028e-06 & 8.1388e-05 & -2.8057e-05 & 5.1924e-09 & 8.8490e-09 & -9.3912e-09 \\ -1.1054e-05 & -2.8057e-05 & 1.0873e-04 & -8.3071e-09 & -9.3909e-09 & 1.8006e-08 \\ 5.7777e-09 & 5.1924e-09 & -8.3071e-09 & 1.9284e-12 & 1.2069e-12 & -1.9311e-12 \\ 3.6027e-09 & 8.8490e-09 & -9.3909e-09 & 1.2069e-12 & 1.9523e-12 & -1.8622e-12 \\ -5.7687e-09 & -9.3912e-09 & 1.8006e-08 & -1.9311e-12 & -1.8622e-12 & 3.7683e-12 \end{bmatrix}$$

Table 11: Final estimated state of Lunar Orbiter (@Moon MCI)

3.4 Lunar Lander Coordinates Estimation

In the final section, the goal is to extend the UKF developed in Section 3.3 to estimate simultaneously the orbiter's state and the coordinates of the lunar lander. The process follows the same steps outlined in Algorithm 2, with the addition of the relative range measurement. The objective is to estimate iteratively the augmented mean state and covariance matrix, which now include both the orbiter's state and the lander's coordinates, at each time step.
Table 12 shows the estimated augmented state and associated covariance matrix at the final time.
The obtained results in terms of time evolution of the error estimate for both the state and the coordinates are reported in Fig. 15 and Fig. 16, with the 3σ of the estimated covariance.

Lander Latitude [deg]	78.2230
Lander Longitude [deg]	15.3955

Position in ECI $[x, y, z]$ **[km]**	[-3479.0396, -2506.1813, 4010.7470]
Velocity in ECI $[v_x, v_y, v_z]$ **[km/s]**	[-0.4501, 0.3485, -0.5578]

$$
\mathbf{P} = \begin{bmatrix}
6.3672\text{e-}05 & 4.5772\text{e-}06 & -7.4467\text{e-}06 & 3.8886\text{e-}09 & 2.7434\text{e-}09 & -4.3989\text{e-}09 & -2.7277\text{e-}07 & -2.8176\text{e-}07 \\
4.5772\text{e-}06 & 8.1923\text{e-}05 & -2.8915\text{e-}05 & 5.2371\text{e-}09 & 8.8929\text{e-}09 & -9.4603\text{e-}09 & -2.9078\text{e-}07 & -2.5651\text{e-}07 \\
-7.4467\text{e-}06 & -2.8915\text{e-}05 & 1.1011\text{e-}04 & -8.3894\text{e-}09 & -9.4665\text{e-}09 & 1.8125\text{e-}08 & 4.7001\text{e-}07 & 4.1281\text{e-}07 \\
3.8886\text{e-}09 & 5.2371\text{e-}09 & -8.3894\text{e-}09 & 1.7858\text{e-}12 & 1.1522\text{e-}12 & -1.8433\text{e-}12 & -5.7810\text{e-}11 & -6.4308\text{e-}11 \\
2.7434\text{e-}09 & 8.8929\text{e-}09 & -9.4665\text{e-}09 & 1.1522\text{e-}12 & 1.9320\text{e-}12 & -1.8298\text{e-}12 & -5.4833\text{e-}11 & -4.6662\text{e-}11 \\
-4.3989\text{e-}09 & -9.4603\text{e-}09 & 1.8125\text{e-}08 & -1.8433\text{e-}12 & -1.8298\text{e-}12 & 3.7165\text{e-}12 & 8.7142\text{e-}11 & 7.3644\text{e-}11 \\
-2.7277\text{e-}07 & -2.9078\text{e-}07 & 4.7001\text{e-}07 & -5.7810\text{e-}11 & -5.4833\text{e-}11 & 8.7142\text{e-}11 & 1.0288\text{e-}08 & 1.1242\text{e-}08 \\
-2.8176\text{e-}07 & -2.5651\text{e-}07 & 4.1281\text{e-}07 & -6.4308\text{e-}11 & -4.6662\text{e-}11 & 7.3644\text{e-}11 & 1.1242\text{e-}08 & 1.3239\text{e-}08
\end{bmatrix}
$$

Table 12: Final Position, Velocity, Latitude, Longitude, and Covariance Matrix (@Moon MCI)

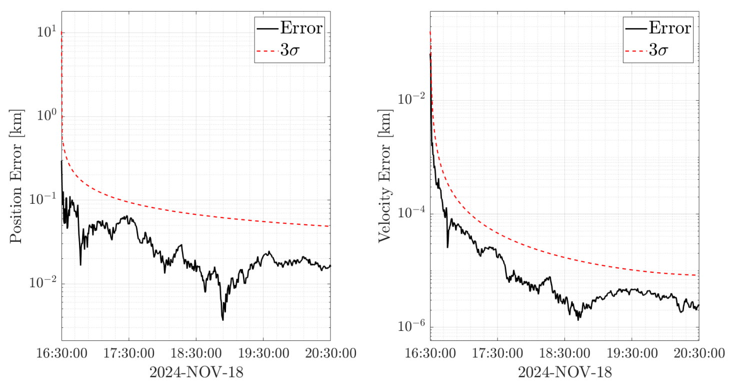


Figure 15: Time evolution of Position and Velocity Errors with 3σ bounds

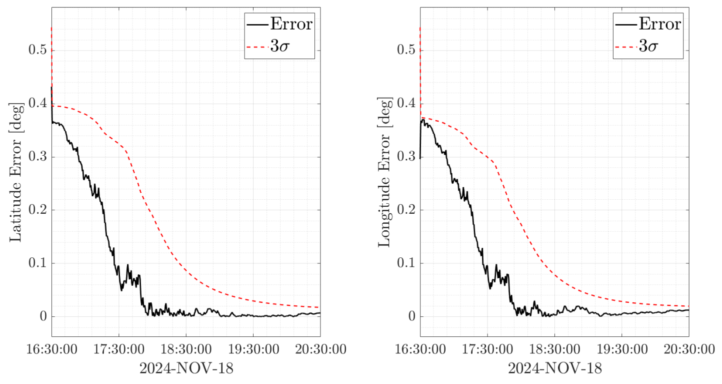


Figure 16: Time evolution of Longitude and Latitude Errors with 3σ bounds

As showcased in the obtained plots, the performance of the UKF is satisfactory as the relative errors decrease over time and stay within the 3σ bounds.
Due to the stochastic nature of the analysis, the analysis was performed multiple times to assess the validity of the adopted models. In some simulations, the error in the coordinates of the lander exceeds the 3σ bounds for short period of times, suggesting occasional deviations from the assumption of Gaussianity. This behaviour could be due to the non-linearities in the relationship between the orbiter's position and the relative range measurements, combined with the random nature of the generated noise. The estimate of the orbiter's state, however, remains consistently within the expected bounds, showing that the position measurements are more robust and less affected by the non-linearities.
These short-term violations do not invalidate the performances of the UKF, but they highlight some potential limitations in the model ability to estimate accurately the lander's coordinates. Overall, the filter remains reliable and robust, but to achieve better performances the dynamic model could be enhanced adding higher order approximations and improving the accuracy of the noise model.

Appendix

Exercise A: Batch Filter Algorithm

Algorithm 1 Computation of Residuals for Least Squares Optimization

```
Require: $\mathbf{x}_0$ (Initial state vector), stations (cell array of station data)
Ensure: Residuals of the cost function ($\mathbf{r}$)
 1: Extract stations measurements:
 2: for $i = 1 : \mathrm{n}_{\mathrm{Stations}}$ do
 3:     Collect visibility epochs and associated measurements
 4:     Compute measurement weights and ground station states
 5: end for
 6: Sort measurements by time
 7:     collect increasing visibility times vector in $\mathbf{t}_{vec}$
 8:     collect sorted measurements in meas
 9: Propagate states at discretized visible epochs using ODE solver
10: for each time in $\mathbf{t}_{vec}$ do
11:     Convert propagated states from ECI to topocentric frame
12:     Compute range, azimuth, and elevation
13: end for
14: Compute residuals:
15: for each time in $\mathbf{t}_{vec}$ do
16:     Calculate weighted differences between predicted and real measurements
17:     Store residuals in $\mathbf{r}$
18: end for
19: return Residuals ($\mathbf{r}$)
```

Exercise B: B: UKF Algorithm

Algorithm 2 General UKF scheme for step $t_{k-1} \to t_k$

```
1: Given: A posteriori estimate $\mathbf{x}k - 1$ and covariance $\mathbf{P}k - 1$ at $t_{k-1}$ and measurements $\mathbf{y_k}$, $\mathbf{R_k}$ at $t_k$
2: Integration Step:
3: Generate sigma points $\boldsymbol{\chi_{k-1}}$ from $\hat{\mathbf{x}}^{+}k - 1$ and $\mathbf{P}^{+}k - 1$
4: Propagate with Keplerian dynamics. Output: $\boldsymbol{\chi_k}$
5: Prediction Step:
6: Compute a priori estimate: $\hat{\mathbf{x}}_k^- = \sum \mathbf{W_i}^{(m)} \boldsymbol{\chi_k}$
7: Compute a priori covariance: $\mathbf{P}_k^- = \sum \mathbf{W_i}^{(c)} (\boldsymbol{\chi_k} - \hat{\mathbf{x}}_k^-)(\boldsymbol{\chi_k} - \hat{\mathbf{x}}_k^-)^\top$
8: Compute sigma points in measurement space: $\boldsymbol{\gamma_k} = \mathbf{h}(\boldsymbol{\chi_k})$
9: Compute predicted measurement mean: $\hat{\mathbf{y}}_k = \sum \mathbf{W_i}^{(m)} \boldsymbol{\gamma_k}$
10: Update Step:
11: Compute measurement covariance: $\mathbf{P}_{yy,k} = \sum \mathbf{W_i}^{(c)} (\boldsymbol{\gamma_k} - \hat{\mathbf{y}}_k)(\boldsymbol{\gamma_k} - \hat{\mathbf{y}}_k)^\top + \mathbf{R}$
12: Compute cross-covariance: $\mathbf{P}_{xy,k} = \sum \mathbf{W_i}^{(c)} (\boldsymbol{\chi_k} - \hat{\mathbf{x}}_k^-)(\boldsymbol{\gamma_k} - \hat{\mathbf{y}}_k)^\top$
13: Compute Kalman gain: $\mathbf{K}k = \mathbf{P}xy, k\mathbf{P}_{yy,k}^{-1}$
14: Update state: $\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{y}_k - \hat{\mathbf{y}}_k)$
15: Update covariance: $\mathbf{P}^{+}k = \mathbf{P}_k^- - \mathbf{K}_k \mathbf{P}yy, k\mathbf{K}_k^\top$
16: Output: A posteriori estimate$\mathbf{x}_k^+$, and covariance $\mathbf{P}_k^+$ at time $t_k$
```