SGN – Assignment #2

Edoardo Mensi Weingrill, 244706

Exercise 1: Uncertainty propagation

You are asked to analyze the state uncertainty evolution along a transfer trajectory in the Planar Bicircular Restricted Four-Body Problem, obtained as optimal solution of the problem stated in Section 3.1 (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. In the first point, initial mean and covariance have been propagated along a reference trajectory using two different approaches: Linearized Covariance (LinCov) and Unscented Transform (UT).

   ***LinCov*:** Through numerical integration of the Planar Bicircular Restricted Four-Body Problem (PBR4BP) dynamics, the Linear Covariance (LinCov) approach has been used to propagate the initial state $\mathbf{x_0}$ and the corresponding covariance matrix $\mathbf{P_0}$ within a time grid from t_0 to t_f. The variational approach has been used to calculate the State Transition Matrix (STM or $\mathbf{\Phi}(t_0, t_i)$) at each time-step. Next, the covariance matrix and state vector are updated through:

   $$\hat{\mathbf{x}}_{LinCov}(t_i) = \boldsymbol{\varphi}(\mathbf{x_0}, t_0, t_i) \qquad \mathbf{P}(t_i) = \mathbf{\Phi}(t_0, t_i)\mathbf{P_0}\mathbf{\Phi}(t_0, t_i)^{\mathsf{T}}$$

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

   ***UT*:** Using the Unscented Transform (UT) technique, the initial state vector $\mathbf{x_0}$ and its associated covariance matrix $\mathbf{P_0}$ are propagated forward in time. A set of sigma points ($\boldsymbol{\chi}_i$) is created around the initial state assuming a Gaussian distribution. These sigma points are then propagated by numerically integrating the Planar Bicircular Restricted Four-Body Problem (PBR4BP) dynamics over the time grid . Afterwards, weighted averages of the propagated sigma points ($\boldsymbol{\gamma} = f(\boldsymbol{\chi})$) are used to recalculate the mean and covariance:

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

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

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

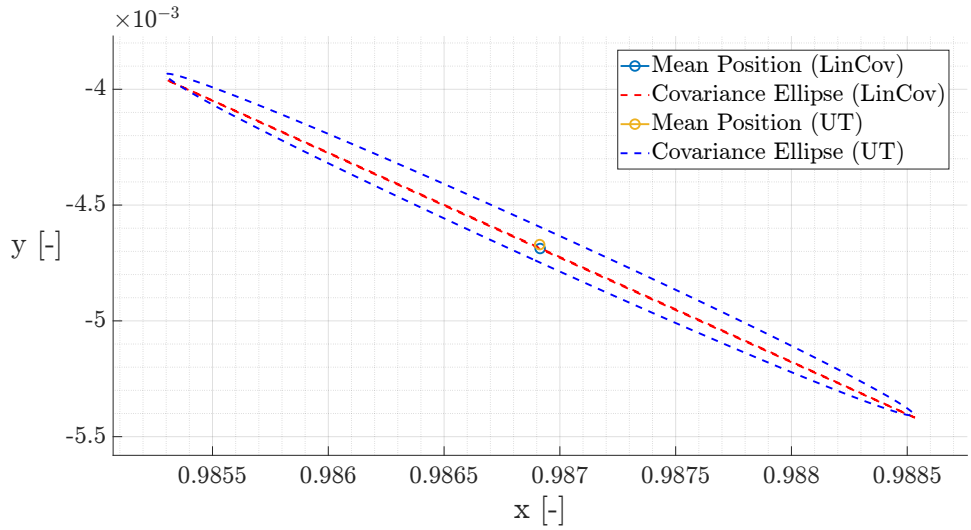

Figure 1: Comparison between LinCov and UT methods @(EMB) Earth-Moon Rotating Frame

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

approximate uncertainties effectively, as it is capable of accounting them in the primary direction of propagation. However, UT remains the more accurate method for uncertainty propagation.
The flattened covariance ellipse in the LinCov method may result from the linear approximation of the dynamics, which does not fully capture the system's non-linearities, leading to a distorted, more elongated or near-linear covariance shape.

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

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

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

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

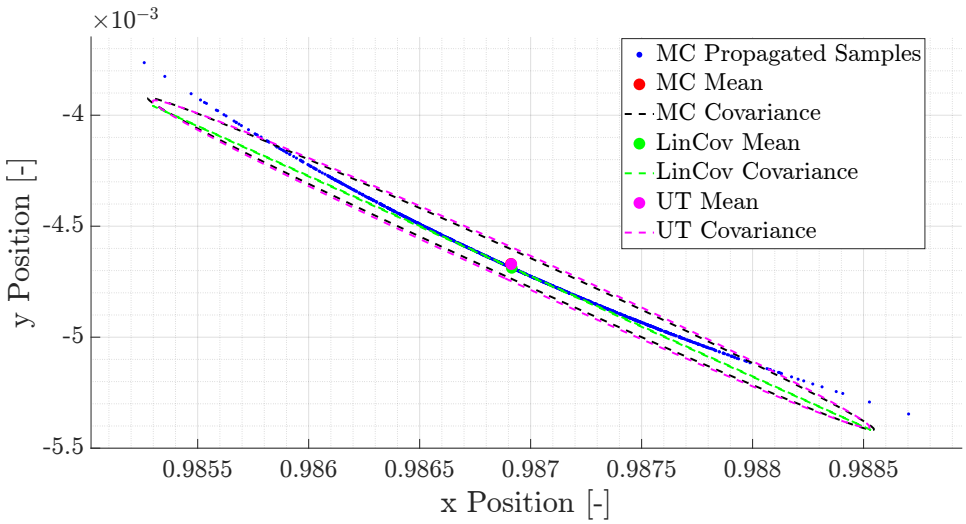

Figure 2: Comparison between the methods @(EMB) Earth-Moon Rotating Frame

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

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

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

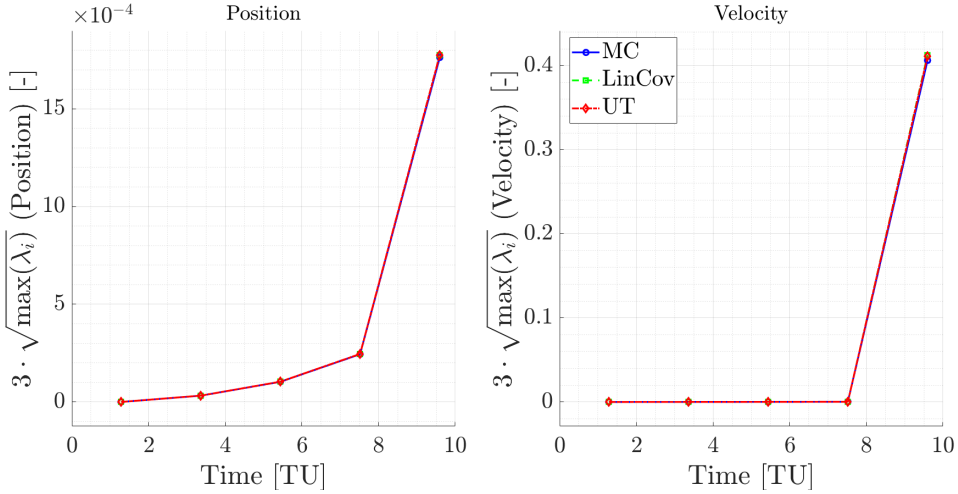

Figure 3: Maximum square root of the eigenvalues of the position and velocity covariance matrix over time, calculated using three different methods: LinCov, UT, and Monte Carlo.

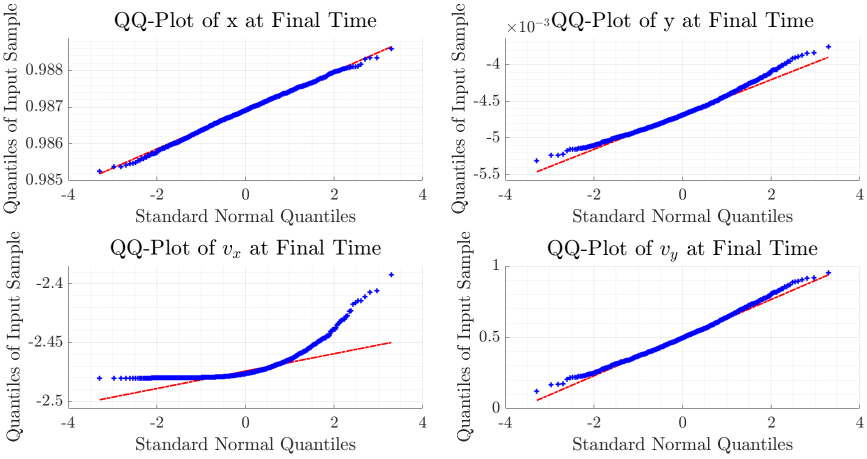

Figure 4: QQ-Plots for each state component (x, y, v_x, v_y) at the final time. These plots compare the theoretical quantiles with the sample quantiles to assess the normality of the propagated distributions.

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

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

Exercise 2: Batch filters

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

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

1. *Compute visibility windows.* The Two-Line Elements (TLE) set of SMOS are reported in Table 3 (and in WeBeep as 36036.3le). Compute the osculating state from the TLE at the reference epoch t_{ref}, then propagate this state assuming Keplerian motion to predict the trajectory of the satellite and compute all the visibility time windows from the available stations in the time interval from $t_0 =$ `2024-11-18T20:30:00.000` (UTC) to $t_f =$ `2024-11-18T22:15:00.000` (UTC). Consider the different time grid for each station depending on the frequency of measurement acquisition. Report the resulting visibility windows and plot the predicted Azimuth and Elevation profiles within these time intervals.

2. *Simulate measurements.* Use SGP4 and the provided TLE to simulate the measurements acquired by the sensor network in Table 2 by:

   (a) Computing the spacecraft position over the visibility windows identified in Point 1 and deriving the associated expected measurements.

   (b) Simulating the measurements by adding a random error to the expected measurements (assume a Gaussian model to generate the random error, with noise provided in Table 2). Discard any measurements (i.e., after applying the noise) that does not fulfill the visibility condition for the considered station.

3. *Solve the navigation problem.* Using the measurements simulated at the previous point:

   (a) Find the least squares (minimum variance) solution to the navigation problem without a priori information using
   - the epoch t_0 as reference epoch;
   - the reference state as the state derived from the TLE set in Table 3 at the reference epoch;
   - the simulated measurements obtained for the KOROU ground station only;
   - pure Keplerian motion to model the spacecraft dynamics.

   (b) Repeat step 3a by using all simulated measurements from the three ground stations.

   (c) Repeat step 3b by using a J2-perturbed motion to model the spacecraft dynamics.

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

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

[‡]Not just estimated state or covariance

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

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

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

Station name	KOUROU	TROLL	SVALBARD
Coordinates	LAT = 5.25144° LON = -52.80466° ALT = -14.67 m	LAT = -72.011977° LON = 2.536103° ALT = 1298 m	LAT = 78.229772° LON = 15.407786° ALT = 458 m
Type	Radar (monostatic)	Radar (monostatic)	Radar (monostatic)
Measurements type	Az, El [deg] Range (one-way) [km]	Az, El [deg] Range (one-way) [km]	Az, El [deg] Range (one-way) [km]
Measurements noise (diagonal noise matrix R)	$\sigma_{Az,El}$ = 125 mdeg 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
```

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

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

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

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

Table 5: Visibility windows for SMOS tracking.

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

Concurrently, for each ground station the Visibility Window plot is shown on the left, where the time evolution over the total time interval of azimuth and elevation is presented. The highlighted sections in the plot represent the periods during which the satellite is visible, defined by elevation exceeding the station's minimum threshold.
The plot on the right displays the satellite's trajectory in terms of azimuth and elevation. This plot represents the satellite's path as it moves throughout the station's field of view, starting at low elevation, reaching a peak, and descending back towards the horizon. Only the visible portions of the trajectory, where elevation exceeds the threshold, are displayed.

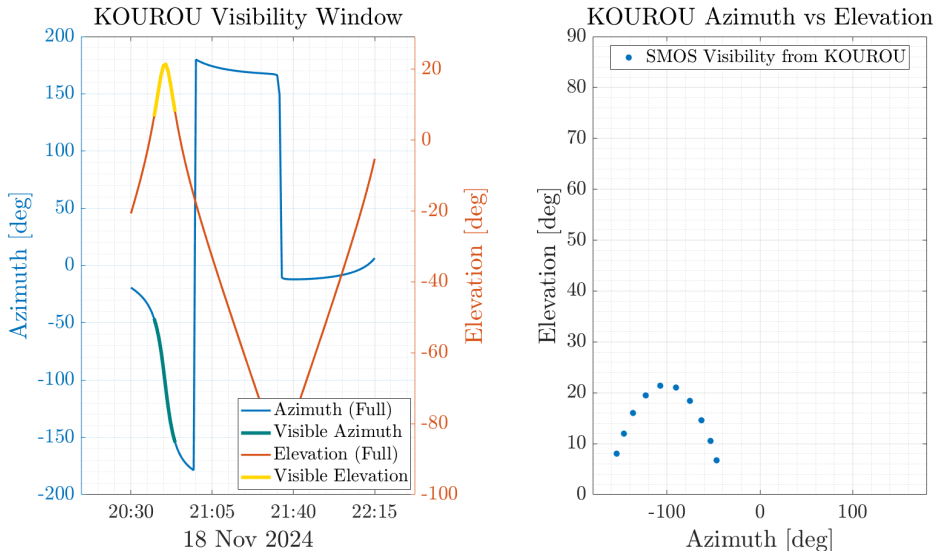

Figure 5: Combined plots for KOUROU station.

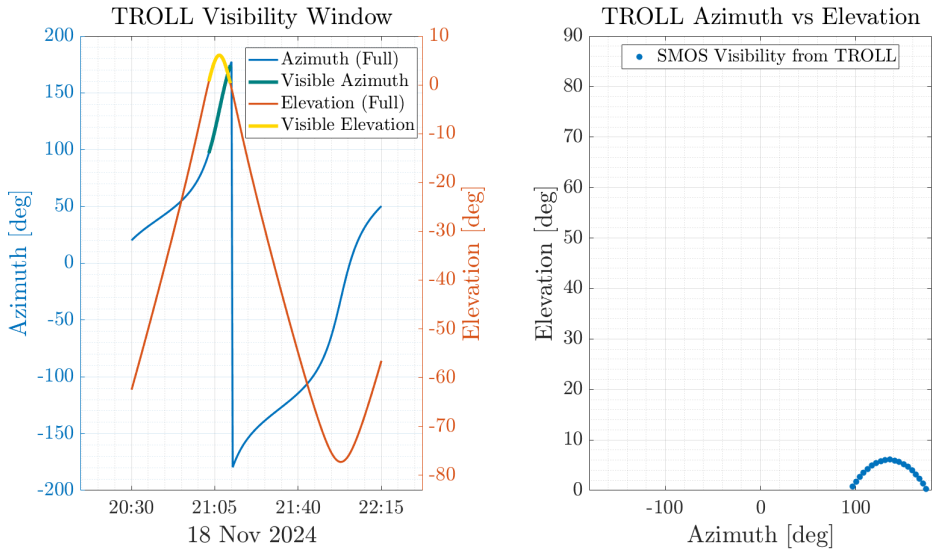

Figure 6: Combined plots for TROLL station.

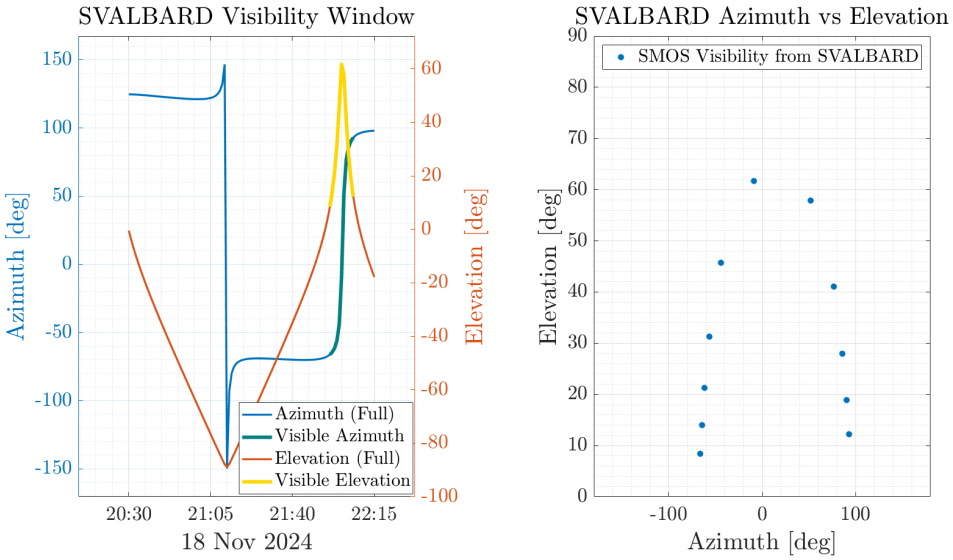

Figure 7: Combined plots for SVALBARD station.

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

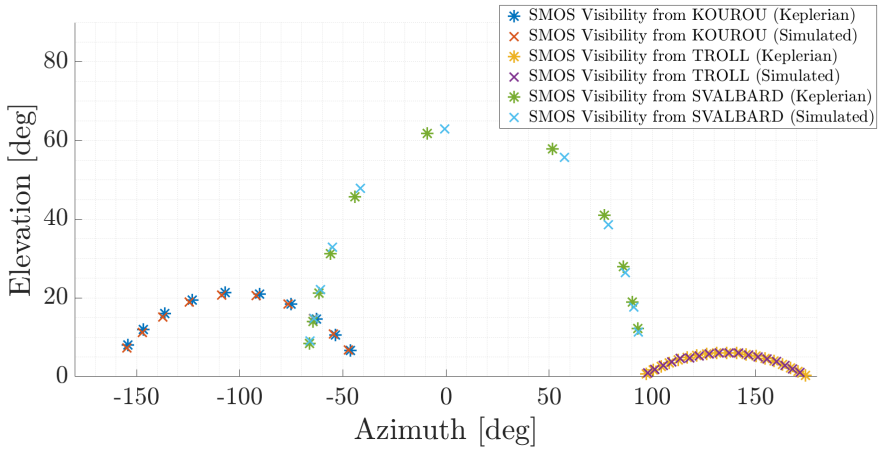

Figure 8: Comparison of visibility windows computed using the Keplerian propagator and noisy simulated measurements propagated with SGP4.

3. In point three the navigation problem is solved under three different configurations: (a) using measurements from Kourou only, (b) using measurements from all three ground stations, and (c) incorporating the J_2-perturbed motion model. The outputs for each point include the navigation solution at time t_0, comprehending the state in Cartesian coordinates and covariance matrix, the square root of the trace of the covariance submatrices for position and velocity, and the standard deviations for the semi-major axis and inclination, computed via a Jacobian transformation to Keplerian elements. The procedure adopted is detailed in Appendix 3.
   In all cases, the navigation problem was solved using least-squares optimization via MATLAB's `lsqnonlin` function, employing the `levenberg-marquardt` algorithm, which is particularly effective when the initial guess is far from the true solution.

   (a) Using Kourou Measurements Only

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

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

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

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

$$\mathbf{P_0} = \begin{bmatrix} 6.6220\text{e+}01 & 2.0189\text{e+}01 & -8.7122\text{e+}00 & -6.7000\text{e-}02 & -3.3100\text{e-}02 & -3.7200\text{e-}02 \\ 2.0189\text{e+}01 & 3.1027\text{e+}01 & -1.9742\text{e+}01 & -1.2000\text{e-}02 & -2.5100\text{e-}02 & -1.6000\text{e-}03 \\ -8.7122\text{e+}00 & -1.9742\text{e+}01 & 1.3159\text{e+}01 & 3.1000\text{e-}03 & 1.4100\text{e-}02 & -2.1000\text{e-}03 \\ -6.7000\text{e-}02 & -1.2000\text{e-}02 & 3.1000\text{e-}03 & 7.0720\text{e-}05 & 2.8383\text{e-}05 & 4.0896\text{e-}05 \\ -3.3100\text{e-}02 & -2.5100\text{e-}02 & 1.4100\text{e-}02 & 2.8383\text{e-}05 & 2.6579\text{e-}05 & 1.3190\text{e-}05 \\ -3.7200\text{e-}02 & -1.6000\text{e-}03 & -2.1000\text{e-}03 & 4.0896\text{e-}05 & 1.3190\text{e-}05 & 2.4923\text{e-}05 \end{bmatrix}$$

Table 7: Covariance Matrix at t_0 in @(Earth) ECI frame using Kourou only. Units: $[km^2, km^2/s, km^2/s^2]$

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

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

(b) Using Measurements from All Ground Stations

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

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

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

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

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

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

$$\mathbf{P_0} = \begin{bmatrix} 5.6700\text{e-}01 & -3.1070\text{e-}01 & -7.0200\text{e-}02 & -6.0000\text{e-}04 & 2.0000\text{e-}04 & -3.0000\text{e-}04 \\ -3.1070\text{e-}01 & 2.9660\text{e-}01 & 4.8900\text{e-}02 & 4.0000\text{e-}04 & -1.0000\text{e-}04 & 2.0000\text{e-}04 \\ -7.0200\text{e-}02 & 4.8900\text{e-}02 & 6.4700\text{e-}02 & 3.0000\text{e-}05 & -3.0000\text{e-}05 & 7.0000\text{e-}05 \\ -6.0000\text{e-}04 & 4.0000\text{e-}04 & 3.0000\text{e-}05 & 6.3200\text{e-}07 & -1.6600\text{e-}07 & 3.1500\text{e-}07 \\ 2.0000\text{e-}04 & -1.0000\text{e-}04 & -3.0000\text{e-}05 & -1.6600\text{e-}07 & 1.2100\text{e-}07 & -1.2400\text{e-}07 \\ -3.0000\text{e-}04 & 2.0000\text{e-}04 & 7.0000\text{e-}05 & 3.1500\text{e-}07 & -1.2400\text{e-}07 & 2.2600\text{e-}07 \end{bmatrix}$$

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

(c) Using J_2-Perturbed Motion

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

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

Table 10: Navigation Solution at time t_0 in @(Earth) ECI frame, using measurements from all the ground stations and J2-perturbed dynamics.

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

$$\mathbf{P_0} = \begin{bmatrix} 3.2000\text{e-}04 & -1.6000\text{e-}04 & -3.9300\text{e-}05 & -3.0900\text{e-}07 & 1.1300\text{e-}07 & -1.9300\text{e-}07 \\ -1.6000\text{e-}04 & 1.7000\text{e-}04 & 3.1600\text{e-}05 & 1.9600\text{e-}07 & -4.6300\text{e-}08 & 1.3200\text{e-}07 \\ -3.9300\text{e-}05 & 3.1600\text{e-}05 & 4.1100\text{e-}05 & 1.5100\text{e-}08 & -1.2100\text{e-}08 & 4.5800\text{e-}08 \\ -3.0900\text{e-}07 & 1.9600\text{e-}07 & 1.5100\text{e-}08 & 3.5700\text{e-}10 & -8.8900\text{e-}11 & 1.9300\text{e-}10 \\ 1.1300\text{e-}07 & -4.6300\text{e-}08 & -1.2100\text{e-}08 & -8.8900\text{e-}11 & 6.8600\text{e-}11 & -6.8800\text{e-}11 \\ -1.9300\text{e-}07 & 1.3200\text{e-}07 & 4.5800\text{e-}08 & 1.9300\text{e-}10 & -6.8800\text{e-}11 & 1.4900\text{e-}10 \end{bmatrix}$$

Table 11: Covariance Matrix at t_0 in @(Earth) ECI frame using measurements from all the ground stations and J2-perturbed dynamics. Units: $[km^2, km^2/s, km^2/s^2]$

The results obtained demonstrate how different configurations of ground station measurements and dynamic models impact the accuracy of the satellite's navigation solution.
In the first case, using only Kourou station, the uncertainties in position and velocity estimation are significantly higher, as can be noted in the covariance analysis. Since SMOS is in a LEO polar orbit, it passes minimal time over the equatorial region where Kourou is located. The limited visibility and reduced observation time window lead to increased errors in the estimation of the satellite's position and velocity, as they limit the variety and volume of measurements. Progressively introducing observations from Troll and Svalbard, followed by the inclusion of J2 perturbations in the dynamic model, results in a significant reduction in position and velocity uncertainties for the SMOS satellite.

This improvement is demonstrated in Table 8 and Table 10. The overall observation time frame is increased by combining the observation periods of the three ground stations, resulting in a more comprehensive coverage of the SMOS satellite's trajectory. The inclusion of the J2-perturbed model in the dynamics captures the perturbative effects of Earth's oblateness, providing a more accurate representation of the satellite's real trajectory. The secular nature of the J2 perturbation's effects would make its impact even more important if a longer time window were taken into account.
In this way, the residuals between the predicted measurements and the actual ones (which are the simulated measurements computed in Ex. 2.2b) are further reduced, improving the precision in position and velocity estimation.

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

Table 12: Trade-off analysis results

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

Analyzing the results, the second combination provides the highest accuracy in semi-major axis estimation, while the third combination achieves the best accuracy for inclination estimation. However, these results are not absolute. Due to the stochastic nature of the process, influenced by random noise in the measurements, the outcomes may vary with each iteration. By repeatedly running the procedure, it is observed that the second combination consistently offers the best accuracy for semi-major axis estimation, while for inclination estimation, this occurs frequently but not always.
Hence, the optimal configuration coming from this trade-off analysis consists of Kourou and Svalbard stations. Its effectiveness is due to the complementary geographical locations of the two stations. In the time window selected, from t_0 to t_f, SMOS passes directly over Kourou, increasing the accuracy of its measurements. Meanwhile, Svalbard's high-latitude location ensures excellent coverage in polar regions, which is particularly advantageous for a polar orbit like that of SMOS. This combination, through its geometric diversity, enhances the observability of the satellite's state, with Kourou contributing to along-track precision during low-latitude passes and Svalbard to cross-track accuracy in high-latitude passes. Indeed, Kourou-Troll combination performs worse as Troll is located in the southern hemisphere with a lower latitude compared to Svalbard station. In addition, as can be seen in Fig. 8, Svalbard complements Kourou more effectively than Troll, enhancing SMOS' coverage in the given observation campaign. Moreover, this combination is the most cost-effective, with an overall cost of 65.000€, remaining within the budget.
In conclusion, the combination of Kourou and Svalbard stations guarantees the best accuracy for state estimation while minimizing the costs.

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

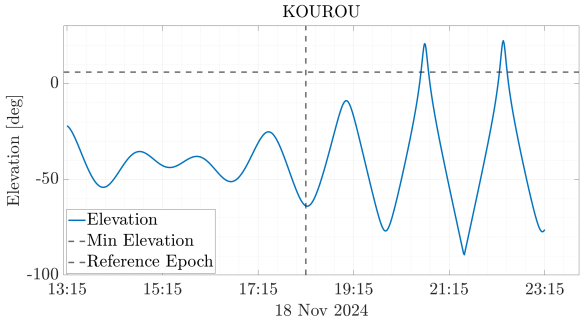

Figure 9: Elevation profile as observed from Kourou over the extended time window.

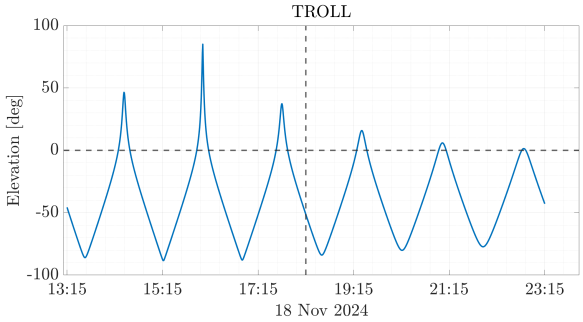

Figure 10: Elevation profile as observed from Troll over the extended time window.

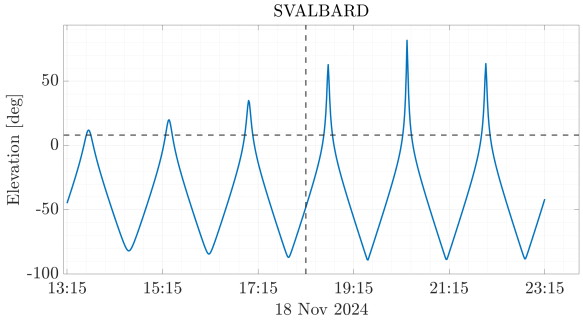

Figure 11: Elevation profile as observed from Svalbard over the extended time window.

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

Table 13: Extended visibility window for each ground station

In the figures above, the evolution of SMOS' elevation over the extended time interval for each ground station is shown, together with the SMOS' minimum elevation needed to be visible from the ground station. As remarked in Table 13, Svalbard and Troll provide noticeably higher visibility opportunities than Kourou during the extended time window. Kourou serves as a backup during equatorial runs, whereas Troll and Svalbard are selected as the principal ground stations because of the focus on the repeatability of passes and the reliability of orbit determination. The rationale behind this choice is based on SMOS' polar orbit and the geographical position of Troll (72°S) and Svalbard (78°N), which ensures excellent repeatability by providing extensive and frequent coverage in polar regions, where SMOS spends the majority of its orbital period. In this way, these two stations complement each other, allowing a complete coverage of high-latitude regions. Moreover, the high density of measurements achieved enhances the precision of orbital parameters determination, in particular inclination and cross-track components.

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

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

Exercise 3: Sequential filters

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

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

1. *Check the visibility window.* Considering the initial state $\mathbf{x}_0$ and the time interval with a time-step of 30 seconds from t_0 to t_f reported in Table 14, predict the trajectory of the satellite in an inertial Moon-centered reference frame assuming Keplerian motion. Use the estimated coordinates given in Table 15 to predict the state of the lunar lander. Finally, check that the lander and the orbiter are in relative visibility for the entire time interval.

2. *Simulate measurements.* Always assuming Keplerian motion to model the lunar orbiter dynamics around the Moon, compute the time evolution of its position vector in an inertial Moon-centered reference frame and the time evolution of the relative range between the satellite and the lunar lander. Finally, simulate the measurements by adding a random error to the spacecraft position vector and to the relative range. Assume a Gaussian model to generate the random error, with noise provided in Table 14 for both the relative range and the components of the position vector. Verify (graphically) that the applied noise level is within the desired boundary.

3. *Estimate the lunar orbiter absolute state.* As a first step, you are asked to develop a sequential filter to narrow down the uncertainty on the knowledge of the lunar orbiter absolute state vector. To this aim, you can exploit the measurements of the components of its position vector computed at the previous point. Using an Unscented Kalman Filter (UKF), provide an estimate of the spacecraft state (in terms of mean and covariance) by sequentially processing the acquired measurements in chronological order. To initialize the filter in terms of initial covariance, you can refer to the first six elements of the initial covariance $\mathbf{P}_0$ reported in Table 14. For the initial state, you can perturb the actual initial state $\mathbf{x}_0$ by exploiting the MATLAB function `mvnrnd` and the previously mentioned initial covariance. We suggest to use $\alpha = 0.01$ and $\beta = 2$ for tuning the UT in this case. Plot the time evolution of the error estimate together with the 3σ of the estimated covariance for both position and velocity.

4. *Estimate the lunar lander coordinates.* To fulfill the goal of your mission, you are asked to develop a sequential filter to narrow down the uncertainty on the knowledge of the lunar lander coordinates (considering a fixed zero altitude). To this aim, you can exploit the measurements of the components of the lunar orbiter position vector together with the measurements of the relative range between the orbiter and the lander computed at the previous point. Using an UKF, provide an estimate of the spacecraft state and the lunar lander coordinates (in terms of mean and covariance) by sequentially processing the acquired measurements in chronological order. To initialize the filter in terms of initial covariance, you can refer to the initial covariance $\mathbf{P}_0$ reported in Table 14. For the initial state, you can perturb the actual initial state, composed by $\mathbf{x}_0$ and the latitude

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

Table 14: Initial conditions for the lunar orbiter.

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

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

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

1. In the first point, starting from the initial state $\mathbf{x_0}$ at time t_0 and assuming an unperturbed Keplerian motion, the trajectory of the satellite has been predicted by numerically integrating the unperturbed dynamics of the 2-Body Problem over the assigned time interval from t_0 to t_f. The predicted trajectory in Moon-Centered Inertial (MCI) reference frame is reported in Fig. 12, overlaying the Moon's surface for context.

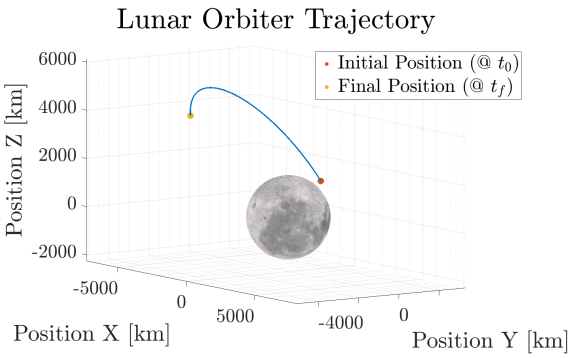

Figure 12: Lunar Orbiter Trajectory @(Moon) Moon-Centered Inertial frame

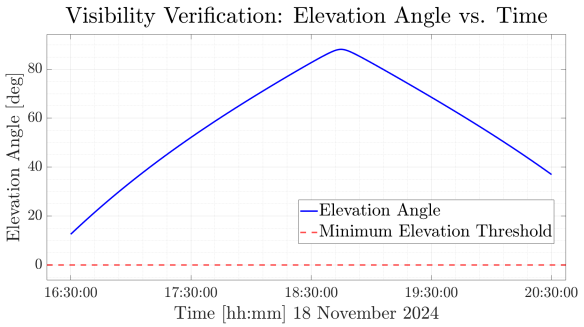

Figure 13: Elevation profile of Lunar Orbiter as observed from Lunar Lander over the time interval.

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

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

The state of the Lunar Lander was predicted starting from the given coordinates and then transformed to Cartesian coordinates in the Moon-Centered Moon-Fixed (MCMF) reference frame through the MATLAB function `cspice_latrec`, and subsequently rotated in the MCI reference frame. In Table 16 the predicted state at t_0 is reported.
Knowing the propagated position of the lunar orbiter and the predicted position of the lunar lander, the relative position of the orbiter with respect to the lander in the MCI frame was computed. After rotating the relative position in the Lunar Lander Topocentric reference frame, range, azimuth and elevation were retrieved using `cspice_xmfsta` function.
Continuous orbiter visibility from the lander was ensured by verifying the relative visibility during the entire time interval. As can be seen in Fig. 13, the calculated elevation angles continuously met the criterion, indicating no visibility interruptions.

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

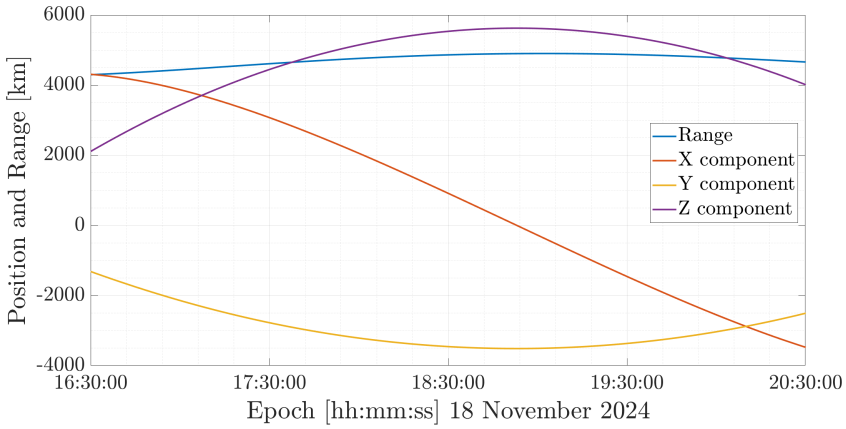

Figure 14: Time evolution of lunar orbiter's position and relative range between orbiter and lander over the selected time interval

Next step consists in simulating real sensor data by adding a random error to the measurements. Noise with a standard deviation of $\sigma_p = 100m$ was applied to the components of the orbiter's position vector in the Moon-Centered Inertial (MCI) reference frame and the relative range. The noise was generated using a multivariate Gaussian random distribution through `mvnrnd` MATLAB function to accurately simulate sensor uncertainties.
To validate Gaussian distribution assumption, in Fig. 17 are reported the errors between the noisy and the true measurements compared with the $\pm 3\sigma$ boundaries. Since the simulated measurements consistently remain within these boundaries, the assumption of Gaussian distribution is validated, and the simulated measurements are therefore suitable for use in subsequent filtering and estimation tasks.

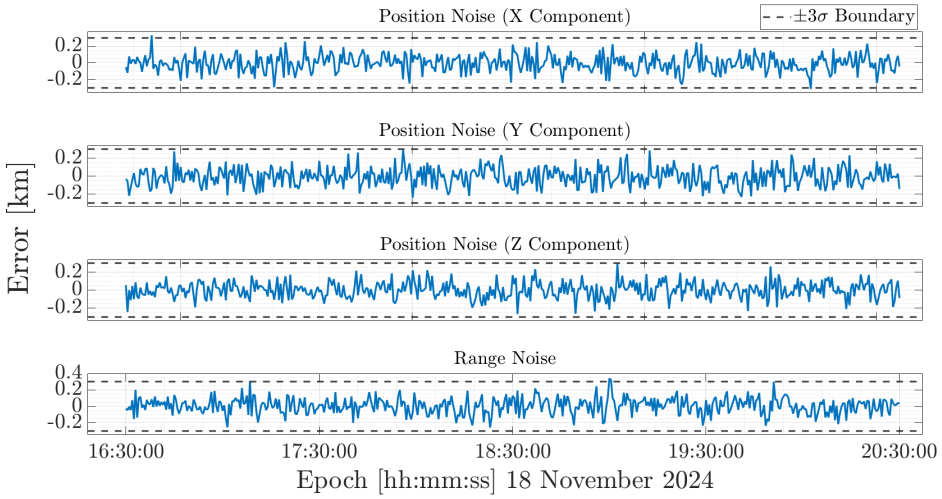

Figure 15: Comparison of the errors between noisy and true measurements for the three position components and relative range with the $\pm 3\sigma$ boundaries.

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

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

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

Algorithm 1 `Unscented Kalman Filter (UKF) for Lunar Orbiter`

Input: Initial state $\mathbf{x}_0$, Initial covariance $\mathbf{P}_0$, Time span t_{span}, Simulated noisy measurements, Measurement noise covariance $\mathbf{R}$, UKF parameters α, β, λ
Output: Estimated state $\mathbf{x}_k^+$, Covariance History$\mathbf{P}_k^+$
Compute weights $W_i^{(m)}$, $W_i^{(c)}$ based on UKF parameters
for each time step k from 1 to length(t_{span}) - 1 **do**
  Prediction Step:
  Generate sigma points $\boldsymbol{\chi}_{k-1}$ from $\hat{\mathbf{x}}_{k-1}^+$ and $\mathbf{P}_{k-1}^+$
  Propagate sigma points using Keplerian dynamics: $\boldsymbol{\chi}_k$
  Compute predicted state mean: $\hat{\mathbf{x}}_k^-$ using $W_i^{(m)}$
  Compute predicted covariance: $\mathbf{P}_k^-$ using $W_i^{(c)}$
  Measurement Update:
  Transform sigma points to measurement space: $\boldsymbol{\gamma}_k$ by extracting position components
  Compute predicted measurement mean: $\hat{\mathbf{y}}_k^-$ using $W_i^{(m)}$
  Compute measurement covariance: $\mathbf{P}_{yy,k} = \mathbf{P}_{ee,k} + \mathbf{R}$
  Compute cross-covariance: $\mathbf{P}_{xy,k}$
  Compute Kalman gain: $\mathbf{K}_k = \mathbf{P}_{xy,k}\mathbf{P}_{yy,k}^{-1}$
  Update state using simulated noisy measurements: $\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{y}_k - \hat{\mathbf{y}}_k^-)$
  Update covariance: $\mathbf{P}_k^+ = \mathbf{P}_k^- - \mathbf{K}_k\mathbf{P}_{yy,k}\mathbf{K}_k^\top$
end for

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

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

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

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

Table 18: Covariance Matrix at t_f @(Moon) MCI frame. Units: $[km^2, km^2/s, km^2/s^2]$

To assess the performance of the UKF in estimating the lunar orbiter's position and velocity, two different errors have been computed.
The position error was defined as the norm of the difference between the true propagated position and the UKF-estimated position. Similarly, the velocity error was computed as the norm of the difference between the true propagated velocity and the UKF-estimated one. These two errors clearly assess the UKF's true effectiveness in estimating the lunar orbiter's state.
Then, from the diagonal elements of the corresponding submatrix, the 3σ bounds for both position and velocity were derived. These bounds are necessary to validate the filter's effectiveness, as they represent the uncertainty in the state estimate.
In Fig. 16, the time evolution of position and velocity errors along with their corresponding 3σ boundaries is shown. This visualization helps in verifying whether the estimation errors remain within the statistical confidence intervals predicted by the UKF.

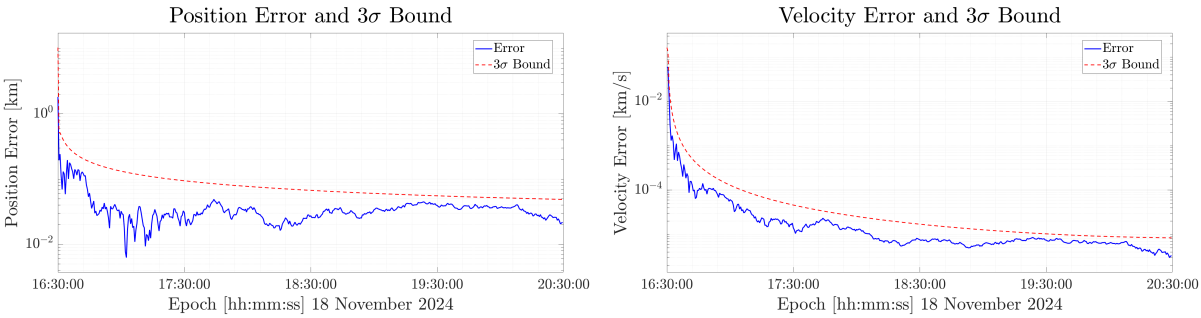

Figure 16: Evolution of the Lunar Orbiter's position and velocity errors over the selected time interval, along with the corresponding 3σ boundaries.

The efficiency of the UKF is evident from the generated plots for position and velocity errors. The perturbed initial state causes the errors to be relatively high at first, but position and velocity errors converge as the filter processes more samples. The errors stay well inside the 3σ boundaries throughout the simulation, confirming the Gaussian assumptions and showing that the filter is correctly tuned.
Because position estimations are more sensitive to measurement noise, the position error decreases more slowly than the velocity error. Since deterministic orbital dynamics dominate velocity propagation, the velocity error exhibits a faster convergence. By balancing the data from the dynamics model and the implemented UKF, these observations confirm its ability to produce accurate state estimations.

4. In the last point, an UKF has been implemented to reduce uncertainties in the knowledge of the lunar lander's coordinates exploiting the measurements of lunar orbiter position and of relative range between the orbiter and the lander. The approach implemented is similar to that used in Ex. 3.3, with some slight differences. The augmented initial state $\mathbf{x_0}$ now includes both the orbiter's initial state and the coordinates of the lander, resulting in a state vector of dimension $[8 \times 1]$. Analogously, the initial covariance matrix **P0** has been updated to $[8 \times 8]$.The relative range, which is dynamically calculated at each algorithmic time step, is now also included in the measurements. The complete method is described below.

Algorithm 2 `Unscented Kalman Filter (UKF) for Lunar Lander`

Input: Initial state $\mathbf{x}_0$ (orbiter state and lander coordinates), Initial covariance $\mathbf{P}_0$, Time span t_{span}, Simulated noisy measurements (orbiter position and relative range), Measurement noise covariance $\mathbf{R}$, UKF parameters α, β, λ
Output: Estimated state $\mathbf{x}_k^+$ (orbiter state and lander coordinates), Covariance $\mathbf{P}_k^+$
Compute weights $W_i^{(m)}$, $W_i^{(c)}$ based on UKF parameters
for each time step k from 1 to length(t_{span}) - 1 **do**
Prediction Step:
Same as algorithm 1.
Measurement Update:
Transform sigma points to measurement space:

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

Same as algorithm 1
end for

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

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

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

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

Table 20: Covariance Matrix at t_f @(Moon) MCI frame Units: $[km^2, km^2/s, km^2/s^2, km\ rad, km/s\ rad, rad^2]$

As in the previous point, to assess the performance of the UKF in estimating lunar orbiter's position and velocity and lunar lander coordinates, four different errors have been computed. Along with the position and velocity errors computed as before, latitude and longitude errors were computed as the absolute value ofthe difference between the true coordinates with the UKF-estimated ones.
In the plots presented hereafter are reported the time evolution of the errors along with their corresponding 3σ boundaries.

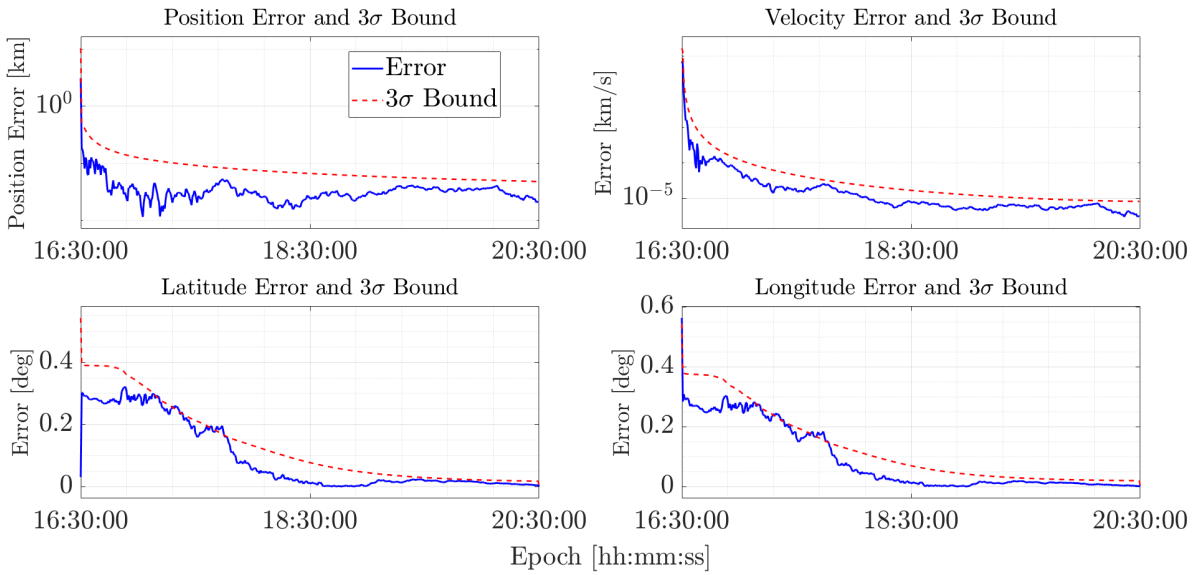

Figure 17: Evolution of the Lunar Orbiter's position and velocity errors and the Lunar Lander's coordinates over the selected time interval, along with the corresponding 3σ boundaries.

The first row presents the errors for the lunar orbiter's position and velocity. The UKF performances are satisfactory as the errors systematically falls below the 3σ boundary. Regarding the different trend, same considerations as Ex. 3.3 can be done.
The bottom row presents the errors in lander's coordinates estimation. It can be noticed they slightly exceed the 3σ bounds, particularly in the early stages of the simulation. This behaviour can be attributed to the non-linearities present in the measurement model adopted to estimate the relative range, along with the random nature of the generated noise. The UKF may not fully represent these non-linearities in the initial phase, where uncertainties are larger.
Despite these occasional violations, as the UKF incorporates additional observations and reduces the uncertainty in the state estimate, the latitude and longitude errors exhibit clear convergence. The filter's overall performance demonstrates its robustness and capability in determining the lander's coordinates based on the orbiter's location and relative range measurements. The efficiency of the UKF in reducing uncertainty within a non-linear system is further validated by the convergence of all errors over time toward their respective 3σ bounds.

Appendix A: Mapping Covariance from Cartesian State to Keplerian Elements

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

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

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

where:

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

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

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

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

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

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

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

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

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

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

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