# SGN – Assignment #2

Marcello Pareschi, 252142

## 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]<br>$\mathbf{v}_i$ = [10.718855256727338, 0.116502348513671] |
| Initial time $t_i$ | 1.282800225339865 |
| Final time $t_f$ | 9.595124551366348 |
| Covariance $\mathbf{P}_0$ | $\begin{bmatrix} +1.041e-15 & +6.026e-17 & +5.647e-16 & +4.577e-15 \\ +6.026e-17 & +4.287e-18 & +4.312e-17 & +1.855e-16 \\ +5.647e-16 & +4.312e-17 & +4.432e-16 & +1.455e-15 \\ +4.577e-15 & +1.855e-16 & +1.455e-15 & +2.822e-14 \end{bmatrix}$ |

*F. Topputo, "On optimal two-impulse Earth–Moon transfers in a four-body model", Celestial Mechanics and Dynamical Astronomy, Vol. 117, pp. 279–313, 2013, DOI: 10.1007/s10569-013-9513-8.

†Use at least 1000 samples drawn from the initial covariance

## 1.1 LinCov and UT Methods

Considering the Planar Bicircular Restricted Four Body Problem (PBRFBP) as the astrodynamical model (Appendix A), the uncertainty propagation was initially carried out using two different techniques: the LinCov and the UT methods

### 1.1.1 Linearized Approach (LinCov)

Using the LinCov method, the propagated mean coincides with the propagated reference state, while the propagated covariance matrix is computed with a linear mapping involving the initial covariance matrix $P_0$ and State Transition Matrix (STM) $\Phi(t_0, t_f)$:

$$\begin{cases} \hat{\mathbf{x}}(t) = \mathbf{x}^*(t), \\ P(t) = \Phi(t_0, t) P_0 \Phi(t_0, t)^T, \end{cases} \tag{1}$$

where $\hat{\mathbf{x}}(t)$ represents the propagated mean, and $P(t)$ is the propagated covariance matrix.

The initial reference state was propagated along with the STM by numerically integrating the PBRFBP equation of motion and the associated STM dynamics. The equations are provided Appendix A. The integration was performed using `ode78` with initial conditions $\mathbf{x}_0^* = [\mathbf{r}_i \;\; \mathbf{v}_i]^T$ and $\Phi_0 = \mathbf{I}_{4\times4}$. Both `RelTol` and `AbsTol` were set to $10^{-12}$, and a `linspace` of 5 elements from $t_i$ to $t_f$ was provided to the solver to obtain the solution at 5 equally spaced time instants.

### 1.1.2 Unscented Transform (UT)

In the UT method the initial probability density function (pdf) is parametrized by a small number of samples, called $\sigma$-points, whose sample mean and covariance are representative of the mean and covariance of the initial pdf. The $\sigma$-points are propagated integrating numerically the PBRFBP equation of motion and the final mean and covariance are computed as weighted sample mean and covariance of the propagated $\sigma$-points. The main steps are the following:

1. Use Cholesky decomposition to compute $\sqrt{(n+\lambda)P_0}$
2. Compute the sigma points $\boldsymbol{\chi}_i$:

$$\boldsymbol{\chi}_i = \begin{cases} \hat{\mathbf{x}}_0, & i = 0, \\ \hat{\mathbf{x}}_0 + \left(\sqrt{(n+\lambda)P_0}\right)_i, & i = 1, \dots, n, \\ \hat{\mathbf{x}}_0 - \left(\sqrt{(n+\lambda)P_0}\right)_i, & i = n+1, \dots, 2n. \end{cases}$$

3. Compute the weights $W_i^{(m)}$ and $W_i^{(c)}$:

$$\begin{aligned} W_0^{(m)} &= \lambda / (n+\lambda), \\ W_0^{(c)} &= \lambda / (n+\lambda) + (1 - \alpha^2 + \beta), \\ W_i^{(m)} &= W_i^{(c)} = 1 / [2(n+\lambda)] \quad \text{for } i = 1, \dots, 2n. \end{aligned}$$

4. Propagate the sigma points: $\mathbf{Y}_i = \varphi(\boldsymbol{\chi}_i, t_i, t_f) \quad i = 0, \dots, 2n$
5. Use the propagated sigma points and weights to compute $\hat{\mathbf{x}}(t)$ and $P(t)$ as

$$\hat{\mathbf{x}}(t) = \hat{\mathbf{Y}} = \sum_{i=0}^{2n} W_i^{(m)} \mathbf{Y}_i \qquad P(t) = P_y = \sum_{i=0}^{2n} W_i^{(c)} \left(\mathbf{Y}_i - \hat{\mathbf{Y}}\right)\left(\mathbf{Y}_i - \hat{\mathbf{Y}}\right)^T$$

Here $n = 4$ is the size of the initial mean $\hat{\mathbf{x}}_0$, which coincides with the initial reference state $\mathbf{x}_0^*$, $\lambda = n\left(\alpha^2 - 1\right)$ is the scaling parameter and the notation $\left(\sqrt{(n+\lambda)P_0}\right)_i$ indicates the $i$-th column of the matrix under brackets.

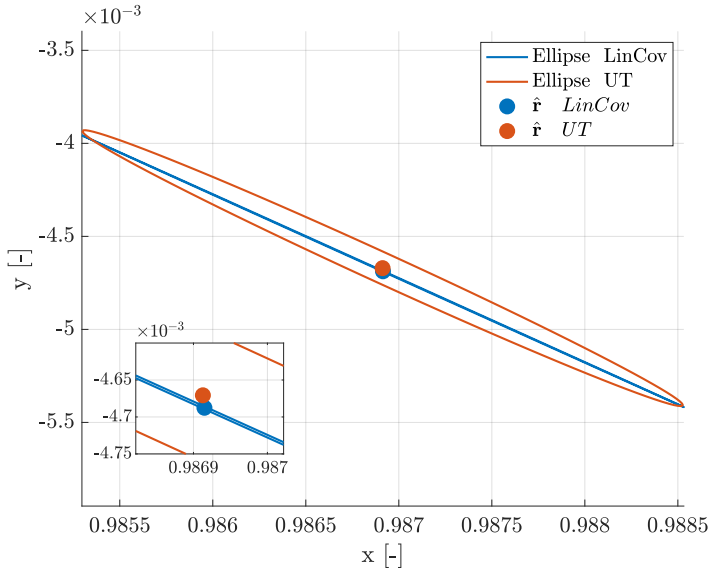



**Figure 1:** Mean and confidence ellipses for LinCov and UT methods in the Earth-Moon rotating frame

In Figure 1, the confidence ellipses of $P_r\left(t_f\right)$, the submatrix of $P\left(t_f\right)$ corresponding to the position, are depicted. These ellipses were obtained from $P_r\left(t_f\right)$ by performing Singular Value Decomposition (SVD) to extract the orthogonal matrix $R$ and the diagonal matrix $D$. The diagonal matrix $\boldsymbol{\sigma}$, containing the principal standard deviations, was computed as $\boldsymbol{\sigma} = \sqrt{\text{diag}(D)}$. Using $R$ and $\boldsymbol{\sigma}$, the unit circle centered at the origin was scaled and rotated to form the ellipse. Finally, the ellipse was translated to the mean vector and plotted. This procedure was repeated for both the LinCov and UT methods.

**Table 2:** Principal standard deviations

| **Method** | LinCov | UT |
|---|---|---|
| $\boldsymbol{\sigma_1}$ | 5.9183e-04 | 5.9134e-04 |
| $\boldsymbol{\sigma_2}$ | 5.4910e-07 | 2.7237e-05 |

From Figure 1, it can be observed that the confidence ellipse for the LinCov method is significantly compressed along one direction. This is due to the fact that, as shown in Table 2, $\sigma_1$ and $\sigma_2$ for the LinCov method differ by three orders of magnitude. Consequently, the size of the ellipse along the second principal direction is much smaller compared to the first.

## 1.2 Monte Carlo simulation and Comparison

The uncertainty was also propagated using the Monte Carlo method. In this approach, $N$ random samples were generated from the initial distribution, modeled as a multivariate normal distribution with mean $\hat{\mathbf{x}}_0 = [\mathbf{r}_i \quad \mathbf{v}_i]^T$ and covariance $P_0$. The MATLAB function `mvnrnd` was used to generate the samples. Each sample was propagated by numerically integrating the equations of the PBRFBP. At each time step, the sample mean and covariance were computed to characterize the propagated uncertainty. A total of $N = 1000$ samples was selected as a balance between computational efficiency and result accuracy.

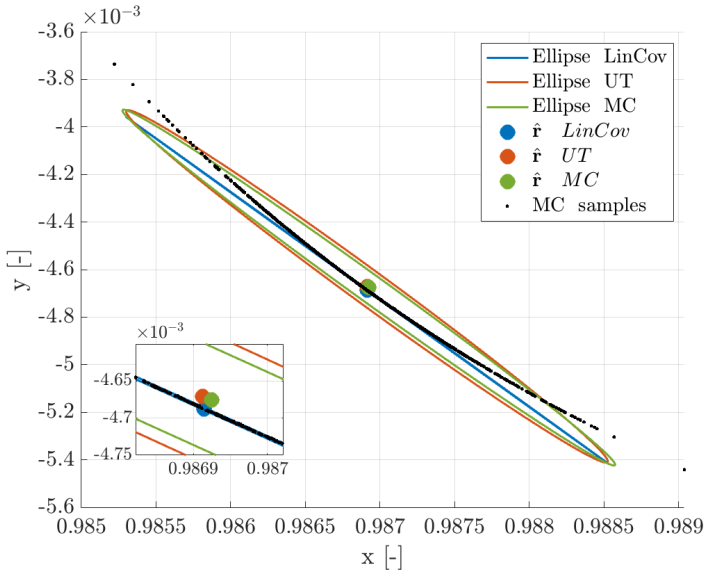
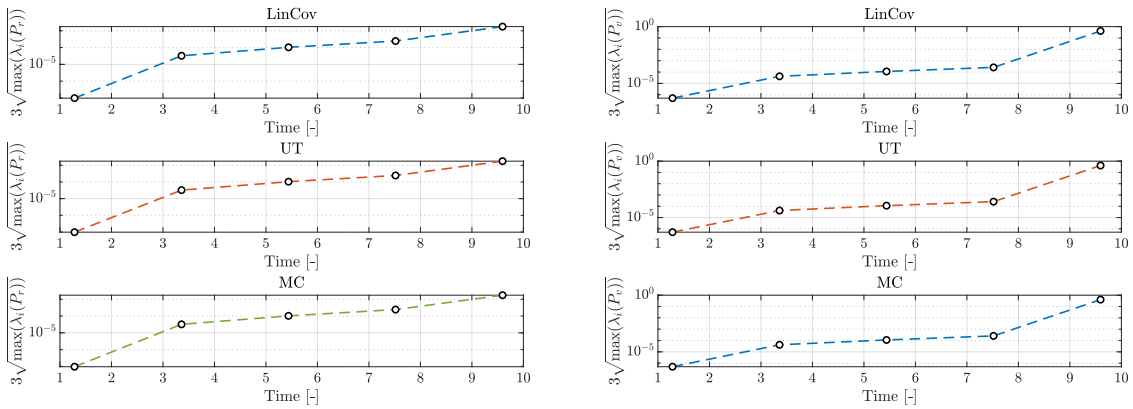



**Figure 2:** Mean and confidence ellipses for LinCov, UT and Monte Carlo methods in the Earth-Moon rotating frame

**Table 3:** Principal standard deviations

| **Method** | LinCov | UT | MC |
|---|---|---|---|
| $\boldsymbol{\sigma_1}$ | 5.9183e-04 | 5.9134e-04 | 6.0864e-04 |
| $\boldsymbol{\sigma_2}$ | 5.4910e-07 | 2.7237e-05 | 2.6516e-05 |

From Figure 2, it is evident that the UT method provides a more accurate representation of the uncertainty propagation obtained from the Monte Carlo simulation compared to the LinCov method. The reason is that LinCov is based on the assumption of linearity, which is inadequate for highly non linear problems, such as the PBRFBP, and long-term propagations. Moreover, Figure 2 shows that the Monte Carlo samples tend to form a *banana shape*, which deviates significantly from the characteristic shape of a bivariate Gaussian distribution.




Uncertainty on the position Uncertainty on the velocity

**Figure 3:** Time history of uncertainty

In Figure 3, the time history of the maximum uncertainty for both the position and the ve-

locity, measured as $3\sqrt{\max\left(\lambda_i\left(P_r\right)\right)}$ and $3\sqrt{\max\left(\lambda_i\left(P_v\right)\right)}$ respectively, is depicted for all three methods. The plots are presented in a semi-logarithmic scale for improved visualization. It can be observed that the uncertainty is strictly growing over time in all cases. The graphs are similar across all methods, indicating that all the three methods provide similar results in terms of maximum uncertainty. This can also be verified by examining Figure 2, as all three ellipses have comparable dimensions along the first principal direction (associated with the maximum eigenvalue). Additionally, it is evident that the uncertainty in velocity increases more rapidly than the uncertainty in position. At $t_i$, both $3\sqrt{\max\left(\lambda_i\left(P_r\right)\right)}$ and $3\sqrt{\max\left(\lambda_i\left(P_v\right)\right)}$ are on the order of $10^{-7}$; however, at $t_f$, the former is on the order of $10^{-3}$, while the latter reaches the order of $10^{-1}$. It is worth noting that $3\sqrt{\max\left(\lambda_i\left(P_r\right)\right)}$ and $3\sqrt{\max\left(\lambda_i\left(P_v\right)\right)}$ are non-dimensional quantities, making them directly comparable.

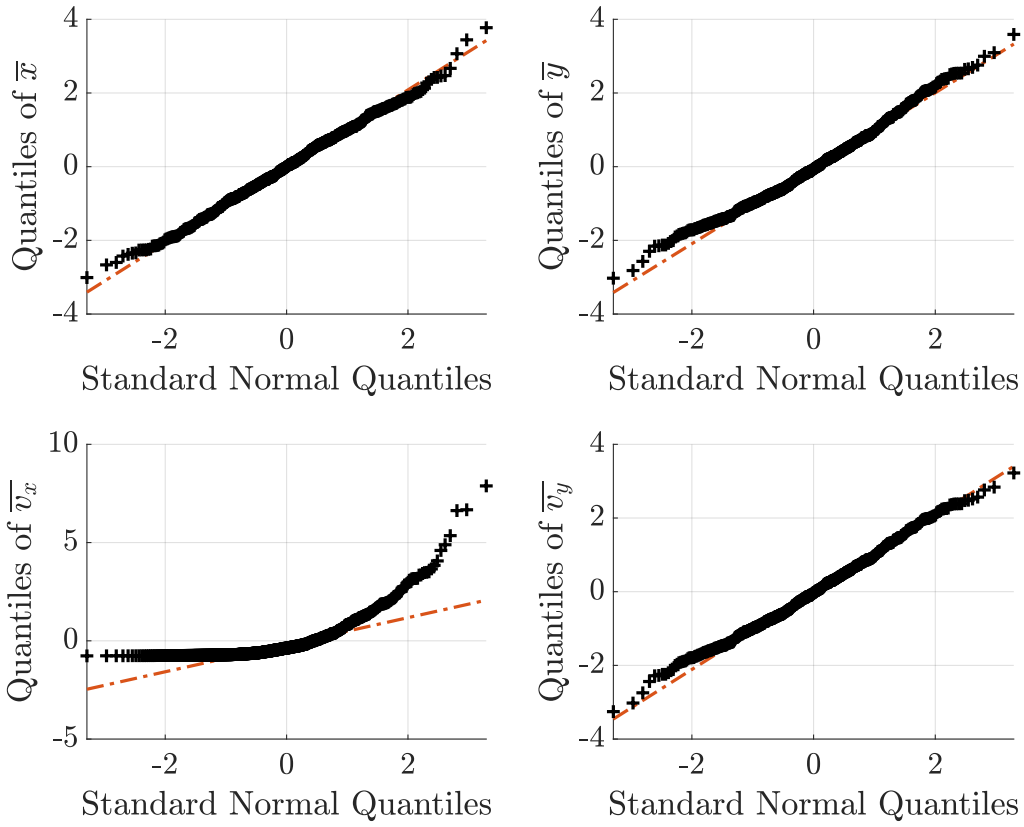



**Figure 4:** Mean and confidence ellipses for LinCov and UT methods

The qq-plots in Figure 4 illustrate the comparison between the quantiles of the standardized components of the state vector at the final time obtained from the Monte Carlo simulation and the quantiles of the Standard Normal Distribution (SND). The standardization of the data was performed as $\overline{\mathbf{x}}_i = \left(\mathbf{x}_i - \mu_i\right)/\sigma_i$, where $\mathbf{x}_i$ represents the $i$-th component of the state vector, and $\mu_i$ and $\sigma_i$ are its corresponding mean and standard deviation. If the standardized components $\overline{\mathbf{x}}_i$ follow a normal distribution, their quantiles should lie on the identity line $y = x$ in the qq-plots.

Looking at Figure 4, the normality hypothesis is confirmed for $x$, $y$, and $v_y$. However, the same cannot be said for $v_x$, where the data show a convex curve that significantly deviates from the $y = x$ line. In particular, both the positive and negative quantiles deviate upward from $y = x$, suggesting that the distribution of $v_x$ at the final time has a right tail that is heavier than that

of the normal distribution, while the left tail is lighter. This implies that the $v_x$ distribution at the final time is asymmetric [1].

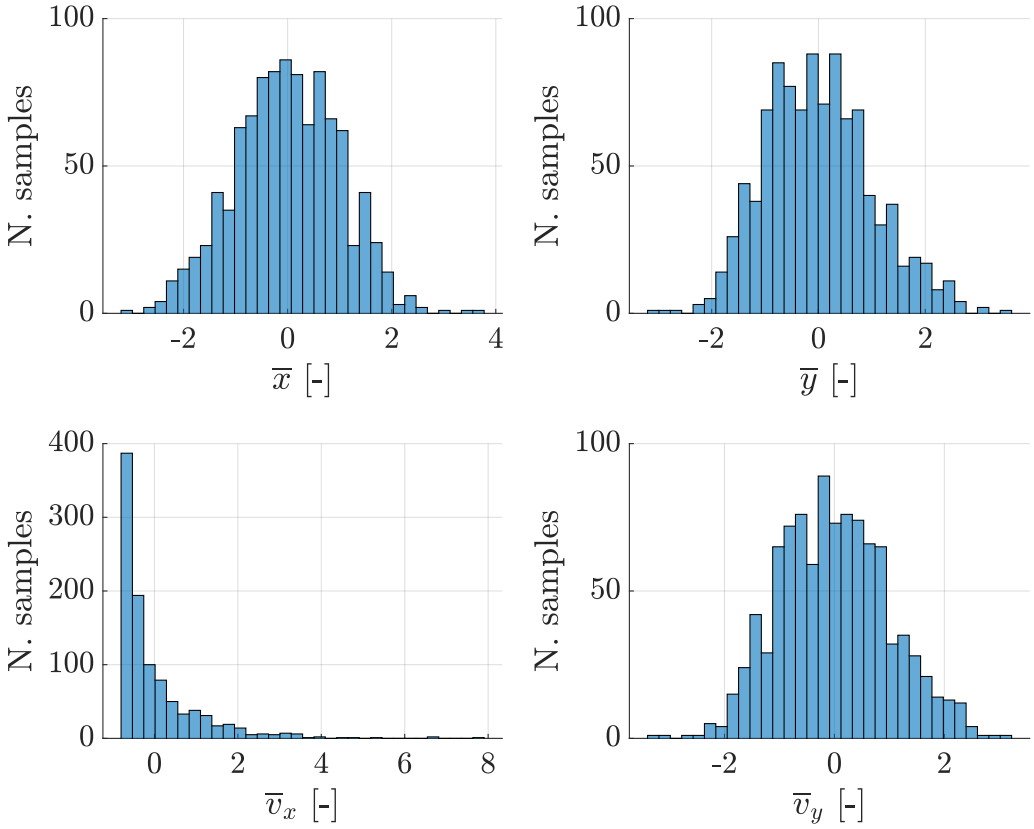



**Figure 5:** Histograms of the standardized components of Monte Carlo samples

The hypotheses made about the distribution of the components of the Monte Carlo samples at $t_f$ are confirmed by the histograms in Figure 5. Specifically, the histogram of the $\bar{v}_x$ distribution clearly demonstrates a *positively skewed* shape, with a high concentration of samples near zero and a long right tail extending toward higher positive velocities.

## 2 Batch filters

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

You have been asked to track SMOS to improve the accuracy of its state estimate. To this aim, you shall schedule the observations from the three ground stations reported in Table 4.

1. Compute visibility windows. The Two-Line Elements (TLE) set of SMOS are reported in Table 5 (and in WeBeep as 36036.3le). Compute the osculating state from the TLE at the reference epoch $t_{ref}$, then propagate this state assuming Keplerian motion to predict the trajectory of the satellite and compute all the visibility time windows from the available stations in the time interval from $t_0 =$ `2024-11-18T20:30:00.000` (UTC) to $t_f =$ `2024-11-18T22:15:00.000` (UTC). Consider the different time grid for each station depending on the frequency of measurement acquisition. Report the resulting visibility windows and plot the predicted Azimuth and Elevation profiles within these time intervals.
2. Simulate measurements. Use SGP4 and the provided TLE to simulate the measurements acquired by the sensor network in Table 4 by:
   (a) Computing the spacecraft position over the visibility windows identified in Point 1 and deriving the associated expected measurements.
   (b) Simulating the measurements by adding a random error to the expected measurements (assume a Gaussian model to generate the random error, with noise provided in Table 4). Discard any measurements (i.e., after applying the noise) that does not fulfill the visibility condition for the considered station.
3. Solve the navigation problem. Using the measurements simulated at the previous point:
   (a) Find the least squares (minimum variance) solution to the navigation problem without a priori information using
      - the epoch $t_0$ as reference epoch;
      - the reference state as the state derived from the TLE set in Table 5 at the reference epoch;
      - the simulated measurements obtained for the KOROU ground station only;
      - pure Keplerian motion to model the spacecraft dynamics.
   (b) Repeat step 3a by using all simulated measurements from the three ground stations.
   (c) Repeat step 3b by using a J2-perturbed motion to model the spacecraft dynamics.

   Provide the results in terms of navigation solution[‡], square root of the trace of the estimated covariance submatrix of the position elements, square root of the trace of the estimated covariance submatrix of the velocity elements. Finally, considering a linear mapping of the estimated covariance from Cartesian state to Keplerian elements, provide the standard deviation associated to the semimajor axis, and the standard deviation associated to the inclination. Elaborate on the results, comparing the different solutions.
4. Trade-off analysis. For specific mission requirements, you are constrained to get a navigation solution within the time interval reported in Point 1. Since the allocation of antenna time has a cost, you are asked to select the passes relying on a budget of 70.000 €. The cost per pass of each ground station is reported in Table 4. Considering this constraint,

[‡]Not just estimated state or covariance

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

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

**Table 4:** Sensor network to track SMOS: list of stations, including their features.

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

**Table 5:** TLE of SMOS.

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

## 2.1 Compute Visibility Windows

The computation of the visibility windows requires the knowledge of Azimuth (Az) and Elevation (El) angles of SMOS as seen from the ground station (GS). To do so, the state of SMOS at the reference epoch $t_{ref}$, obtained from the two-line orbital elements (TLE), was propagated until $t_f$ by integrating the equations of the Restricted Two-Body Problem (R2BP). The integration was performed using `ode78` with `RelTol` and `AbsTol` both set to $10^{-12}$. For each station, the integration was carried out using a specific time grid consistent with its acquisition frequency. For each of the epoch, Az, El and range were computed using the following procedure:

1. Retrieve GS position vector in *J2000* from SPK kernels using `cspice_spkpos`.
2. Compute SMOS state relative to the station: $\mathbf{r}_{\text{rel}} = \mathbf{r}_{\text{SMOS}} - \mathbf{r}_{\text{GS}}$.
3. Compute the rotation matrix $R^{\text{topo}}_{\text{J2000}}$ from *J2000* to GS topocentric frame with `cspice_pxform`.
4. Compute SMOS position vector in the topocentric frame: $\mathbf{r}_{\text{topo}} = R^{\text{topo}}_{\text{J2000}}\mathbf{r}_{\text{rel}}$.
5. Compute Az, El, and range using `cspice_reclat`.

The visibility windows in Table 6 were identified extracting from the measurements time series, those with El greater or equal to the minimum elevation allowed from the GS, the results in terms of Azimuth and El are shown in Figure 6

**Table 6:** Visibility Windows

| Station | Start | End |
|---|---|---|
| Kourou | 2024-11-18T20:40:00 | 2024-11-18T20:49:00 |
| Troll | 2024-11-18T21:02:30 | 2024-11-18T21:11:30 |
| Svalbard | 2024-11-18T21:56:00 | 2024-11-18T22:06:00 |

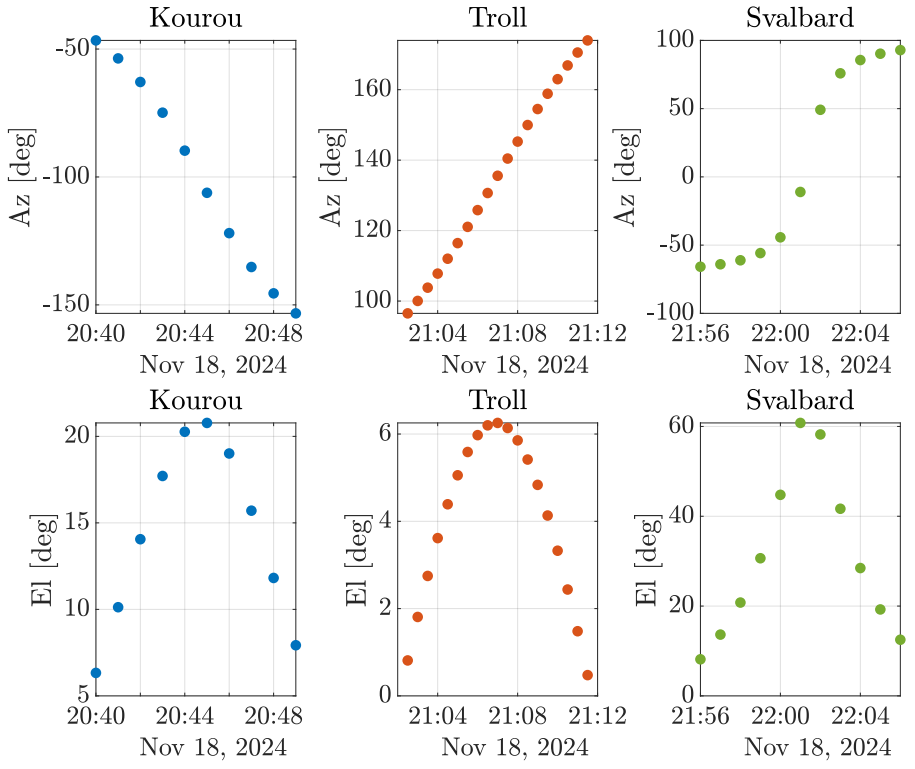



**Figure 6:** Azimuth and Elevation within the visibility windows

## 2.2 Simulate Measurements

The measurements within the visibility windows identified in subsection 2.1 were simulated as follows:

- The spacecraft TLE was read from the provided file using `read_3LE`, and the TLE reference epoch was converted to ephemeris time using `cspice_str2et`. To propagate the spacecraft orbit, the time vector in the visibility window was converted into minutes elapsed since the reference epoch, and the orbit was propagated using `sgp4`. To convert from *TEME* to *J2000*, corrections for Earth's nutation and precession were applied. The nutation parameters $\delta_\psi = -0.114752''$ and $\delta_\varepsilon = -0.007529''$, corresponding to the date 2024-11-18, were obtained from the Earth-Orientation-Parameters file available on the CelesTrak website (`https://celestrak.org/SpaceData/EOP-Last5Years.txt`). For precession correction, the centuries elapsed since 2000-01-01 were computed using `cspice_unitim`. Finally, the conversion from *TEME* to *J2000* was performed using `cspice_teme2eci`.
- The position vector in *J2000*, obtained with `sgp4`, was used to compute the expected measurements in terms of Az, El, and range following the procedure described in subsection 2.1. The real measurements were simulated by adding Gaussian random errors to the expected values, with a mean of the null vector and covariance $R = \text{diag}\left(\left[\sigma_{Az}^2 \quad \sigma_{El}^2 \quad \sigma_r^2\right]\right)$. The simulated measurements were then filtered to discard those with El smaller than the minimum elevation allowed from the GS.

The so simulated measurements are shown in Figure 7.

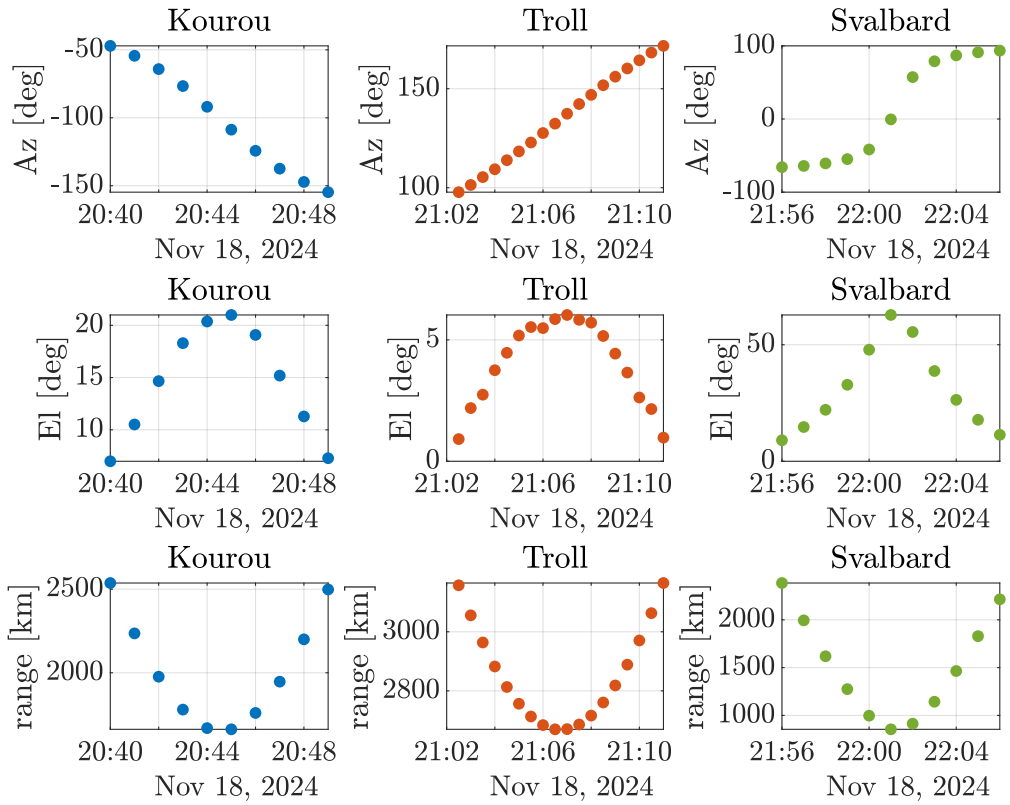



**Figure 7:** Simulated Measurements

## 2.3 Solve the Navigation Problem

The least squares solution to the navigation problem was computed using MATLAB `lsqnonlin`. The cost function was designed to work with one, two, or three stations and to handle different dynamical models, including or excluding the $J2$ perturbation to the Keplerian dynamics based on a flag. The function takes as input an array of *station*, a struct containing the time grid, the station weight matrix $W_m$ (as defined in Equation 2), and the measurements computed in subsection 2.2.

$$W_m = \begin{bmatrix} \frac{1}{\sigma_{Az}} & 0 & 0 \\ 0 & \frac{1}{\sigma_{El}} & 0 \\ 0 & 0 & \frac{1}{\sigma_{range}} \end{bmatrix} \tag{2}$$

For each station in the array, the initial state, which is an input to the cost function, is propagated over the corresponding time grid by numerically integrating the equations of the supplied dynamical model, and the expected measurements are computed as described in subsection 2.1. The residuals $\boldsymbol{\rho}$ are computed as $\boldsymbol{\rho} = W_m \begin{bmatrix} dAz & dEl & dr \end{bmatrix}^T$, where $dr$ is the difference between the predicted and measured range, and $dAz$ and $dEl$ are calculated using MATLAB `angdiff`.

### Pure Keplerian Motion-Only Kourou Station

In this case, only the measurements from the Kourou GS were considered, and pure Keplerian motion was used to propagate the initial state within the cost function. The initial guess was generated by propagating the reference state from the TLE at the reference epoch $t_{ref}$ to $t_0$ using the pure Keplerian model.

### Pure Keplerian Motion-All Stations

The data from all three ground stations (Kourou, Troll, and Svalbard) were used to estimate the state at $t_0$, with the propagation performed assuming pure Keplerian motion. The initial guess was generated using the same procedure as in the case with only the Kourou GS.

### J2 Perturbation-All Stations

In this case, the measurements from all three stations were considered, but a more refined model of the spacecraft motion was used. Specifically, the acceleration $\mathbf{a}_{J2}$ due to the $J2$ perturbation was added to the pure Keplerian model. The acceleration $\mathbf{a}_{J_2}$ was computed as follows:

1. Compute the rotation matrix $R^{\text{ECEF}}_{J2000}$ from the *J2000* frame to *ITRF93* using `cspice_pxform`.
2. Convert the ECI position vector to ECEF: $\mathbf{r}_{\text{ECEF}} = R^{\text{ECEF}}_{J2000}\mathbf{r}_{J2000}$.
3. Compute $\mathbf{a}_{J_2}$ in the *ECEF* frame:

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

4. Convert $\mathbf{a}_{J_2}$ back to the *J2000* frame.

The initial guess was generated by propagating the reference state to $t_0$ using the Keplerian model perturbed by the $J_2$ term.

The obtained results in terms of estimated mean and covariance are reported in Appendix B. The estimation error for the three cases was computed as $\boldsymbol{\varepsilon} = \hat{\mathbf{x}}_0 - \mathbf{x}_{0,SGP4}$. Here, $\mathbf{x}_{0,SGP4}$ was obtained by propagating with `sgp4` the reference state from the TLE at $t_{ref}$ to $t_0$.

$$\begin{aligned}
\mathbf{x}_{0,SGP4} &= \begin{bmatrix}\mathbf{r}_{0,SGP4} & \mathbf{v}_{0,SGP4}\end{bmatrix}^T \quad \text{with} \\
\mathbf{r}_{0,SGP4} &= \begin{bmatrix}3932.7688 & -1414.9188 & 5778.5029\end{bmatrix} \quad [km] \quad \text{and} \\
\mathbf{v}_{0,SGP4} &= \begin{bmatrix}4.8798 & -3.7632 & -4.2328\end{bmatrix} \quad [km/s]
\end{aligned}$$

Table 7 presents the norm of the estimation error in position and velocity, as well as the trace of the position and velocity covariance submatrices.

**Table 7:** Uncertainty estimate in the three cases

| **Case** | $\varepsilon_r$ $[km]$ | $\varepsilon_v$ $[km/s]$ | $\sqrt{Tr(P_{rr})}$ $[km]$ | $\sqrt{Tr(P_{vv})}$ $[km/s]$ |
|---|---|---|---|---|
| Kourou-Keplerian motion | 46.2414 | 0.0605 | 5.5798 | 0.0062 |
| All stations-Keplerian motion | 7.2409 | 0.0090 | 1.0203 | 0.0011 |
| All stations-J2 perturbation | 0.0328 | 2.3762e-05 | 0.0157 | 1.5915e-05 |

From Table 7, it can be observed that using measurements from all stations reduces the estimation error in both position and velocity compared to the first case, where only Kourou measurements were processed, as more measurements are utilized. However, the uncertainty remains high because the pure Keplerian model does not accurately represent the spacecraft dynamics. Adding the $J2$ perturbation significantly reduces both the errors and the uncertainty in position and velocity by two orders of magnitude.

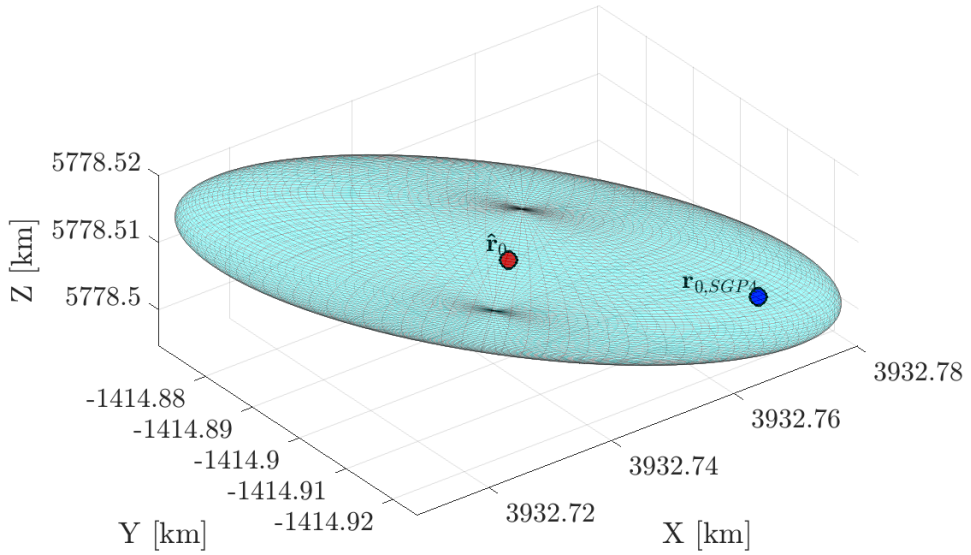



**Figure 8:** $3\sigma$-confidence ellipsoid using all stations and including $J2$ in the Earth-centered equatorial J2000 reference frame

In Figure 8, the $3\sigma$-confidence ellipsoid associated with the position covariance is illustrated, obtained by processing the measurements from all stations and including the $J2$ perturbation

in the model. It can be observed that the estimated position $\hat{\mathbf{r}}_0$ is very close to the reference $\mathbf{r}_{0,SGP4}$, which lies within the ellipsoid. This indicates that the $J2$-perturbed model provides a good representation of the reference spacecraft dynamics.

Finally, the standard deviations $\sigma_{aa}$ and $\sigma_{ii}$, associated with the semi-major axis and inclination, respectively, were computed using the following procedure. A function, `car2kep`, was implemented to compute the orbital elements from a given Cartesian orbital state. The Jacobian $J$ of this non-linear transformation was calculated numerically using centered finite differences by perturbing the estimated state for the three analyzed cases. The covariance matrix in orbital elements, $P_{kep}$, was obtained through the linear mapping described in Equation 3, using the Cartesian covariance matrix $P_{car}$. Finally, $\sigma_{aa}$ and $\sigma_{ii}$ were extracted from the diagonal elements of $P_{kep}$.

$$P_{kep} = JP_{car}J^T \tag{3}$$

The so computed $\sigma_{aa}$ and $\sigma_{ii}$ are reported in Table 8. It can be noted that, as expected, the third method is the one that performs best, since it processes all the available measurements and utilizies a more refined model of the spacecraft dynamics.

**Table 8:** Uncertainty estimate in the three cases

| **Case** | $\sigma_{aa}$ $[km]$ | $\sigma_{ii}$ $[deg]$ |
|---|---|---|
| Kourou-Keplerian motion | 4.1954 | 0.0418 |
| All stations-Keplerian motion | 0.1434 | 0.0020 |
| All stations-J2 perturbation | 0.0022 | 2.9893e-05 |

## 2.4 Trade off Analysis

Since a constraint budget of 70000 € to get the telemetry data in the $[t_0,\ t_f]$ time interval was imposed, the entire set of GS cannot be used for the mission as the total cost would have exeded the budget constraint. Therefore the trade off analysis described in this section was performed. The single stations and all the possible couples were taken into consideration. For all the analyzed cases $\sigma_{aa}$ and $\sigma_{ii}$ were computed with the procedure described in subsection 2.3, then they were normalized with respect to the semi-majoraxis and the inclination, in Table 9, obtained from the provided TLE.

**Table 9:** Orbital Elements of SMOS at $t_{ref}$

| $a$ $[km]$ | $e$ $[-]$ | $i$ $[deg]$ | $\Omega$ $[deg]$ | $\omega$ $[deg]$ | $\theta$ $[deg]$ |
|---|---|---|---|---|---|
| 7143.71346 | 0.00124363 | 98.43420 | 148.46890 | 69.29293 | 290.84031 |

The normalization allows the computation of $\overline{\sigma}_{aa}$ and $\overline{\sigma}_{ii}$ for all the analyzed cases, resulting in non-dimensional quantities that are more comparable than their corresponding dimensional counterparts. All the analyzed cases are represented in the $\overline{\sigma}_{aa}$-$\overline{\sigma}_{ii}$ plane, as shown in Figure 9. The plot uses a logarithmic scale for improved visualization. To evaluate the trade-off, the inverse of the distance from the origin was used as a performance metric, as maximizing it corresponds to minimizing the overall uncertainty. The results, including the performance index and associated cost, are summarized in Table 10.

**Table 10:** Trade-off analysis results

| **Case** | Kourou | Troll | Svalbard | Kourou+Troll | Kourou+Svalbard | Svalbard+Troll |
|---|---|---|---|---|---|---|
| **Perf. [-]** | 937.8 | 6142.3 | 18691.9 | 5124.4 | 271588.0 | 140479.6 |
| **Cost [€]** | 30000 | 35000 | 35000 | 65000 | 65000 | 70000 |

From Table 10, it can be observed that the combination offering the best performance is the couple *Kourou+Svalbard*. Additionally, using measurements solely from *Svalbard* achieves a relatively high performance index at a significantly lower cost. However, this analysis prioritized reducing uncertainty, leading to the conclusion that the couple *Kourou+Svalbard* is the optimal combination.

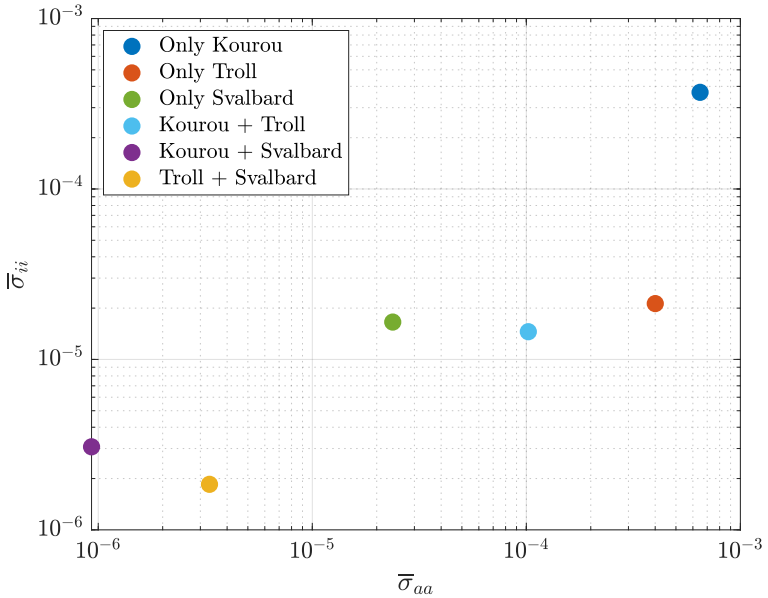



**Figure 9:** Trade-off analysis

## 2.5 Long-Term Analysis

The long-term analysis presented in this section was conducted to select the best combination of ground stations for extended operations. Visibility windows were recalculated over a three-day time span starting from the epoch 2024-11-18T20:30:00, following the same methodology outlined in subsection 2.1. The propagation was carried out using the $J_2$-perturbed model, which provides a more accurate representation of the spacecraft dynamics.

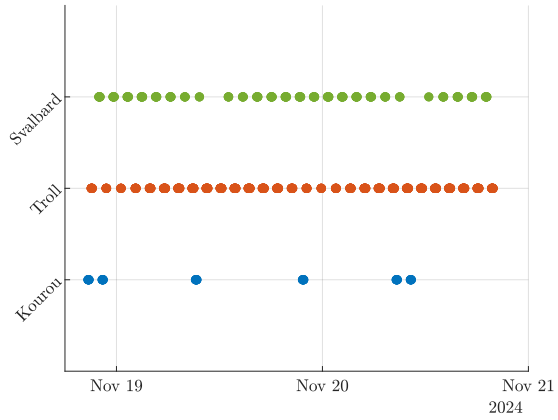



**Figure 10:** Visibility windows over a 3-day timeframe

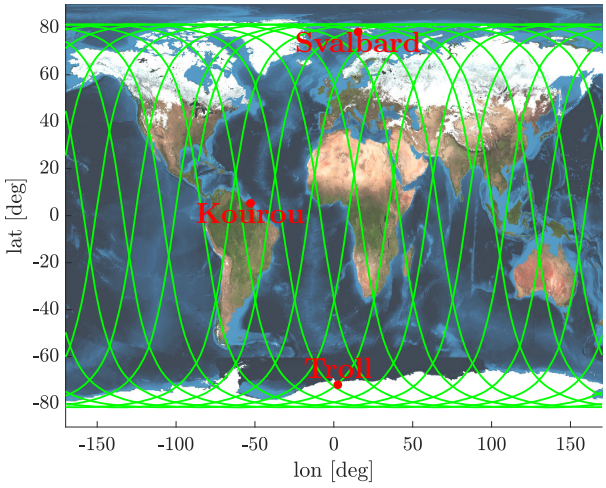



**Figure 11:** Groundtrack over a 1-day timeframe

In Figure 10, the visibility windows for all ground stations are depicted. It is evident that Kourou is not a suitable choice for long-term operations, as it offers significantly fewer visibility windows compared to the other two stations. This outcome is expected, given that SMOS operates in a polar orbit, as indicated by the inclination in Table 9.

The groundtrack of SMOS from 2024-11-18T00:00:00 to 2024-11-18T23:59:00 is shown in Figure 11, illustrating that the spacecraft rarely passes within Kourou's field of view during this timeframe.

Additionally, Figure 10 shows that Troll provides more visibility windows than Svalbard. This is because Troll operates with a lower minimum elevation angle compared to Svalbard, allowing for longer acquisition windows. For this reason, and due to its higher acquisition frequency, Troll was selected as the primary ground station, with Svalbard designated as the backup.

# 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 11, predict the trajectory of the satellite in an inertial Moon-centered reference frame assuming Keplerian motion. Use the estimated coordinates given in Table 12 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 11 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 11. 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 $3\sigma$ 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 11. For the initial state, you can perturb the actual initial state, composed by $\mathbf{x}_0$ and the latitude

and longitude given in Table 12, 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 $3\sigma$ of the estimated covariance for both position and velocity.

**Table 11:** 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]<br>$\mathbf{v}_0$ = [-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 12:** Lunar lander - initial guess coordinates and horizon mask

| | |
|---|---|
| Lander name | MOONLANDER |
| Coordinates | LAT = 78°<br>LON = 15°<br>ALT = 0 m |
| Minimum elevation | 0 deg |

### 3.1 Check the visibility windows

Given the initial state of the orbiter in the Moon-Centered Inertial (MCI) frame, the S/C trajectory over the timespan $[t_0,\ t_f]$ was computed via numerical integration of the restricted two-body problem (R2BP) equations. The estimated coordinates of the lander were transformed to the Moon-Centered Moon-Fixed (MCMF) frame using `cspice_pgrrec`. The S/C trajectory in MCMF and the estimated position of the lander are shown in Figure 12.

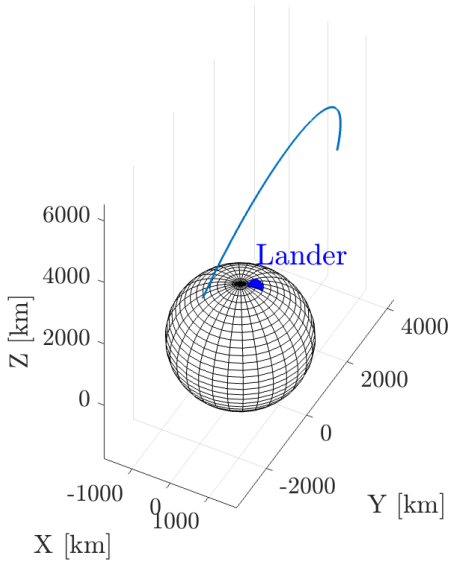



**Figure 12:** S/C trajectory in MCMF frame

To ensure that the lander and the orbiter remain in relative visibility, it was necessary to verify that the elevation angle ($El$) of the spacecraft (S/C) relative to the lander satisfies the condition $El > El_{min} \forall t \in [t_0,\ t_f]$. This guarantees continuous visibility of the orbiter from the lander. The elevation was computed following these steps:

- Compute the rotation matrix $R_{MCI}^{MCMF}$ from MCI to MCMF frame using `cpsice_pxform`
- Compute the S/C position in MCMF: $\mathbf{r}_{SC}^{MCMF} = R_{MCI}^{MCMF}\mathbf{r}_{SC}^{MCI}$
- Compute the S/C position with respect to the lander: $\mathbf{r}_{rel}^{MCMF} = \mathbf{r}_{SC}^{MCMF} - \mathbf{r}_{L}^{MCMF}$
- Compute the rotation matrix $R_{MCMF}^{TOPO}$ from the MCMF to the topocentric frame with `cspice_eul2m`
- Compute the relative position in the topocentric frame: $\mathbf{r}_{rel}^{TOPO} = R_{MCMF}^{TOPO}\mathbf{r}_{rel}^{MCMF}$
- Compute range ($\rho$), Azimuth ($Az$) and Elevation ($El$) using `cspice_reclat`

It was obtained that the S/C and the lander are always in relative visibility, as shown in Figure 13.

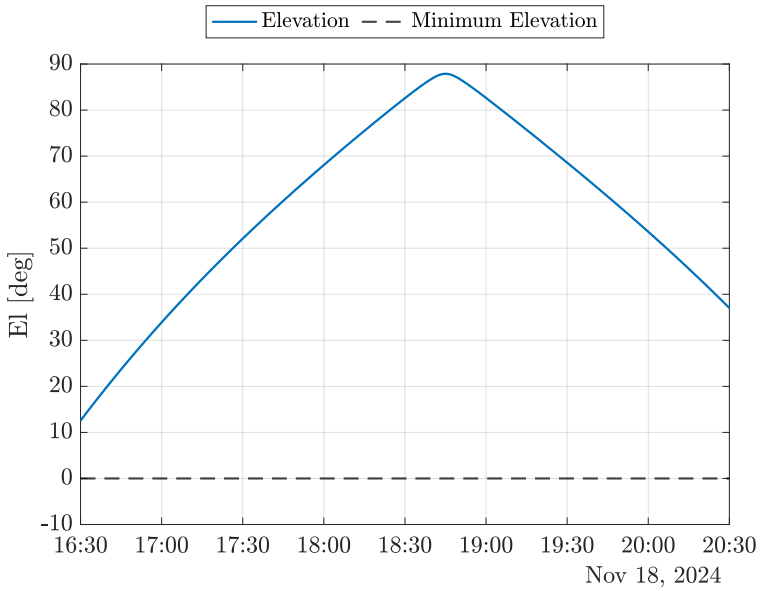



**Figure 13:** S/C elevation over the timespan

### 3.2 Simulate Measurements

The expected measurements were computed using the same procedure described in subsection 3.1, with the only difference being that the lander position was derived from the coordinates provided in the supplied kernels. The expected measurements of the spacecraft position were obtained by integrating the R2BP equations, assuming they represent the true dynamics of the orbiter. The real measurements were simulated by adding Gaussian random noise with zero mean and covariance

$$R = \text{diag}\left(\begin{bmatrix}\sigma_P^2 & \sigma_P^2 & \sigma_P^2 & \sigma_P^2\end{bmatrix}\right),$$

to the computed spacecraft position and relative range.

To verify that the noise level is within the desired boundaries, the following procedure was implemented. Each measurement $y_i$, a Gaussian random variable with mean $\mu_i$ and standard deviation $\sigma_i$, was standardized as shown in Equation 4, resulting in a random variable $z_i$ distributed according to the Standard Normal Distribution (SND):

$$z_i = \frac{y_i - \mu_i}{\sigma_i} \quad \text{for} \quad i = 1, \dots, 4. \tag{4}$$

The standardized variables $z_i$ were then organized into a vector $\mathbf{Z}$, composed of random variables distributed as the SND. To confirm that the applied noise level remained within the desired boundaries of $\pm 3\sigma$, the percentage of values in $\mathbf{Z}$ exceeding the $\pm 3$ interval was computed for each simulation. Although some variability was observed due to the limited number of samples, the percentage rarely exceeded $\approx 0.27\%$, consistent with the expected behavior of a Gaussian distribution. In Figure 14, the histogram for a single simulation is shown. It can be observed graphically that only a limited number of samples exceeds the $\pm 3\sigma$ interval.

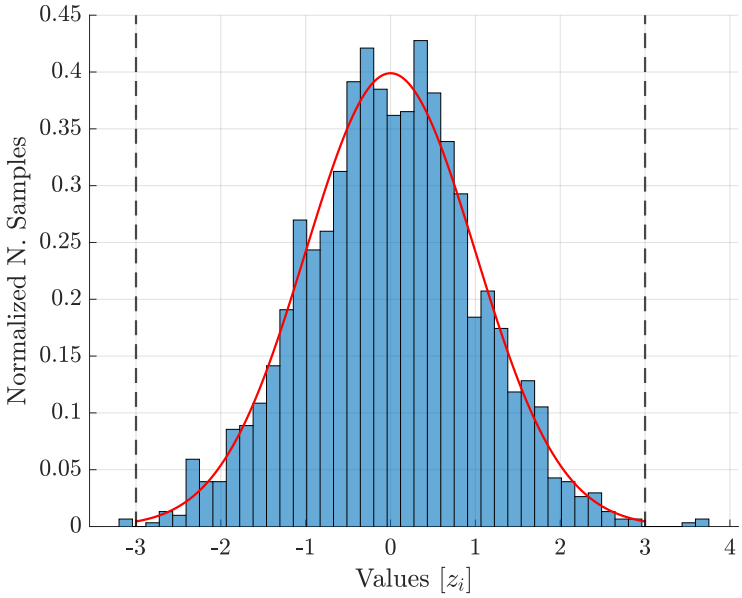



**Figure 14:** Histogram of standardized variables **Z**

### 3.3 Estimate Orbiter State

An Unscented Kalman Filter (UKF) was implemented to sequentially estimate the spacecraft state and uncertainty using the available simulated measurements at each time instant $t_k$. The covariance was initialized as

$$P_0 = \text{diag}\left([10, 1, 1, 0.001, 0.001, 0.001]\right),$$

while the initial guess $\hat{\mathbf{x}}_0^+$ was generated using MATLAB's `mvnrnd`, with the mean equal to the nominal orbiter state at $t_0$ and covariance $P_0$. The parameters $\alpha$ and $\beta$ were set respectively to 0.01 and 2, as suggested. At each time instant $t_k$, the estimated state and covariance were computed using the following steps:

- **Generate the $\sigma$-points**: Using the a-posteriori estimated mean $\hat{\mathbf{x}}_{k-1}^+$ and covariance $P_{k-1}^+$ from the previous time step, the $\sigma$-points $\boldsymbol{\chi}_{i,\,k-1}$ and the weights $W_i^{(m)}$ and $W_i^{(c)}$ were generated as described in subsubsection 1.1.2.
- **Perform the prediction step**: The $\sigma$-points $\boldsymbol{\chi}_{i,\,k-1}$ were propagated from $t_{k-1}$ to $t_k$ by numerically integrating the R2BP equations. The a-priori mean $\hat{\mathbf{x}}_k^-$ and covariance $P_k^-$ were computed as the weighted sample mean and covariance of the propagated $\sigma$-points. These $\sigma$-points $\boldsymbol{\chi}_{i,\,k}$ were then mapped into the measurement space using a non-linear transformation, which, in this case, consisted of extracting the first three elements of the $\sigma$-points since only the orbiter position measurements were required to estimate its state. The mean in the measurement space, $\hat{\mathbf{y}}_k$, was computed as the weighted mean of the $\sigma$-points mapped to the measurement space.
- **Perform the update step**: The measurement weighted covariance $P_{yy,k}$ and the cross-covariance $P_{xy,k}$ were computed. The Kalman gain was obtained as $K_k = P_{xy,k}P_{yy,k}^{-1}$. The a-posteriori mean and covariance were updated as:

$$\begin{cases} \hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + K_k\left(\mathbf{y}_k - \hat{\mathbf{y}}_k\right), \\ P_k^+ = P_k^- - K_k P_{yy,k} K_k^T, \end{cases} \tag{5}$$

  where $\mathbf{y}_k$ represents the measurement at the $k$-th time instant.

The detailed algorithm is provided in Appendix C.

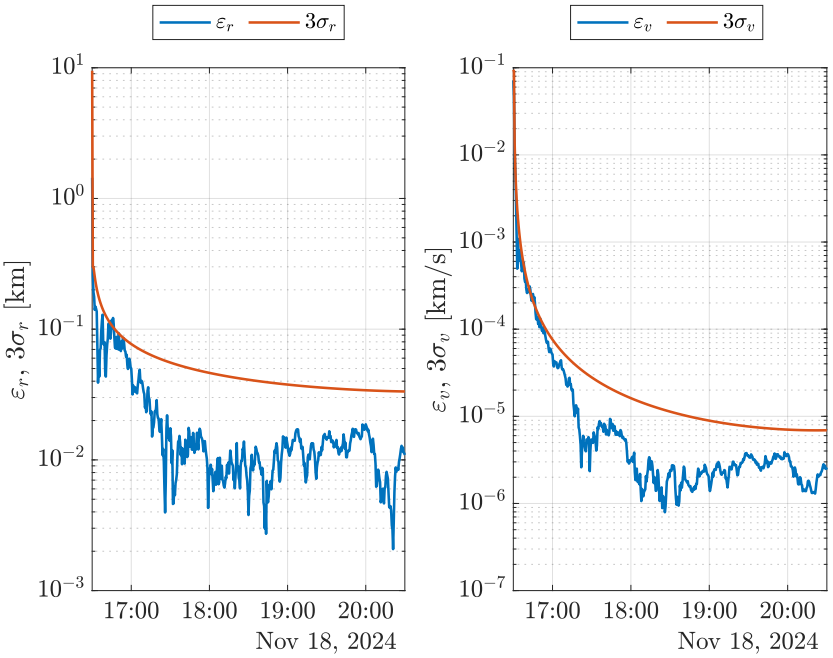



**Figure 15:** Error and $3\sigma$ time history for position and velocity

In Figure 15, the time evolution of the position and velocity errors, along with their corresponding $3\sigma$ confidence levels, are shown. The $3\sigma$ bounds are defined as $3\sigma_r = 3\sqrt{\max(\text{eig}(P_{rr}))}$ and $3\sigma_v = 3\sqrt{\max(\text{eig}(P_{vv}))}$. From the plot, it can be observed that the implemented UKF effectively reduces both the estimation error and the associated uncertainty as more measurements are processed. Furthermore, the errors remain below their respective $3\sigma$ bounds and tend to stabilize at significantly lower levels as additional measurements are processed.

### 3.4 Estimate Lander Position

Finally, the lander latitude and longitude were estimated by utilizing both the measurements of the spacecraft position and the relative range. An UKF was implemented, following the same procedure described in subsection 3.3. The only difference was in the prediction step, where the non-linear mapping from the state to the measurement space not only involved extracting the first three components of the $\sigma$-points but also included computing the relative range, as described in subsection 3.1, using the a-priori estimated longitude and latitude at the $k$-th time step. The initial guess was generated by drawing a random sample using `mvnrnd`, with the mean vector composed of the nominal orbiter state at $t_0$ and the first estimates of latitude and longitude. The covariance matrix was defined as:

$$P_0 = \text{diag}\left([10, 1, 1, 0.001, 0.001, 0.001, 0.00001, 0.00001]\right).$$

The parameters $\alpha$ and $\beta$ were set to 0.01 and 2, respectively, as suggested.

In Figure 16, the time evolution of the error in terms of estimated position, velocity, latitude, and longitude is depicted. It can be observed that the time history of $\varepsilon_r$ and $\varepsilon_v$ closely resembles the one presented in subsection 3.3, suggesting that the estimation of the orbiter state is minimally influenced by the estimation of the lander coordinates, despite the coupling of the equations. For both longitude and latitude, the estimation error and $3\sigma$ bounds tend to decrease over

time as more measurements become available, albeit at a slower rate compared to $\varepsilon_r$ and $\varepsilon_v$. Additionally, the error consistently remains below the $3\sigma$ threshold, further validating the accuracy of the implemented UKF. However, it is also evident that the estimation errors for latitude and longitude exhibit greater oscillations compared to the orbiter's position and velocity.

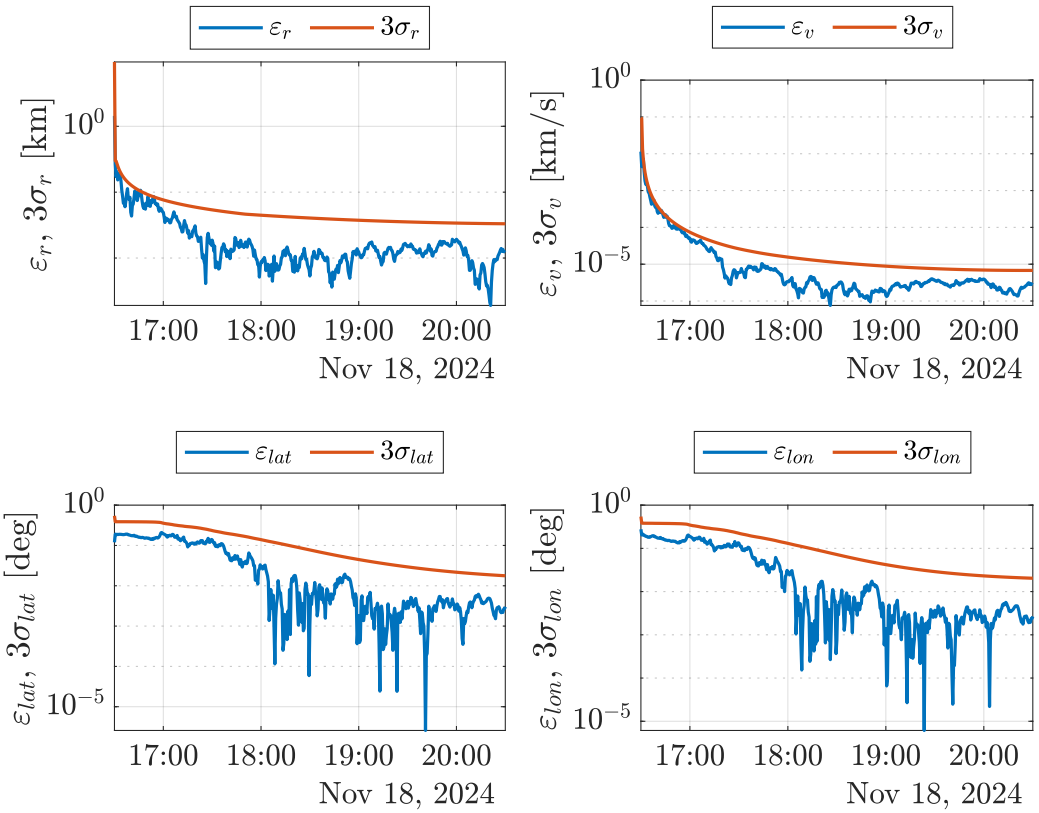



**Figure 16:** Error and $3\sigma$ time history for position, velocity, latitude and longitude

# A Planar Bicircular Restricted Four-Body Problem

The Planar Bicircular Restricted Four-Body Problem (PBRFBP) is an extension of the Circular Restricted Three-Body Problem (CRTBP) that incorporates the gravitational influence of a third primary body (the Sun) into the system of two main bodies (Earth and Moon). The equations describing the PBRFBP are:

$$\dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x}) = \begin{bmatrix} v_x \\ v_y \\ 2v_y + \frac{\partial \Omega_4}{\partial x} \\ -2v_x + \frac{\partial \Omega_4}{\partial y} \end{bmatrix}$$

where

$$\Omega_4(x, y, t) = \frac{1}{2}(x^2 + y^2) + \frac{1-\mu}{r_1} + \frac{\mu}{r_2} + \frac{1}{2}\mu(1-\mu) + \frac{m_s}{r_3(t)} - \frac{m_s}{\rho^2}(x\cos(\omega_s t) + y\sin(\omega_s t)),$$

with

$$\begin{aligned} r_1 &= \sqrt{(x+\mu)^2 + y^2} \\ r_2 &= \sqrt{(x+\mu-1)^2 + y^2} \\ r_3(t) &= \sqrt{(x - \rho\cos(\omega_s t))^2 + (y - \rho\sin(\omega_s t))^2} \end{aligned}$$

$\mathbf{x} = [x \;\; y \;\; v_x \;\; v_y]^T$ is the state vector at time t, $m_s$ the scaled mass of the Sun and, $\rho$ is the distance of the Sun from the Earth-Moon barycenter and $\omega_s$ is the Sun angular velocity in the Earth-Moon rotating frame. The STM dynamics is described by the following variational equation:

$$\begin{cases} \dot{\Phi} = A(t)\Phi(t_0, t) \\ \Phi(t_0, t) = \mathbf{I}_{4\times 4} \end{cases}$$

where

$$A(t) = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} = \begin{bmatrix} \mathbf{0}_{2\times 2} & \mathbf{I}_{2\times 2} \\ \begin{bmatrix} \frac{\partial^2 \Omega}{\partial x^2} & \frac{\partial^2 \Omega}{\partial x \partial y} \\ \frac{\partial^2 \Omega}{\partial y \partial x} & \frac{\partial^2 \Omega}{\partial y^2} \end{bmatrix} & \begin{bmatrix} 0 & 2 \\ -2 & 0 \end{bmatrix} \end{bmatrix}$$

$$\begin{aligned} \frac{\partial^2 \Omega}{\partial x^2} &= 1 - \frac{(1-\mu)}{r_1^3} + \frac{3(1-\mu)(x+\mu)^2}{r_1^5} - \frac{\mu}{r_2^3} + \frac{3\mu(x+\mu-1)^2}{r_2^5} - \frac{m_s}{r_3^3} + \frac{3m_s(x - \rho\cos(\omega_s t))^2}{r_3^5}, \\ \frac{\partial^2 \Omega}{\partial y^2} &= 1 - \frac{(1-\mu)}{r_1^3} + \frac{3(1-\mu)y^2}{r_1^5} - \frac{\mu}{r_2^3} + \frac{3\mu y^2}{r_2^5} - \frac{m_s}{r_3^3} + \frac{3m_s(y - \rho\sin(\omega_s t))^2}{r_3^5}, \\ \frac{\partial^2 \Omega}{\partial x \partial y} &= \frac{\partial^2 \Omega}{\partial y \partial x} = \frac{3(1-\mu)(x+\mu)y}{r_1^5} - \frac{3\mu(x+\mu-1)y}{r_2^5} + \frac{3m_s(x - \rho\cos(\omega_s t))(y - \rho\sin(\omega_s t))}{r_3^5}. \end{aligned}$$

# B Minimum Variance Results

Here are presented the obtained results for the three analyzed cases

## B.1 Only Kourou-Pure Keplerian Motion

$$\hat{\mathbf{r}}_0 \quad [km] = \begin{bmatrix} 3896.9443 & -1390.5843 & 5762.2947 \end{bmatrix}^T \tag{6}$$

$$\hat{\mathbf{v}}_0 \quad [km/s] = \begin{bmatrix} 4.9283 & -3.7648 & -4.1967 \end{bmatrix}^T \tag{7}$$

$$P_{rr} \quad \left[km^2\right] = \begin{bmatrix} 16.1405 & 1.7516 & -0.0703 \\ 1.7516 & 11.8455 & -5.8830 \\ -0.0703 & -5.8830 & 3.1486 \end{bmatrix} \tag{8}$$

$$P_{rv} \quad \left[km^2/s\right] = \begin{bmatrix} -0.0172 & -0.0062 & -0.0102 \\ 0.0019 & -0.0096 & 0.0044 \\ -0.0018 & 0.0041 & -0.0029 \end{bmatrix} \tag{9}$$

$$P_{vv} \quad \left[km^2/s^2\right] = \begin{bmatrix} 0.1952 & 0.0364 & 0.1253 \\ 0.0364 & 0.1000 & 0.0007 \\ 0.1253 & 0.0007 & 0.0910 \end{bmatrix} \cdot 10^{-4} \tag{10}$$

## B.2 All Stations-Pure Keplerian Motion

$$\hat{\mathbf{r}}_0 \quad [km] = \begin{bmatrix} 3926.8683 & -1411.0112 & 5780.0346 \end{bmatrix}^T \tag{11}$$

$$\hat{\mathbf{v}}_0 \quad [km/s] = \begin{bmatrix} 4.8886 & -3.7650 & -4.2334 \end{bmatrix}^T \tag{12}$$

$$P_{rr} \quad \left[km^2\right] = \begin{bmatrix} 0.6084 & -0.3640 & -0.0920 \\ -0.3640 & 0.3547 & 0.0873 \\ -0.0920 & 0.0873 & 0.0780 \end{bmatrix} \tag{13}$$

$$P_{rv} \quad \left[km^2/s\right] = \begin{bmatrix} -0.6182 & 0.2070 & -0.3799 \\ 0.4227 & -0.1035 & 0.2778 \\ 0.0609 & -0.0245 & 0.0976 \end{bmatrix} \cdot 10^{-3} \tag{14}$$

$$P_{vv} \quad \left[km^2/s^2\right] = \begin{bmatrix} 0.7155 & -0.1748 & 0.3931 \\ -0.1748 & 0.1167 & -0.1254 \\ 0.3931 & -0.1254 & 0.2869 \end{bmatrix} \cdot 10^{-6} \tag{15}$$

## B.3 All Stations-J2-Perturbed Motion

$$\hat{\mathbf{r}}_0 \quad [km] = \begin{bmatrix} 3932.7443 & -1414.8975 & 5778.5073 \end{bmatrix}^T \tag{16}$$

$$\hat{\mathbf{v}}_0 \quad [km/s] = \begin{bmatrix} 4.8798 & -3.7632 & -4.2329 \end{bmatrix}^T \tag{17}$$

$$P_{rr} \quad \left[km^2\right] = \begin{bmatrix} 0.1437 & -0.0883 & -0.0262 \\ -0.0883 & 0.0842 & 0.0227 \\ -0.0262 & 0.0227 & 0.0182 \end{bmatrix} \cdot 10^{-3} \tag{18}$$

$$P_{rv} \quad \left[km^2/s\right] = \begin{bmatrix} -0.1422 & 0.0493 & -0.0911 \\ 0.0982 & -0.0260 & 0.0657 \\ 0.0186 & -0.0078 & 0.0247 \end{bmatrix} \cdot 10^{6} \tag{19}$$

$$P_{vv} \quad \left[km^2/s^2\right] = \begin{bmatrix} 0.1590 & -0.0412 & 0.0908 \\ -0.0412 & 0.0268 & -0.0308 \\ 0.0908 & -0.0308 & 0.0675 \end{bmatrix} \cdot 10^{-9} \tag{20}$$

# C Algorithm UKF

**Algorithm 1** Unscented Kalman Filter (UKF)

**Require:** Initial state $\mathbf{x}_0$, scaling parameters $\alpha$, $\beta$, initial covariance $\mathbf{P}_0$, number of measurements $n_{\text{meas}}$, measurement noise covariance $\mathbf{R}$, measurements $\mathbf{y}$, time span $t_{\text{span}}$, dynamics flag

**Ensure:** Estimated states $\hat{\mathbf{x}}$, covariance matrices $\mathbf{P}$

Dimensions: $n$ number of states, $m$ number of time steps

**for** $k = 2$ to $m$ **do**

Compute sigma points and weights with $\alpha$, $\beta$, $\mathbf{x}_0$, $\mathbf{P}_0$: $[(\mathbf{X}_i)_{k-1}, \mathbf{W}_m, \mathbf{W}_c]$

Perform prediction step:

Propagate the $2n+1$ sigma points from $t_{k-1}$ to $t_k$ and obtain $(\mathbf{X}_i)_k$

Compute a priori mean and covariance:

$$\hat{\mathbf{x}}_k^- = \sum_{i=0}^{2n} (W_m)_i (\mathbf{X}_i)_k$$

$$\mathbf{P}_k^- = \sum_{i=0}^{2n} (W_c)_i ((\mathbf{X}_i)_k - \hat{\mathbf{x}}_k^-)((\mathbf{X}_i)_k - \hat{\mathbf{x}}_k^-)^T$$

Compute sigma points in measurements space $(\mathbf{Y}_i)_k$ and mean of measurements:

$$\hat{\mathbf{y}}_k = \sum_{i=0}^{2n} (W_m)_i (\mathbf{Y}_i)_k$$

Perform update step:

Compute measurements covariance and cross-covariance:

$$\mathbf{P}_{yy,k} = \sum_{i=0}^{2n} (W_c)_i ((\mathbf{Y}_i)_k - \hat{\mathbf{y}}_k)((\mathbf{Y}_i)_k - \hat{\mathbf{y}}_k)^T + \mathbf{R}$$

$$\mathbf{P}_{xy,k} = \sum_{i=0}^{2n} (W_c)_i ((\mathbf{X}_i)_k - \hat{\mathbf{x}}_k^-)((\mathbf{Y}_i)_k - \hat{\mathbf{y}}_k)^T$$

Compute the Kalman gain: $\mathbf{K}_k = \mathbf{P}_{xy,k} \mathbf{P}_{yy,k}^{-1}$

Compute a posteriori mean and covariance:

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{y}_k - \hat{\mathbf{y}}_k)$$

$$\mathbf{P}_k^+ = \mathbf{P}_k^- - \mathbf{K}_k \mathbf{P}_{yy,k} \mathbf{K}_k^T$$

Symmetrize covariance:

$$\mathbf{P}_k^+ = \frac{\mathbf{P}_k^+ + \mathbf{P}_k^{+T}}{2}$$

Update time and state for next iteration: $\mathbf{x}_0 \leftarrow \hat{\mathbf{x}}_k^+$, $\mathbf{P}_0 \leftarrow \mathbf{P}_k^+$

**end for**

## References

[1] Douglas C. Montgomery and George C. Runger. Applied Statistics and Probability for Engineers. 7th. John Wiley & Sons, 2020. ISBN: 9781119496576.