SGN – Assignment #2

Piercarlo Fontana, 10730242

## 1: Uncertainty propagation

You are asked to analyse the state uncertainty evolution along a transfer trajectory in the Planar Bicircular Restricted Four-Body Problem, obtained as optimal solution of the problem stated in Section 3.1 (Topputo, 2013)*. The mean initial state $\mathbf{x}_i$ at initial time $t_i$ with its associated covariance $\mathbf{P}_0$ and final time $t_f$ for this optimal transfer are provided in Table 1.

1. Propagate the initial mean and covariance within a time grid of 5 equally spaced elements going from $t_i$ to $t_f$, using both a Linearized Approach (LinCov) and the Unscented Transform (UT). We suggest to use $\alpha = 1$ and $\beta = 2$ for tuning the UT in this case. Plot the mean and the ellipses associated with the position elements of the covariances obtained with the two methods at the final time.

2. Perform the same uncertainty propagation process on the same time grid using a Monte Carlo (MC) simulation †. Compute the sample mean and sample covariance and compare them with the estimates obtained at Point 1). Provide the following outputs.

   - Plot of the propagated samples of the MC simulation, together with the mean and the covariance obtained with all methods, in terms of ellipses associated with the position elements at the final time.
   - Plot of the time evolution (for the time grid previously defined) for all three approaches (MC, LinCov, and UT) of $3\sqrt{\max(\lambda_i(P_r))}$ and $3\sqrt{\max(\lambda_i(P_v))}$, where $P_r$ and $P_v$ are the 2x2 position and velocity covariance submatrices.
   - Plot resulting from the use of the MATLAB function `qqplot`, for each component of the previously generated MC samples at the final time.

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

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

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

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

†Use at least 1000 samples drawn from the initial covariance

### 1.1 Plot the mean and the ellipses associated with LinCov and UT

The goal is to evaluate three different uncertainty propagation approaches: each approach is applied to the same initial state and covariance to evaluate their relative accuracy and performance, specifically through the analysis of propagated mean and error ellipses at the final time step.

#### 1.1.1 Linear Covariance

In the *Linear Covariance* method, the propagated mean coincides with the mean of the final distribution, found by simply propagating the initial state $[\mathbf{r}_0,\ \mathbf{v}_0]$ from $t_i$ to $t_f$, following a 5 spaces discretization grid. The propagated covariance is instead computed using the state transition matrix $\boldsymbol{\Phi}$ at each time instant $k$, as it follows:

$$\hat{\mathbf{x}}_k = \mathbf{x}_k, \quad \mathbf{P}_k = \boldsymbol{\Phi}_{k-1}\mathbf{P}_0\boldsymbol{\Phi}_{k-1}^\top$$

This approach provides computational efficiency and is particularly well-suited for systems where nonlinearity is limited, or the dynamics can be adequately captured by linearization. However, as we'll see in this case, it fails to account for higher-order nonlinear effects.

#### 1.1.2 Unscented Transform

The *Unscented Transform* addresses such limitations by adopting a nonlinear propagation scheme through the generation and processing of sigma points, which capture the mean and covariance of the state distribution before and after a non-linear transformation. The Unscented Transform has three main steps, which are tuned for the suggested values of $\alpha$, $\beta$, $k$:

1. **Generation of the $\boldsymbol{\chi}$-points:** The number of sigma points depends on the system state dimension $N$. For this case ($N = 5$), the value of $\lambda$ determines the sigma points:

$$\boldsymbol{\chi}_0 = \hat{\mathbf{x}}, \quad \boldsymbol{\chi}_i = \hat{\mathbf{x}} + \sqrt{(N+\lambda)\mathbf{P}}_i, \quad \boldsymbol{\chi}_{i+N} = \hat{\mathbf{x}} - \sqrt{(N+\lambda)\mathbf{P}}_i, \quad i = 1, \ldots, N$$

2. **Propagation of the $\boldsymbol{\chi}$-points:** Each sigma point is propagated through the system's dynamics; in this case, using the Planar Bicircular Restricted Four-Body Problem dynamics.

3. **Weighted sample mean and covariance:** The weights $W_i^{(m)}$ (mean) and $W_i^{(c)}$ (covariance) are calculated from $\alpha$, $\beta$, and $k$, and used to retrieve the state estimate and covariance at each time step:

$$W_0^{(m)} = 1 - \frac{N}{\alpha^2 N}, \quad W_0^{(c)} = (2 - \alpha^2 + \beta) - \frac{N}{\alpha^2 N}, \quad W_i^{(m)} = W_i^{(c)} = \frac{1}{2\alpha^2 N}, \quad i = 1, \ldots, 2N$$

$$\hat{\mathbf{x}}_k = \sum_{i=0}^{2N} W_i^{(m)}\boldsymbol{\chi}_i, \quad \mathbf{P}_k = \sum_{i=0}^{2N} W_i^{(c)}(\boldsymbol{\chi}_i - \hat{\mathbf{x}}_k)(\boldsymbol{\chi}_i - \hat{\mathbf{x}}_k)^\top$$

The comparison between the Linear Covariance and Unscented Transform is carried out by analysing the mean and the error ellipses associated with the position elements of the covariances obtained at the final time.‡

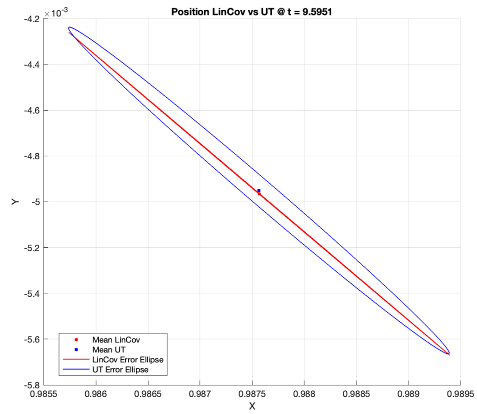


**Figure 1:** Position Covariance Ellipses LinCov vs UT

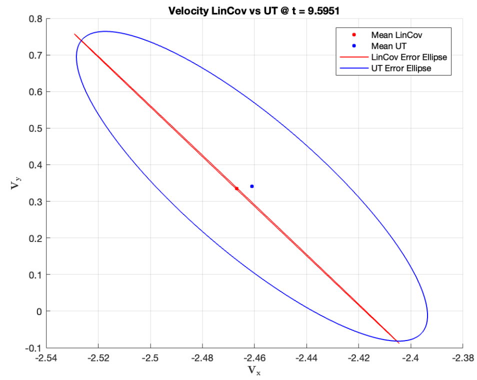


**Figure 2:** Velocity Covariance Ellipses LinCov vs UT

### 1.2 Perform the same uncertainty propagation process on the same time grid using a Monte Carlo

Finally, a Monte Carlo analysis was carried out; contrary to the Unscented Approach, where only a small portion of points is considered, a large population of 1000 samples is drawn from the nominal state $\boldsymbol{x_0}$ and nominal covariance $\boldsymbol{P_0}$, following a Gaussian distribution. Each sample is then propagated, assuming Planar Bicircular Restricted Four-Body dynamics, for the defined time grid. At each epoch, the mean of the state and covariance is computed.
The results from this computation are plotted against the previously found for the former methods, together with the all the samples generated during the simulation, for the position elements and the velocity elements:

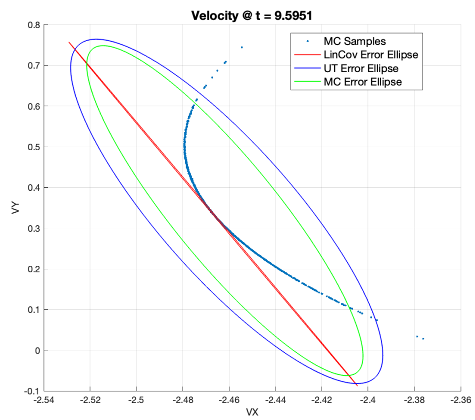


**Figure 3:** Position Covariance Ellipses LinCov vs UT vs MC

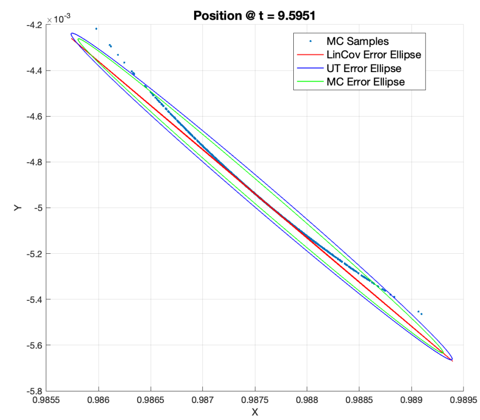


**Figure 4:** Velocity Covariance Ellipses-LinCov vs UT vs MC

Furthermore, the evolution in time of the $3\sqrt{\max(\lambda_i(P))}$, for position and velocity elements, is plotted. They typically represent an estimate of the uncertainty in the position and velocity

‡The error ellipses were drawn considering a $3\sigma$ confidence range.

estimates: the higher, the higher the uncertainty.

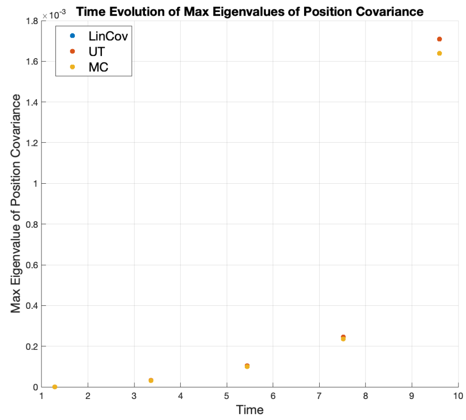


**Figure 5:** $3\sqrt{\max(\lambda_i(P_{pos}))}$ Evolution

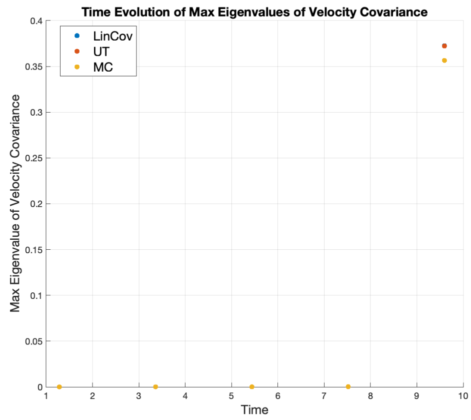


**Figure 6:** $3\sqrt{\max(\lambda_i(P_{vel}))}$ Evolution

Lastly, Quantile-Quantile plots, comparing the distribution of the Monte Carlo samples at the final time step with a theoretical normal distribution, are shown for each of the four states.

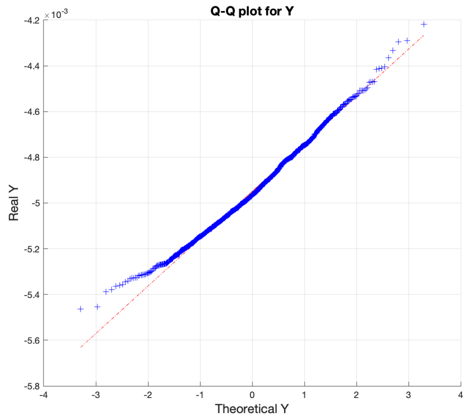


**Figure 7:** Q-Q for $X$

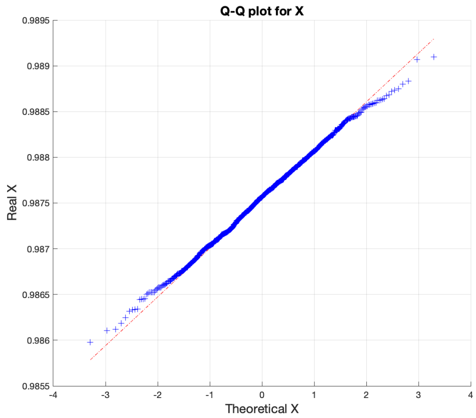


**Figure 8:** Q-Q for $Y$

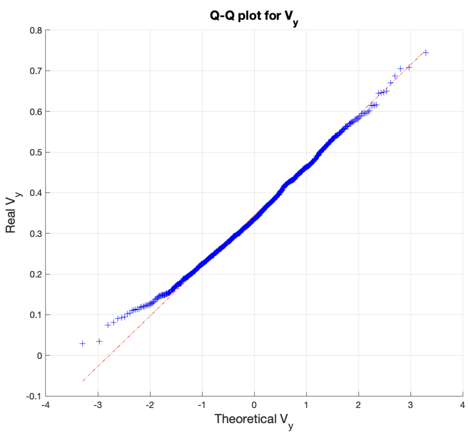


**Figure 9:** Q-Q for $V_y$

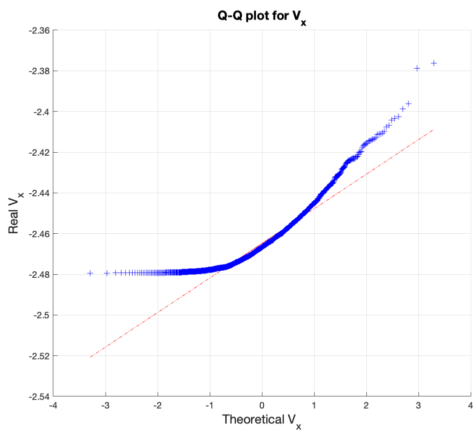


**Figure 10:** Q-Q for $V_x$

### 1.2.1 Conclusions

- At the end of the propagation, the MC samples, when plotted in the position subspace at the final time, did not scatter uniformly in an elliptical fashion but instead tended to lie along a curved manifold. This curved structure is a direct indication of the nonlinearity of the underlying dynamics.
- LinCov, which relies on first-order approximations, inherently assumes that uncertainties evolve linearly around a nominal trajectory, which is not the case in a PBRFBP.
- The Unscented Transform performed better in capturing some of this nonlinearity, providing a covariance estimate and corresponding ellipses that better approximated the actual spread of the MC samples.
- The Q-Q plots of the MC samples at the final time show divergence from the straight line that would correspond to a perfect match with a Gaussian distribution. For $V_x$ component, this divergence was more pronounced, indicating the emergence of non-Gaussian features such as skewness or heavier tails. This was confirmed by performing a *Kolmogorov–Smirnov* test, using the function `ktest`, which, assuming `rng(42)`, yielded negative for the $V_x$ component, indicating that they were not normally distributed.11

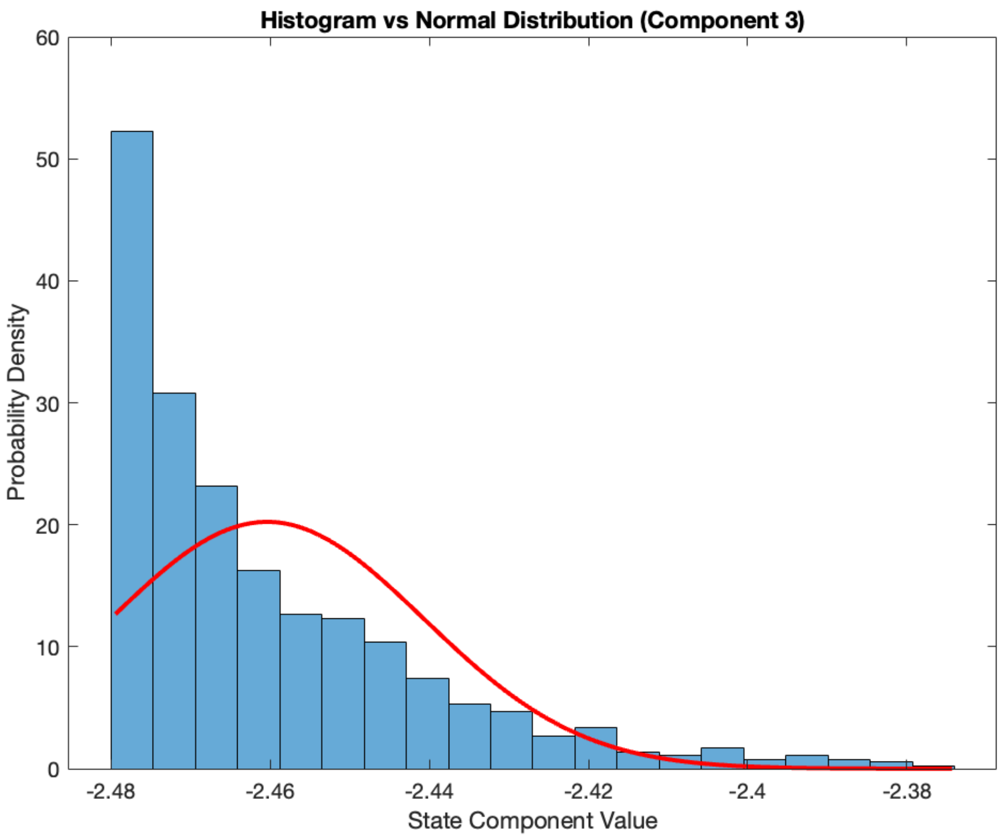


**Figure 11:** PDF for $V_x$

## 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 fulfil 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°<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 (monostatic) | Radar (monostatic) | Radar (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 (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 3:** TLE of SMOS.

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

## 2.1 Compute visibility windows

The main focus of the task is to track the SMOS (Soil Moisture and Ocean Salinity) satellite in order to improve the accuracy of its state estimate. This is accomplished by taking ideal measurements of its the elevation, azimuth and range, simulating real measurements by perturbing the former ones and then solving the navigation problem by means of least-squares.
Above all, the Two-Line Elements (TLE) of the satellite are used to determine its orbital parameters and, after converting the state from TEME to ECI, the TLE position and velocity of SMOS. As the epoch $t_{ref}$ of the TLE is `2024-11-18T18:15:16.064`, we need to propagate the state, assuming Keplerian motion, until the starting epoch $t_0$. From this point, we propagate again until the final epoch $t_f$, simultaneously gathering measurements from the three ground stations. Following are the osculating orbital elements, computed from the state vector in the TEME reference frame and $t_{ref}$:

**Table 4:** Osculating Orbital Elements at $t_{ref}$

| Element | Symbol | Value |
|---|---|---:|
| Semi-major axis | $a$ [km] | 7134.8293 |
| Eccentricity | $e$ [-] | 0.0012436 |
| Inclination | $i$ [rad] | 1.7180 |
| RAAN | $\Omega$ [rad] | 2.5913 |
| Argument of Perigee | $\omega$ [rad] | 1.2094 |
| True Anomaly | $\nu$ [rad] | 5.0761 |
| Epoch Time | $t$ [s] | 785225785.2475 |
| Gravitational Parameter | $\mu$ [km$^3$/s$^2$] | 398600.8 |

### 2.1.1 The Ground Stations

The three stations are placed in three strategic places:

- **Kourou:** is located near the equator, in French Guiana. It has a frequency of measurements of one per minute, having a minimum elevation over the horizon of 6 deg. It has a cost per pass of 30000€.
- **Troll:** resides in Antarctica, having easy tracking for polar orbits. It is able to provide double the measurement frequency with respect to the other stations, with two measurements each minute, and zero degree minimum elevation. However, it has the joint highest cost per pass of 35000€.
- **Svalbard:** is located near the Arctic region, also having easy tracking for satellites in polar orbits. Same frequency of measurements acquisition as Kourou, 8 degrees of minimum elevation and cost per pass of 35000€.

the three stations have the same type of Radar measurements, yielding the same amount of measurements noise. After the position of the satellite relative to the station is transformed from the ECI reference frame to the topocentric frame of the ground station, the function `cspice_xfmsta` is used to find range, azimuth and elevation. The measurements were then filtered to include only those acquired when the satellite was above the minimum elevation.

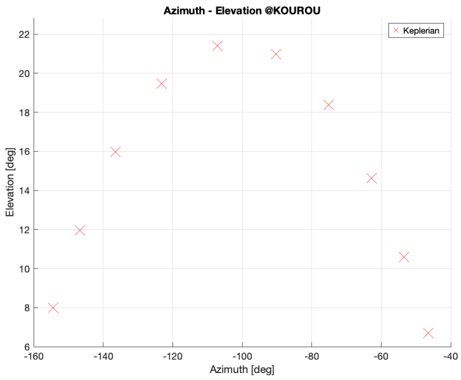


**Figure 12:** Az - El @ Kourou w/ Keplerian Motion

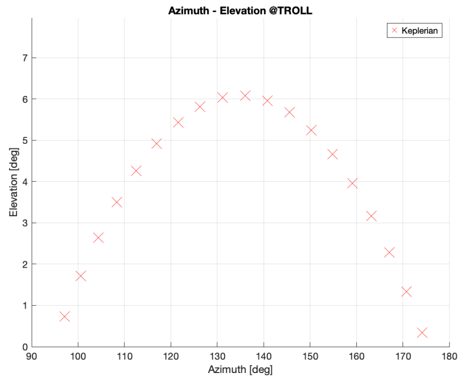


**Figure 13:** Az-El @ Troll w/ Keplerian Motion

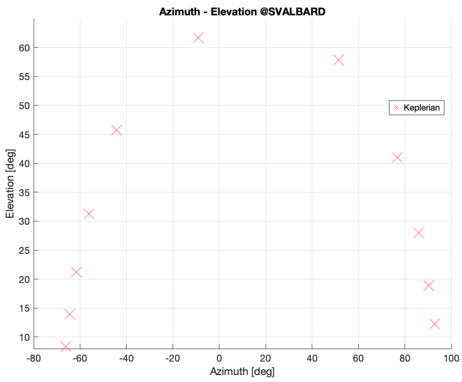


**Figure 14:** Az-El @ Svalbard w/ Keplerian Motion

| Station | Start | End |
|---|---|---|
| Kourou | 18-11-2024 20:40:00 | 18-11-2024 20:49:00 |
| Troll | 18-11-2024 21:02:30 | 18-11-2024 21:11:30 |
| Svalbard | 18-11-2024 21:56:00 | 18-11-2024 22:06:00 |

**Table 5:** Visibility Windows

## 2.2 Simulate measurements

The propagation from $t_0$ to $t_f$ of the initial state found at $t_0$ was repeated using SGP4. At each time step, the state is converted from True Equator Mean Equinox into ECI, using the function `teme2eci`, and the measurements are computed as in the previous point. Then, the latter were then perturbed by adding a random Gaussian noise centred around each nominal value: these were filtered by only including the values acquired during the visibility window.

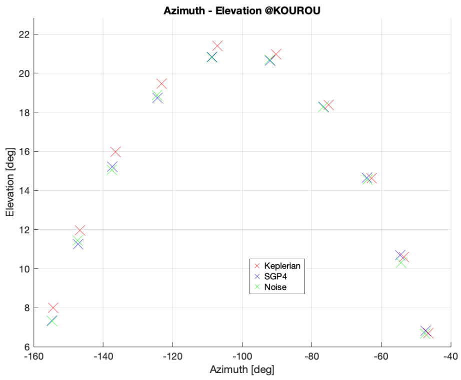


**Figure 15:** Az - El @ Kourou Kep vs SGP4 vs Noise

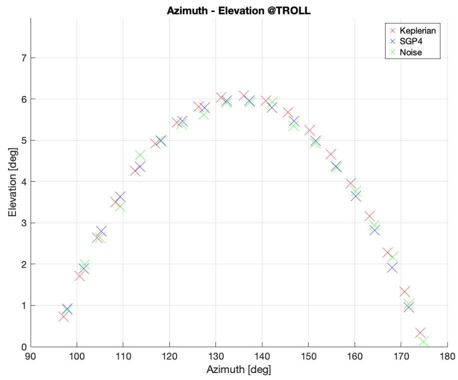


**Figure 16:** Az-El @ Troll Kep vs SGP4 vs Noise

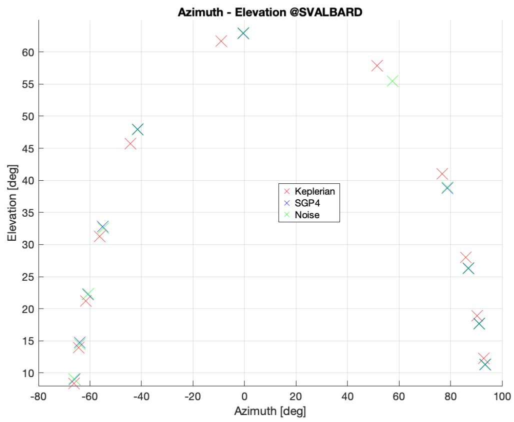


**Figure 17:** Az-El @ Svalbard Kep vs SGP4 vs Noise

## 2.3 Solve the navigation problem.

The main task is to solve the *navigation problem*, which is the process of determining the best satellite's state at each time step based on available, noisy measurements and a mathematical model of its motion. In mathematical terms, the most accurate estimate of the satellite's state, $\hat{\boldsymbol{x}}_{\boldsymbol{k}}$, by minimizing the residuals, $\boldsymbol{\epsilon_i}$, between predicted and observed measurements over the time interval $t_0$ to $t_f$. The residuals represent the differences between the simulated real measurements $y_{\text{real}}$ and the predicted measurements $y_{\text{predicted}}$, which comes from the propagator. For a linearized system, the relationship between these quantities can be expressed as:

$$\boldsymbol{\delta y} = \boldsymbol{H}\boldsymbol{\delta x_k} + \boldsymbol{\epsilon}, \tag{1}$$

where $H$ is the Jacobian matrix, containing the partial derivatives of the measurements with respect to the state variables. To solve the problem, the MATLAB function `lsqnonlin` was utilized. This function minimizes the weighted square of the residuals, defined as:

$$\boldsymbol{\epsilon} = \boldsymbol{W_m}\left(y_{\text{predicted}} - y_{\text{real}}\right), \tag{2}$$

where $\boldsymbol{W_m}$ is the weight matrix, a diagonal matrix made up of the inverse of the measurements noises.

Firstly, the measurements from only *Kourou* were exploited, as well as using a perturbation-free Keplerian propagation: the accuracy of the solution was computed in terms of position and velocity. The same process was repeated using measurements from all three stations and, lastly, making use of a more accurate Keplerian propagation which takes into account also the J2 effects.

| **Station & Model** | **Position Vector [km]** | **Velocity Vector [km/s]** |
|---|---|---|
| Kourou & Kep | [3932.199, −1415.334, 5780.094] | [4.88209, −3.76256, −4.23679] |
| All & Kep | [3926.839, −1411.020, 5780.049] | [4.88860, −3.76497, −4.23333] |
| All & J2 | [3932.782, −1414.926, 5778.496] | [4.87978, −3.76321, −4.23288] |

**Table 6:** Estimated Position and Velocity at $t_0$ @ ECI

In addition, in order to estimate the uncertainty on the semi-major axis and inclination, the covariance matrix of a satellite's state in Cartesian coordinates has to be mapped to the covariance matrix of the Keplerian orbital elements. This is done by finding the matrix $\boldsymbol{T}$, which describes how small changes in Cartesian state variables affect the Keplerian elements.

$$\epsilon = \frac{\boldsymbol{v^2}}{2} - \frac{\mu}{\boldsymbol{r}} \qquad a = -\frac{\mu}{2\epsilon} \qquad \mathbf{T} = \frac{\partial(a,i)}{\partial(x,y,z,v_x,v_y,v_z)}$$

$$\boldsymbol{H} = \boldsymbol{r} \times \boldsymbol{v} \qquad i = \arccos\left(\frac{H_z}{\boldsymbol{H}}\right) \qquad \boldsymbol{P_{kep}} = \boldsymbol{T P_{cart} T^T}$$

The results are presented in 7, where, among the state vector and square root of the trace of the estimated covariance submatrix of the former components, are also present the standard deviations associated to the semi-major axis and inclination. [¶]

| **Station & Model** | $\sqrt{tr(\boldsymbol{P_r})}$ **[km]** | $\sqrt{tr(\boldsymbol{P_v})}$ **[km/s]** | $\boldsymbol{\sigma_a}$ **[km]** | $\boldsymbol{\sigma_i}$ [°] |
|---|---|---|---|---|
| Kourou & Kep | 4.4222 | 0.0044 | 2.5505 | 0.03777 |
| All & Kep | 0.16846 | 0.00019 | 0.01071 | 0.000268 |
| All & J2 | 0.0039 | $10^{-7}$ | 0.0007 | $10^{-7}$ |

**Table 7:** Stations Traces & Uncertainties on $a$ & $i$

The results can be summarized in few points:

- The navigation solution using only Kourou observations and Keplerian propagation have modest accuracy and comparatively high uncertainty.
- When combining measurements from all three stations with Keplerian propagation, the solution improves significantly in confidence, but the estimated position error norm rises.
- With full-station visibility and J2 perturbation modelling the position error drops to just 16 meters, and the velocity error to 2 cm/s, while the position covariance trace is only 3.9 meters.
- The navigation solution is provided in the ECI frame.

[¶] The covariance matrices are instead available in B.

## 2.4 Trade-off analysis

The trade-off analysis is focused on identifying the most effective combination of two ground stations (out of the three available) for orbit determination, under the constraint that the overall pass cost should not exceed €70.000. The problem was formulated as a minimization task where the optimality criterion is not based on just one metric (e.g., lowest uncertainty), but instead considers a weighted sum of three factors: the total operational cost, the standard deviation of the semi-major axis, and the standard deviation of the inclination. The assigned weights were chosen in order to bring all three components into a comparable range of magnitude (e.g: as the uncertainty on the inclination was generally orders of magnitude lower with respect to the others, an higher weight was assigned). From 9, the choice falls on a combination of Kourou & Svalbard, which yield the lowest joint cost and best performances in terms of standard deviations, with three stations achieving, in the time span considered, only one pass each.

| Stations | $\sigma_a$ [km] | $\sigma_i$ [°] | Cost [€] | Score |
|---|---|---|---|---|
| Kourou & Troll | 0.07018 | 0.000172 | 65000 | 72 |
| Kourou & Svalbard | 0.00145 | 0.000087 | 65000 | 65 |
| Svalbard & Troll | 0.02494 | 0.0004 | 70000 | 73 |

**Table 8:** Trade-Off Scores

## 2.5 Long Term Analysis

Over the 72-hour simulation using SGP4 propagation, of the three candidate ground stations, Troll emerges as the most capable station, with a total of 43 passes and the longest average pass duration at 11.5 minutes. This suggests a combination of both frequent contact opportunities and sustained tracking time, making Troll the clear choice for the prime ground station. Svalbard closely follows Troll, with 40 passes over the same period. Its average pass duration is slightly shorter at 7.8 minutes, but provides dense coverage in the polar regions; thus, it is recommended as the backup station. This yields a difference outcome from the previous point, in which Kourou and Svalbard were the chosen station. [‖]

| Stations | Number of Passes | Average Pass Duration [min] |
|---|---|---|
| Kourou | 10 | 8.9 |
| Troll | 43 | 11.5 |
| Svalbard | 40 | 7.8 |

**Table 9:** Long Term Analysis

[‖] Elevation profiles for three stations are available in A

# 3: Sequential filters

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

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

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

1. *Check the visibility window.* Considering the initial state $\mathbf{x}_0$ and the time interval with a time-step of 30 seconds from $t_0$ to $t_f$ reported in Table 10, predict the trajectory of the satellite in an inertial Moon-centered reference frame assuming Keplerian motion. Use the estimated coordinates given in Table 11 to predict the state of the lunar lander. Finally, check that the lander and the orbiter are in relative visibility for the entire time interval.

2. *Simulate measurements.* Always assuming Keplerian motion to model the lunar orbiter dynamics around the Moon, compute the time evolution of its position vector in an inertial Moon-centered reference frame and the time evolution of the relative range between the satellite and the lunar lander. Finally, simulate the measurements by adding a random error to the spacecraft position vector and to the relative range. Assume a Gaussian model to generate the random error, with noise provided in Table 10 for both the relative range and the components of the position vector. Verify (graphically) that the applied noise level is within the desired boundary.

3. *Estimate the lunar orbiter absolute state.* As a first step, you are asked to develop a sequential filter to narrow down the uncertainty on the knowledge of the lunar orbiter absolute state vector. To this aim, you can exploit the measurements of the components of its position vector computed at the previous point. Using an Unscented Kalman Filter (UKF), provide an estimate of the spacecraft state (in terms of mean and covariance) by sequentially processing the acquired measurements in chronological order. To initialize the filter in terms of initial covariance, you can refer to the first six elements of the initial covariance $\mathbf{P}_0$ reported in Table 10. For the initial state, you can perturb the actual initial state $\mathbf{x}_0$ by exploiting the MATLAB function `mvnrnd` and the previously mentioned initial covariance. We suggest to use $\alpha = 0.01$ and $\beta = 2$ for tuning the UT in this case. Plot the time evolution of the error estimate together with the $3\sigma$ 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 10. For the initial state, you can perturb the actual initial state, composed by $\mathbf{x}_0$ and the latitude

and longitude given in Table 11, by exploiting the MATLAB function `mvnrnd` and the previously mentioned initial covariance. We suggest to use $\alpha = 0.01$ and $\beta = 2$ for tuning the UT in this case. Plot the time evolution of the error estimate together with the $3\sigma$ of the estimated covariance for both position and velocity.

**Table 10:** Initial conditions for the lunar orbiter.

| Parameter | Value |
|---|---|
| Initial state $\mathbf{x}_0$ [km, km/s] | $\mathbf{r}_0$ = [4307.844185282820, -1317.980749248651, 2109.210101634011]<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 11:** 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 window

The trajectory of the lunar orbiter was predicted using Keplerian motion, assuming a Moon-centered inertial frame. The initial state vector of the orbiter, defined by its position $\boldsymbol{v_0}$ and velocity $\boldsymbol{r_0}$, was propagated over the time interval from $t_0$ to $t_f$ using non-perturbed Keplerian propagation. The time interval spanned 4 hours with a time step of 30 seconds. The lander position was considered static in time for this point, and can be estimated from the latitude and longitude in 11, using `cspice_pgrrec`, which yielded:

| Component | Value |
|---|---|
| X | 348.917 km |
| Y | 93.492 km |
| Z | 1699.434 km |

**Table 12:** Lander Position @ MCIF

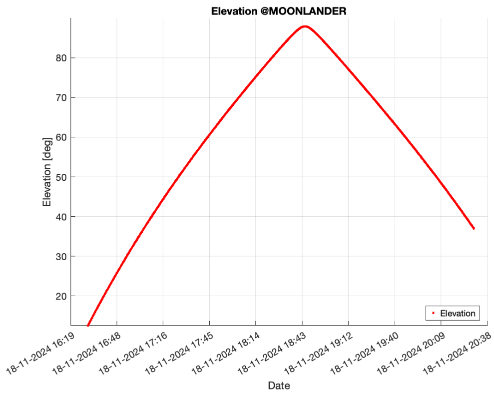


**Figure 18:** Elevation Visibility

Furthermore, as it is clear from 18, the orbiter was ensured to be always visible within the given time frame.

## 3.2 Simulate measurements

The time evolution of the orbiter position vector in a MCMF (Moon Centered Moon Fixed) frame and the relative range between the Moonlander and satellite, was then computed. Differently from the previous point, the position of the lander wasn't computed from the estimated coordinates, but rather from exploiting its kernel: thanks to the function `cspice_spkezr` the position and velocity of the lander was available at each time step. The relative range was then computed as the norm of the difference between the satellite position and the lander one.
Lastly, to simulate measurements, both the position vector components and relative range, were corrupted by Gaussian noise, with standard deviation of 0.1 km. The noise was then graphically verified to stay with the bounds.

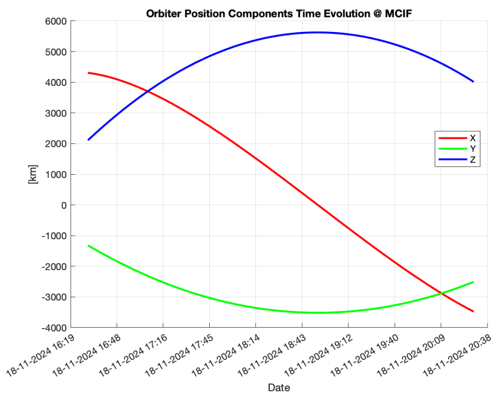


**Figure 19:** Orbiter's PosVec @ MCIF

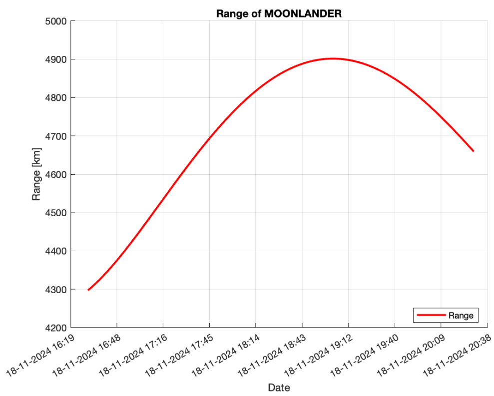


**Figure 20:** Relative Range Moonlander

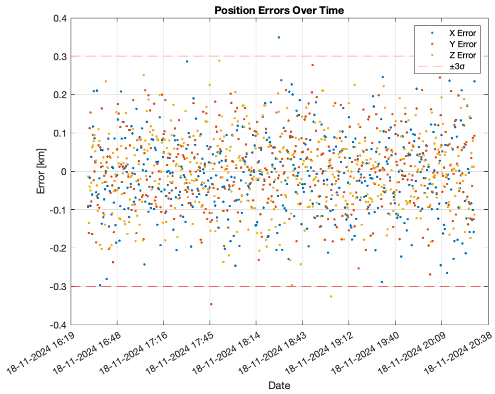


**Figure 21:** Orbiter's Position Vector Noise

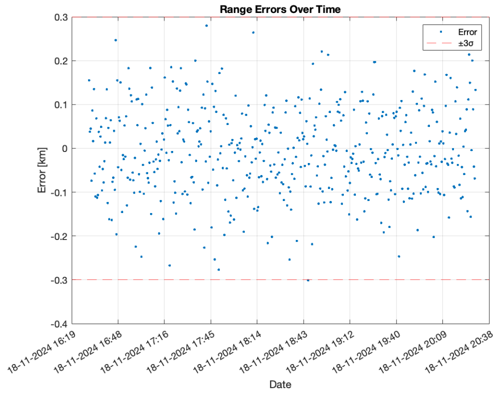


**Figure 22:** Orbiter's Range Noise

### 3.3 Estimate the lunar orbiter absolute state

The state vector $\mathbf{x}$ of the lunar orbiter consists of its position and velocity components, where $\mathbf{r} = [r_x, r_y, r_z]$, $\mathbf{v} = [v_x, v_y, v_z]$, therefore being a *6 states* vector. The main objective is to sequentially process the acquired measurements, which are the components of only the position vector, in chronological order to update the spacecraft's six components state, and, by doing so, narrow down the uncertainty over time. This can be accomplished by making use of an Unscented Kalman Filter, whose steps are detailed below:

1. **Initialization:** The initial state of the orbiter is perturbed using the MatLab function `mvrnd`, which takes in input the mean state $\mathbf{x}_0$ and the first six rows & columns of $\mathbf{P}_0$.

2. $\boldsymbol{\chi}$ **Points Generation & Propagation:** At each discretized time step, a new set of $\boldsymbol{\chi}$ is generated, using the suggested values of $\alpha$, $\beta$, $k$. Outside the loop, the *weights* are computed, taking as dimension of the state $n = 6$. Each point is then propagated through the dynamics of the system.

3. **Measurement Function:** The measurement function is then applied to the propagated points. In this case, the measurement function takes the first three rows of the $\boldsymbol{\chi}$ points, as the measurements are the position vector components of the orbiter, which are now called $\boldsymbol{\gamma}$ points.

4. **Prediction Step:** The predicted state mean and covariance for the $\boldsymbol{\chi}$ points is computed as following:

$$\mathbf{x}^- = \sum_{i=0}^{2n} W_i^m \hat{\boldsymbol{\chi}}_i \qquad \mathbf{P}^- = \sum_{i=0}^{2n} W_i^c (\hat{\boldsymbol{\chi}}_i - \mathbf{x}^-)(\hat{\boldsymbol{\chi}}_i - \mathbf{x}^-)^\top$$

   The same is done for the measurements $\boldsymbol{\gamma}$ points, introducing the measurement noise covariance $\mathbf{R}$:

$$\mathbf{y}^- = \sum_{i=0}^{2n} W_i^m \boldsymbol{\gamma}_i \qquad \mathbf{P}_{ee} = \sum_{i=0}^{2n} W_i^c (\boldsymbol{\gamma}_i - \mathbf{y}^-)(\boldsymbol{\gamma}_i - \mathbf{y}^-)^\top + \mathbf{R}$$

   In addition, the cross-variance matrix is computed:

$$\mathbf{P}_{xy} = \sum_{i=0}^{2n} W_i^c (\hat{\boldsymbol{\chi}}_i - \mathbf{x}^-)(\boldsymbol{\gamma}_i - \mathbf{y}^-)^\top$$

5. **Update Step:** In the update step the best estimate for the subsequent time is computed, before starting the cycle once again for the next time step. The covariance matrix $\mathbf{P}_0$ is used to compute a new set of $\boldsymbol{\chi}$ points

$$\mathbf{K}_k = \mathbf{P}_{xy}\mathbf{P}_{yy}^{-1}$$
$$\mathbf{x}^+ = \mathbf{x}^- + \mathbf{K}_k(\mathbf{y}_k - \mathbf{y}^-)$$
$$\mathbf{P}^+ = \mathbf{P}^- - \mathbf{K}_k\mathbf{P}_{ee}\mathbf{K}_k^\top$$

The time evolution of the error, together with the $3\sigma$ of the estimated covariance, is plotted: the figures below reveal a consistent pattern where the errors rapidly decrease over time, consistently staying below the $3\sigma$ threshold for both position and velocity. **

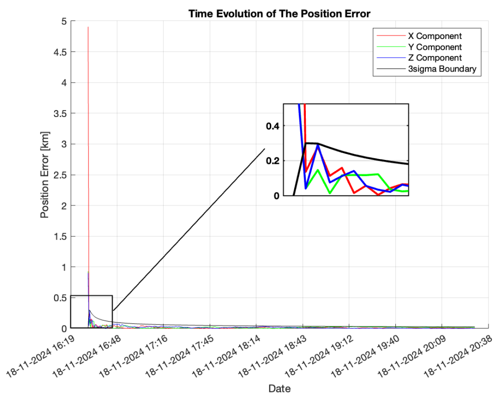


**Figure 23:** Position Error

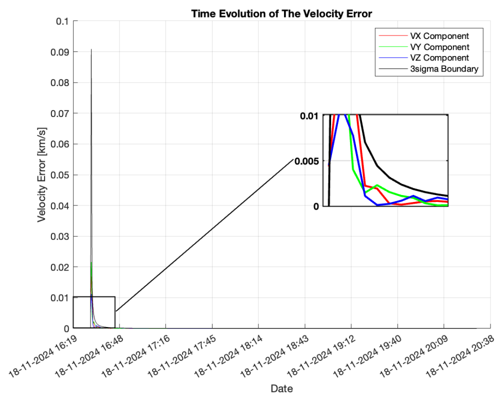


**Figure 24:** Velocity Error

**The $3\sigma$ boundary was chosen by selecting data from the second component ($Y$ & $V_Y$), but it was checked that the threshold wasn't crossed by using the other component's boundaries.

### 3.4 Estimate the lunar lander coordinates

The last point adds another complexity to the filter: the simulatenous estimation of the lander's position on the Moon's surface, described by its latitude and longitude, and the orbiter's position and velocity. The main differences from the previous point lie in:

- The state was augmented by adding the *Moonlander* position, represented by latitude and longitude, making it an 8 *state vector*.
- The measurement model now takes as input, at each step, the propagated $\chi$ points, together with latitude and longitude, and outputs the relative range between the lunar lander and the orbiter.
- Likewise the previous point, the time evolution for position and velocity error over time, always stays under the $3\sigma$ threshold, showing a decreasing behaviour.

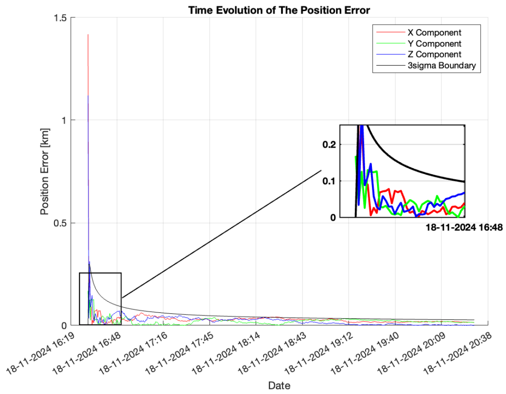


**Figure 25:** Position Error

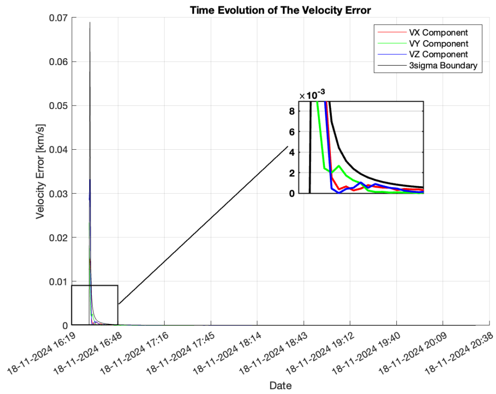


**Figure 26:** Velocity Error

# Appendices

## A: Image Appendix

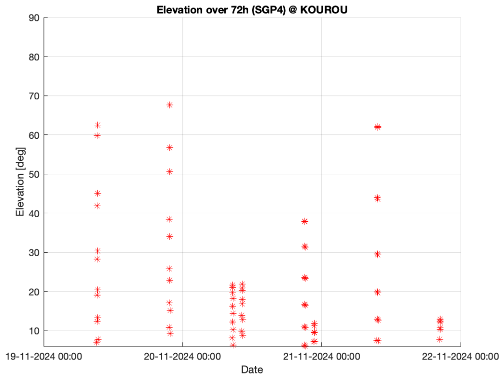


**Figure 27:** Elevation Over 72 Hours @ Kourou

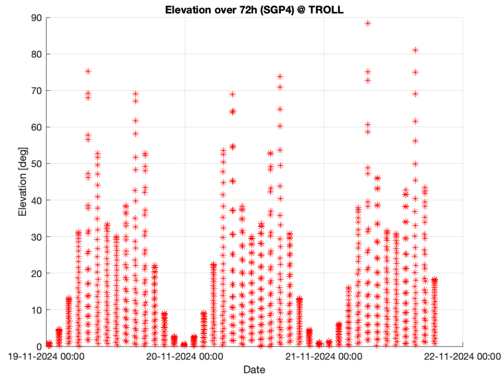


**Figure 28:** Elevation Over 72 Hours @ Troll

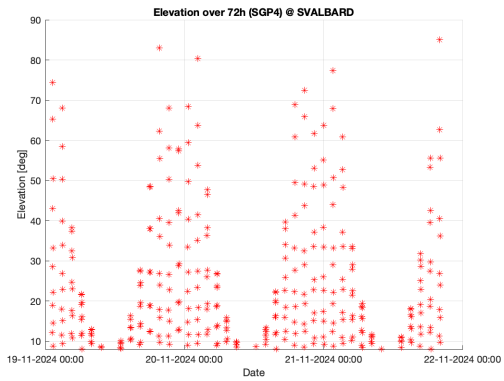


**Figure 29:** Elevation Over 72 Hours @ Svalbard

# Appendices

## B: Tables Appendix

**Table 13:** Cartesian covariance matrix $\mathbf{P}_a$ — Kourou measurements and Keplerian propagation.

$$\mathbf{P}_a = \begin{bmatrix} 12.3030 & 5.2794 & -2.6012 & -0.0121 & -0.0068 & -0.0051 \\ 5.2794 & 5.5915 & -3.0135 & -0.0043 & -0.0045 & -0.0005 \\ -2.6012 & -3.0135 & 1.6615 & 0.0021 & 0.0023 & 0.0001 \\ -0.0121 & -0.0043 & 0.0021 & 1.2051\text{e}{-5} & 6.2407\text{e}{-6} & 5.4522\text{e}{-6} \\ -0.0068 & -0.0045 & 0.0023 & 6.2407\text{e}{-6} & 4.6388\text{e}{-6} & 2.0745\text{e}{-6} \\ -0.0051 & -0.0005 & 0.0001 & 5.4522\text{e}{-6} & 2.0745\text{e}{-6} & 3.0570\text{e}{-6} \end{bmatrix}$$

**Table 14:** Cartesian covariance matrix $\mathbf{P}_b$ — All stations and Keplerian propagation.

$$\mathbf{P}_b = \begin{bmatrix} 1.9306\text{e}{-2} & -9.5890\text{e}{-3} & -4.9218\text{e}{-3} & -1.9391\text{e}{-5} & 7.0842\text{e}{-6} & -1.4096\text{e}{-5} \\ -9.5890\text{e}{-3} & 6.4838\text{e}{-3} & 3.1005\text{e}{-3} & 1.0613\text{e}{-5} & -2.9727\text{e}{-6} & 7.9858\text{e}{-6} \\ -4.9218\text{e}{-3} & 3.1005\text{e}{-3} & 2.5870\text{e}{-3} & 4.3778\text{e}{-6} & -1.6520\text{e}{-6} & 4.5158\text{e}{-6} \\ -1.9391\text{e}{-5} & 1.0613\text{e}{-5} & 4.3778\text{e}{-6} & 2.1222\text{e}{-8} & -6.3815\text{e}{-9} & 1.4336\text{e}{-8} \\ 7.0842\text{e}{-6} & -2.9727\text{e}{-6} & -1.6520\text{e}{-6} & -6.3815\text{e}{-9} & 3.3322\text{e}{-9} & -4.9519\text{e}{-9} \\ -1.4096\text{e}{-5} & 7.9858\text{e}{-6} & 4.5158\text{e}{-6} & 1.4336\text{e}{-8} & -4.9519\text{e}{-9} & 1.1315\text{e}{-8} \end{bmatrix}$$

**Table 15:** Cartesian covariance matrix $\mathbf{P}_c$ — All stations and J2 propagation.

$$\mathbf{P}_c = \begin{bmatrix} 8.2086\text{e}{-6} & -5.3829\text{e}{-6} & -1.3737\text{e}{-6} & -7.9816\text{e}{-9} & 3.1784\text{e}{-9} & -5.4028\text{e}{-9} \\ -5.3829\text{e}{-6} & 5.7408\text{e}{-6} & 1.4257\text{e}{-6} & 6.2429\text{e}{-9} & -1.5247\text{e}{-9} & 4.4276\text{e}{-9} \\ -1.3737\text{e}{-6} & 1.4257\text{e}{-6} & 1.2434\text{e}{-6} & 8.0354\text{e}{-10} & -4.4223\text{e}{-10} & 1.5629\text{e}{-9} \\ -7.9816\text{e}{-9} & 6.2429\text{e}{-9} & 8.0354\text{e}{-10} & 9.1965\text{e}{-12} & -2.4881\text{e}{-12} & 5.3434\text{e}{-12} \\ 3.1784\text{e}{-9} & -1.5247\text{e}{-9} & -4.4223\text{e}{-10} & -2.4881\text{e}{-12} & 1.8337\text{e}{-12} & -1.9438\text{e}{-12} \\ -5.4028\text{e}{-9} & 4.4276\text{e}{-9} & 1.5629\text{e}{-9} & 5.3434\text{e}{-12} & -1.9438\text{e}{-12} & 4.2881\text{e}{-12} \end{bmatrix}$$