SGN – Assignment #2

Mina Baniamein, 232521

Exercise 1: Uncertainty propagation

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

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

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

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

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

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

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

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

†Use at least 1000 samples drawn from the initial covariance

1.1 Point 1

1.1.1 LinCov

The problem is subject to the following dynamic:

$$\begin{cases} \dot{\boldsymbol{x}} = \mathbf{f}(\boldsymbol{x}, t) \\ \boldsymbol{x}(t_0) = \boldsymbol{x_0} \end{cases} \tag{1}$$

with $\mathbf{f}(\boldsymbol{x}, t)$ representing the dynamics of the Planar Bicircular Restricted Four-Body Problem. The LinCov approach is based on two simplifying hypothesis:

- Uncertainties can be described accurately by a gaussian distribution
- A linearized dynamic is considered to be accurate enough to describe the motion of displaced trajectories

Under this assumptions, the propagated mean and covariance $\hat{\boldsymbol{x}}(t)$ and $\boldsymbol{P}$ can be computed as

$$\begin{cases} \hat{x}(t) = \boldsymbol{x}^*(t) \\ \boldsymbol{P} = \boldsymbol{\Phi}(t_0, t) \cdot \boldsymbol{P}_0 \cdot \boldsymbol{\Phi}^T(t_0, t) \end{cases} \tag{2}$$

Where $\boldsymbol{x}^*(t)$ is the mean at the final distribution, $\boldsymbol{P}_0$ is the initial covariance and $\boldsymbol{\Phi}(t_0, t)$ is the State Transition Matrix.

Eq. (2) shows that in the LinCov approach, the propagated mean coincided with the mean at the final distribution, while the covariance $\boldsymbol{P}$ is obtained using the State Transition Matrix $\boldsymbol{\Phi}(t_0, t)$. The method could lose accuracy for high non-linearities in the dynamical system or with long propagation time.

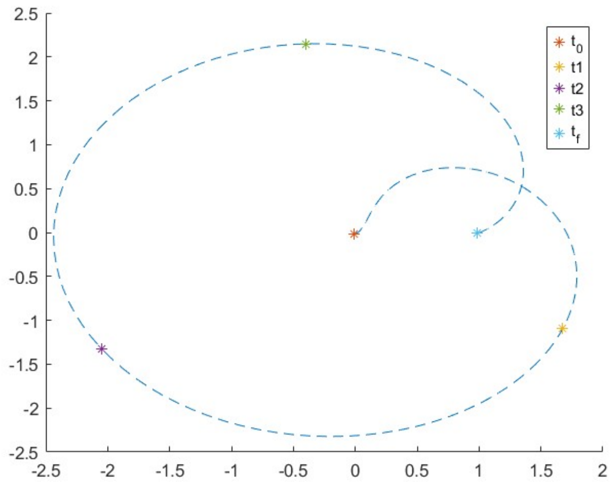

Figure 1: Propagated LinCov points in the rotating frame compared with PBRFBP propagated trajectory from initial mean state

1.1.2 UT

The Unscented Transform approach computes the propagated mean and covariance through a set of weighted samples called sigma points in 3 steps:

- **Step 1** Creation of $2n+1$ sigma points from $\hat{\boldsymbol{x}}$ and P_x and their weights.

$$\begin{cases} \boldsymbol{\chi}_0 = \hat{\boldsymbol{x}} \\ \boldsymbol{\chi}_i = \hat{\boldsymbol{x}} + \left(\sqrt{(n+\lambda)\mathbf{P}_\mathbf{x}}\right)_i \quad (i = 1, \dots, n) \\ \boldsymbol{\chi}_i = \hat{\boldsymbol{x}} - \left(\sqrt{(n+\lambda)\mathbf{P}_\mathbf{x}}\right)_i \quad (i = 1+n, \dots, 2n) \end{cases} \tag{3}$$

where $n = 6$, and $\lambda = \alpha^2(n+k) - n$ with $\alpha = 0.1$, $k = 0$. While the weights are computed as follows.

$$\begin{cases} W_0^{(m)} = \dfrac{\lambda}{n+\lambda} \\ W_0^{(c)} = \dfrac{\lambda}{n+\lambda} + 1 - \alpha^2 + \beta \\ W_i^{(m)} = W_i^{(c)} = \dfrac{1}{2(n+\lambda)} \quad (i = 1, \dots, 2n) \end{cases} \tag{4}$$

With $\beta = 2$, and the notation (m) indicating to weights used for the mean while (c) is used for the covariance.

- **Step 2** Propagation of sigma points using a Keplerian propagator.

$$\mathbf{Y}_i = \mathbf{g}(\mathbf{x_i}) \quad (i = 1, \dots, 2n)$$

- **Step 3** Computation of the sample mean and sample covariance.

$$\begin{cases} \hat{\mathbf{y}} = \sum_{i=0}^{2n} W_i^{(m)} \mathbf{Y}_i \\ \mathbf{P}_y = \sum_{i=0}^{2n} W_i^{(c)} (\mathbf{Y}_i - \hat{\mathbf{y}})(\mathbf{Y}_i - \hat{\mathbf{y}})^T \end{cases} \tag{5}$$

The UT does not rely on the hypothesis of Gaussian distribution and linearization, but rather calculates the propagated mean and covariance by integration of the full equation of motion for a relatively small set of points.

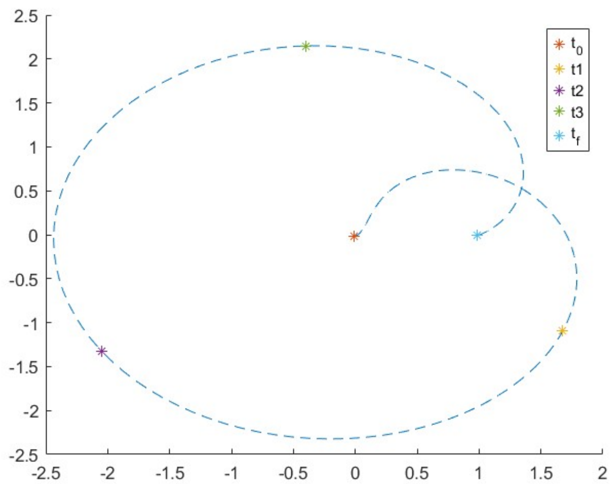

Figure 2: Propagated UT points in the rotating frame compared with PBRFBP propagated trajectory from initial mean state

1.1.3 Covariance ellipses

The points of a 2x2 covariance matrix P associated ellipse can be computed as:

$$\begin{bmatrix} x(\theta) \\ y(\theta) \end{bmatrix} = \begin{bmatrix} v_{1x} & v_{2x} \\ v_{1y} & v_{2y} \end{bmatrix} \begin{bmatrix} \sqrt{\lambda_1}\cos(\theta) \\ \sqrt{\lambda_2}\sin(\theta) \end{bmatrix} \tag{6}$$

where v_1 and v_2 are the eigenvectors of $\mathbf{P}$, λ_1 and λ_2 are the eigenvalues and θ is the parametric angle that locates the points in the plot. The following figure contains the plot of the ellipses associated to the position elements of $\mathbf{P}$ obtained at the final time with LinCov and UT method, as well as their mean states:

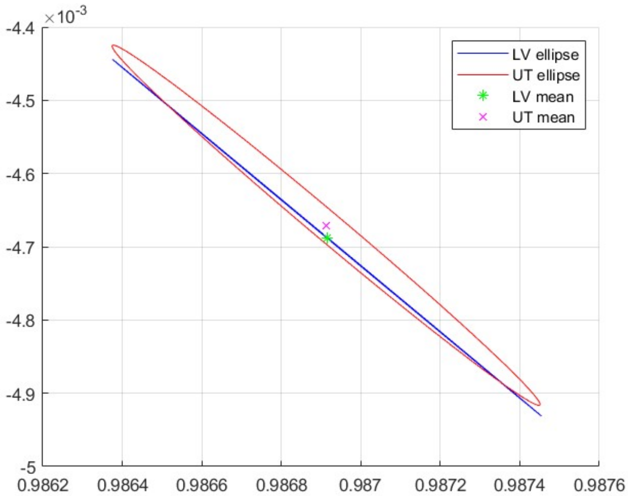

Figure 3: LinCov and UT mean states and ellipses associated with the position elements of the covariances

It can be noticed that LinCov method guarantees a more confident and restricted range of values for state estimation around it's mean state, meaning it's more reliable than the UT method. However it's still required to check if all the assumptions required for the LinCov method to work are verified.

1.2 Point 2

A Montecarlo simulation was also implemented to study the uncertainty propagation.

- **Step 1:** Generation of N gaussian random samples using the initial covariance P_0
- **Step 2:** Propagation of the samples using keplerian dynamics
- **Step 3:** Computation of sample mean and covariance

$$\begin{cases} \hat{\boldsymbol{x}} = \dfrac{1}{N}\displaystyle\sum_{i=1}^{N} \boldsymbol{x}_i \\ \boldsymbol{P} = \dfrac{1}{N-1}\displaystyle\sum_{i=1}^{N} (\boldsymbol{x}_i - \hat{\boldsymbol{x}})(\boldsymbol{x}_i - \hat{\boldsymbol{x}})^T \end{cases} \tag{7}$$

In figure 4 it can be noticed how there are not significative differences in terms of propagated samples with the propagated PBRFBP trajectory from initial mean state.

In figure 5 it can be observed how the MonteCarlo ellipse yields pretty similar results to the UT one, but with an higher computational effort.

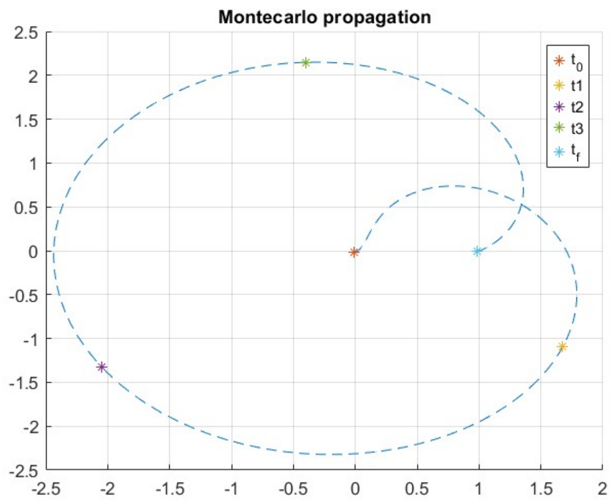

Figure 4: Propagated MC points in the rotating frame compared with PBRFBP propagated trajectory from initial mean state

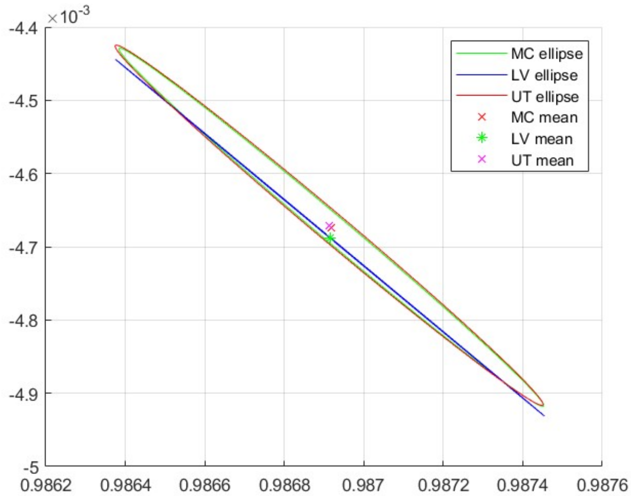

Figure 5: MC, LinCov and UT mean states and ellipses associated with the position elements of the covariances

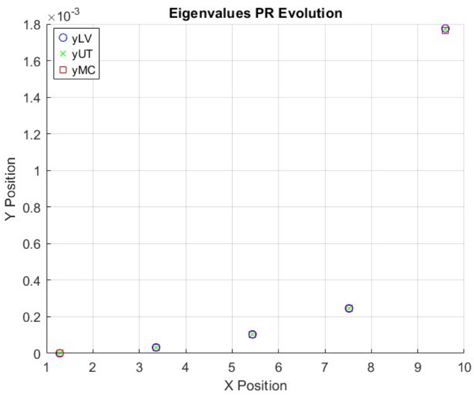

Figure 6: Time evolution of $\boldsymbol{P}_r$ eigenvalues

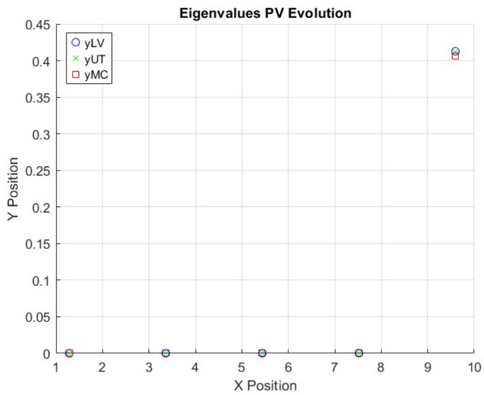

Figure 7: Time evolution of $\boldsymbol{P}_v$ eigenvalues

From the eigenvalues evolution in time of $\boldsymbol{P}_r$ (10) and $\boldsymbol{P}_v$ (11) it can observed how the uncertainty grows in time, and how the Montecarlo method seems to be a little bit more robust than the other methods in this sense (but not in a significative way).
Here follow the QQplots of the state components at the final time propagated with the Montecarlo method:

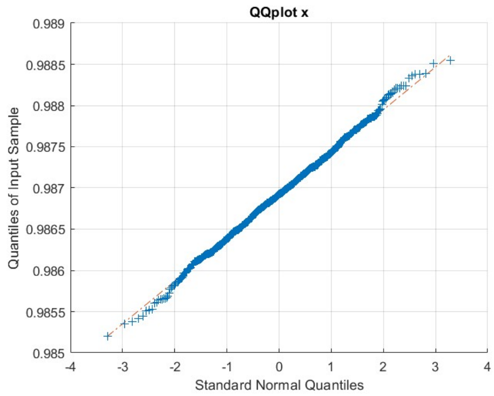

Figure 8: QQplot of x coordinate samples at final time

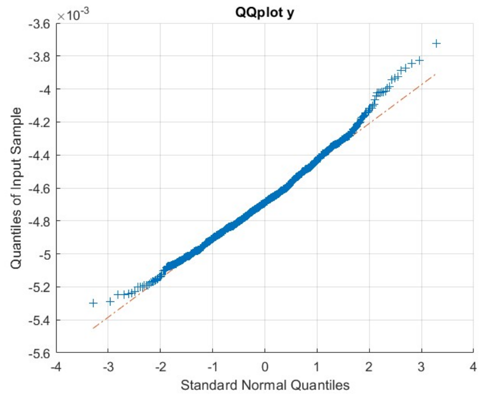

Figure 9: QQplot of y coordinate samples at final time

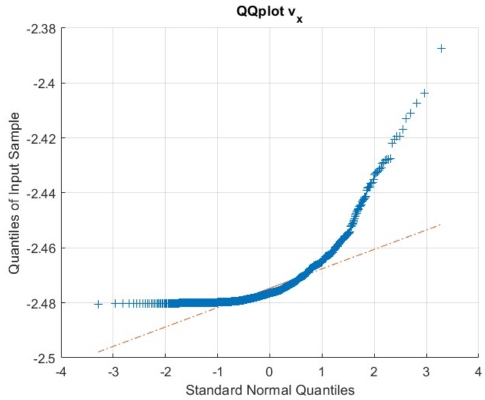

Figure 10: QQplot of v_x component samples at final time

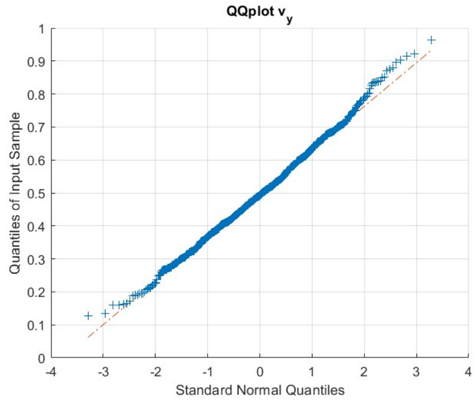

Figure 11: QQplot of v_y component samples at final time

The v_x component is the only one to exhibit a non-uniform distribution around the diagonal, meaning that uncertainty Gaussian assumption is not reliable for it's propagation. This means that the LinCov method it's not suited for correctly propagating the velocity in this case, while for the position it's reliable.

Exercise 2: Batch filters

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

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

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

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

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

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

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

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

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

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

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

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

[‡]Not just estimated state or covariance

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

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

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

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

Table 3: TLE of SMOS.

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

2.1 Point 1

To compute the Visibility Windows, the initial state of SMOS must be retrieved from it's TLE (Table 4), then propagated from t_{ref} to t_0 and then from t_0 to t_f using Keplerian dynamics. The Azimuth, Elevation, and Range can then be computed by rotating the relative state of the spacecraft and the ground station from ECI to the topocentric frame of the ground stations. The Visibility Windows are determined by identifying the time steps for which the predicted Elevation exceeds the minimum Elevation of the ground station . These windows are reported in Table 5 as well as in the Elevation and Azimuth graphs in Fig. 12 and Fig. 13. All predicted measurement outside the visibility windows are discarded, as the ground station would not be able to acquire them.

SMA [km]	**e** [-]	**i** [deg]	Ω [deg]	ω [deg]
7143.7134	0.00124363	98.43420	148.46890	69.29293

Table 4: Osculating state of SMOS

Station name	**Visibility window start**	**Visibility window ending**
KOUROU	2024 NOV 18 20:41:00.000	2024 NOV 18 20:50:00.000
TROLL	2024 NOV 18 21:02:30.000	2024 NOV 18 21:12:30.000
SVALBARD	2024 NOV 18 21:57:00.000	2024 NOV 18 22:07:00.000

Table 5: Predicted time windows

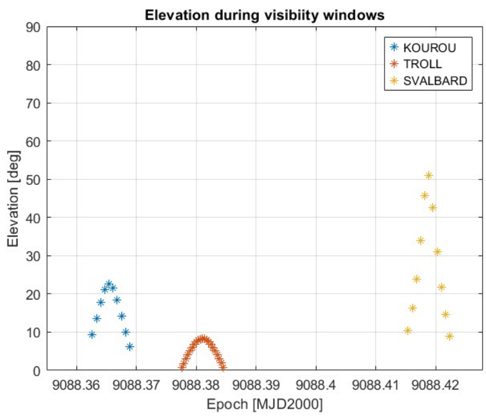

Figure 12: Predicted elevation during the visbility windows

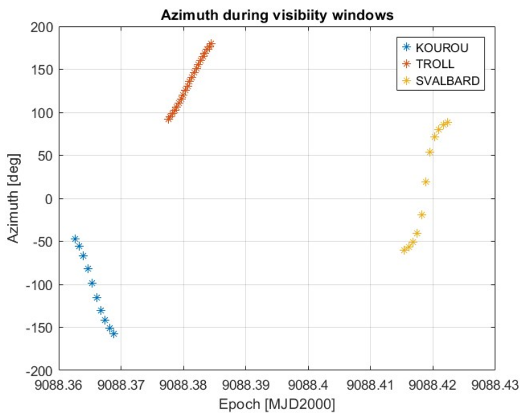

Figure 13: Predicted azimuth during the visbility windows

2.2 Point 2

To simulate measurements from the ground station, the state of SMOS is first retrieved using the SGP4 model, which accounts for spacecraft perturbations. SGP4 returns the state in the TEME reference frame, which is then rotated into the ECI frame and transformed to compute Azimuth, Elevation, and Range in the topocentric reference frame of the ground station. To ensure accurate simulation of measurements, random errors are introduced assuming a Gaussian distribution with a noise matrix R for both ground stations, as depicted in Eq. 8, with values taken from Table 2.

$$R = \begin{bmatrix} \sigma_{Az}^2 & 0 & 0 \\ 0 & \sigma_{EL}^2 & 0 \\ 0 & 0 & \sigma_R^2 \end{bmatrix} \tag{8}$$

After this procedure, a visibility check is performed, discarding all values with elevation below the minimum elevation of the ground stations. The updated visibility windows are shown in table 8. Comparing them with the predicted measurements, it is observed that the last element of the first Kourou visibility window has been removed, resulting in shortened visibility windows, which are shown below.

Station name	**Visibility window start**	**Visibility window ending**
KOUROU	2024 NOV 18 20:41:00.000	2024 NOV 18 20:49:00.000
TROLL	2024 NOV 18 21:02:30.000	2024 NOV 18 21:12:30.000
SVALBARD	2024 NOV 18 21:57:00.000	2024 NOV 18 22:07:00.000

Table 6: Measuread time windows

2.3 Point 3

The navigation problem is solved using a weighted least squares approach aimed at minimizing variance. The MATLAB function *lsqnonlin* it's used to minimize the cost function, which calculates the residual resulting by the weighted difference between expected and simulated measurements. At each iteration, the cost function propagates the spacecraft trajectory using selected dynamics (Keplerian for point A and including J2 perturbations for point C) from the initial state and reference time up to the visibility window. From this data, it computes the predicted measurement and the associated weighted residual, as shown in Eq.9. The *lsqnonlin* function computes the initial state $\hat{\boldsymbol{x}}$ at t_0 that minimizes this weighted residual.

$$residual = \begin{bmatrix} \frac{1}{\sigma_{Az,El}} & 0 & 0 \\ 0 & \frac{1}{\sigma_{Az,El}} & 0 \\ 0 & 0 & \frac{1}{\sigma_R} \end{bmatrix} \begin{bmatrix} Az_{meas} - Az_{pred} \\ El_{meas} - El_{pred} \\ R_{meas} - R_{pred} \end{bmatrix} \tag{9}$$

Once a solution is found the *lsqnonlin* function outputs: the least square solution to the navigation problem $\hat{\boldsymbol{x}}_0$, the squared norm of the residual ε^2, the residual ε, the Jacobian J of the measurements with respect to the variable to estimate. Given this, outputs the covariance matrix can then be computed as:

$$\boldsymbol{P}(t_0) = \frac{\varepsilon^2}{\text{length}(e) - \text{length}(x_0)} (\boldsymbol{J}^T \boldsymbol{J})^{-1} \qquad tra_r = \sqrt{\text{tr}(\boldsymbol{P_{rr}})} \qquad tra_v = \sqrt{\text{tr}(\boldsymbol{P_{vv}})} \tag{10}$$

The results of the three methods are reported hereafter:

Case a: Only Kourou measurement and keplerian dynamics

$$\hat{\boldsymbol{x}}_{\mathbf{0}} = \begin{bmatrix} 3.6447\text{e}{+}03 \text{ [km]} \\ -1.4601\text{e}{+}03 \text{ [km]} \\ 5.9619\text{e}{+}03 \text{ [km]} \\ 5.0799 \text{ [km/s]} \\ -3.7384 \text{ [km/s]} \\ -4.0119 \text{ [km/s]} \end{bmatrix} \qquad \begin{aligned} err_r &= 28.4003 \text{ [km]} \\ tra_r &= 9.5317 \text{ [km]} \\ err_v &= 0.0342 \text{ [km/s]} \\ tra_v &= 0.0106 \text{ [km/s]} \end{aligned}$$

$$\boldsymbol{P} = \begin{bmatrix} 265.6382 & 0.0012 & -0.0041 & -0.0062 & 14.8967 & -14.9076 \\ 0.0012 & 2.1886\text{e}{-}07 & -7.4728\text{e}{-}08 & -3.6629\text{e}{-}08 & 1.1751\text{e}{-}04 & -1.1778\text{e}{-}04 \\ -0.0041 & -7.4728\text{e}{-}08 & 1.0405\text{e}{-}06 & -3.1399\text{e}{-}08 & -2.9105\text{e}{-}04 & 2.9165\text{e}{-}04 \\ -0.0062 & -3.6629\text{e}{-}08 & -3.1399\text{e}{-}08 & 1.6013\text{e}{-}07 & -3.4021\text{e}{-}04 & 3.4042\text{e}{-}04 \\ 14.8967 & 1.1751\text{e}{-}04 & -2.9105\text{e}{-}04 & -3.4021\text{e}{-}04 & 0.8495 & -0.8492 \\ -14.9076 & -1.1778\text{e}{-}04 & 2.9165\text{e}{-}04 & 3.4042\text{e}{-}04 & -0.8492 & 0.8499 \end{bmatrix}$$

Case b: All stations measurements and Keplerian dynamics

$$\hat{\boldsymbol{x}}_{\mathbf{0}} = \begin{bmatrix} 3.6620\text{e}{+}03 \text{ [km]} \\ -1.4651\text{e}{+}03 \text{ [km]} \\ 5.9373\text{e}{+}03 \text{ [km]} \\ 5.0771 \text{ [km/s]} \\ -3.7252 \text{ [km/s]} \\ -4.0439 \text{ [km/s]} \end{bmatrix} \qquad \begin{aligned} err_r &= 30.4596 \text{ [km]} \\ tra_r &= 1.2225 \text{ [km]} \\ err_v &= 0.0346 \text{ [km/s]} \\ tra_v &= 0.0012\text{e} - 05 \text{ [km/s]} \end{aligned}$$

$$\boldsymbol{P} = \begin{bmatrix} 0.0348 & -7.8936\text{e}{-}06 & 1.8477\text{e}{-}06 & -3.9134\text{e}{-}06 & -0.0015 & 0.0015 \\ -7.8936\text{e}{-}06 & 3.4774\text{e}{-}09 & -1.2351\text{e}{-}09 & 8.0051\text{e}{-}10 & 1.5764\text{e}{-}07 & -1.6261\text{e}{-}07 \\ 1.8477\text{e}{-}06 & -1.2351\text{e}{-}09 & 1.3994\text{e}{-}09 & 1.6988\text{e}{-}10 & -2.0344\text{e}{-}08 & 2.1550\text{e}{-}08 \\ -3.9134\text{e}{-}06 & 8.0051\text{e}{-}10 & 1.6988\text{e}{-}10 & 2.8883\text{e}{-}09 & 3.0557\text{e}{-}08 & -3.2726\text{e}{-}08 \\ -0.0015 & 1.5764\text{e}{-}07 & -2.0344\text{e}{-}08 & 3.0557\text{e}{-}08 & 2.0205\text{e}{-}04 & -2.0346\text{e}{-}04 \\ 0.0015 & -1.6261\text{e}{-}07 & 2.1550\text{e}{-}08 & -3.2726\text{e}{-}08 & -2.0346\text{e}{-}04 & 2.0488\text{e}{-}04 \end{bmatrix}$$

Case c: All stations measurements and J2 perturbed dynamics

$$\hat{\boldsymbol{x}}_{\mathbf{0}} = \begin{bmatrix} 3.6692\text{e}{+}03 \text{ [km]} \\ -1.4697\text{e}{+}03 \text{ [km]} \\ 5.9357\text{e}{+}03 \text{ [km]} \\ 5.0668 \text{ [km/s]} \\ -3.7229 \text{ [km/s]} \\ -4.0445 \text{ [km/s]} \end{bmatrix} \qquad \begin{aligned} err_r &= 37.1097 \text{ [km]} \\ tra_r &= 0.03502 \text{ [km]} \\ err_v &= 0.0383 \text{ [km/s]} \\ tra_v &= 3.6051\text{e} - 05 \text{ [km/s]} \end{aligned}$$

$$\boldsymbol{P} = \begin{bmatrix} 3.9490\text{e}{-}05 & -4.4133\text{e}{-}10 & 3.0410\text{e}{-}10 & -4.1091\text{e}{-}09 & -6.7762\text{e}{-}06 & 6.8002\text{e}{-}06 \\ -4.4133\text{e}{-}10 & 8.0396\text{e}{-}13 & -4.1533\text{e}{-}13 & 2.9628\text{e}{-}13 & 7.6834\text{e}{-}10 & -7.6795\text{e}{-}10 \\ 3.0410\text{e}{-}10 & -4.1533\text{e}{-}13 & 1.3725\text{e}{-}12 & 3.0135\text{e}{-}13 & -9.4650\text{e}{-}10 & 9.4684\text{e}{-}10 \\ -4.1091\text{e}{-}09 & 2.9628\text{e}{-}13 & 3.0135\text{e}{-}13 & 2.6533\text{e}{-}12 & 4.3259\text{e}{-}10 & -4.3405\text{e}{-}10 \\ -6.7762\text{e}{-}06 & 7.6834\text{e}{-}10 & -9.4650\text{e}{-}10 & 4.3259\text{e}{-}10 & 2.5955\text{e}{-}06 & -2.5993\text{e}{-}06 \\ 6.8002\text{e}{-}06 & -7.6795\text{e}{-}10 & 9.4684\text{e}{-}10 & -4.3405\text{e}{-}10 & -2.5993\text{e}{-}06 & 2.6031\text{e}{-}06 \end{bmatrix}$$

It can be noticed that with an increasing number of measuring stations and with more precise dynamics prediction provided by the SGP4 model the covariance elements diminish in value, leading to less uncertainty in the state estimation. In order to linearly map the covariance of the state to the keplerian elements it's necessary to compute the Jacobian matrix of the variations of the keplerian elements with respect to the state elements:

$$\boldsymbol{J}_{kep,cart} = \begin{bmatrix} \frac{\partial a}{\partial x} & \frac{\partial a}{\partial y} & \cdots & \frac{\partial a}{\partial v_z} \\ \frac{\partial e}{\partial x} & \frac{\partial e}{\partial y} & \cdots & \frac{\partial e}{\partial v_z} \\ \frac{\partial i}{\partial x} & \frac{\partial i}{\partial y} & \cdots & \frac{\partial i}{\partial v_z} \\ \frac{\partial \Omega}{\partial x} & \frac{\partial \Omega}{\partial y} & \cdots & \frac{\partial \Omega}{\partial v_z} \\ \frac{\partial \omega}{\partial x} & \frac{\partial \omega}{\partial y} & \cdots & \frac{\partial \omega}{\partial v_z} \\ \frac{\partial \theta}{\partial x} & \frac{\partial \theta}{\partial y} & \cdots & \frac{\partial \theta}{\partial v_z} \end{bmatrix} \tag{11}$$

For computing it was exploited the variational approach. Then the resulting covariance associated to keplerian elements it's computed as:

$$\boldsymbol{P}_{kep} = \boldsymbol{J}_{kep,cart}\boldsymbol{P}\boldsymbol{J}_{kep,cart}^{T} \tag{12}$$

2.4 Point 4

To assess this point the step 3c was repeated considering only two stations during the visibility window instead of all to respect the economical budget constraint. The resulting standard deviations are listed in the table hereafter:

Station couple	σ_a^2	σ_i^2
KOUROU - TROLL	0.55007	1.4186e-09
KOUROU - SVALBARD	0.00011746	4.6872e-11
TROLL - SVALBARD	0.010815	4.2864e-10

Table 7: Measuread time windows

In this time window case the couple of stations with the best performance is Kourou-Svalbard. Those stations are the most distant in terms of observing times, allowing the sequential filter to provide a better innovation step in the state estimation; this can be more effective than just having a high number of measurements

2.5 Point 5

In this point it's required to choose a couple of stations for minimizing the standard deviations during long period of times. In this case one of the stations present in the couple it's expected to be Troll, since it has a higher sampling frequency that provides more data and a more reliable estimation in long term periods of state estimation. In order to choose a prime and a backup station the procedure done in the previous section is repeated for an extended period of time, wich is 10 days long starting from SMOS TLE time. In the table hereafter there are the resulting standard deviations:

Station couple	σ_a^2	σ_i^2
KOUROU - TROLL	1.9440e-05	8.0419e-11
KOUROU - SVALBARD	4.3135e-04	1.5522e-10
TROLL - SVALBARD	9.4329e-06	2.9665e-11

Table 8: Measuread time windows in long period of time

It can be deduced so that the set of stations that yields the best state estimation for long period of times is Troll-Svalbard.

Exercise 3: Sequential filters

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

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

1. *Check the visibility window.* Considering the initial state $\mathbf{x}_0$ and the time interval with a time-step of 30 seconds from t_0 to t_f reported in Table 9, predict the trajectory of the satellite in an inertial Moon-centered reference frame assuming Keplerian motion. Use the estimated coordinates given in Table 10 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 9 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 9. For the initial state, you can perturb the actual initial state $\mathbf{x}_0$ by exploiting the MATLAB function `mvnrnd` and the previously mentioned initial covariance. We suggest to use $\alpha = 0.01$ and $\beta = 2$ for tuning the UT in this case. Plot the time evolution of the error estimate together with the 3σ of the estimated covariance for both position and velocity.

4. *Estimate the lunar lander coordinates.* To fulfill the goal of your mission, you are asked to develop a sequential filter to narrow down the uncertainty on the knowledge of the lunar lander coordinates (considering a fixed zero altitude). To this aim, you can exploit the measurements of the components of the lunar orbiter position vector together with the measurements of the relative range between the orbiter and the lander computed at the previous point. Using an UKF, provide an estimate of the spacecraft state and the lunar lander coordinates (in terms of mean and covariance) by sequentially processing the acquired measurements in chronological order. To initialize the filter in terms of initial covariance, you can refer to the initial covariance $\mathbf{P}_0$ reported in Table 9. For the initial state, you can perturb the actual initial state, composed by $\mathbf{x}_0$ and the latitude

and longitude given in Table 10, by exploiting the MATLAB function `mvnrnd` and the previously mentioned initial covariance. We suggest to use $\alpha = 0.01$ and $\beta = 2$ for tuning the UT in this case. Plot the time evolution of the error estimate together with the 3σ of the estimated covariance for both position and velocity.

Table 9: Initial conditions for the lunar orbiter.

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

Table 10: Lunar lander - initial guess coordinates and horizon mask

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

3.1 Point 1

To compute the Visibility Window the initial state of the satellite $\boldsymbol{x}_0$ (9) has been propagated t_0 to t_f in a Moon Centered Inertial frame, while the geodetic data for the lander 10 were converted into rectangular and then rotated from a Moon fixed reference frame into the same reference frame of the satellite. Then the same procedure 2.1 was repeated, making it right to conclude that the satellite and the lander are in relative visibility for the entire time window, as the given time window coincides with the computed one:

Source	**Visibility window start**	**Visibility window ending**
DATA	2024 NOV 18 16:30:00.000	2024 NOV 18 20:30:00.000
COMPUTED	2024 NOV 18 16:30:00.000	2024 NOV 18 20:30:00.000

Table 11: Visibility windows between satellite and lander

3.2 Point 2

To assess this point a more precise kernel, 'MOONLANDER', was exploited to obtain the rotation matrix between the lander topocentric frame and the inertial frame. To ensure accurate simulation of measurements, random errors are introduced assuming a Gaussian distribution with the assigned noise parameter σ_p in table 10, by exploting MATLAB's function *randn*. To verify that the errors fall into the correct boundaries 10000 samples of the randomly assigned error were plotted with histograms, and compared to a normal probability density function centered on a zero mean and with a deviation equal to σ_p (in MATLAB *normpdf*), resulting in the following plot:

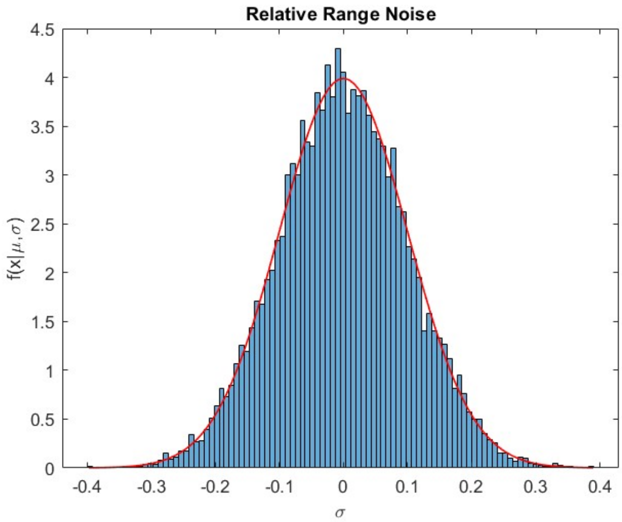

Figure 14: Error samples compared to normal probability density function

As most of the samples fall into the $[-3\sigma, 3\sigma]$ range and there's an overall resemblance to the gaussian distribution it can be concluded that the random errors are correctly applied to the measurements.

3.3 Point 3

In order to estimate the lunar orbiter absolute state an Unscented Kalman Filter (UKF) has been implemented. The UKF sequentially processes the acquired measurements in chronological order, so its inputs are:

- The mean at the previous time step $\hat{\boldsymbol{x}}_{k-1}^{+}$
- The covariance at previous time step $\boldsymbol{P}_{k-1}^{+}$
- The real measurements at current time step $\boldsymbol{y}_k^{r}$
- The noise matrix at current time step $\boldsymbol{R}_k$

The procedure for the UKF is:

1. **Computation of the 2n+1 sigma points:**

$$\begin{cases} \boldsymbol{\chi}_0 = \hat{\mathbf{x}} \\ \boldsymbol{\chi}_i = \hat{\mathbf{x}} + \left(\sqrt{(n+\lambda)\mathbf{P}_x}\right)_i \quad (i = 1, \dots, n) \\ \boldsymbol{\chi}_i = \hat{\mathbf{x}} - \left(\sqrt{(n+\lambda)\mathbf{P}_x}\right)_i \quad (i = 1+n, \dots, 2n) \end{cases} \tag{13}$$

With $n = length(\hat{x}) = 6$, $\alpha = 0.1$, $k = 0$ and $\lambda = \alpha^2(n+k) - n$.

2. **Computation of the weights**

$$
\begin{aligned}
w_0^{(m)} &= \frac{\lambda}{n+\lambda} \\
w_0^{(c)} &= \frac{\lambda}{n+\lambda} + 1 - \alpha^2 + \beta \ (\beta = 2) \\
w_i^{(m)} &= w_i^{(c)} = \frac{1}{2(n+\lambda)} \ (i = 1, ..., 2n)
\end{aligned}
\tag{14}
$$

3. **Prediction step:** Propagation of sigma points with J2-perturbed dynamics. Calculate measurement associated with sigma points and their mean:

$$
\begin{cases}
\boldsymbol{\chi}_{i,k} = f(\boldsymbol{\chi}_{i,k-1}) \\
\boldsymbol{\gamma}_{i,k} = h(\boldsymbol{\chi}_{i,k-1}) \\
\hat{\mathbf{y}}_k = \sum_{i=0}^{2n} W_i^{(m)} \boldsymbol{\gamma}_{i,k}
\end{cases}
\tag{15}
$$

This allows us to compute the a priori mean and covariance:

$$
\begin{cases}
\hat{\mathbf{x}}_k^- = \sum_{i=0}^{2n} W_i^{(m)} \boldsymbol{\chi}_{i,k} \\
\mathbf{P}_k^- = \sum_{i=0}^{2n} W_i^{(c)} (\boldsymbol{\chi}_{i,k} - \hat{\mathbf{x}}_k^-)(\boldsymbol{\chi}_{i,k} - \hat{\mathbf{x}}_k^-)^T
\end{cases}
\tag{16}
$$

4. **Update step.** Compute measurements covariance and cross covariance:

$$
\begin{cases}
\mathbf{P}_{yy,k} = \sum_{i=0}^{2n} W_i^{(c)} \left(\boldsymbol{\gamma}_{i,k} - \hat{\mathbf{y}}_k^-\right) \left(\boldsymbol{\gamma}_{i,k} - \hat{\mathbf{y}}_k^-\right)^T + \mathbf{R}_k \\
\mathbf{P}_{yx,k} = \sum_{i=0}^{2n} W_i^{(c)} \left(\boldsymbol{\chi}_{i,k} - \hat{\mathbf{x}}_k^-\right) \left(\boldsymbol{\gamma}_{i,k} - \hat{\mathbf{y}}_k^-\right)^T
\end{cases}
\tag{17}
$$

Finally the Kalman gain K_k is computed in order to find the a posteriori mean and covariance:

$$
\begin{cases}
\mathbf{K}_k = \mathbf{P}_{yx,k} \mathbf{P}_{yy,k}^{-1} \\
\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k \left(\mathbf{y}_k - \hat{\mathbf{y}}_k^-\right) \\
\mathbf{P}_k^+ = \mathbf{P}_k^- - \mathbf{K}_k \mathbf{P}_{yy,k} \mathbf{K}_k^T
\end{cases}
\tag{18}
$$

By initializing the filter with the assigned $\boldsymbol{x}_0$,$\boldsymbol{P}_0$, α and β the mean is obtained with the UKF, and the error on position and velocity can be computed by comparing it with the reference state as shown in Eq.19), while the 3σ of the estimated covariance can be computed starting from the covariance matrix as displayed in Eq.20.

$$
\begin{cases}
err_r = \|\mathbf{r}_{ref} - \mathbf{r}_{UKF}\| \\
err_v = \|\mathbf{v}_{ref} - \mathbf{v}_{UKF}\|
\end{cases}
\tag{19}
$$

$$
\begin{cases}
3\sigma_r = 3\sqrt{\mathrm{tr}(\mathbf{P}_{rr})} \\
3\sigma_v = 3\sqrt{\mathrm{tr}(\mathbf{P}_{vv})}
\end{cases}
\tag{20}
$$

The time evolution of the error and the 3σ are reported in Fig. 17 and 18. As expected, the error is below the 3σ threshold of the estimated covariance, and both values decrease over time. This is explained by the UKF's capability to refine its estimate over time.

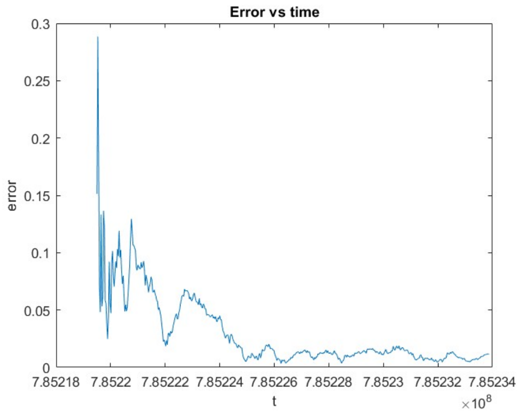

Figure 15: Error evolution in time

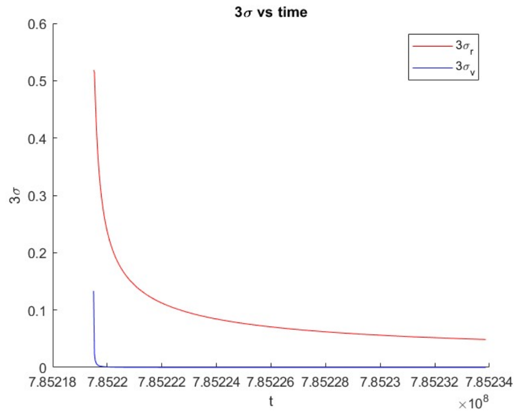

Figure 16: 3σ evolution in time

3.4 Point 4

To asses this point the procedure illustrated in 3.3 is repeated by initializing the filter with the assigned $\boldsymbol{x}_0$, ϕ, λ as the initial state to estimate, with ϕ and λ being respectively the latitude and longitude of the lander, and $\boldsymbol{P}_0$,α and β. The results in error and 3σ are reported in the following plots:

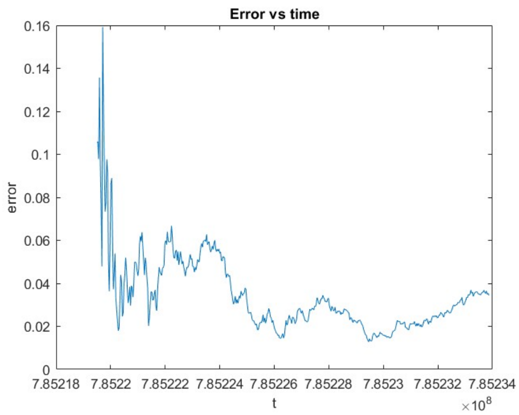

Figure 17: Error evolution in time

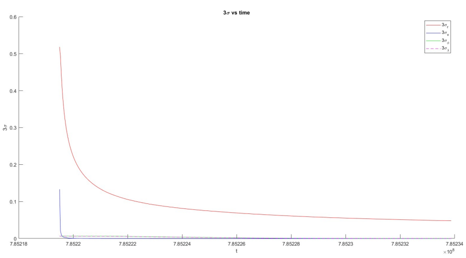

Figure 18: 3σ evolution in time