SGN – Assignment #2

Torre Federico, 233133

Disclaimer: The story plot contained in the following three exercises is entirely fictional.

Exercise 1: Uncertainty propagation

The Prototype Research Instruments and Space Mission Technology Advancement (PRISMA) is a technology in-orbit test-bed mission for demonstrating Formation Flying (FF) and rendezvous technologies, as well as flight testing of new sensors and actuator equipment. It was launched on June 15, 2010, and it involves two satellites: Mango (Satellite 1, ID 36599), the chaser, and Tango (Satellite 2, ID 36827), the target.

You have been provided with an estimate of the states of Satellites 1 and 2 at the separation epoch t_{sep} = `2010-08-12T05:27:39.114` (UTC) in terms of mean and covariance, as reported in Table 1. Assume Keplerian motion can be used to model the spacecraft dynamics.

1. Propagate the initial mean and covariance for both satellites within a time grid going from t_{sep} to $t_{sep} + N\,T_1$, with a step equal to T_1, where T_1 is the orbital period of satellite 1 and $N = 10$, using both a Linearized Approach (LinCov) and the Unscented Transform (UT). We suggest to use $\alpha = 0.1$ and $\beta = 2$ for tuning the UT in this case.

2. Considering that the two satellites are in close formation, you have to guarantee a sufficient accuracy about the knowledge of their state over time to monitor potential risky situations. For this reason, at each revolution, you shall compute:

   - the norm of the relative position (Δr), and
   - the sum of the two covariances associated to the position elements of the states of the two satellites (P_{sum})

   The critical conditions which triggers a collision warning is defined by the following relationship:

   $$\Delta r < 3\sqrt{\max(\lambda_i(P_{\text{sum}}))}$$

   where $\lambda_i(P_{\text{sum}})$ are the eigenvalues of P_{sum}. Identify the revolution N_c at which this condition occurs and elaborate on the results and the differences between the two approaches (UT and LinCov).

3. 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 plots of:

   - the time evolution for all three approaches (MC, LinCov, and UT) of $3\sqrt{\max(\lambda_i(P_{r,\text{i}}))}$ and $3\sqrt{\max(\lambda_i(P_{v,\text{i}}))}$, where $i = 1, 2$ is the satellite number and P_r and P_v are the 3x3 position and velocity covariance submatrices.
   - the propagated samples of the MC simulation, together with the mean and covariance obtained with all methods, projected on the orbital plane.

   Compare the results and discuss on the validity of the linear and Guassian assumption for uncertainty propagation.

*Use at least 100 samples drawn from the initial covariance

Table 1: Estimate of Satellite 1 and Satellite 2 states at t_0 provided in ECI J2000.

Parameter	Value
Ref. epoch t_{sep} [UTC]	2010-08-12T05:27:39.114
Mean state $\hat{\boldsymbol{x}}_{0,\text{sat1}}$ [km, km/s]	$\hat{\boldsymbol{r}}_{0,\text{sat1}} = [4622.232026629, 5399.3369588058, -0.0212138165769957]$ $\hat{\boldsymbol{v}}_{0,\text{sat1}} = [0.812221125483763, -0.721512914578826, 7.42665302729053]$
Mean state $\hat{\boldsymbol{x}}_{0,\text{sat2}}$ [km, km/s]	$\hat{\boldsymbol{r}}_{0,\text{sat2}} = [4621.69343340281, 5399.26386352847, -3.09039248714313]$ $\hat{\boldsymbol{v}}_{0,\text{sat2}} = [0.813960847513811, -0.719449862738607, 7.42706066911294]$
Covariance P_0 [km^2, km^2/s, km^2/s^2]	$\begin{bmatrix} +5.6e-7 & +3.5e-7 & -7.1e-8 & 0 & 0 & 0 \\ +3.5e-7 & +9.7e-7 & +7.6e-8 & 0 & 0 & 0 \\ -7.1e-8 & +7.6e-8 & +8.1e-8 & 0 & 0 & 0 \\ 0 & 0 & 0 & +2.8e-11 & 0 & 0 \\ 0 & 0 & 0 & 0 & +2.7e-11 & 0 \\ 0 & 0 & 0 & 0 & 0 & +9.6e-12 \end{bmatrix}$

1. **LinCov and UT propagation**: The propagation of mean states and covariance for both Tango and Mango spacecraft is performed using two methods: LinCov and UT. In the function `LinCov`, the mean states are propagated starting from the mean of the previous step, resulting in the propagated mean representing the average of the final distribution as shown in Equation1a. The dynamics are modeled using Keplerian motion, with the STM extracted at each step to estimate the covariance for the subsequent time step as shown in Equation 1b.

$$\underline{\hat{x}}(t) = \underline{x}^*(t) \qquad \text{(1a)} \qquad P(t) = \Phi(t_0, t)\, P_0 \Phi^\top(t_0, t) \qquad \text{(1b)}$$

For the UT method, a helper function `UT` is employed to implement the underlying logic, as described in Equations 2. The main steps involve composing `2n+1` sigma points for each time step where n is 6, followed by propagation from the mean ($\hat{x}$) and covariance (P_x) obtained from the previous step. The propagation starts from the initial values provided in the Table 1, utilizing a nonlinear map, specifically the Keplerian propagator.

$$\underline{\chi}_0 = \hat{x} \tag{2a}$$

$$\underline{\chi}_i = \begin{cases} \underline{\hat{x}} + \left(\sqrt{(n+\lambda)P_x}\right)_i, & \text{for } i = 1, \ldots, n \\ \underline{\hat{x}} - \left(\sqrt{(n+\lambda)P_x}\right)_{i-n}, & \text{for } i = n+1, \ldots, 2n \end{cases} \tag{2b}$$

$$\lambda = \alpha^2(n+K) - n \quad \text{where } \alpha = 0.1, \quad K = 0 \tag{2c}$$

$$\begin{aligned} W_0^{(m)} &= \tfrac{\lambda}{n+\lambda} \\ W_0^{(c)} &= \tfrac{\lambda}{n+\lambda} + \left(1 - \alpha^2 + \beta\right) \quad \text{where } \beta = 2, \\ W_i^{(m)} &= W_i^{(c)} = \tfrac{1}{[2(n+1)]} \quad i = 1, \ldots, 2n \end{aligned} \tag{3}$$

The mean state and covariance (Equations 4) at each time step are calculated given the weights as shown in Equations 3.

$$\begin{aligned} \underline{\hat{y}} &= \sum_{i=0}^{2n} W_i^{(m)} \underline{\gamma}_i \\ P_y &= \sum_{i=0}^{2n} W_i^{(c)} \left(\underline{\gamma}_i - \hat{y}\right) \left(\underline{\gamma}_i - \hat{y}\right)^\top \end{aligned} \tag{4}$$

2. **Collision check**: The script computes the norm of the relative position Δr and the sum of covariances associated with the position elements P_{sum} for Mango and Tango satellites using the LinCov and UT approaches. It then checks for the critical condition where Δr is less than $3\sqrt{\lambda_{\max}(P_{\text{sum}})}$, where $\lambda_{\max}(P_{\text{sum}})$ represents the maximum eigenvalue of P_{sum}. The revolution number (Nc) at which this condition is met is recorded. It plots the evolution of Δr and the critical condition boundary Δr_{lim} for both approaches over time as shown in Figure 1 and 2.

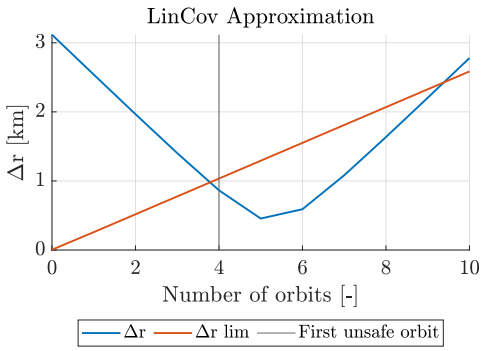

Figure 1: LinCov collision check.

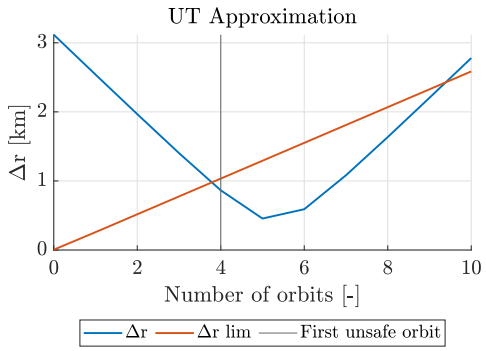

Figure 2: UT collision check.

The collision warning analysis reveals that a critical condition, indicating a potential collision risk between the satellites, is met at revolution $Nc = 4$ using both the LinCov and UT approaches. This suggests that, in this specific scenario, both approaches provided consistent estimates of the collision risk.

However, in more complex scenarios with significant nonlinearities or uncertainties, the UT approach is expected to provide more accurate and reliable estimates compared to the LinCov approach due to its ability to directly handle nonlinearities.

3. **Monte Carlo Computation and Plot**:

The `mvnrnd` function is utilized to generate samples that follow a white Gaussian distribution, representing the uncertainty in the spacecraft's state. The number of samples used for this approximation is set to 500, balancing the trade-off between accuracy and computational efficiency; higher sample counts generally yield more accurate results but come at the cost of increased computational resources.

Given the generated samples, we propagate each sample over the entire time grid using the Keplerian orbit propagator defined in the `MonteCarlo` function. At each time step, the sample mean and covariance are computed from the states of the propagated samples. To improve computational efficiency, a parallel loop `parfor` is employed to propagate the states of the samples. This parallelization strategy leverages the computational power of multi-core processors, enabling faster execution of the Monte Carlo simulation. To evaluate the time evolution of uncertainty in the position and velocity covariances for all three

methods, is calculated the maximum eigenvalues of the covariance matrices using the formula $3 \times \sqrt{\lambda_{\max}(P_{\mathrm{r},i})}$ and $3 \times \sqrt{\lambda_{\max}(P_{\mathrm{v},i})}$, where $\lambda_{\max}$ denotes the maximum eigenvalue, and $P_{\mathrm{r},i}$ and $P_{\mathrm{v},i}$ represent the position and velocity covariance submatrices, respectively. As shown in Figure 3, the plots depict the evolution of these eigenvalues over the time grid for all three methods. A close fit between the LinCov and UT methods is observed, with overlapping curves indicating consistent estimates throughout the propagation. However, slight discrepancies are evident in the Monte Carlo representation. This discrepancy can be attributed to the stochastic nature of the Monte Carlo method, where random sampling introduces variability in the results. While the Monte Carlo approach provides a more comprehensive representation of uncertainty, it may exhibit slight deviations from the deterministic estimates obtained by the LinCov and UT methods.

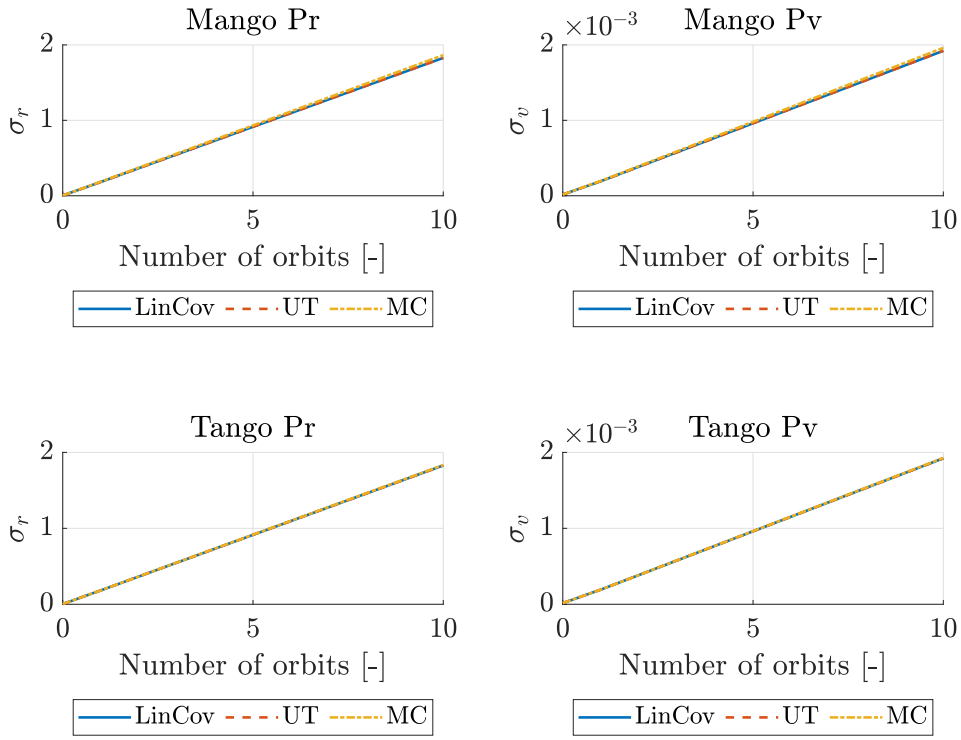

Figure 3: Comparison of LinCov, UT, and MC methods.

4. **Propagated Samples and Comparison**: The propagated samples of the Monte Carlo simulation, along with the mean and covariance obtained from both the LinCov and UT methods, are projected onto the orbital plane to visualize the spread of uncertainty. This visualization is achieved using the `rotation` function, which calculates the orbital plane based on the initial conditions defined in Table 1 and rotates the samples accordingly to align with this reference plane. In Figures 4 and 5, the black points represent the propagated samples, providing insights into the distribution of possible positions. Additionally, the covariance ellipses surrounding the mean positions illustrate the uncertainty associated with each estimation method.

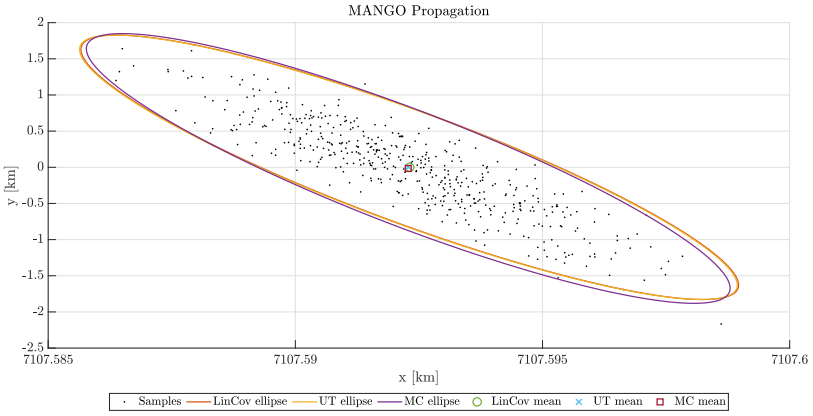

Figure 4: LinCov,UT and MC method comparison on orbital plane.

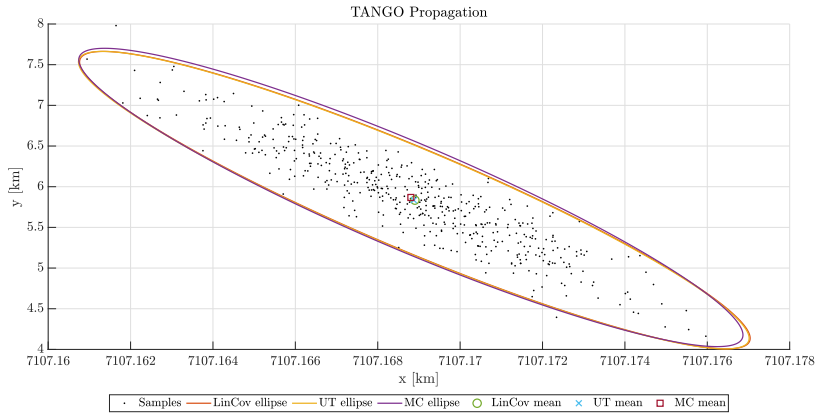

Figure 5: LinCov,UT and MC method comparison on orbital plane.

In comparing the results, it is evident that the LinCov and UT methods produce similar mean positions and ellipses, indicating consistency in their estimation of uncertainty. However, the Monte Carlo representation exhibits a wider spread of samples and ellipses, reflecting the stochastic nature of the simulation and the inherent variability introduced by random sampling.

5. **Validity of the assumptions:** For the reasons introduced in Point 3 and Point 4, the linear approximation holds for uncertainty propagation due to the almost complete overlapping of the covariance representation within the time grid. To validate the Gaussian assumption of the propagation, a Gaussian test is performed. Both mean position and velocity are plotted using the `qqplot`. Figures 6 and 7 depict the results for Mango and

Tango spacecraft, respectively. The blue distribution closely matches the theoretical red distribution, confirming the validity of the Gaussian assumption.

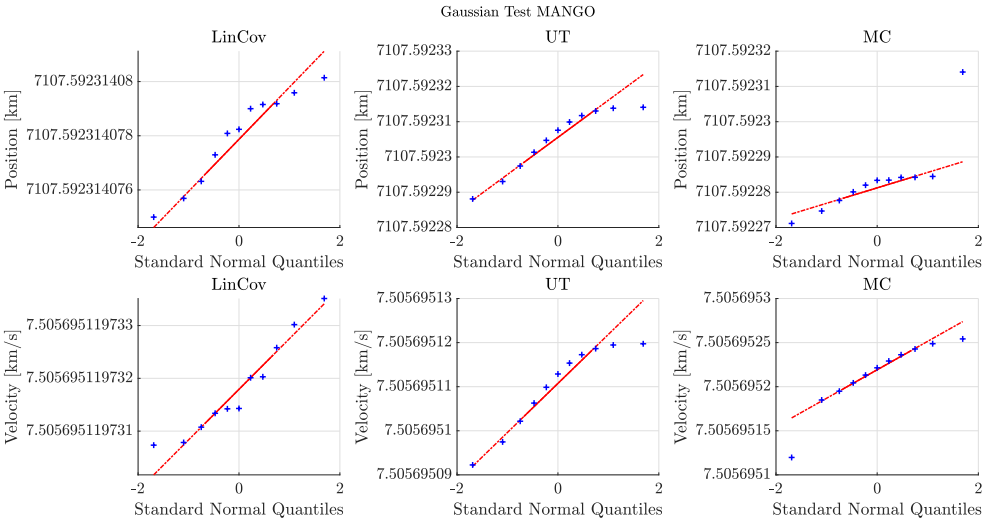

Figure 6: Gaussian test for Mango spacecraft.

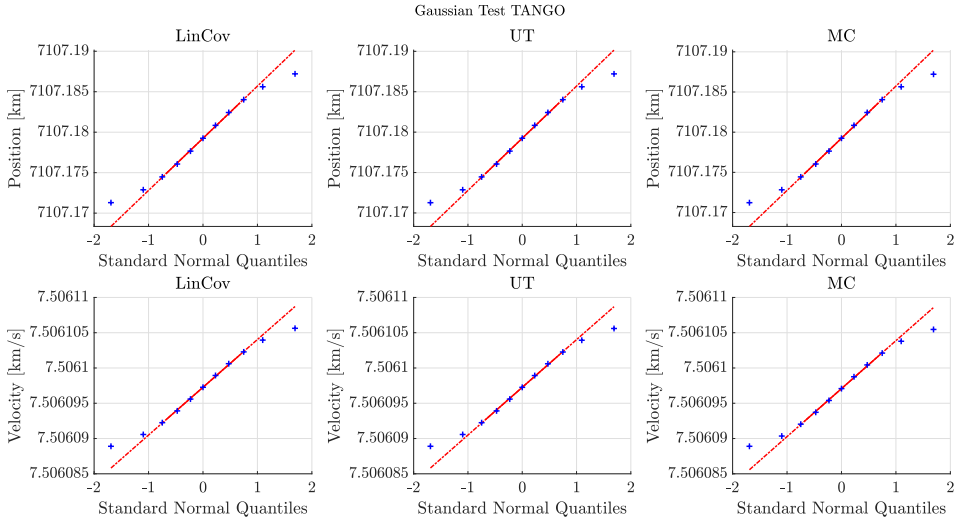

Figure 7: Gaussian test for Tango spacecraft.

Exercise 2: Batch filters

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

1. *Compute visibility windows.* By using the mean state reported in Table 1 and by assuming Keplerian motion, predict the trajectory of the satellite over a uniform time grid (with a time step of 60 seconds) and compute all the visibility time windows from the available stations in the time interval from t_0 = `2010-08-12T05:30:00.000` (UTC) to t_f = `2010-08-12T11:00:00.000` (UTC). Plot the resulting predicted Azimuth and Elevation profiles in the visibility windows.

2. *Simulate measurements.* The Two-Line Elements (TLE) set of Mango are reported in Table 3 (and in WeBeep as 36599.3le). Use SGP4 and the provided TLEs 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 both ground stations.

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

4. Provide the obtained navigation solutions and elaborate on the results, comparing the different solutions.

5. Select the best combination of dynamical model and ground stations and perform the orbit determination for the other satellite.

Table 2: Sensor network to track Mango and Tango: list of stations, including their features.

Station name	KOUROU	SVALBARD
Coordinates	LAT = 5.25144° LON = -52.80466° ALT = -14.67 m	LAT = 78.229772° LON = 15.407786° ALT = 458 m
Type	Radar (monostatic)	Radar (monostatic)
Provided measurements	Az, El [deg] Range (one-way) [km]	Az, El [deg] Range (one-way) [km]
Measurements noise (diagonal noise matrix R)	$\sigma_{Az,El}$ = 100 mdeg σ_{range} = 0.01 km	$\sigma_{Az,El}$ = 125 mdeg σ_{range} = 0.01 km
Minimum elevation	10 deg	5 deg

Table 3: TLE of Mango.

```
1 36599U 10028B   10224.22752732 -.00000576  00000-0 -16475-3 0  9998
2 36599 098.2803 049.5758 0043871 021.7908 338.5082 14.40871350  8293
```

Table 4: TLE of Tango.

```
1 36827U 10028F   10224.22753605  .00278492  00000-0  82287-1 0  9996
2 36827 098.2797 049.5751 0044602 022.4408 337.8871 14.40890217    55
```

1. **Compute visibility windows**: After propagating the state of the Mango spacecraft until t0 starting from t_{sep}, the `measurements` function is employed to acquire azimuth, range, and elevation data for every minute and for each station during the time span of the visibility window. Subsequently, utilizing the `visibility_window` function, it becomes feasible to identify which of the propagated states are visible from the stations. Specifically, within the `visibility_window` function, we verify that the elevation of a measurement exceeds the minimum elevation observable from the respective station, as specified in Table 2.

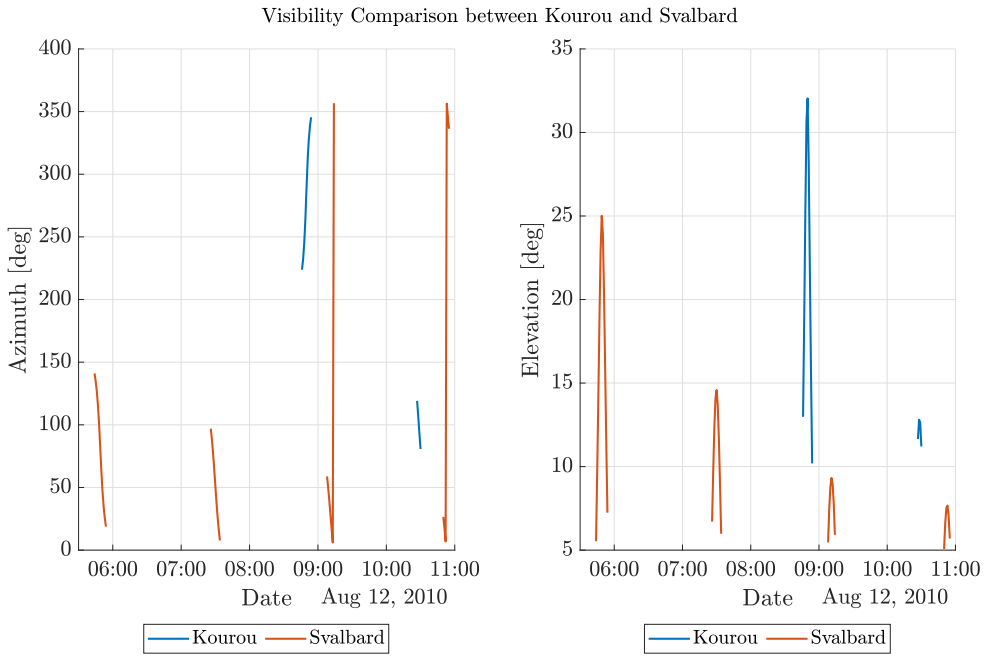

Figure 8: LinCov,UT and MC method comparison on orbital plane.

The visibility within the time grid is depicted in Figure 8.

2. **Simulate measurements**: To simulate measurements, the `propagateSatelliteTLE` function is employed, starting from the TLE defined in Table 3. This function propagates the satellite states in the time instant where the visibility is satisfied from Point 1 using the SGP4 model.

   Once the states are defined, a `measurements` function is utilized to calculate the azimuth, range, and elevation. To introduce noise into the measurements, a function `noise_function` is adopted. The noise is represented by random Gaussian white noise characterized by parameters outlined in Table 2, and it is generated using the Matlab function `mvnrnd`, which returns random vectors from the multivariate normal distribution. Inside the `noise_function` is implemented a logic in which is verified that the visibility condition is satisfied $El_{\text{measure}} \geq El_{\text{station}}$.

3. **Solve the navigation problem**: To solve the navigation problem, we employ the least squares method as depicted in Figure 9. The "true" state of the Mango satellite at t_0 in ECI coordinates is obtained using the `TLE2Cartesian` function, propagating with spg4 relative to the reference epoch defined by the TLE. The initial guess is derived from keplerian propagation from t_{sep} to t_0, considering the mean state as initial condition defined in Table 1.

   Using the least squares method, we compare real measurements with predicted ones using the `costfunction`. Predicted measurements are propagated with `keplerian_ rhs` and `perturbed_ keplerian_ rhs` (if $J2$ perturbation is considered), only at time instants with visibility found in Point 1. Within the `costfunction`, the `measurements` function translates predicted states into range, azimuth, and elevation for comparison with real measurements from our stations obtained at Point 2, calculating residuals as differences.

   The objective is to minimize the sum of squares of residuals by adjusting the initial state guess as shown in Equation 5.

$$\underline{x}_0^* \text{ that minimizes } J = \frac{1}{2}\sum_{i=1}^{l} \underline{\varepsilon}_i^T \cdot \underline{\varepsilon}_j \tag{5}$$

The Levenberg-Marquardt algorithm, combining features of gradient descent and Gauss-Newton methods, is specified through the 'Algorithm' option. Upon running `lsqnonlin`, estimated parameters, final residual norm, vector, exit flag, and optionally, the Jacobian matrix are obtained.

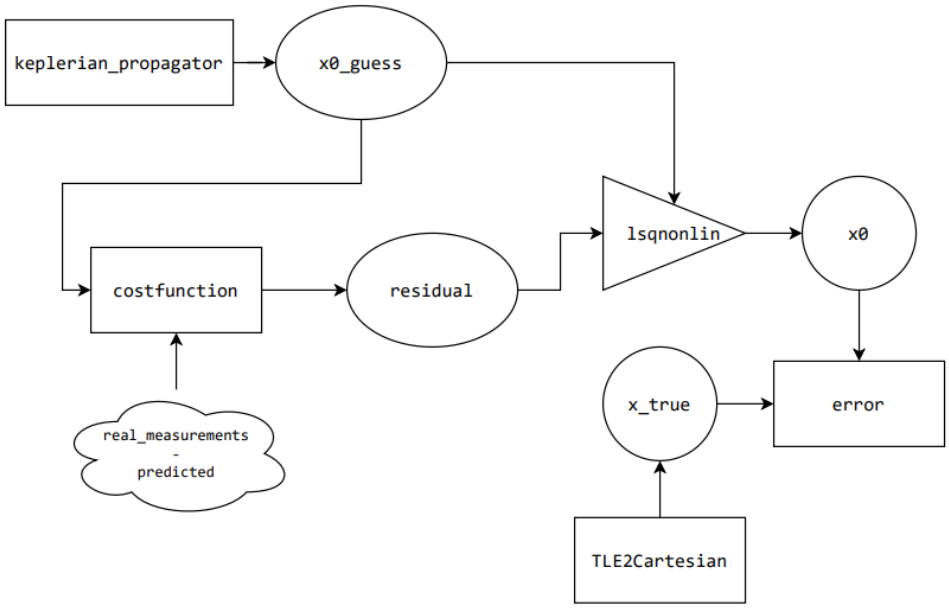

Figure 9: Flowchart for the implementation of the algorithm.

An important note is that the value obtained from the `TLE2Cartesian` function is referred to as "true," but it is not intended to represent the actual measurement. Instead, we rely more on the results given by `lsqnonlin`, as we input the real measurements within the time grid. Additionally, the TLE provides values for a limited time period.

Table 5 displays errors (calculated as difference between $x0$ and x_{true} as shown in Figure 9) for different propagation methods and numbers of stations. Minor errors occur when predicting values with $J2$ perturbation and both stations.

Navigation solutions vary across scenarios due to factors such as the count of ground stations and consideration of $J2$ perturbation. Solely relying on Kourou station measurements may lead to inaccuracies due to limited coverage. Incorporating both Kourou and Svalbard stations significantly improves navigation accuracy, with Svalbard's additional measurements reducing errors.

In conclusion, comprehensive data input and accurate dynamical modeling are essential for precise navigation outcomes. Utilizing multiple ground stations enhances accuracy and reliability, while advanced dynamical models like $J2$ perturbation further refine solutions, particularly for high-precision scenarios. Combining multiple ground stations with advanced dynamical modeling emerges as the optimal approach for accurate navigation solutions, as shown in Table 5.

Comparing Solutions: Least Squares Method	Position Error	Velocity Error
KOUROU without J2	27.8746 km	0.0280 km/s
KOUROU and SVALBARD without J2	21.8900 km	0.0148 km/s
KOUROU and SVALBARD with J2	0.1685 km	0.0002 km/s

Best combination : KOUROU and SVALBARD with J2

Table 5: Comparison of Navigation Solutions: MANGO.

Covariance matrix for KOUROU station without J2 perturbation:

$$P_K = \begin{bmatrix} 27.4488 & -75.9899 & 368.6301 & -0.0311 & -0.3006 & -0.0725 \\ -75.9899 & 254.3227 & -1.1926 \times 10^3 & 0.2525 & 1.1270 & 0.2495 \\ 368.6301 & -1.1926 \times 10^3 & 5.6262 \times 10^3 & -1.0690 & -5.1896 & -1.1646 \\ -0.0311 & 0.2525 & -1.0690 & 6.8339 \times 10^{-4} & 0.0015 & 2.7048 \times 10^{-4} \\ -0.3006 & 1.1270 & -5.1896 & 0.0015 & 0.0053 & 0.0011 \\ -0.0725 & 0.2495 & -1.1646 & -2.7048 \times 10^{-4} & 0.0011 & 2.4691 \times 10^{-4} \end{bmatrix}$$

Covariance matrix for KOUROU and SVALBARD stations without J2 perturbation:

$$P_{KS} = \begin{bmatrix} 0.3081 & 0.0839 & 0.1405 & -2.6387 \times 10^{-4} & -3.8221 \times 10^{-4} & -3.8146 \times 10^{-4} \\ 0.0839 & 0.2368 & -9.9746 \times 10^{-4} & -7.7836 \times 10^{-5} & -2.7895 \times 10^{-4} & -3.0706 \times 10^{-4} \\ 0.1405 & -9.9746 \times 10^{-4} & 0.9104 & -5.8962 \times 10^{-4} & -4.1453 \times 10^{-4} & -3.0237 \times 10^{-4} \\ -2.6387 \times 10^{-4} & -7.7836 \times 10^{-5} & -5.8962 \times 10^{-4} & 5.425 \times 10^{-7} & 4.9060 \times 10^{-7} & 4.2457 \times 10^{-7} \\ -3.8221 \times 10^{-4} & -2.7895 \times 10^{-4} & -4.1453 \times 10^{-4} & 4.9060 \times 10^{-7} & 8.6527 \times 10^{-7} & 7.2556 \times 10^{-7} \\ -3.8146 \times 10^{-4} & -3.0706 \times 10^{-4} & -3.0237 \times 10^{-4} & 4.2457 \times 10^{-7} & 7.2556 \times 10^{-7} & 7.0489 \times 10^{-7} \end{bmatrix}$$

Covariance matrix for KOUROU and SVALBARD stations with J2 perturbation:

$$P_{J2} = \begin{bmatrix} 9.9629 \times 10^{-5} & 2.6782 \times 10^{-5} & 4.1820 \times 10^{-5} & -8.3501 \times 10^{-8} & -1.2233 \times 10^{-7} & -1.2312 \times 10^{-7} \\ 2.6782 \times 10^{-5} & 7.4443 \times 10^{-5} & -5.6738 \times 10^{-6} & -2.1602 \times 10^{-8} & -8.5134 \times 10^{-8} & -9.5460 \times 10^{-8} \\ 4.1820 \times 10^{-5} & -5.6738 \times 10^{-6} & 2.8980 \times 10^{-4} & -1.8944 \times 10^{-7} & -1.3517 \times 10^{-7} & -9.2504 \times 10^{-8} \\ -8.3501 \times 10^{-8} & -2.1602 \times 10^{-8} & -1.8944 \times 10^{-7} & 1.7559 \times 10^{-10} & 1.5907 \times 10^{-10} & 1.3484 \times 10^{-10} \\ -1.2233 \times 10^{-7} & -8.5134 \times 10^{-8} & -1.3517 \times 10^{-7} & 1.5907 \times 10^{-10} & 2.7521 \times 10^{-10} & 2.2994 \times 10^{-10} \\ -1.2312 \times 10^{-7} & -9.5460 \times 10^{-8} & -9.2504 \times 10^{-8} & 1.3484 \times 10^{-10} & 2.2994 \times 10^{-10} & 2.2395 \times 10^{-10} \end{bmatrix}$$

As outlined in Table 5 the best strategy is to employ the dynamical model with J2 perturbation and both stations. In table 6 are reported the errors and the covariance matrix for TANGO.

Comparing Solutions: Least Squares Method	Position Error	Velocity Error
KOUROU and SVALBARD with J2	3.1532 km	0.0029 km/s

Table 6: Comparison of Navigation Solutions: TANGO.

Covariance matrix for KOUROU and SVALBARD stations with J2 perturbation for TANGO:

$$P_T = \begin{bmatrix} 9.9349 \times 10^{-5} & 2.5580 \times 10^{-5} & 4.8731 \times 10^{-5} & -8.8912 \times 10^{-8} & -1.2849 \times 10^{-7} & -1.2385 \times 10^{-7} \\ 2.5580 \times 10^{-5} & 7.2295 \times 10^{-5} & -2.2841 \times 10^{-6} & -2.3693 \times 10^{-8} & -8.5348 \times 10^{-8} & -9.3576 \times 10^{-8} \\ 4.8731 \times 10^{-5} & -2.2841 \times 10^{-6} & 3.0742 \times 10^{-4} & -2.0657 \times 10^{-7} & -1.5968 \times 10^{-7} & -1.0594 \times 10^{-7} \\ -8.8912 \times 10^{-8} & -2.3693 \times 10^{-8} & -2.0657 \times 10^{-7} & 1.9173 \times 10^{-10} & 1.8081 \times 10^{-10} & 1.4589 \times 10^{-10} \\ -1.2849 \times 10^{-7} & -8.5348 \times 10^{-8} & -1.5968 \times 10^{-7} & 1.8081 \times 10^{-10} & 3.0250 \times 10^{-10} & 2.4290 \times 10^{-10} \\ -1.2385 \times 10^{-7} & -9.3576 \times 10^{-8} & -1.0594 \times 10^{-7} & 1.4589 \times 10^{-10} & 2.4290 \times 10^{-10} & 2.2707 \times 10^{-10} \end{bmatrix}$$

Exercise 3: Sequential filters

According to the Formation Flying In Orbit Ranging Demonstration experiment (FFIORD), PRISMA's primary objectives include testing and validation of GNC hardware, software, and algorithms for autonomous formation flying, proximity operations, and final approach and recede operations. The cornerstone of FFIORD is a Formation Flying Radio Frequency (FFRF) metrology subsystem designed for future outer space formation flying missions.

FFRF subsystem is in charge of the relative positioning of 2 to 4 satellites flying in formation. Each spacecraft produces relative position, velocity and line-of-sight (LOS) of all its companions.

You have been asked to track Mango to improve the accuracy of the estimate of its absolute state and then, according to the objectives of the PRISMA mission, validate the autonomous formation flying navigation operations by estimating the relative state between Mango and Tango by exploiting the relative measurements acquired by the FFRF subsystem. The Two-Line Elements (TLE) set of Mango and Tango satellites are reported in Tables 3 and 4 (and in WeBeep as 36599.3le, and 36827.3le).

The relative motion between the two satellites can be modelled through the linear, Clohessy-Wiltshire (CW) equations[†]

$$\begin{aligned} \ddot{x} &= 3n^2 x + 2n\dot{y} \\ \ddot{y} &= -2n\dot{x} \\ \ddot{z} &= -n^2 z \end{aligned} \tag{6}$$

where x, y,and z are the relative position components expressed in the LVLH frame, whereas n is the mean motion of Mango, which is assumed to be constant and equal to:

$$n = \sqrt{\frac{GM}{R^3}} \tag{7}$$

where R is the position of Mango at t_0.

The unit vectors of the LVLH reference frame are defined as follows:

$$\hat{\boldsymbol{i}} = \frac{\boldsymbol{r}}{r}, \quad \hat{\boldsymbol{j}} = \hat{\boldsymbol{k}} \times \hat{\boldsymbol{i}}, \quad \hat{\boldsymbol{k}} = \frac{\boldsymbol{h}}{h} = \frac{\boldsymbol{r} \times \boldsymbol{v}}{\|\boldsymbol{r} \times \boldsymbol{v}\|} \tag{8}$$

To perform the requested tasks you should:

1. *Estimate Mango absolute state.* You are asked to develop a sequential filter to narrow down the uncertainty on the knowledge of Mango absolute state vector. To this aim, you shall schedule the observations from the SVALBARD ground station[‡] reported in Table 2, and then proceed with the state estimation procedure by following these steps:

   (a) By using the mean state reported in Table 1 and by assuming Keplerian motion, predict the trajectory of the satellite over a uniform time grid (with a time step of 5 seconds) and compute the first visibility time window from the SVALBARD station in the time interval from t_0 = `2010-08-12T05:30:00.000` (UTC) to t_f = `2010-08-12T06:30:00.000` (UTC).

   (b) Use SGP4 and the provided TLE to simulate the measurements acquired by the SVALBARD station for the Mango satellite only. For doing it, compute the spacecraft position over the visibility window using a time-step of 5 seconds, and derive the associated expected measurements. Finally, simulate the measurements by adding a random error (assume a Gaussian model to generate the random error, with noise provided in Table 2).

[†]Notice that the system is linear, therefore it has an analytic solution of the state transition matrix $\boldsymbol{\Phi}$

[‡]Note that these are the same ones computed in Exercise 2

(c) 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. Plot the time evolution of the error estimate together with the 3σ of the estimated covariance for both position and velocity.

2. *Estimate the relative state.* To validate the formation flying operations, you are also asked to develop a sequential filter to narrow down the uncertainty on the knowledge of the relative state vector. To this aim, you can exploit the relative azimuth, elevation, and range measurements obtained by the FFRF subsystem, whose features are reported in Table 7, and then proceed with the state estimation procedure by following these steps:

   (a) Use SGP4 and the provided TLEs to propagate the states of both satellites at epoch t_0 in order to compute the relative state in LVLH frame at that specific epoch.

   (b) Use the relative state as initial condition to integrate the CW equations over the time grid defined in Point 1a. Finally, simulate the relative measurements acquired by the Mango satellite through its FFRF subsystem by adding a random error to the expected measurements. Assume a Gaussian model to generate the random error, with noise provided in Table 7.

   (c) Consider a time interval of 20 minutes starting from the first epoch after the visibility window (always with a time step of 5 seconds). Use an UKF to provide an estimate of the spacecraft relative state in the LVLH reference frame (in terms of mean and covariance) by sequentially processing the measurements acquired during those time instants in chronological order. Plot the time evolution of the error estimate together with the 3σ of the estimated covariance for both relative position and velocity.

3. *Reconstruct Tango absolute covariance.* Starting from the knowledge of the estimated covariance of the absolute state of Mango, computed in Point 1, and the estimated covariance of the relative state in the LVLH frame, you are asked to provide an estimate of the covariance of the absolute state of Tango. You can perform this operation as follows:

   (a) Pick the estimated covariance of the absolute state of Mango at the last epoch of the visibility window, and propagate it within the time grid defined in Point 2c.

   (b) Rotate the estimated covariance of the relative state from the LVLH reference frame to the ECI one within the same time grid.

   (c) Sum the two to obtain an estimate of the covariance of the absolute state of Tango. Plot the time evolution of the 3σ for both position and velocity and elaborate on the results.

Table 7: Parameters of FFRF.

Parameter	Value
Measurements noise $\sigma_{Az,El}$	1 deg
(diagonal noise matrix R) σ_{range}	1 cm

1. **Estimate Mango absolute state** To estimate Mango's absolute state, the initial state at time t_0 is computed and propagated using Keplerian dynamics, incorporating J2 perturbation, from the separation time t_{sep} to t_0. This initial state is then projected forward within a specified time grid using the same method. The resulting state is transformed into observable quantities such as range, azimuth, and elevation using the `measurements` function. Subsequently, the `visibility_window` function is utilized to ascertain whether these measurements fall within the visibility range of the ground station Svalbard, ensuring that the minimum elevation of the measure is higher than the elevation of the Svalbard station. Furthermore, the `propagateSatelliteTLE` function is employed to propagate the spacecraft's state during the visibility window obtained in the previous step. Finally, the states acquired from the SGP4 propagation are translated into measurable quantities using the `measurements` function once more.

   Once these measurements are obtained, Gaussian white noise is introduced using Matlab's `mvnrnd` function within the `noise_measure` function. An additional verification is then performed to ensure that the introduced noise, as specified in Table 2, does not impede visibility in the previously computed time steps.

   The implementation of the UKF commences with the definition of two auxiliary functions: `UKFprop` and `UKFupdate` as shown in Algorithm 1.

   To compute the initial mean and covariance at t_0, the `UKFprop` function is used. Once these initial conditions are established, the `UKF` function, acting as a wrapper for the aforementioned functions, is employed to propagate the mean and covariance.

Algorithm 1 Procedure to implement the UKF method.

`UKFprop` % Implement the UT method as in 9 for $\hat{\underline{\mathbf{x}}}_k^-$ and $\mathbf{P}_k^-$.
`UKFupdate` % "updating" measurement by assessing disparities with the predicted measurement γ_i as shown in Eqs 9 for $\hat{\underline{\mathbf{y}}}_k^-$, $\mathbf{P}_{ee,k}$, and $\mathbf{P}_{xy,k}$, and Eqs 10.
`UKF` % wrapper of `UKFupdate` and `UKFprop` functions.
`Input:` $(UTDATA, SC, CASE)$ % The $CASE$ is relative or absolute.
for $j = 2 : length(time_{grid})$ **do**
  $[sigma_{points}, x_{mean}, Pprop, UTDATA] = UKFprop(t_{grid}, UTDATA, SC, CASE)$; % UT implementation with propagation of sigma points, mean and covariance.
  $[x_{mean}(:, j), P(:, :, j)] = UKFupdate(sigma_{points}, x_{mean}, Pprop, measure, UTDATA,$ % UKF implementation: correction with the measurement.
end for
`Output:` $[sigma_{points}, x_{mean}, P]$

$$
\begin{aligned}
\hat{\underline{\mathbf{x}}}_k^- &= \sum_{i=0}^{2L} W_i^{(m)} \underline{\chi}_{i,k} \quad \mathbf{P}_k^- = \sum_{i=0}^{2L} W_i^{(c)} \left[\underline{\chi}_{i,k} - \hat{\underline{\mathbf{x}}}_k^-\right] \left[\underline{\chi}_{i,k} - \hat{\underline{\mathbf{x}}}_k^-\right]^{\mathrm{T}} \\
\hat{\underline{\mathbf{y}}}_k^- &= \sum_{i=0}^{2L} W_i^{(m)} \underline{\gamma}_{i,k} \quad \mathbf{P}_{ee,k} = \sum_{i=0}^{2L} W_i^{(c)} \left[\underline{\gamma}_{i,k} - \hat{\underline{\mathbf{y}}}_k^-\right] \left[\underline{\gamma}_{i,k} - \hat{\underline{\mathbf{y}}}_k^-\right]^{\mathrm{T}} + \mathbf{R}_k \\
\mathbf{P}_{xy,k} &= \sum_{i=0}^{2L} W_i^{(c)} \left[\underline{\chi}_{i,k} - \hat{\underline{\mathbf{x}}}_k^-\right] \left[\underline{\gamma}_{i,k} - \hat{\underline{\mathbf{y}}}_k^-\right]^{\mathrm{T}}
\end{aligned}
\tag{9}
$$

$$\begin{aligned} \mathbf{K}_k &= \mathbf{P}_{xy,k}\mathbf{P}_{ee,k}^{-1} \\ \hat{\underline{\mathbf{x}}}_k^+ &= \hat{\underline{\mathbf{x}}}_k^- + \mathbf{K}_k\left(\underline{\mathbf{y}}_k - \hat{\underline{\mathbf{y}}}_k^-\right) \\ \mathbf{P}_k^+ &= \mathbf{P}_k^- - \mathbf{K}_k\mathbf{P}_{ee,k}\mathbf{K}_k^{\mathrm{T}} \end{aligned} \tag{10}$$

The 3σ deviation is calculated as $\sigma = 3 \times \sqrt{\mathrm{trace}(MANGO.P_{\mathrm{abs}})}$, while the error, computed with respect to the SPG4 propagation, is represented by $x_{\mathrm{MANGO}}-$MANGO.x_mean_abs.

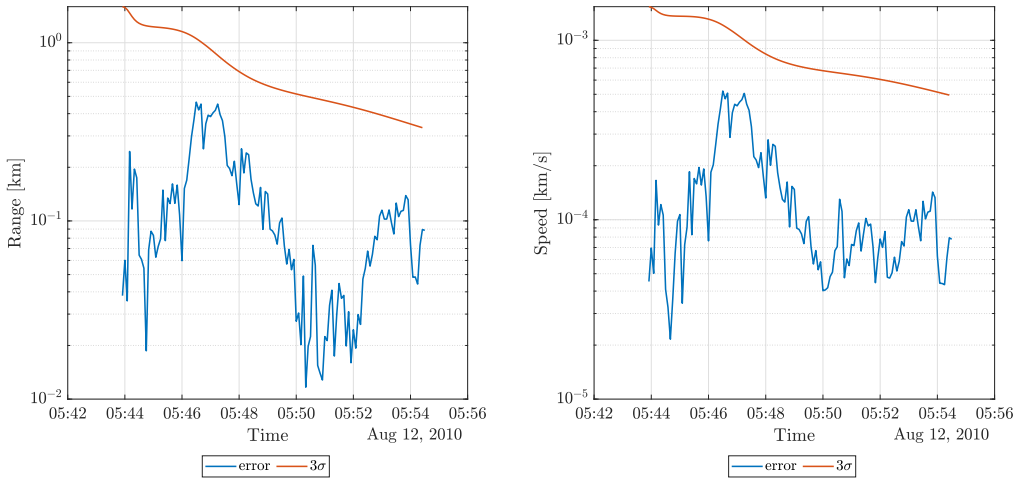

Figure 10: Error estimate with 3σ error for Mango absolute state.

The resulting plot illustrates the behavior of both the 3σ deviation and the error trend. As demonstrated in Figure 10, both the error and the 3σ deviation exhibit a decreasing trend; the error is shown in blue, exhibiting an oscillating behavior due to the presence of noise in the measurements. The observed trend, where the error curve consistently lies below the 3σ curve and both curves decrease over time, signifies the UKF's effectiveness in estimating the spacecraft's state. This suggests that the filter is converging efficiently, refining its estimate with each sequential measurement while maintaining a conservative estimation of uncertainty. The robustness of the filter in handling measurement noise is evident, reflecting its reliability in providing accurate state estimates. Overall, the results demonstrate the UKF's capability in tracking the spacecraft's position and velocity accurately, making it a valuable tool for real-time state estimation in dynamic systems.

2. **Estimate the relative state** To compute the state in ECI coordinates at time t_0, the `TLE2Cartesian` function is utilized for both the Tango and Mango spacecraft. Utilizing the definition of mean motion (as depicted in Equation 7) along with the state vector at t_0, the rotation matrix is calculated to facilitate the transformation of the reference system from ECI to the LVLH frame using the `eci_to_lvlh` function.

   The dynamics of the Tango spacecraft are propagated using the CW equations (Equations 6). The `noise_measureFFRF` function is employed to transform the states obtained from the `CW` function into measurements with the noise matrix defined in Table 7.

   The time window for this point is calculated from the last measurement epoch in Point 1 for the subsequent 20 minutes. Additional logic is applied to verify the consistency of the time window with the Tango states. The initial and final states of Tango are determined using the Matlab function `find` identifying the initial and final index of the previously propagated states.

   To estimate the relative state of the Tango spacecraft, the `UKF` function is utilized, initializing the initial state of Tango as the first state in the time window and considering the covariance matrix as diag([0.01, 0.01, 0.1, 0.0001, 0.0001, 0.001]). The function outputs the relative mean and covariance of Tango with respect to the Mango spacecraft. The 3σ deviation is calculated as $\sigma = 3 \times \sqrt{\text{trace}(TANGO.P_{\text{rel}})}$, while the error is based on the state of Tango propagated with the `CW` function. Figure 11 illustrates the behavior of both the error and the σ deviation, showing a decreasing trend.

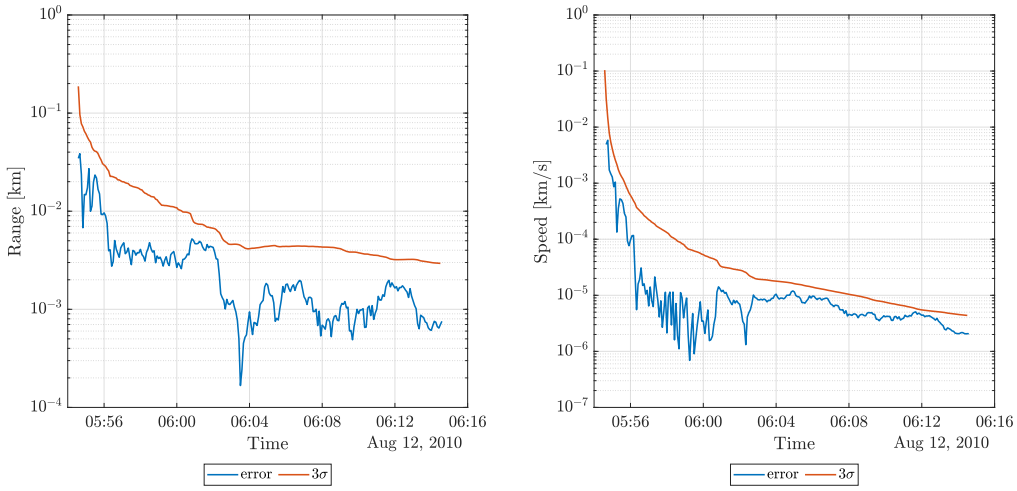

Figure 11: Error estimate with 3σ error for Tango relative state.

   the observed behavior of the error and uncertainty curves demonstrates the effectiveness, convergence, and robustness of the UKF in estimating the spacecraft's relative state in the LVLH reference frame. It provides confidence in the filter's ability to produce accurate estimates with decreasing uncertainty over time, making it a valuable tool for real-time state estimation in dynamic systems.

3. **Reconstruct Tango absolute covariance** To reconstruct the absolute covariance of Tango, the initial condition for the propagation with the `UKFprop` function is taken from the last state of the Mango spacecraft obtained in Point 1. Once both the state and covariance of Tango are obtained within the time grid, a rotation is applied. This rotation is achieved by calculating the rotation matrix R using the function `eci_to_lvlh`, and then

the covariance matrix is rotated and the absolute covariance is constructed as shown in Eq. 11.

$$\begin{aligned} P_{\text{rot}} &= R' \cdot \text{TANGO.P\_rel} \cdot R \\ \text{TANGO.P\_abs} &= \text{TANGO.P} + P_{\text{rot}} \end{aligned} \tag{11}$$

where TANGO.P is the covariance matrix obtained at the previous propagation within the time grid, and TANGO.P_rel is the one obtained in Point 2.

In Figure 12, the behavior of the 3σ for both position and velocity is illustrated.

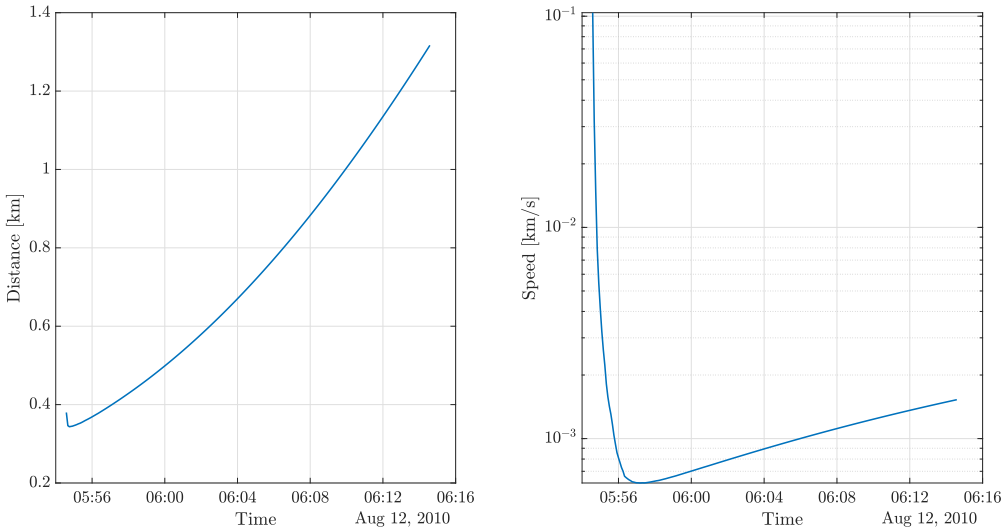

Figure 12: Evolution of the 3σ for both position and velocity.

The temporal progression of the 3σ boundaries for both position and velocity yields valuable insights into the spacecraft's state estimation process.

This behavior arises from the summation of the relative and absolute covariance of Tango Equation 11. Noteworthy observations include the gradual augmentation of both distance and velocity over time. The expanding 3σ boundary indicates an increasing uncertainty in the spacecraft's state estimation as time elapses, potentially attributable to factors such as modeling errors.