SGN – Assignment #2

Jules GOMEL, 219436

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

Exercise 1: Uncertainty propagation

After its launch on November 11, 2022, the upper stage of the Ariane 5 launcher (ID: 87654) is cruising along its highly-elliptical transfer orbit before releasing its embarked payload, the in-orbit servicer unit Orbital Repair Satellite (ORS).

You have been provided with an estimate of the Ariane 5 upper stage state at the pericenter epoch $t_0 =$ `2022-11-11T19:08:49.824` (UTC) in terms of its mean and covariance, as reported in Table 8. Assume Keplerian motion can be used to model the spacecraft dynamics.

1. Propagate the initial mean and covariance to all apocenter and pericenter epochs included in the subsequent four revolutions of the upper stage around the Earth using both a linearized approach (LinCov) and the unscented transform (UT). Compare the results in terms of both propagated mean and covariance. Elaborate on the results and the differences between the two approaches.

2. Perform the same uncertainty propagation process to the same epochs using a Monte Carlo (MC) simulation (using at least 100 samples drawn from the initial covariance). Compute the sample mean and sample covariance, and compare them with the results obtained at the previous point. Plot the propagated samples of the MC simulation, and the mean and covariance obtained with all methods, on the equatorial plane in the ECI reference frame. Compare the results and discuss on the validity of the linear and Guassian assumption for uncertainty propagation.

Table 1: Estimate of the ORS state at t_0 provided in ECI J2000.

Parameter	Value
Ref. epoch t_0 [UTC]	2022-11-11T19:08:49.824
Mean state $\hat{\boldsymbol{x}}_0$ [km, km/s]	$\hat{\boldsymbol{r}}_0 =$ [6054.30795817484, -3072.03883303992, -133.115352431876] $\hat{\boldsymbol{v}}_0 =$ [4.64750094824087, 9.18608475681236, -0.62056520749034]
Covariance P_0 [km^2, km^2/s, km^2/s^2]	$\begin{bmatrix} +5.6e-3 & +3.5e-3 & -7.1e-4 & 0 & 0 & 0 \\ +3.5e-3 & +9.7e-3 & +7.6e-4 & 0 & 0 & 0 \\ -7.1e-4 & +7.6e-4 & +8.1e-4 & 0 & 0 & 0 \\ 0 & 0 & 0 & +2.8e-7 & 0 & 0 \\ 0 & 0 & 0 & 0 & +2.7e-7 & 0 \\ 0 & 0 & 0 & 0 & 0 & +9.6e-8 \end{bmatrix}$

0) I have first computed the period using Kepler's law and I found $T = 6.8347.10^4$ s. This data allows me to compute the time of apogee and perigee of the orbit with full and half multiples of the period.

1) Let's now detail the methods we will implement - Linear Covariance (LinCov), Unscented Transform (UT) and Monte Carlo (MC) - to compute mean state and covariance at apogee and perigee and then compare the results we got :

LinCov : This method is based on propagating uncertainty using linear method. In our case of orbital motions, it implies using the state-transition matrix which we will call STM and note Φ. As we have seen in class we have, noting with a hat the mean state and covariance of our distribution and without, the propagated state we have :

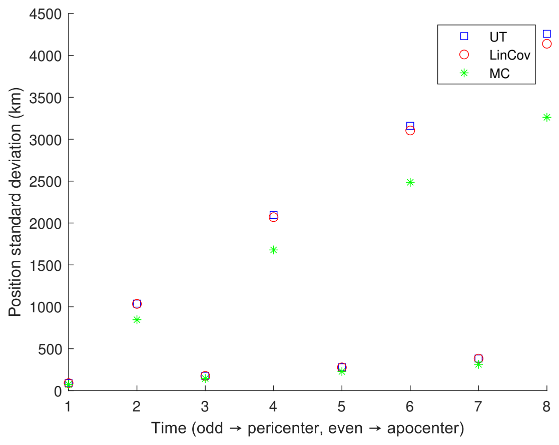
$$\hat{x}(t_k) = x(t_k) \ ; \ \hat{P}(t_k) = \Phi(t_0, t_k) P_0 \Phi(t_0, t_k)^T \tag{1}$$

UT : We first need to generate $2n + 1$-sigma points called x_p (with $n = 6$ in our case, as the state is in six dimension) and then propagate these sigma points into y_p and then compute the weighted sample mean and variance using the following formulas we studied in class, for $p = 1...2n$, with $\alpha = 10^{-3}$; $\beta = 2$; $\lambda = (\alpha^2 - 1)n$. :

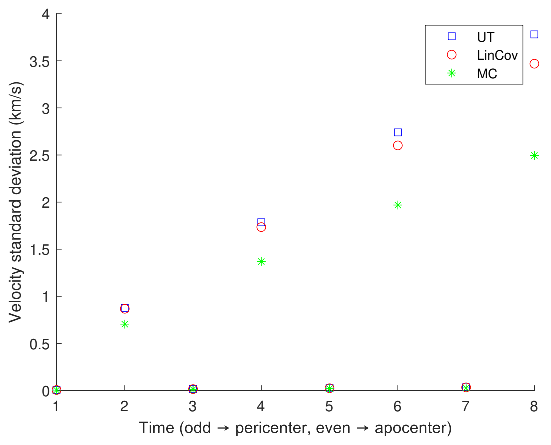
$$\begin{cases} x_0 = \hat{x}_0 \\ x_p = \sqrt{nP_0}_i \text{ if } p <= n \\ x_p = \sqrt{nP_0}_{i-n} \text{ else} \\ W_{0m} = \lambda/(n+\lambda) \\ W_{pm} = 1/2(n+\lambda)) \\ W_{0c} = W_{0m} + (1 - \alpha^2 + \beta) \\ W_{pc} = W_{pm} \\ \hat{x}_{UT} = \Sigma_{p=0}^{2n} W_{pm} y_p \\ \hat{P}_{UT} = \Sigma_{p=0}^{2n} W_{pc}(y_p - \hat{x}_{UT})(y_p - \hat{x}_{UT})^{\mathrm{T}} \end{cases} \tag{2}$$

MC Monte Carlo method is based on generating n points using the data of the mean state $\hat{x}_0$ and the covariance matrix P_0 using the assumption that it is a multivariate gaussian random variable. Thus, I generated 100 points using the function `mvnrnd` of `MatLab`, then I propagated it integrated the right-hand side of the Keplerian equation of motion and I then computed the samples mean and covariance.

Comparison The following plot shows the comparison of the square root of the trace of the covariance matrix obtained with UT, LinCov and Monte Carlo.



(a) Square root of the trace of the position covariance matrix obtained with UT (squares), LinCov (circles) and MC (stars)



(b) Square root of the trace of the velocity covariance matrix obtained with UT (squares), LinCov (circles) and MC (stars)

Figure 1: Comparison of the square root of the trace of the covariance matrix with the three methods

We can see that UT and LinCov deliver two results that are close in term of standard deviation and mean state. It can be due to the fact that the dynamics studied is not highly non-linear and that the propagation time is not so long to make LinCov too approximated. However, in term of mean state, LinCov is the reference, thus we see that UT is drifting from the reference over time, which is coherent because uncertainty is growing over time as it is propagating. In term of standard deviation, UT delivers a higher standard deviation over time than LinCov, it is mainly due to the case we studied, but in general LinCov will deliver a poor result compared to UT.

In term of computation time, we can see that LinCov is performed in a third of second on average, and UT in around five seconds. Thus, in this case, LinCov delivers a coherent first approximation with Keplerian dynamics.

	Mean state value [km, km, km, km/s, km/s, km/s]
First apogee	LinCov = [−5.837e+4, +2.962e+4, 1.283e+3, −4.820e-1, −9.528e-1, +6.436e-2]
	UT = [−5.837e+4, +2.962e+4, 1.283e+3, −4.822e-1, −9.527e-1, +6.437e-2]
	MC = [−5.837e+4, +2.961e+4, 1.284e+3, −4.820e-1, −9.528e-1, +6.436e-2]
First perigee	LinCov = [+6.054e+3, −3.072e+3, −1.331e+2, 4.648e0, +9.186e0, −6.206e-1]
	UT = [+5.996e+3, −3.092e+3, −1.296e+2, 4.650e0, +9.111e0, −6.173e-1]
	MC = [+6.030e+3, −3.057e+3, −1.327e+2, 4.626e0, +9.149e0, −6.179e-1]
Second apogee	LinCov = [−5.837e+4, +2.962e+4, 1.283e+3, −4.820e-1, −9.528e-1, +6.436e-2]
	UT = [−5.837e+4, +2.963e+4, 1.283e+3, −4.827e-1, −9.524e-1, +6.438e-2]
	MC = [−5.837e+4, +2.962e+4, 1.284e+3, −4.820e-1, −9.528e-1, +6.436e-2]
Second perigee	LinCov = [+6.054e+3, −3.072e+3, −1.331e+2, 4.648e0, +9.186e0, −6.206e-1]
	UT = [+5.824e+3, −3.141e+3, −1.197e+2, 4.653e0, +8.888e0, −6.074e-1]
	MC = [+5.956e+3, −3.016e+3, −1.312e+2, 4.569e0, +9.041e0, −6.105e-1]
Third apogee	LinCov = [−5.837e+4, +2.962e+4, 1.283e+3, −4.820e-1, −9.528e-1, +6.436e-2]
	UT = [−5.836e+4, +2.963e+4, 1.282e+3, −4.833e-1, −9.520e-1, +6.440e-2]
	MC = [−5.837e+4, +2.962e+4, 1.284e+3, −4.820e-1, −9.528e-1, +6.436e-2]
Third perigee	LinCov = [+6.054e+3, −3.072e+3, −1.331e+2, 4.648e0, +9.186e0, −6.206e-1]
	UT = [+5.639e+3, −3.025e+3, −1.166e+2, 4.489e0, +8.601e0, −5.872e-1]
	MC = [+5.838e+3, −2.953e+3, −1.288e+2, 4.486e0, +8.879e0, −5.995e-1]
Fourth apogee	LinCov = [−5.837e+4, +2.962e+4, 1.283e+3, −4.820e-1, −9.528e-1, +6.436e-2]
	UT = [−5.836e+4, +2.963e+4, 1.283e+3, −4.828e-1, −9.522e-1, +6.438e-2]
	MC = [−5.837e+4, +2.962e+4, 1.284e+3, −4.819e-1, −9.528e-1, +6.436e-2]
Fourth perigee	LinCov = [+6.054e+3, −3.072e+3, −1.331e+2, 4.648e0, +9.186e0, −6.206e-1]
	UT = [+5.380e+3, −2.863e+3, −1.123e+2, 4.260e0, +8.199e0, −5.589e-1]
	MC = [+5.682e+3, −2.872e+3, −1.254e+2, 4.387e0, +8.681e0, −5.862e-1]

Table 2: Summary table of mean states $\hat{\boldsymbol{x}}_0$ for each time step and each method

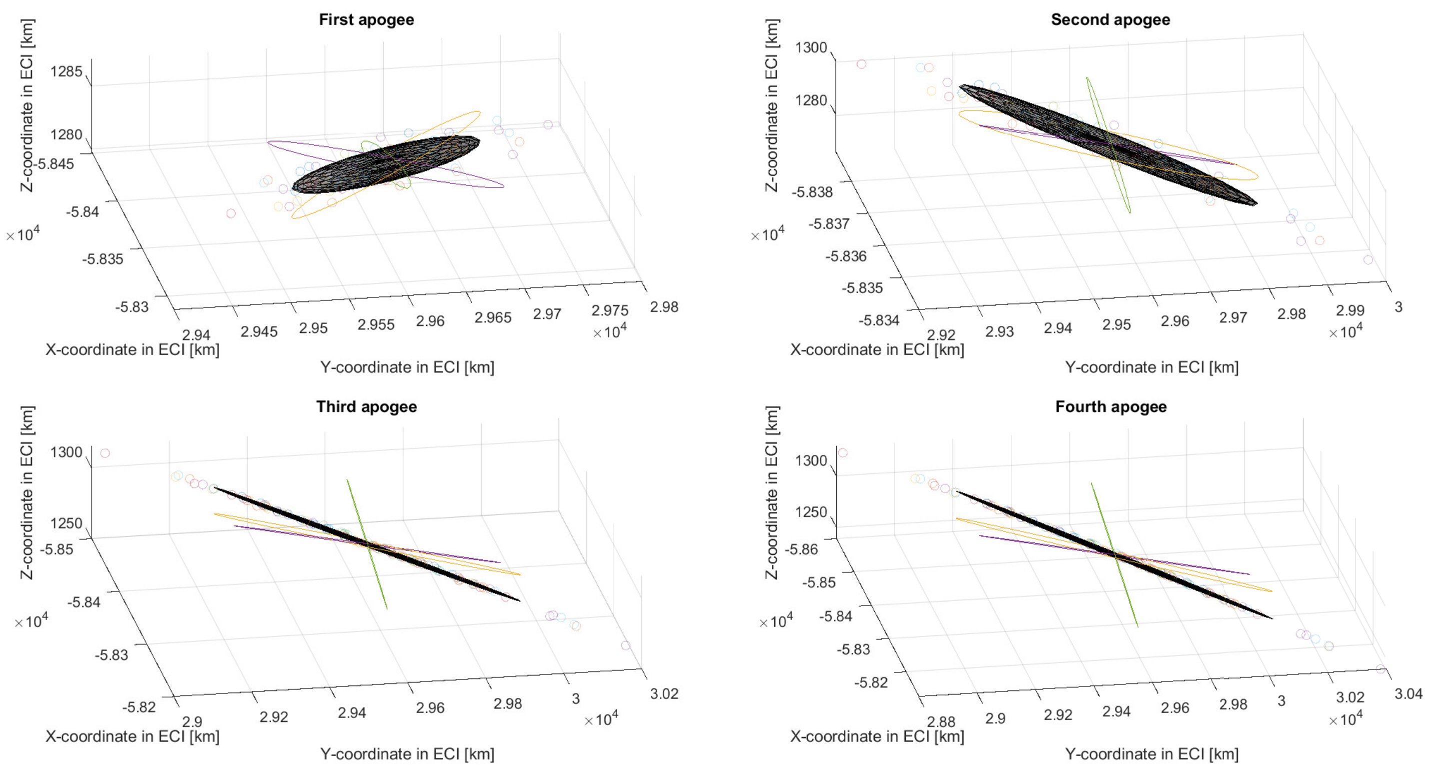


Figure 2: Sample states used for MC method, with sample covariance plotted for apogee

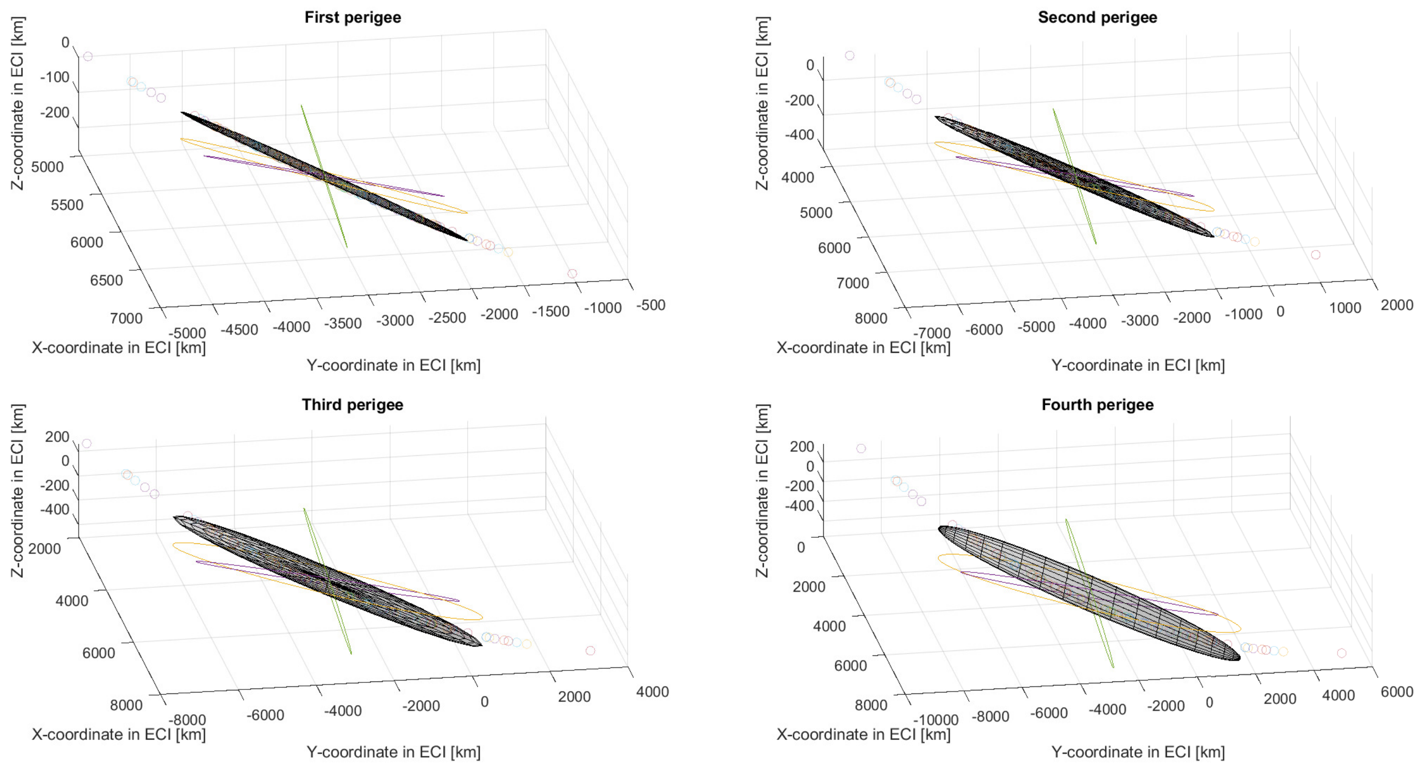


Figure 3: Sample states used for MC method, with sample covariance plotted for perigee

Exercise 2: Batch filters

You have been asked to track the Ariane 5 upper stage to improve the accuracy of its state estimate. To this aim, you are allowed to task the observations of the two ground stations reported in Table 3.

1. *Compute visibility windows.* By using the mean state reported in Table 8 and by assuming Keplerian motion, predict the upper stage trajectory over a uniform time grid (one point per minute) and compute all the visibility time windows from the available stations in the time interval from t_0 = `2022-11-12T04:30:00.000` (UTC) to t_f = `2022-11-14T16:30:00.000` (UTC). Plot the resulting predicted Azimuth and Elevation profiles in the visibility windows.

2. *Simulate measurements.* The latest available Two-Line Elements (TLE) set of the upper stage are reported in Table 4 (and in WeBeep as 87654.tle). Use SGP4 and the provided TLE to simulate the measurements acquired by the sensor network in Table 3 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 3).

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 4 at the reference epoch;
      - the simulated measurements obtained for the PERTH 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.

Table 3: Sensor network to track the Ariane 5 upper stage: list of stations, including their features.

Station name	KOUROU	PERTH

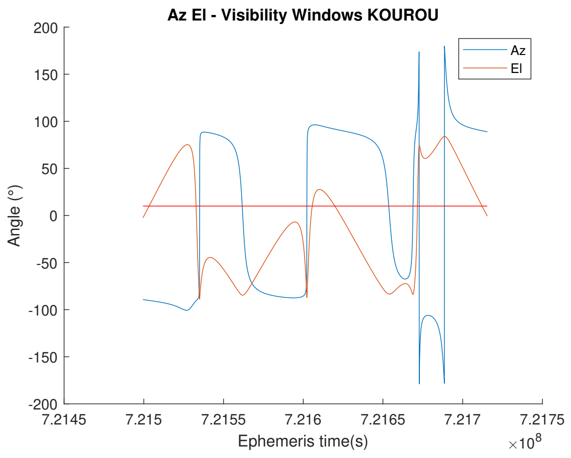
Coordinates	LAT = 5.25144°
	LON = -52.80466°
	ALT = -14.67 m
		LAT = -31.80252°
		LON = 115.88516°
		ALT = 22.16 m

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	$\sigma_{Az,El}$ = 100 mdeg σ_{range} = 0.01 km
Minimum elevation	10 deg	5 deg

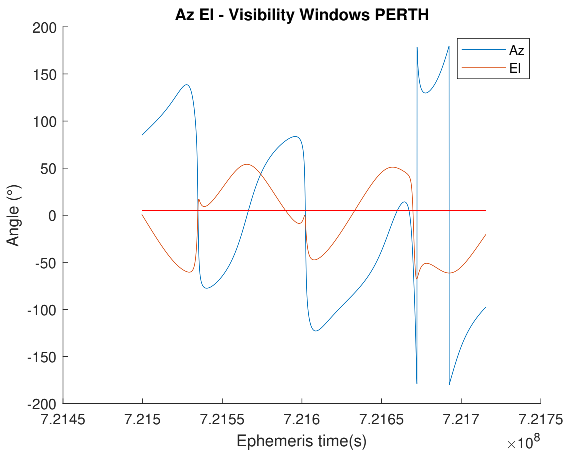
Table 4: Latest available TLE of the Ariane 5 upper stage.

```
1␣87654U␣22110B␣␣␣22316.00967942␣␣.00000002␣␣00000-0␣␣32024-3␣0␣␣9990
2␣87654␣␣␣3.6309␣137.1541␣8138191␣196.1632␣␣96.6141␣␣1.26411866␣␣␣834
```

1) I use the same process as in `Lab 4` : first I computed the two vectors between the station - Kourou or Perth - and the satellite in ECI. Then I use the function `cspice_xfmsta` to compute the azimuth and elevation of the satellite from the two stations.



(a) Azimuth and Elevation of the satellite from Kourou



(b) Azimuth and Elevation of the satellite from Perth

Figure 4: Visibility windows of the satellite from Kourou and Perth. In red is the limit of visibility

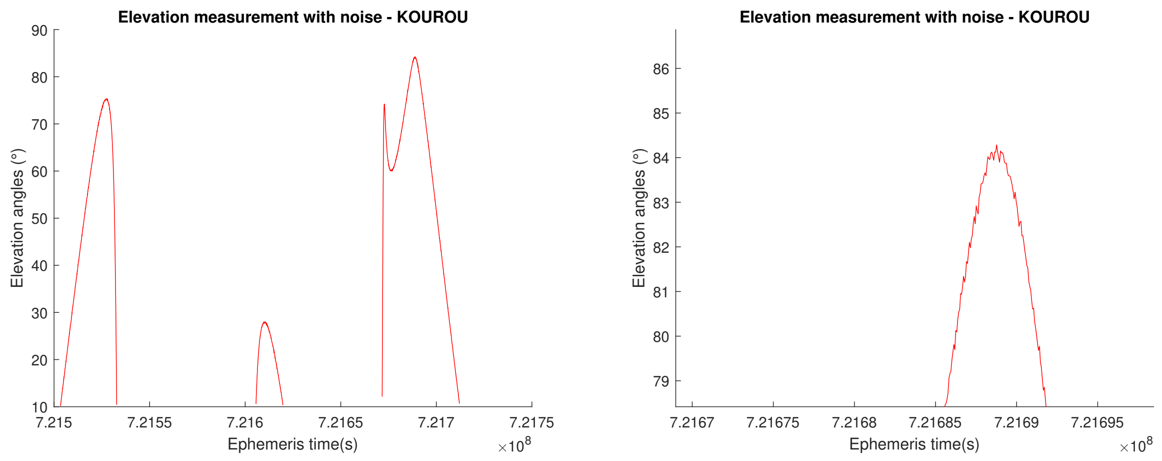
We can compute the visibility with the following table, rounded to the superior minute as all time where ended by 59.9999s, for the sake of conciseness :

Station Name	Start time (UTC)	End time (UTC)
KOUROU	2022-NOV-12-05:34:00	2022-NOV-12-13:45:00
PERTH	2022-NOV-12-14:17:00	2022-NOV-13-05:34:00
KOUROU	2022-NOV-13-10:00:00	2022-NOV-13-13:55:00
PERTH	2022-NOV-13-17:38:00	2022-NOV-14-03:46:00
KOUROU	2022-NOV-14-04:20:00	2022-NOV-14-15:34:00

Table 5: Details of the visibility windows of each stations

2) We will need the TLE as we did in `Lab.4` to derive the satellite position and convert it to azimuth, elevation and range as seen by stations using `sgp4`, which constitute the expected measurements of the station in KOUROU and PERTH. Thus, we find that the satellite number ID is `87654` and the reference epoch of the TLE is `UTC 2022-11-12T00:13:56.301888`

However, these measurements are perfect and not affected by uncertainty. As we have the data of standard deviation of measurements of each station, we can generate random gaussian noise, which is zero mean and with the standard deviation given. The following plot shows a part of the measurement we got for KOUROU, with the presence of Gaussian noise.



(a) Simulated measurements for KOUROU during the visibility windows

(b) Zoom on the measurements noise

Figure 5: Noisy measurements simulated for KOUROU station, during the visibility windows

3) For the three following solutions of the navigation, I used the same process as the example in the slides of the assignment 2 but without a priori information. I will only describe the difference between the approaches.

   (a) For this case, in my objective function, I used only the measurements of PERTH station and thus, I computed the residuals of the measurements between predicted and noisy measurements only with PERTH data. The motion considered for orbit propagation is keplerian. In the end, here are the results of `lsqnonlin` function of MatLab

	Value
Mean state $\hat{\boldsymbol{x}}_0$ [km, km/s]	$\hat{\boldsymbol{r}}_0 = [-5.826\text{e}{+}4,\ +2.984\text{e}{+}4,\ 1.240\text{e}{+}3]$ $\hat{\boldsymbol{v}}_0 = [-5.194e-1,\ -9.331\text{e-}1,\ +6.765e-2]$
Covariance P_0 [km^2, km^2/s, km^2/s^2]	$\begin{bmatrix} +1.058\text{e}0 & +2.188\text{e}0 & -3.080\text{e-}1 & -3.073\text{e-}5 & +2.633\text{e-}5 & +5.226\text{e-}5 \\ +2.188\text{e}0 & +4.591\text{e}0 & -7.738\text{e-}1 & -6.817\text{e-}5 & +6.081\text{e-}5 & -1.059\text{e-}4 \\ -3.080\text{e-}1 & -7.738\text{e-}1 & +6.515\text{e-}1 & +3.164\text{e-}5 & -2.567\text{e-}5 & +2.558\text{e-}5 \\ -3.073\text{e-}5 & -6.817\text{e-}5 & +3.164\text{e-}5 & +2.435\text{e-}9 & -1.397\text{e-}9 & +5.015\text{e-}9 \\ +2.633\text{e-}5 & +6.081\text{e-}5 & -2.567\text{e-}5 & -1.397\text{e-}9 & +1.373\text{e-}9 & +9.369\text{e-}10 \\ +5.226\text{e-}5 & -1.059\text{e-}4 & +2.558\text{e-}5 & +5.015\text{e-}9 & +9.369\text{e-}10 & +4.478\text{e-}8 \end{bmatrix}$
Square root of tr(P_{rr}) [km]	2.510
Square root of tr(P_{vv}) [km/s]	2.204e-4

Table 6: Initial state and covariance matrix which are solution of the navigation problem with keplerian motion and only PERTH station

(b) For this case, I used the same approach as before but using also KOUROU measurements. Here are the results :

	Value
Mean state $\hat{\boldsymbol{x}}_0$ [km, km/s]	$\hat{\boldsymbol{r}}_0 = [-5.825\text{e}{+4},\ +2.984\text{e}{+4},\ 1.278\text{e}{+3}]$ $\hat{\boldsymbol{v}}_0 = [-5.193\text{e-}1,\ -9.338\text{e-}1,\ +7.002\text{e-}2]$
Covariance P_0 [km^2, km^2/s, km^2/s^2]	$\begin{bmatrix} +7.455\text{e-}1 & +1.458\text{e}0 & -1.773\text{e-}1 & -2.547\text{e-}5 & +1.441\text{e-}5 & -1.559\text{e-}5 \\ +1.458\text{e}0 & +2.904\text{e}0 & -3.319\text{e-}1 & -4.968\text{e-}5 & +2.988\text{e-}5 & -3.490\text{e-}5 \\ -1.773\text{e-}1 & -3.319\text{e-}1 & +2.096\text{e-}1 & +5.265\text{e-}6 & -6.212\text{e-}6 & -2.363\text{e-}5 \\ -2.547\text{e-}5 & -4.968\text{e-}5 & +5.265\text{e-}6 & +9.435\text{e-}10 & -4.858\text{e-}10 & +1.208\text{e-}9 \\ +1.441\text{e-}5 & +2.988\text{e-}5 & -6.212\text{e-}6 & -4.858\text{e-}10 & +4.283\text{e-}10 & +2.106\text{e-}10 \\ -1.559\text{e-}5 & -3.490\text{e-}5 & -2.363\text{e-}5 & +1.208\text{e-}9 & +2.106\text{e-}10 & +1.179\text{e-}8 \end{bmatrix}$
Square root of tr(P_{rr}) [km]	1.964
Square root of tr(P_{vv}) [km/s]	1.147e-4

Table 7: Initial state and covariance matrix which are solution of the navigation problem with keplerian motion and both PERTH and KOUROU stations

(c) Last we use the same case as before, with the two stations. However, we will consider a J2-perturbed movement as described in the slides of the assignment. I proceeded as follows to compute the acceleration to add to keplerian acceleration du to J2 perturbation. First, I computed the matrix to convert position from ECI frame to ECEF frame. Then with the position converted to ECEF I computed the acceleration du to J2 as $\boldsymbol{a}_{J_2} = \frac{3}{2}\mu J_2 \frac{\boldsymbol{r}}{r^3}\left(\frac{R_E}{r}\right)^2 \left(5\left(\frac{z}{r}\right)^2 - [1\ 1\ 3]^T\right)$. I then converted it to ECI and added it to keplerian acceleration.

	Value
Mean state $\hat{\boldsymbol{x}}_0$ [km, km/s]	$\hat{\boldsymbol{r}}_0 = [-5.812\text{e}{+4},\ +3.010\text{e}{+4},\ 1.240\text{e}{+3}]$ $\hat{\boldsymbol{v}}_0 = [-5.235\text{e-}1,\ -9.311\text{e-}1,\ +6.663\text{e-}2]$
Covariance P_0 [km^2, km^2/s, km^2/s^2]	$\begin{bmatrix} +5.792\text{e-}2 & +1.109\text{e-}1 & +8.797\text{e-}3 & -2.006\text{e-}6 & +1.152\text{e-}6 & -4.029\text{e-}6 \\ +1.109\text{e-}1 & +2.164\text{e-}1 & +1.576\text{e-}2 & -3.814\text{e-}6 & +2.354\text{e-}6 & -7.762\text{e-}6 \\ +8.797\text{e-}3 & +1.576\text{e-}2 & +4.302\text{e-}3 & -3.583\text{e-}7 & +1.707\text{e-}7 & -1.195\text{e-}6 \\ -2.006\text{e-}6 & -3.814\text{e-}6 & -3.583\text{e-}7 & +7.486\text{e-}11 & -4.026\text{e-}11 & +1.807\text{e-}10 \\ +1.152\text{e-}6 & +2.354\text{e-}6 & +1.707\text{e-}7 & -4.026\text{e-}11 & +3.143\text{e-}11 & -7.491\text{e-}11 \\ -4.029\text{e-}6 & -7.762\text{e-}6 & -1.195\text{e-}6 & +1.807\text{e-}10 & -7.491\text{e-}11 & +7.834\text{e-}10 \end{bmatrix}$
Square root of tr(P_{rr}) [km]	5.279e-1
Square root of tr(P_{vv}) [km/s]	2.982e-5

Table 8: Initial state and covariance matrix which are solution of the navigation problem with perturbed motion and both PERTH and KOUROU stations

Exercise 3: Sequential filters

After the release from the Ariane 5 upper stage, the OSR unit unfurls its solar panels and uses its low-thrust plasma engines to reach its target spacecraft SGN-I. SGN-I is moving on a circular geostationary (GEO) orbit. Once in the GEO regime, the OSR unit approaches the target SGN-I to start its in-orbit servicing mission. The target can be modeled as a parallelepiped with properties reported in Table 9.

Table 9: Parameters of SGN-I.

Parameter	Value
Size [m]	$[l, h, d] = [10, 5, 3]$
Density [kg/m^3]	$\rho = 1420$
Mass [kg]	$m = lhd\rho = 213000$
Matrix of Inertia [kg/m^2]	$\mathbf{J} = \frac{m}{12}\text{diag}\left([d^2+h^2, l^2+h^2, l^2+d^2]\right)$
Vertices [m]	$\mathbf{p} = \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \\ p_6 \\ p_7 \\ p_8 \end{bmatrix} = \begin{bmatrix} l/2 & -d/2 & -h/2 \\ l/2 & d/2 & -h/2 \\ l/2 & d/2 & h/2 \\ l/2 & -d/2 & h/2 \\ -l/2 & -d/2 & -h/2 \\ -l/2 & d/2 & -h/2 \\ -l/2 & d/2 & h/2 \\ -l/2 & -d/2 & h/2 \end{bmatrix}$

SGN-I has lost its capability of stabilizing its attitude and therefore it is tumbling according to Euler's equation of rigid body motion:

$$\mathbf{J}\dot{\boldsymbol{\omega}} = -\boldsymbol{\omega} \wedge \mathbf{J}\boldsymbol{\omega} \tag{3}$$

Where $\boldsymbol{\omega}$ represents the target's angular velocity with respect to the inertial frame, expressed in the target's body frame. In this equation $\boldsymbol{J}$ represents the matrix of inertia of the target. To complete the description of the dynamics, the target's attitude with respect to the inertial frame can be expressed via quaternions, which evolve according to the following differential equation:

$$\dot{\mathbf{q}} = \frac{1}{2}\begin{bmatrix} 0 & -\boldsymbol{\omega}(1) & -\boldsymbol{\omega}(2) & -\boldsymbol{\omega}(3) \\ \boldsymbol{\omega}(1) & 0 & \boldsymbol{\omega}(3) & -\boldsymbol{\omega}(2) \\ \boldsymbol{\omega}(2) & -\boldsymbol{\omega}(3) & 0 & \boldsymbol{\omega}(1) \\ \boldsymbol{\omega}(3) & \boldsymbol{\omega}(2) & -\boldsymbol{\omega}(1) & 0 \end{bmatrix}\mathbf{q} \tag{4}$$

The quaternion $\mathbf{q}$ can be used to extract the director cosine matrix necessary to express in the target body frame a point whose coordinates are given in the inertial frame (i.e., through matlab's *quat2dcm*):

$$\mathbf{C}_{T,I} = \begin{bmatrix} \mathbf{q}(1)^2+\mathbf{q}(2)^2-\mathbf{q}(3)^2-\mathbf{q}(4)^2 & 2(\mathbf{q}(2)\mathbf{q}(3)+\mathbf{q}(1)\mathbf{q}(4)) & 2(\mathbf{q}(2)\mathbf{q}(4)-\mathbf{q}(1)\mathbf{q}(3)) \\ 2(\mathbf{q}(2)\mathbf{q}(3)-\mathbf{q}(1)\mathbf{q}(4)) & \mathbf{q}(1)^2-\mathbf{q}(2)^2+\mathbf{q}(3)^2-\mathbf{q}(4)^2 & 2(\mathbf{q}(3)\mathbf{q}(4)+\mathbf{q}(1)\mathbf{q}(2)) \\ 2(\mathbf{q}(2)\mathbf{q}(4)+\mathbf{q}(1)\mathbf{q}(3)) & 2(\mathbf{q}(3)\mathbf{q}(4)-\mathbf{q}(1)\mathbf{q}(2)) & \mathbf{q}(1)^2-\mathbf{q}(2)^2-\mathbf{q}(3)^2+\mathbf{q}(4)^2 \end{bmatrix} \tag{5}$$

It is assumed that SGN-I can correctly estimate its attitude and provide it to the chaser via a direct link with the OSR. The initial value of the states for the target's attitude are reported in Table 10.

Therefore, the only state that needs to be estimated to retrieve the relative pose is the relative position and velocity expressed in the target's LVLH frame. Based on previous tracking

Table 10: Initial quaternion and angular velocity of the target body frame with respect to the inertial one.

Parameter	Value
Ref. epoch t_0 [UTC]	2023-04-01T14:55:12.023
Initial quaternion [-]	$\mathbf{q}(t_0) = \begin{bmatrix} 0.674156352338764 \\ 0.223585877389611 \\ 0.465489474399161 \\ 0.528055032413102 \end{bmatrix}$
Initial Angular velocity [rad/s]	$\boldsymbol{\omega}(t_0) = \begin{bmatrix} -0.001262427155865 \\ 0.001204540074343 \\ -0.000039180139156 \end{bmatrix}$

campaigns, you receive an initial estimate of this state (i.e., OSR with respect to SGN-I), provided in terms of mean vector and covariance matrix expressed in the LVLH reference frame centered at SGN-I (see Table 11).

Table 11: Estimate of the initial relative state state at t_0 in the target relative LVLH frame.

Parameter	Value
Ref. epoch t_0 [UTC]	2023-04-01T14:55:12.023
Mean state $\hat{\boldsymbol{x}}_0$ [m, m/s]	$\hat{\boldsymbol{r}}_0 =$ [15.792658268071492 -59.044939772661586 3.227106250277039] $\hat{\boldsymbol{v}}_0 =$ [-0.053960274403210 -0.053969644762889 -0.089140748762173]
Covariance P_0 [m^2, m^2/s, m^2/s^2]	$\text{diag}\left([10, 10, 10, 0.1, 0.1, 0.1]\right)$

The motion is assumed to be correctly described by 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 the target along its GEO trajectory. Remember that the LVLH frame will rotate with respect to the inertial frame with a director cosine matrix that evolves as:

$$\mathbf{C}_{L,I} = \begin{bmatrix} \cos(nt) & \sin(nt) & 0 \\ -\sin(nt) & \cos(nt) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{7}$$

You are asked to develop a sequential filter to narrow down the uncertainty on the knowledge of the OSR relative state vector before executing the rendezvous procedure. To this aim, you can exploit the measurements obtained by a stereo camera onboard the OSR, whose features are reported in Table 12. It is assumed that the camera is pointed towards the V-axis of the LVLH frame during the entire navigation window, hence the needed Director Cosine Matrix

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

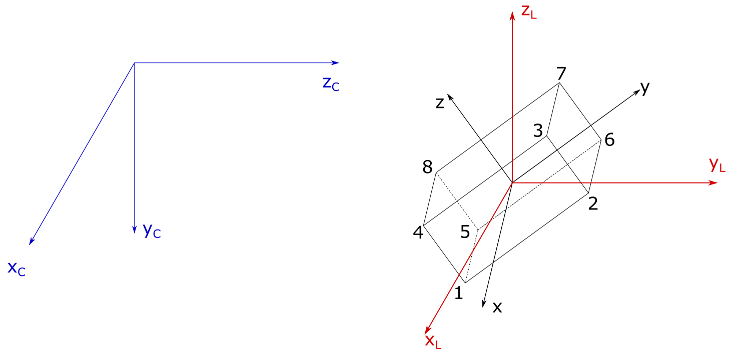


Figure 6: Pay attention to the three different reference frames, in blue the camera frame (C pedices), in red the LVLH frame (L pedices), and in black the target body frame.

that allows to rotate a vector from LVLH frame to camera frame can be expressed as:

$$\mathbf{C}_{\text{cam},L} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \tag{8}$$

The overall geometry of the observation is described in Figure 6.

The camera provides measurements of the vertices of the RSO in terms of horizontal pixel, vertical pixel, and disparity according to the model:

$$\mathbf{y} = \left[u_0 - Dfoc\frac{Y}{Z}, v_0 + Dfoc\frac{X}{Z}, \frac{bDfoc}{Z} \right] \tag{9}$$

where $[X, Y, Z]$ represent the coordinates of any vertex of the RSO expressed **in a reference frame fixed with the camera and centered in the chaser**. In the framework of this exercise, you are provided with a tool (i.e., *meas_sim*) to simulate the measurements acquired by the stereo camera during the entire navigation window. This tool will provide a set of stereo measurement for each *visible* vertex of the RSO and the ID of the corresponding vertex as indicated in Table 9. Visibility is automatically computed by taking into account both illumination conditions and relative position between target and chaser. For some configurations it may be impossible to view any vertex[†].

Table 12: Parameters of the stereo camera.

Parameter	Value
Focal length [mm]	$foc = 30$
Pixel Density [pix/mm]	$D = 54$
Baseline [m]	$b = 1$
Center Pixel [pix]	$[u_0, v_0] = [960, 600]$
Sensor Size [pix]	[1920,1200]
Measurement Noise [pix^2]	$\sigma_u^2 = \sigma_v^2 = \sigma_d^2 = 10$

1. Use the function *meas_sim* to simulate the measurements acquired by the stereo camera during a navigation window of 1 day when following the **nominal trajectory** (reference for error estimation) obtained by setting $\mathbf{r}_0 = [12.0, -60.0, 0.0]$ and $\mathbf{v}_0 = [1e -$

[†]In this case, the filter can only perform the prediction step, and not the correction.

$4, -2n\mathbf{r}_0(1), -1.2e-3]$. Compute the number of visible corners at each measurement instant and visualize the relative nominal trajectory. Assume that the camera is providing measurements with a frequency of 1Hz.
The provided function *meas_ sim* requires at each time instant t: the mean motion n, the relative state $\mathbf{r}$, the quaternion of the target $\mathbf{q}$, the time t passed since the initial epoch, and a structure *Camera* containing the camera parameters indicated with the following nomenclature: *Camera.f* for the focal length, *Camera.d* for pixel density, *Camera.p0* for the central pixel coordinates, *Camera.Cframe* for the director cosine matrix responsible for the rotation from LVLH to camera frame, and finally the parameter *Camera.R* to indicate the variance of the measurements.

2. Verify that the visible features of the target SGN-I lie inside the field-of-view of the camera during the entire navigation window.

3. Using both an extended Kalman filter (EKF) and an unscented Kalman filter (UKF) update sequentially the spacecraft state (in terms of mean and covariance) by processing the acquired measurements in chronological order[‡]

4. Compute the error along the navigation window between the estimated mean states and the true trajectory. Check the consistency between the covariance matrices estimated by the filter during the navigation window and the computed error. Elaborate on the comparison of the results obtained with the two filters.

[‡]Keep in mind that you can simulate measurements regardless of visibility, and then only use those components that are actually visible from real measurements.

1) Before anything, as we have to simulate the measurements at each time for a windows of 1 day we need to compute a time span of 1 day in ephemeris time, with a time step of 1 second, as the camera has a refreshment rate of 1Hz. Thus we have a time span of 86401 points we will loop on.

   Let's look at the required inputs to the `meas_sim` function we have to use at each time to simulate measurements.

   - First, we need the data of the mean motion n which can be computed, for a GEO orbit as $n = \sqrt{\dfrac{\mu}{R_3}}$ where $\mu = 398600\text{km}^3\text{s}^{-2}$ the standard gravitation parameter for Earth and $R = 42241.08$ m for a GEO orbit, taking into account the radius of Earth and of the orbit.
   - Then, we need the relative position of the chaser in LVLH frame. To obtain it at each time step of the time span, we have to integrate CW-equations, with the begin state as given in the question. Extracting the position of the state we computed that way gives us the data we need at each time step.
   - We also need a quaternion describing the orientation of the target with respect to the inertial frame. Here the equations (3) and (4) describing the evolution of angular velocity of a rigid body with Euler's equations, and linking quaternion evolution over time to angular velocity of the body. We encapsulate the two equations in a single `MatLab` function and I integrated it and eventually, I obtain the quaternion we need at each time step.
   - It is necessary to provide the reference epoch, which is "2023-04-01T14:55:12.023" in UTC, so I only had to convert it to ephemeris time and voilà. I only had to specify the duration between current time and reference epoch which can be computed substracting the reference ephemeris time to the current time step of the time span.
   - Finally, we need a structure representing the camera which can be easily provided compiling the data we have on the camera to the supposed field of the structure.

   Integrating Clohessy-Wiltshire (abreviated CW) I computed the nominal relative trajectory in LVLH frame as follows :

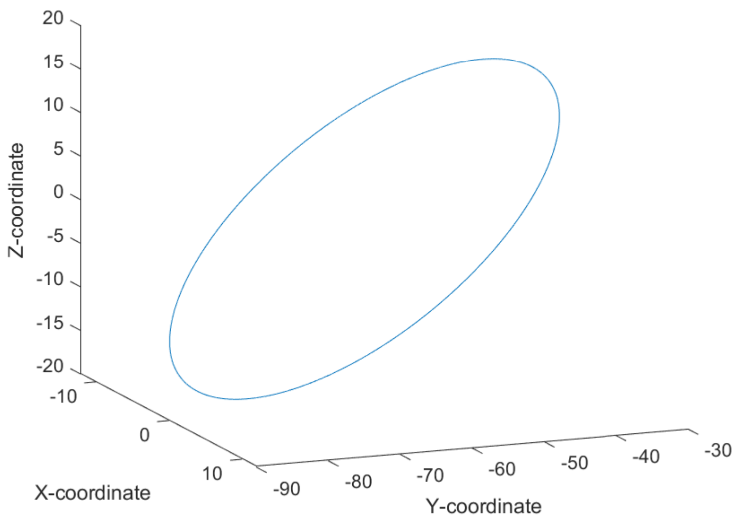


Figure 7: Nominal relative trajectory in LVLH frame - coordinates are in meters

Eventually, after having simulated the measurements as asked, here are the visible vertices at each time :

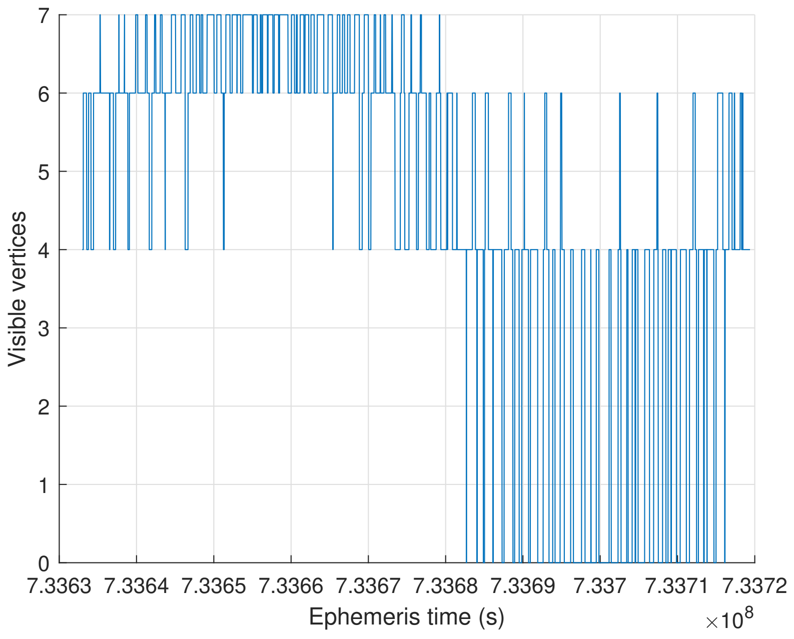


Figure 8: Number of visible vertices

2) We can see that the spacecraft is in the field of view of the camera by plotting the visible vertices simulated on the camera screen as follows :

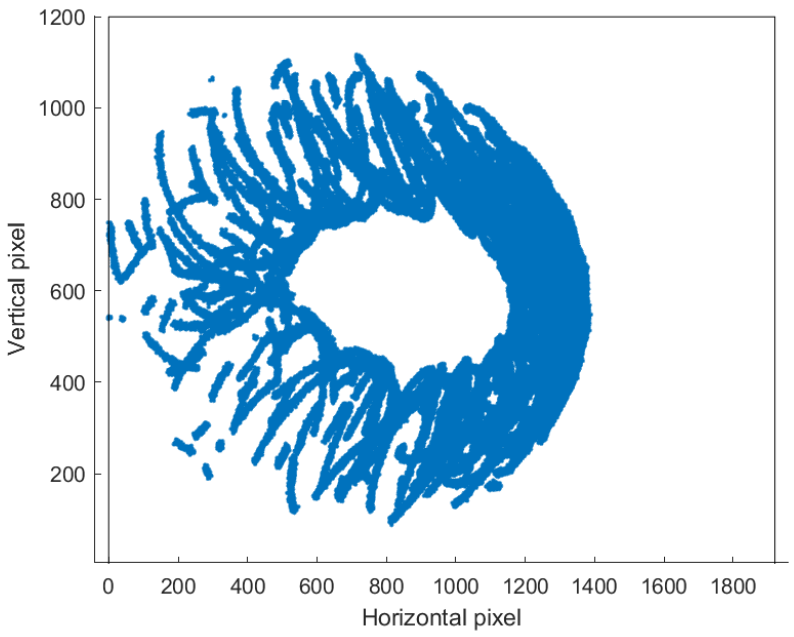


Figure 9: Screen of the camera

We can see that only five points are out of the field of view of the camera

3) For the implementation of the filters, we have to propagate the mean state $\hat{x}_0$ and covariance P_0 given in Table 11 using the techniques of UKF and EKF.

UKF Let's detail the update step to get $\hat{x}_k^+$ and P_k^+ from $\hat{x}_{k-1}^+$ and P_{k-1}^+. I will first, as I did for UT, generate n sigma points, propagate them by integrating CW equations, compute the "sigma"-measurements using the measurements equations. For the last point, I will have to convert the coordinates of the vertex to camera frame. Then I will have to compute UT weights to compute means and covariances as detailed on the slides, before computing Kalman gain and then $\hat{x}_k^+$ and P_k^+. The equations are detailed on the slides.

EKF Let's detail the update step to get $\hat{x}_k^+$ and P_k^+ from $\hat{x}_{k-1}^+$ and P_{k-1}^+. We first have to use $\hat{x}_{k-1}^+$ in our propagator to get the a priori state $\hat{x}_k^-$ and the STM between time $k-1$ and k, noted $\Phi_{k-1,k}$. This STM will be used to compute the a priori covariance $P_k^- = \Phi_{k-1,k} P_{k-1}^+ \Phi_{k-1,k}^T$. The state transition matrix have an analytic form as follows, with $h = t_k - t_{k-1}$ the time step :

$$\Phi_{k-1,k} = \begin{bmatrix} 4-3\cos nh & 0 & 0 & \frac{1}{n}\sin nh & \frac{2}{n}(1-\cos nh) & 0 \\ 6(\sin nh - nh) & 1 & 0 & \frac{2}{n}(\cos nh - 1) & \frac{1}{n}(4\sin nh - 3nh) & 0 \\ 0 & 0 & \cos nh & 0 & 0 & \frac{1}{n}\sin nt \\ 3n\sin nh & 0 & 0 & \cos nh & 2\sin nt & 0 \\ 6n(\cos nh - 1) & 0 & 0 & -2\sin nh & 4\cos nh - 3 & 0 \\ 0 & 0 & -n\sin nh & 0 & 0 & \cos nh \end{bmatrix}$$

Now we have to use our measurements. We can compute then with the propagated mean state, with the following equation $\mathbf{y} = \left[u_0 - Dfoc\frac{Y}{Z}, v_0 + Dfoc\frac{X}{Z}, \frac{bDfoc}{Z}\right]$ with X, Y and Z the coordinate of a vertex in the camera frame, by multiplying the coordinates of the vertex $[p_x - x, p_y - y, p_z - z]$ by $C_{cam,L}$, with $\boldsymbol{p} = [p_x, p_y, p_z]$ the vertex measurements. We will also need the jacobian of the measurement equation which can be computed as follows, noting $\boldsymbol{X} = [X, Y, Z]$ and $\boldsymbol{x} = [x, y, z]$:

$$\frac{\partial y}{\partial \boldsymbol{X}} = \begin{bmatrix} 0 & -\frac{D_{foc}}{2} & \frac{D_{foc}Y}{4} \\ \frac{D_{foc}}{2} & 0 & -\frac{D_{foc}X}{4} \\ 0 & 0 & \frac{-D_{foc}b}{4} \end{bmatrix}$$

$$\boldsymbol{X} = C_{cam,L}(\boldsymbol{p} - \boldsymbol{x}) \implies \frac{\partial \boldsymbol{X}}{\partial \boldsymbol{x}} = -C_{cam,L}$$

And eventually, extending the matrix to the whole state vector, for one vertex :

$$H_k = \frac{\partial y}{\partial \boldsymbol{X}}\frac{\partial \boldsymbol{X}}{\partial \boldsymbol{x}} = \begin{bmatrix} 0 & -\frac{D_{foc}Y}{4} & -\frac{D_{foc}}{2} & 0 & 0 & 0 \\ -\frac{D_{foc}}{2} & \frac{D_{foc}X}{4} & 0 & 0 & 0 & 0 \\ 0 & \frac{D_{foc}b}{4} & 0 & 0 & 0 & 0 \end{bmatrix}$$

In the end, we can compute the Kalman Gain K_k to compute the updated mean state $\hat{x}_k^+$ and covariance matrix P_k^+ as detailed in the slides of the assignment.

The following plot shows the results with the two filters compared to the reference trajectory :

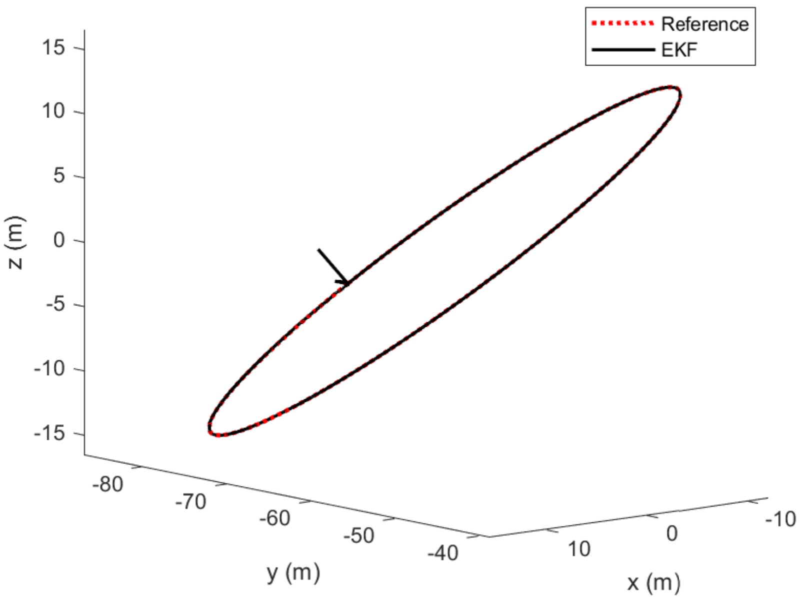


Figure 10: EKF trajectory compared to reference trajectory in LVLH frame

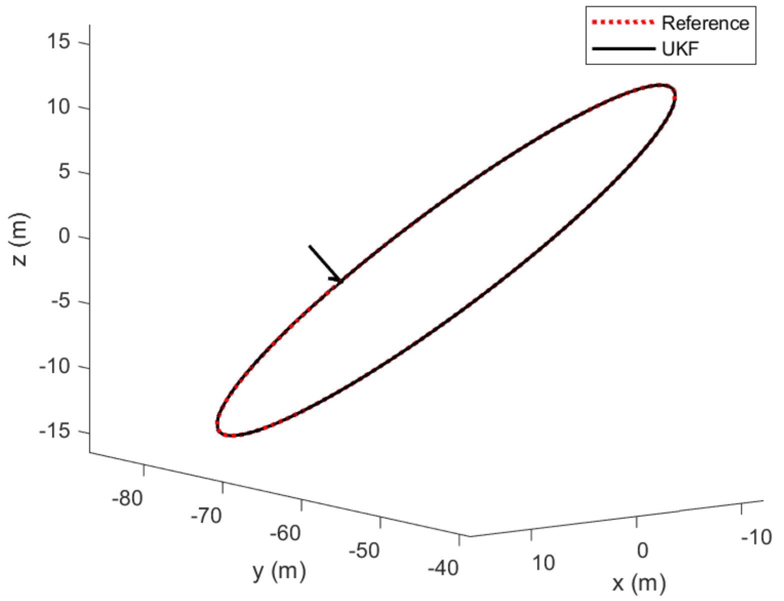


Figure 11: UKF trajectory compared to reference trajectory in LVLH frame