



# SGN – Assignment #2

Matteo Luciardello Lecardi, 977701

June 15, 2023

**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 1. 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 Gaussian 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 $\hat{\mathbf{x}}_0$ [km, km/s]	$\hat{\mathbf{r}}_0 = [6054.30795817484, -3072.03883303992, -133.115352431876]$ $\hat{\mathbf{v}}_0 = [4.64750094824087, 9.18608475681236, -0.62056520749034]$
Covariance $P_0$ [km <sup>2</sup> , km <sup>2</sup> /s, km <sup>2</sup> /s <sup>2</sup> ]	$\begin{bmatrix} +5.6e-3 & +3.5e-3 & -7.1e-4 & 0 & 0 & 0 \\ +3.5e-3 & +9.7e-3 & +7.6e-4 & 0 & 0 & 0 \\ -7.1e-4 & +7.6e-4 & +8.1e-4 & 0 & 0 & 0 \\ 0 & 0 & 0 & +2.8e-7 & 0 & 0 \\ 0 & 0 & 0 & 0 & +2.7e-7 & 0 \\ 0 & 0 & 0 & 0 & 0 & +9.6e-8 \end{bmatrix}$



- 1) Propagated **mean** and **covariance** at apocenters and pericenters of each revolution, expressed in ECI J2000 reference frame (see A 1.1 for brief discussion on the results):

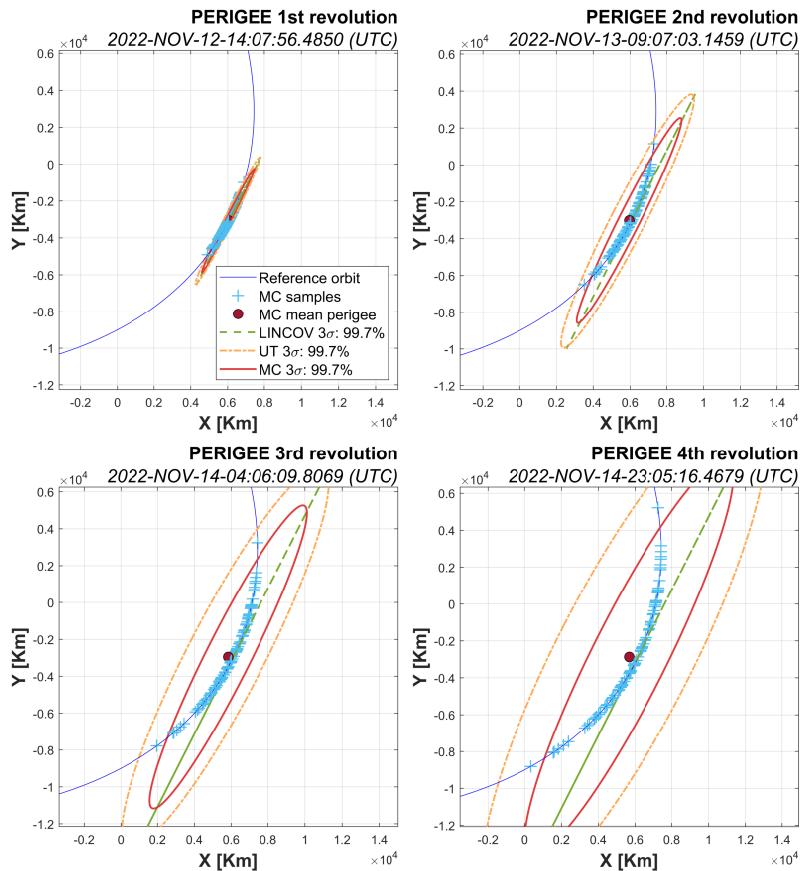
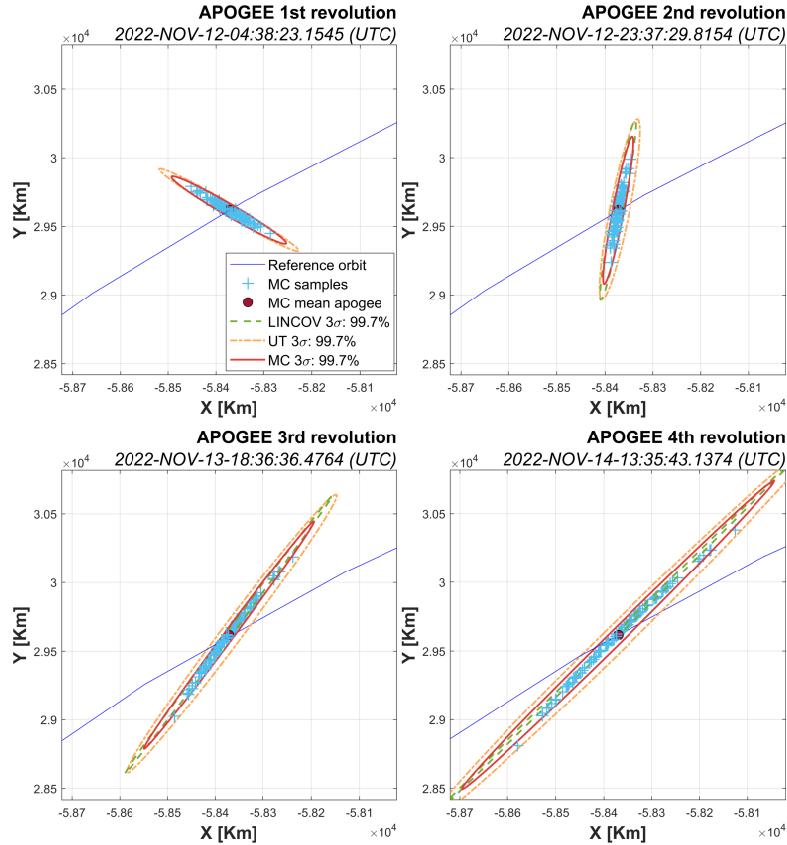
	apocenters mean $\hat{\mathbf{x}}$ [Km, Km/s] and covariance $\mathbf{P}$ [km <sup>2</sup> , km <sup>2</sup> /s, km <sup>2</sup> /s <sup>2</sup> ]					
	1st: 2022-NOV-12-04:38:23.1545 (UTC)			2nd: 2022-NOV-12-23:37:29.8154 (UTC)		
	LINCOV	UT	MC (n: 100)	LINCOV	UT	MC
$\hat{x}$	$-5.837\,283 \times 10^4$	$-5.837\,270 \times 10^4$	$-5.837\,285 \times 10^4$	$-5.837\,283 \times 10^4$	$-5.836\,918 \times 10^4$	$-5.837\,246 \times 10^4$
$\hat{y}$	$2.961\,917 \times 10^4$	$2.961\,926 \times 10^4$	$2.961\,849 \times 10^4$	$2.961\,917 \times 10^4$	$2.962\,496 \times 10^4$	$2.961\,788 \times 10^4$
$\hat{z}$	$1.283\,437 \times 10^3$	$1.283\,427 \times 10^3$	$1.283\,510 \times 10^3$	$1.283\,437 \times 10^3$	$1.283\,014 \times 10^3$	$1.283\,520 \times 10^3$
$\hat{u}$	$-4.820\,291 \times 10^{-1}$	$-4.820\,343 \times 10^{-1}$	$-4.820\,072 \times 10^{-1}$	$-4.820\,291 \times 10^{-1}$	$-4.825\,343 \times 10^{-1}$	$-4.819\,678 \times 10^{-1}$
$\hat{v}$	$-9.527\,616 \times 10^{-1}$	$-9.527\,573 \times 10^{-1}$	$-9.527\,766 \times 10^{-1}$	$-9.527\,616 \times 10^{-1}$	$-9.524\,798 \times 10^{-1}$	$-9.527\,852 \times 10^{-1}$
$\hat{w}$	$6.436\,373 \times 10^{-2}$	$6.436\,377 \times 10^{-2}$	$6.436\,216 \times 10^{-2}$	$6.436\,373 \times 10^{-2}$	$6.437\,370 \times 10^{-2}$	$6.436\,078 \times 10^{-2}$
$\sqrt{tr(\mathbf{P}_{rr})}$	$8.915\,554 \times 10^1$	$8.914\,955 \times 10^1$	$7.307\,917 \times 10^1$	$1.761\,379 \times 10^2$	$1.763\,567 \times 10^2$	$1.442\,673 \times 10^2$
$\sqrt{tr(\mathbf{P}_{vv})}$	$4.797\,317 \times 10^{-3}$	$4.796\,305 \times 10^{-3}$	$3.930\,698 \times 10^{-3}$	$1.404\,479 \times 10^{-2}$	$1.406\,430 \times 10^{-2}$	$1.151\,417 \times 10^{-2}$

	3rd: 2022-NOV-13-18:36:36.4764 (UTC)			4th: 2022-NOV-14-13:35:43.1374 (UTC)		
	LINCOV	UT	MC	LINCOV	UT	MC
	$-5.837\,283 \times 10^4$	$-5.836\,628 \times 10^4$	$-5.837\,151 \times 10^4$	$-5.837\,283 \times 10^4$	$-5.836\,399 \times 10^4$	$-5.837\,001 \times 10^4$
$\hat{x}$	$2.961\,917 \times 10^4$	$2.962\,694 \times 10^4$	$2.961\,699 \times 10^4$	$2.961\,917 \times 10^4$	$2.962\,532 \times 10^4$	$2.961\,581 \times 10^4$
$\hat{y}$	$1.283\,436 \times 10^3$	$1.282\,795 \times 10^3$	$1.283\,518 \times 10^3$	$1.283\,436 \times 10^3$	$1.282\,765 \times 10^3$	$1.283\,503 \times 10^3$
$\hat{z}$	$-4.820\,291 \times 10^{-1}$	$-4.827\,545 \times 10^{-1}$	$-4.819\,237 \times 10^{-1}$	$-4.820\,291 \times 10^{-1}$	$-4.826\,851 \times 10^{-1}$	$-4.818\,751 \times 10^{-1}$
$\hat{u}$	$-9.527\,616 \times 10^{-1}$	$-9.523\,299 \times 10^{-1}$	$-9.527\,848 \times 10^{-1}$	$-9.527\,616 \times 10^{-1}$	$-9.523\,139 \times 10^{-1}$	$-9.527\,753 \times 10^{-1}$
$\hat{v}$	$6.436\,373 \times 10^{-2}$	$6.437\,683 \times 10^{-2}$	$6.435\,880 \times 10^{-2}$	$6.436\,373 \times 10^{-2}$	$6.437\,300 \times 10^{-2}$	$6.435\,620 \times 10^{-2}$
$\sqrt{tr(\mathbf{P}_{rr})}$	$2.777\,746 \times 10^2$	$2.780\,481 \times 10^2$	$2.275\,805 \times 10^2$	$3.825\,072 \times 10^2$	$3.827\,087 \times 10^2$	$3.135\,051 \times 10^2$
$\sqrt{tr(\mathbf{P}_{vv})}$	$2.336\,168 \times 10^{-2}$	$2.338\,345 \times 10^{-2}$	$1.915\,444 \times 10^{-2}$	$3.268\,404 \times 10^{-2}$	$3.269\,451 \times 10^{-2}$	$2.680\,371 \times 10^{-2}$

	pericenters mean $\hat{\mathbf{x}}$ [Km, Km/s] and covariance $\mathbf{P}$ [km <sup>2</sup> , km <sup>2</sup> /s, km <sup>2</sup> /s <sup>2</sup> ]					
	1st: 2022-NOV-12-14:07:56.4850 (UTC)			2nd: 2022-NOV-13-09:07:03.1459 (UTC)		
	LINCOV	UT	MC (n: 100)	LINCOV	UT	MC
$\hat{x}$	$6.054\,308 \times 10^3$	$6.006\,611 \times 10^3$	$6.029\,910 \times 10^3$	$6.054\,308 \times 10^3$	$5.868\,757 \times 10^3$	$5.956\,099 \times 10^3$
$\hat{y}$	$-3.072\,039 \times 10^3$	$-3.069\,948 \times 10^3$	$-3.056\,530 \times 10^3$	$-3.072\,039 \times 10^3$	$-3.053\,325 \times 10^3$	$-3.016\,103 \times 10^3$
$\hat{z}$	$-1.331\,154 \times 10^2$	$-1.310\,758 \times 10^2$	$-1.327\,249 \times 10^2$	$-1.331\,154 \times 10^2$	$-1.256\,523 \times 10^2$	$-1.312\,366 \times 10^2$
$\hat{u}$	4.647 501	4.632 447	4.625 878	4.647 501	4.578 676	4.569 143
$\hat{v}$	9.186 085	9.119 759	9.148 579	9.186 085	8.925 154	9.040 706
$\hat{w}$	$-6.205\,652 \times 10^{-1}$	$-6.169\,131 \times 10^{-1}$	$-6.179\,011 \times 10^{-1}$	$-6.205\,652 \times 10^{-1}$	$-6.057\,674 \times 10^{-1}$	$-6.105\,176 \times 10^{-1}$
$\sqrt{tr(\mathbf{P}_{rr})}$	$1.034\,993 \times 10^3$	$1.036\,964 \times 10^3$	$8.463\,406 \times 10^2$	$2.070\,079 \times 10^3$	$2.086\,265 \times 10^3$	$1.678\,559 \times 10^3$
$\sqrt{tr(\mathbf{P}_{vv})}$	$8.675\,206 \times 10^{-1}$	$8.726\,567 \times 10^{-1}$	$7.042\,501 \times 10^{-1}$	1.735 110	1.776 233	1.368 949

	3rd: 2022-NOV-14-04:06:09.8069 (UTC)			4th: 2022-NOV-14-23:05:16.4679 (UTC)		
	LINCOV	UT	MC	LINCOV	UT	MC
	$6.054\,308 \times 10^3$	$5.664\,379 \times 10^3$	$5.837\,945 \times 10^3$	$6.054\,308 \times 10^3$	$5.401\,640 \times 10^3$	$5.681\,906 \times 10^3$
$\hat{x}$	$-3.072\,039 \times 10^3$	$-2.975\,524 \times 10^3$	$-2.953\,511 \times 10^3$	$-3.072\,039 \times 10^3$	$-2.820\,402 \times 10^3$	$-2.872\,255 \times 10^3$
$\hat{y}$	$-1.331\,154 \times 10^2$	$-1.199\,976 \times 10^2$	$-1.287\,583 \times 10^2$	$-1.331\,154 \times 10^2$	$-1.152\,020 \times 10^2$	$-1.254\,218 \times 10^2$
$\hat{z}$	4.647 501	4.447 047	4.486 094	4.647 501	4.224 013	4.386 457
$\hat{u}$	9.186 085	8.622 102	8.878 739	9.186 085	8.217 497	8.680 530
$\hat{v}$	$-6.205\,652 \times 10^{-1}$	$-5.862\,662 \times 10^{-1}$	$-5.995\,267 \times 10^{-1}$	$-6.205\,652 \times 10^{-1}$	$-5.581\,125 \times 10^{-1}$	$-5.861\,657 \times 10^{-1}$
$\hat{w}$	$3.105\,094 \times 10^3$	$3.155\,846 \times 10^3$	$2.485\,740 \times 10^3$	$4.139\,959 \times 10^3$	$4.255\,616 \times 10^3$	$3.261\,346 \times 10^3$
$\sqrt{tr(\mathbf{P}_{rr})}$	2.602 641	2.736 581	1.968 791	3.470 045	3.778 656	2.495 100

- 2) Lincov, UT and MC samples and covariance ellipses at each revolution in ECI J2000 on equatorial plane (see A 1.2 for separate plots with samples, and validation):





## 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 2.

1. *Compute visibility windows.* By using the mean state reported in Table 1 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 3 (and in WeBeep as 87654.tle). 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).
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 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 2:** Sensor network to track the Ariane 5 upper stage: list of stations, including their features.

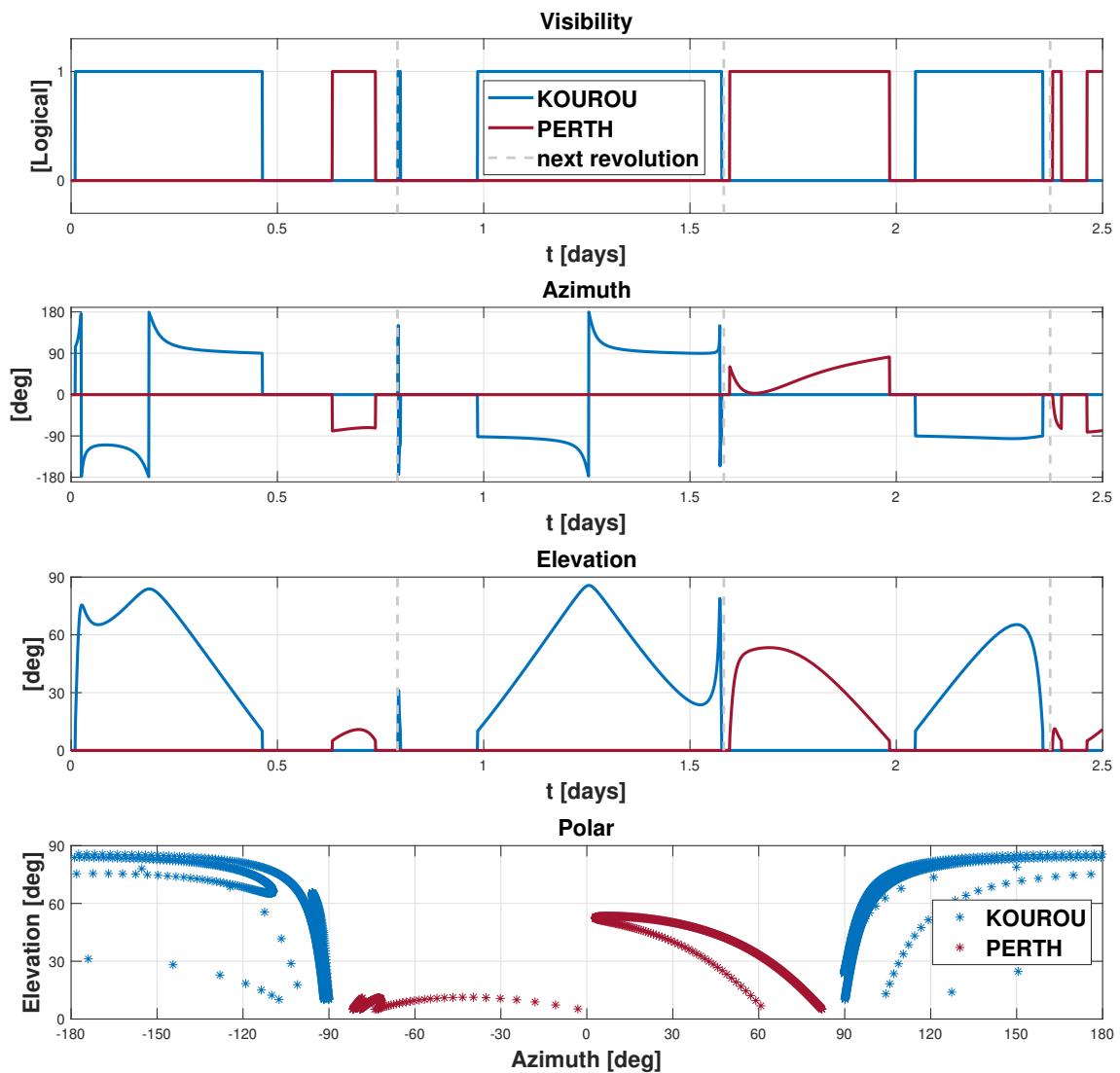
Station name	KOUROU	PERTH
Coordinates	LAT = 5.25144° LON = -52.80466° ALT = -14.67 m	LAT = -31.80252° LON = 115.88516° ALT = 22.16 m
Type	Radar (monostatic)	Radar (monostatic)
Provided measurements	Az, El [deg] Range (one-way) [km]	Az, El [deg] Range (one-way) [km]
Measurements noise (diagonal noise matrix R)	$\sigma_{Az,El} = 100$ mdeg $\sigma_{range} = 0.01$ km	$\sigma_{Az,El} = 100$ mdeg $\sigma_{range} = 0.01$ km
Minimum elevation	10 deg	5 deg

**Table 3:** Latest available TLE of the Ariane 5 upper stage.

1_87654U_22110B_22316.00967942_0.0000002_00000-0_32024-3_0_9990
2_87654_3.6309_137.1541_8138191_196.1632_96.6141_1.26411866_834

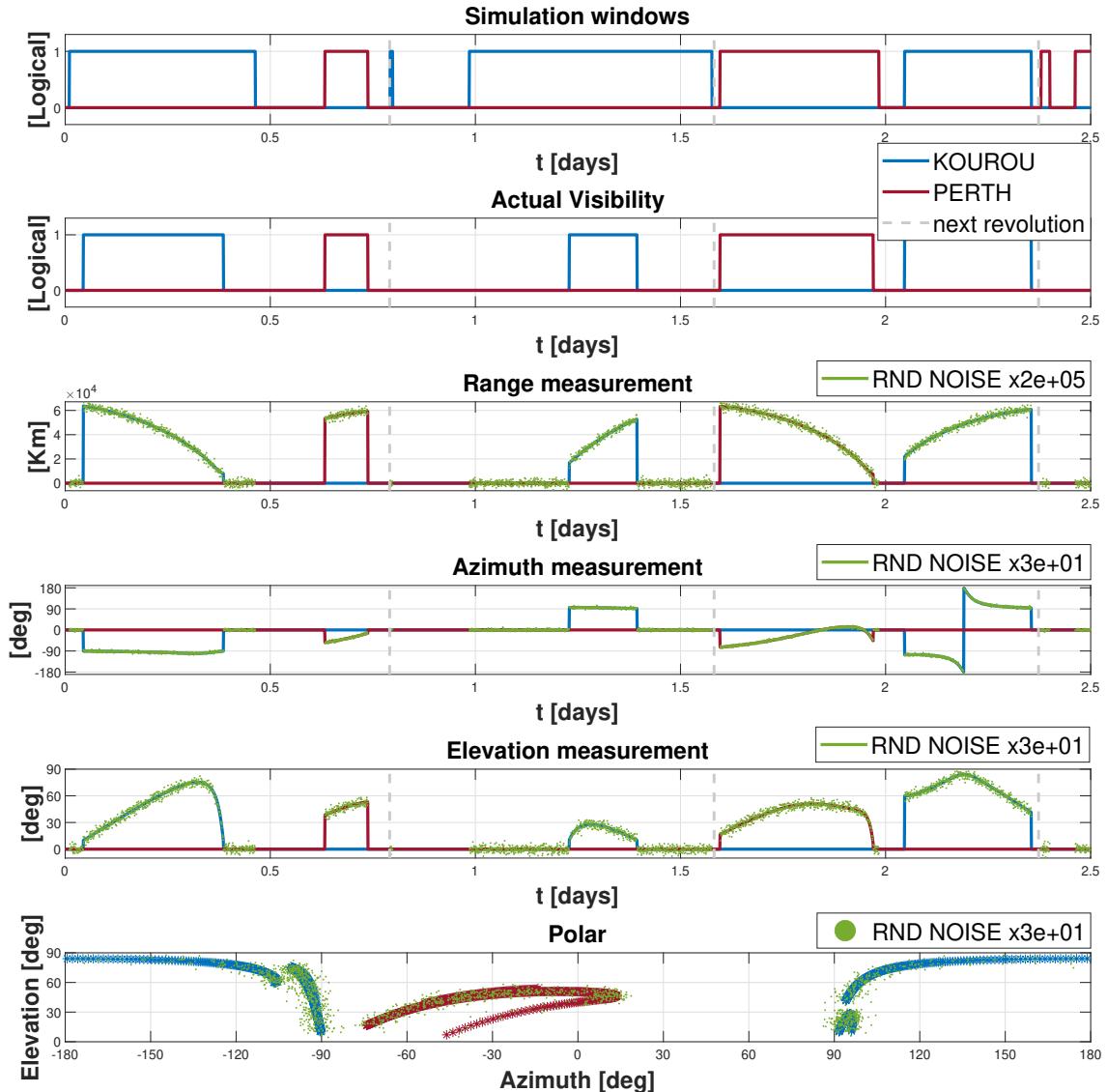
1. Visibility windows prediction (propagation plots in A 2.1):

Pass#	Station Name	Start Time(UTC)	End Time(UTC)
1	KOUROU	2022-NOV-12-04:45:00.000	2022-NOV-12-15:37:00.000
2	PERTH	2022-NOV-12-19:42:00.000	2022-NOV-12-22:12:00.000
3	KOUROU	2022-NOV-12-23:31:00.000	2022-NOV-12-23:39:00.000
4	KOUROU	2022-NOV-13-04:09:00.000	2022-NOV-13-18:20:00.000
5	PERTH	2022-NOV-13-18:49:00.000	2022-NOV-14-04:06:00.000
6	KOUROU	2022-NOV-14-05:37:00.000	2022-NOV-14-13:01:00.000
7	PERTH	2022-NOV-14-13:36:00.000	2022-NOV-14-14:06:00.000
8	PERTH	2022-NOV-14-15:36:00.000	2022-NOV-14-16:30:00.000





2. Expected measurements and Gaussian multivariate random noise, which magnitude has been augmented exclusively for readability purposes [(a) (b)]:



(propagation plots in A 2.2)

3. Least-squares solution for (a), (b) and (c) cases (@ECI J2000):

	$\mathbf{x}_0$ [Km, Km/s] and $\mathbf{P}_0$ [ $\text{km}^2$ , $\text{km}^2/\text{s}$ , $\text{km}^2/\text{s}^2$ ]		
	(a)	(b)	(c)
$x_0$	$-5.829\ 012 \times 10^4$	$-5.813\ 493 \times 10^4$	$-5.812\ 615 \times 10^4$
$y_0$	$2.976\ 785 \times 10^4$	$3.006\ 828 \times 10^4$	$3.009\ 585 \times 10^4$
$z_0$	$1.260\ 863 \times 10^3$	$1.261\ 985 \times 10^3$	$1.249\ 570 \times 10^3$
$u_0$	$-5.192\ 588 \times 10^{-1}$	$-5.232\ 290 \times 10^{-1}$	$-5.244\ 531 \times 10^{-1}$
$v_0$	$-9.332\ 352 \times 10^{-1}$	$-9.311\ 881 \times 10^{-1}$	$-9.310\ 048 \times 10^{-1}$
$w_0$	$6.576\ 284 \times 10^{-2}$	$6.846\ 346 \times 10^{-2}$	$5.609\ 232 \times 10^{-2}$
$\sqrt{\text{tr}(\mathbf{P}_{0\text{rr}})}$	$7.778\ 530 \times 10^{-2}$	$2.137\ 082 \times 10^5$	$3.512\ 373 \times 10^3$
$\sqrt{\text{tr}(\mathbf{P}_{0\text{vv}})}$	$4.410\ 754 \times 10^{-10}$	$4.330\ 511 \times 10^{-3}$	$5.855\ 269 \times 10^{-6}$

(Example plot in A 2.3)



Covariance $\mathbf{P}_0$ [km <sup>2</sup> , km <sup>2</sup> /s, km <sup>2</sup> /s <sup>2</sup> ] for (a)					
$1.122 \times 10^{-2}$	$2.535 \times 10^{-2}$	$2.459 \times 10^{-3}$	$-2.875 \times 10^{-7}$	$5.289 \times 10^{-7}$	$1.759 \times 10^{-6}$
$2.535 \times 10^{-2}$	$5.754 \times 10^{-2}$	$4.726 \times 10^{-3}$	$-6.450 \times 10^{-7}$	$1.211 \times 10^{-6}$	$4.084 \times 10^{-6}$
$2.459 \times 10^{-3}$	$4.726 \times 10^{-3}$	$9.028 \times 10^{-3}$	$-8.510 \times 10^{-8}$	$1.106 \times 10^{-7}$	$1.144 \times 10^{-7}$
$-2.875 \times 10^{-7}$	$-6.450 \times 10^{-7}$	$-8.510 \times 10^{-8}$	$8.189 \times 10^{-12}$	$-1.350 \times 10^{-11}$	$-3.905 \times 10^{-11}$
$5.289 \times 10^{-7}$	$1.211 \times 10^{-6}$	$1.106 \times 10^{-7}$	$-1.350 \times 10^{-11}$	$2.747 \times 10^{-11}$	$9.724 \times 10^{-11}$
$1.759 \times 10^{-6}$	$4.084 \times 10^{-6}$	$1.144 \times 10^{-7}$	$-3.905 \times 10^{-11}$	$9.724 \times 10^{-11}$	$4.054 \times 10^{-10}$

Covariance $\mathbf{P}_0$ [km <sup>2</sup> , km <sup>2</sup> /s, km <sup>2</sup> /s <sup>2</sup> ] for (b)					
$3.669 \times 10^4$	$7.407 \times 10^4$	$5.123 \times 10^3$	$-1.229$	$9.997 \times 10^{-1}$	$8.469 \times 10^{-1}$
$7.407 \times 10^4$	$1.538 \times 10^5$	$1.495 \times 10^4$	$-2.398$	$2.228$	$2.442$
$5.123 \times 10^3$	$1.495 \times 10^4$	$2.325 \times 10^4$	$1.929 \times 10^{-1}$	$9.385 \times 10^{-1}$	$8.117$
$-1.229$	$-2.398$	$1.929 \times 10^{-1}$	$4.978 \times 10^{-5}$	$-1.882 \times 10^{-5}$	$1.371 \times 10^{-4}$
$9.997 \times 10^{-1}$	$2.228$	$9.385 \times 10^{-1}$	$-1.882 \times 10^{-5}$	$6.439 \times 10^{-5}$	$3.802 \times 10^{-4}$
$8.469 \times 10^{-1}$	$2.442$	$8.117$	$1.371 \times 10^{-4}$	$3.802 \times 10^{-4}$	$4.216 \times 10^{-3}$

Covariance $\mathbf{P}_0$ [km <sup>2</sup> , km <sup>2</sup> /s, km <sup>2</sup> /s <sup>2</sup> ] for (c)					
$4.206 \times 10^2$	$7.744 \times 10^2$	$3.949 \times 10^2$	$-1.467 \times 10^{-2}$	$6.438 \times 10^{-3}$	$-1.508 \times 10^{-2}$
$7.744 \times 10^2$	$1.482 \times 10^3$	$5.895 \times 10^2$	$-2.687 \times 10^{-2}$	$1.369 \times 10^{-2}$	$-3.362 \times 10^{-2}$
$3.949 \times 10^2$	$5.895 \times 10^2$	$1.610 \times 10^3$	$-1.279 \times 10^{-2}$	$-1.241 \times 10^{-3}$	$-2.887 \times 10^{-2}$
$-1.467 \times 10^{-2}$	$-2.687 \times 10^{-2}$	$-1.279 \times 10^{-2}$	$5.312 \times 10^{-7}$	$-2.316 \times 10^{-7}$	$5.840 \times 10^{-7}$
$6.438 \times 10^{-3}$	$1.369 \times 10^{-2}$	$-1.241 \times 10^{-3}$	$-2.316 \times 10^{-7}$	$1.896 \times 10^{-7}$	$-1.875 \times 10^{-7}$
$-1.508 \times 10^{-2}$	$-3.362 \times 10^{-2}$	$-2.887 \times 10^{-2}$	$5.840 \times 10^{-7}$	$-1.875 \times 10^{-7}$	$5.134 \times 10^{-6}$



### 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 4.

**Table 4:** Parameters of SGN-I.

Parameter	Value
Size [m]	$[l, h, d] = [10, 5, 3]$
Density [kg/m <sup>3</sup> ]	$\rho = 1420$
Mass [kg]	$m = lhd\rho = 213000$
Matrix of Inertia [kg/m <sup>2</sup> ]	$\mathbf{J} = \frac{m}{12} \text{diag}([d^2 + h^2, l^2 + h^2, l^2 + d^2])$
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:

$$\mathbf{J}\dot{\boldsymbol{\omega}} = -\boldsymbol{\omega} \wedge \mathbf{J}\boldsymbol{\omega} \quad (1)$$

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  $\mathbf{J}$  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:

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

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*):

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

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 5.

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 5:** 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 6).

**Table 6:** 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 $\hat{\mathbf{x}}_0$ [m, m/s]	$\hat{\mathbf{r}}_0 = [15.792658268071492 \ -59.044939772661586 \ 3.227106250277039]$ $\hat{\mathbf{v}}_0 = [-0.053960274403210 \ -0.053969644762889 \ -0.089140748762173]$
Covariance $P_0$ [m <sup>2</sup> , m <sup>2</sup> /s, m <sup>2</sup> /s <sup>2</sup> ]	diag ([10, 10, 10, 0.1, 0.1, 0.1])

The motion is assumed to be correctly described by the linear, Clohessy-Wiltshire (CW) equations\*

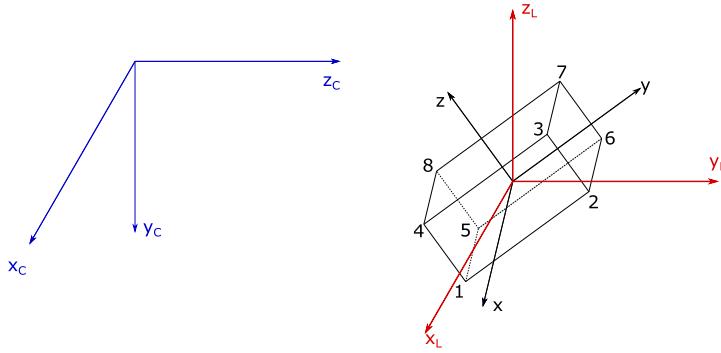
$$\begin{aligned}\ddot{x} &= 3n^2x + 2ny \\ \ddot{y} &= -2nx \\ \ddot{z} &= -n^2z\end{aligned}\tag{4}$$

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:

$$\mathbf{C}_{L,I} = \begin{bmatrix} \cos(nt) & \sin(nt) & 0 \\ -\sin(nt) & \cos(nt) & 0 \\ 0 & 0 & 1 \end{bmatrix}\tag{5}$$

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 7. 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  $\Phi$



**Figure 1:** 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:

$$\mathbf{C}_{\text{cam},L} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \quad (6)$$

The overall geometry of the observation is described in Figure 1.

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

$$\mathbf{y} = \left[ u_0 - Dfoc \frac{Y}{Z}, v_0 + Dfoc \frac{X}{Z}, \frac{bDfoc}{Z} \right] \quad (7)$$

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 4. 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<sup>†</sup>.

**Table 7:** 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 <sup>2</sup> ]	$\sigma_u^2 = \sigma_v^2 = \sigma_d^2 = 10$

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  $\mathbf{v}_0 = [1e -$

<sup>†</sup>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.

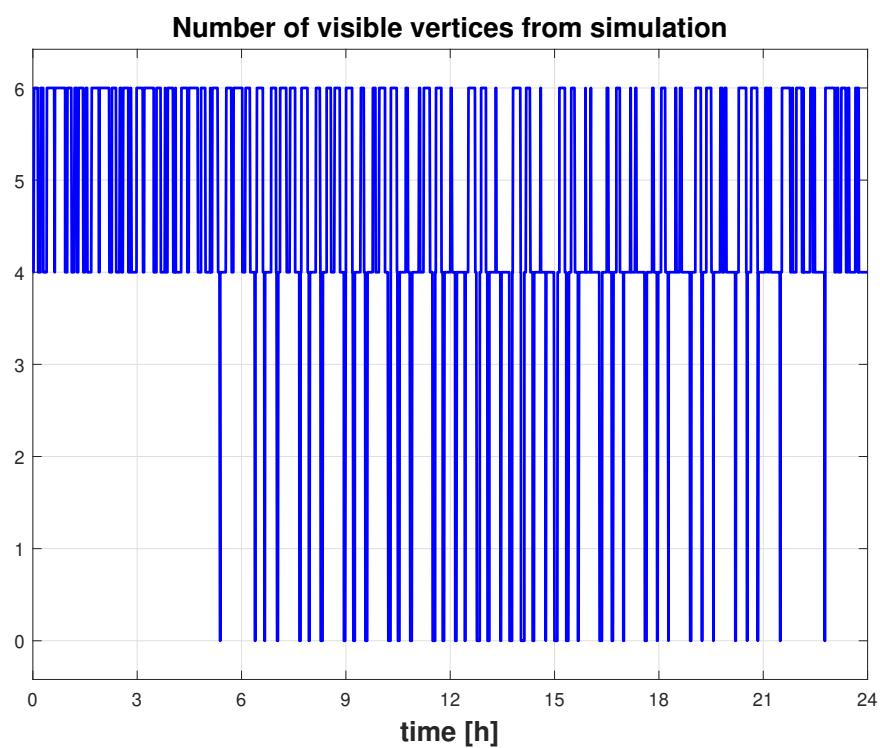
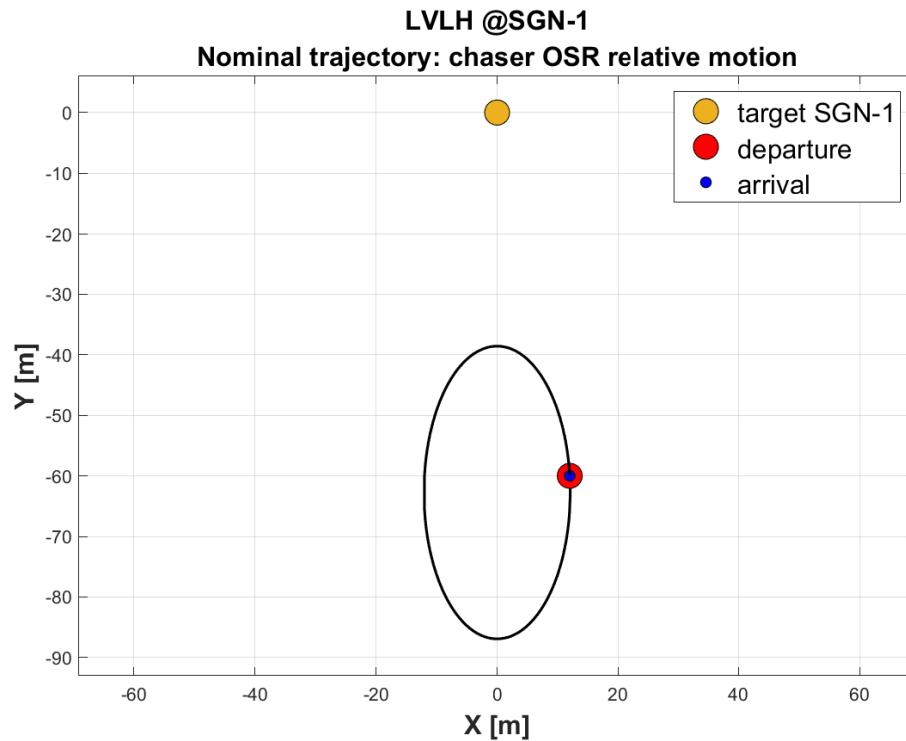
2. Verify that the visible features of the target SGN-I lie inside the field-of-view of the camera during the entire navigation window.
3. Using both an extended Kalman filter (EKF) and an unscented Kalman filter (UKF) update sequentially the spacecraft state (in terms of mean and covariance) by processing the acquired measurements in chronological order<sup>‡</sup>
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.

---

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

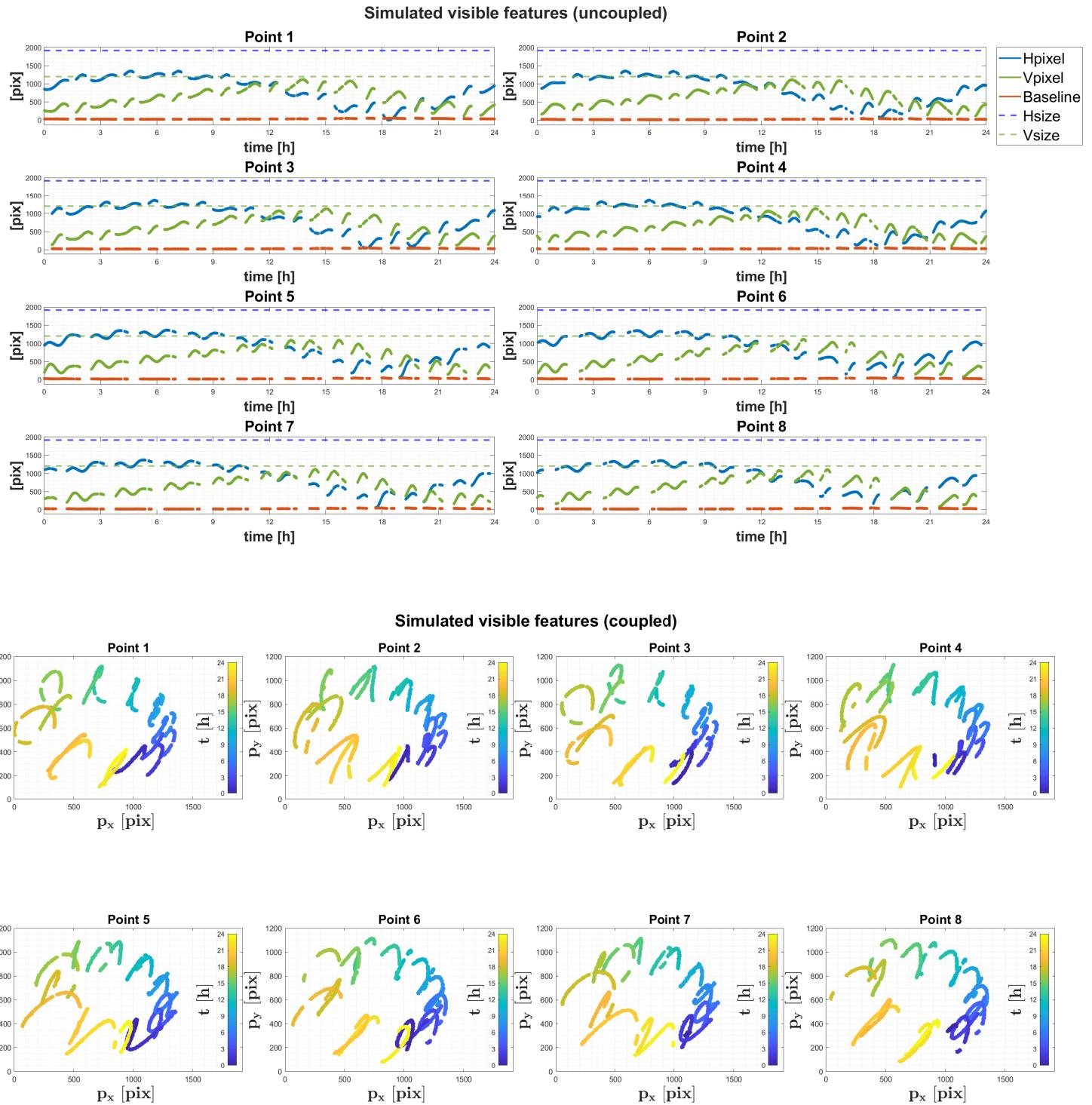


## 1. Nominal trajectory





## 2. Visible features within chaser camera Field-of-View





3. Extended Kalman Filter Jacobian of Measurements function:

$$H_k = \frac{\partial h(\hat{x}_k^-)}{\partial \hat{x}} = \frac{\partial h(\hat{x}_k^-)}{\partial \underline{x}_{cccf}} \frac{\partial \underline{x}_{cccf}}{\partial \hat{x}} \text{ Chain rule}$$

where  $\underline{x}_{cccf}$  is Chaser Centered Camera Fixed position of the 8 vertices:

- $\underline{x}_{cccf} = C_{CL} [\underline{r}_{TCLF} - \hat{x}_{kTCLF}^-]$  (Target Centered LVLH Fixed *vertices-chaser* positions)

$$\bullet H_k(\underline{x}_{cccf}) = \begin{bmatrix} 0 & -DfY/Z^2 & -Df/Z & 0 & 0 & 0 \\ -Df/Z & DfX/Z^2 & 0 & 0 & 0 & 0 \\ 0 & bDf/Z^2 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \quad (8)$$

$[3M \times 6]$  where M is number of visible vertices;  $f$  is the focal length

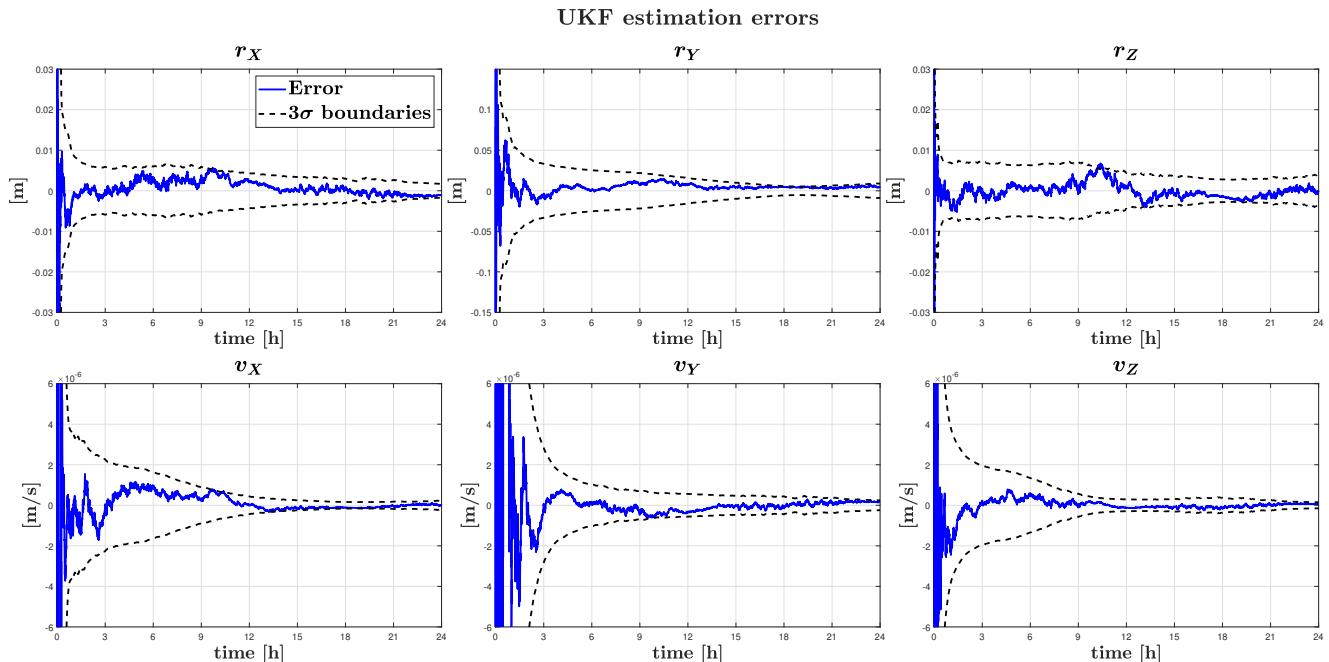
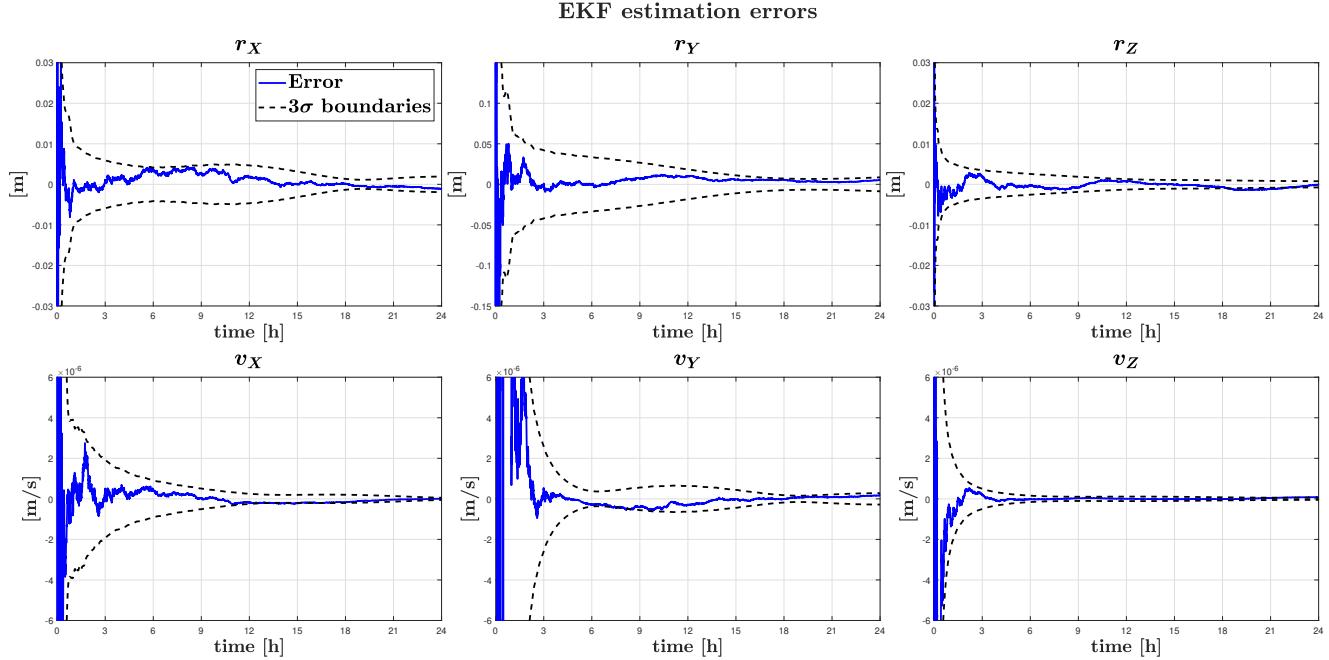
- Extra:

Analytic solution of Clohessy-Wiltshire linear equations: STM (for EKF)

$$\Phi(\Delta t = 1s) : \begin{bmatrix} 4 - 3\cos(nt) & 0 & 0 & \sin(nt)/n & 2(1 - \cos(nt))/n & 0 \\ 6(\sin(nt) - nt) & 1 & 0 & 2(\cos(nt) - 1)/n & (4\sin(nt) - 3nt)/n & 0 \\ 0 & 0 & \cos(nt) & 0 & 0 & \sin(nt)/n \\ 3n\sin(nt) & 0 & 0 & \cos(nt) & 2\sin(nt) & 0 \\ 6n(\cos(nt) - 1) & 0 & 0 & -2\sin(nt) & 4\cos(nt) - 3 & 0 \\ 0 & 0 & -n\sin(nt) & 0 & 0 & \cos(nt) \end{bmatrix} \quad (9)$$

where "n" is the mean motion of target SGN-1 along its GEO orbit [rad/s]

4. Error between nominal propagation (of chaser state, LVLH Fixed, Target centered) and filters `mean` estimation, with  $3\sigma$  boundaries, retrieved from estimated covariance matrices at each sample:



After a few runs of the provided Kalman filters it is understandable that the EKF keeps consistently under-estimating the error, at times causing the  $3\sigma$  boundaries to be slightly crossed quite late in the time window, after an initial quick convergence, whereas this inaccurate behaviour is efficiently avoided by UKF (default tuning of UT weights). In general the simple linearization around a single point (first order lin.), makes the EKF not as robust as the UKF (second order lin. around 13 "sigma points"), which instead consistently provides slightly larger confidence intervals, more accurately predicting the uncertainties: therefore the higher "Kalman Gain" achieved leads to higher corrections, because measurements are more certain. In case the sequential filter trusts the measures



too much, applying larger corrections may lead to a fast decrease of covariance, not coupled with the necessary decrease of the error. Such an occurrence could eventually cause the filter to diverge as the high certainty won't match the low accuracy of the measurements.

## Appendix

### A 1.1: LinCov vs Unscented Transform for ex 1.1

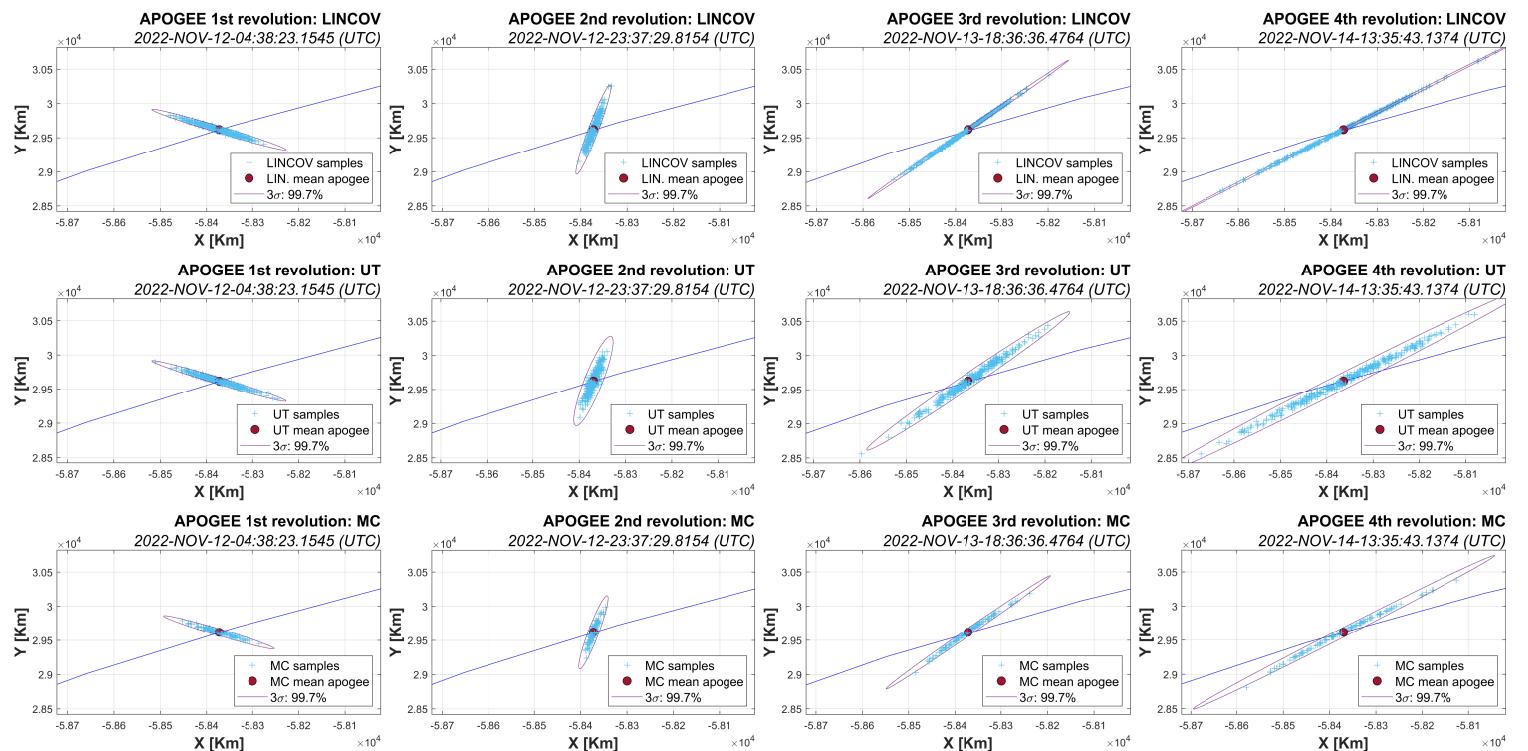
Assuming that the initial and propagated uncertainties are correctly represented by a Gaussian (multivariate) normal distribution, the Linear (LinCov) mean and covariance result to be wrongly estimated, i.e. first order approximation is incapable of representing highly non-linear dynamics, and the error keeps expanding for longer propagation times; on the other hand, UT can provide a better estimate, much closer to the ideal Monte Carlo Gaussian distribution propagation.

### A 1.2 Distinct plots and validation for ex 1.2

It is clearly understandable from the samples positions (especially at perigees) for each method, that Linear assumption is the least accurate model to represent strong non-linearities after longer propagations, causing the uncertainty to be portrayed the furthest from the Gaussian distribution of MC samples. On the other hand, UT model can better estimate the final mean and covariance, with low computational time, close to LinCov model. This pattern is better noticeable at perigees and at 4th revolution, whereas at 1st rev. and at apogees, LinCov samples show similar patterns, but with over-confident covariance (narrow ellipses).

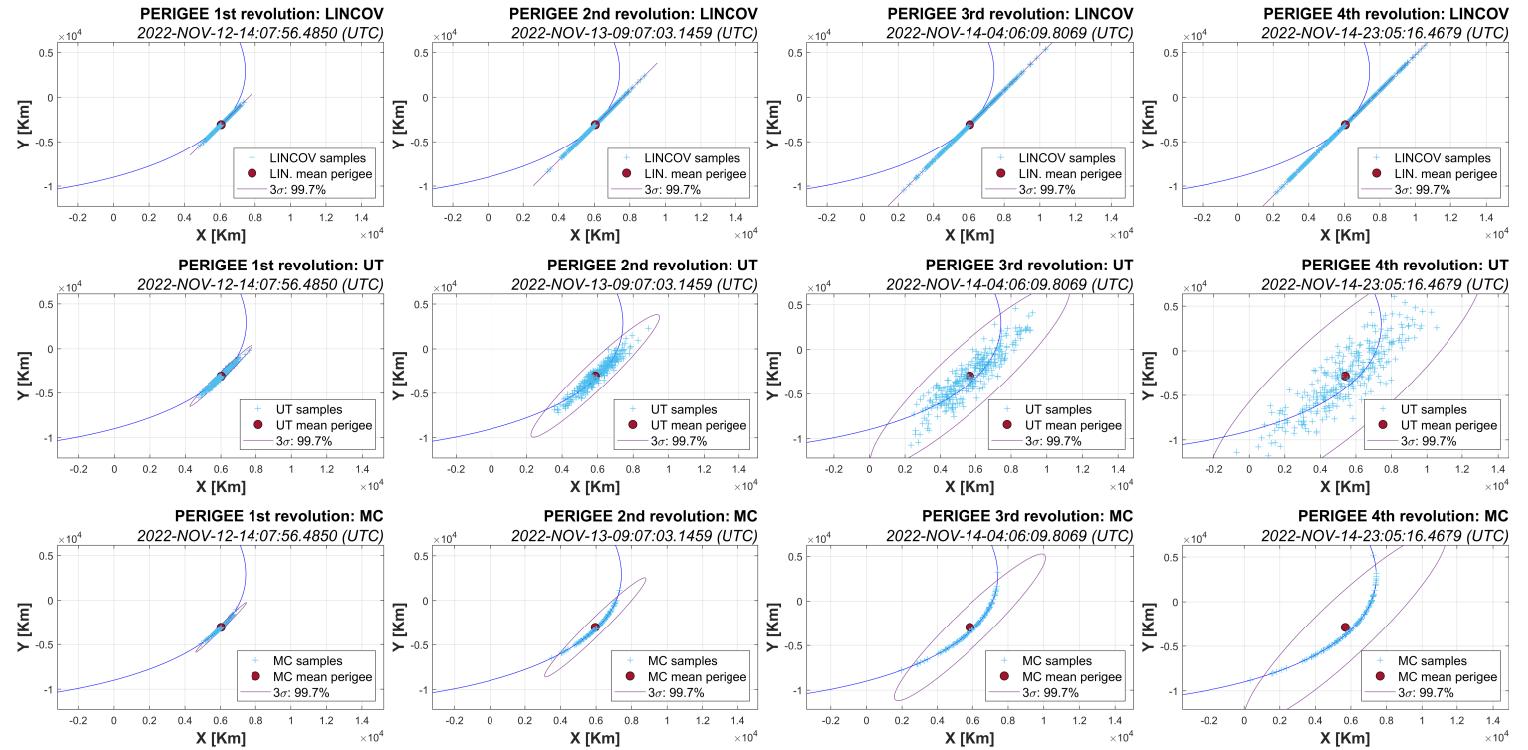
Confidence interval for representing uncertainty at each epoch is  $3\sigma$ : 99.7%.

All plots are ECI J2000 on equatorial plane:



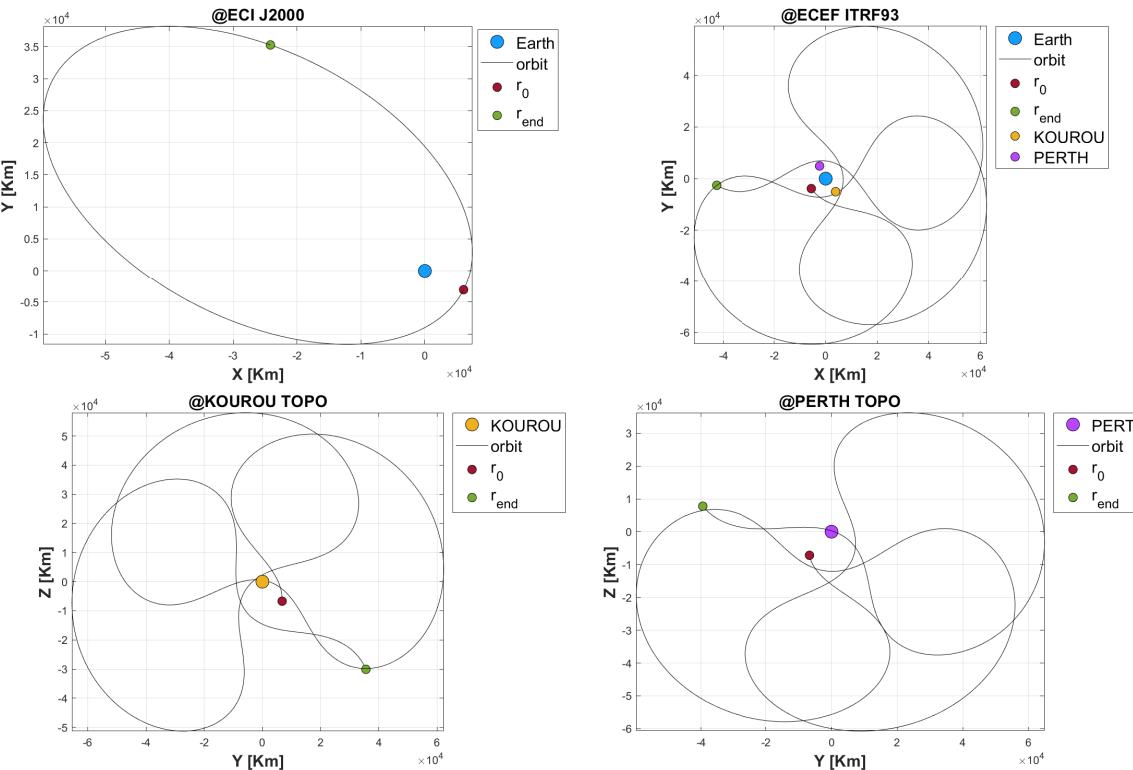
Covariance ellipses/-oids adapted from:

AJ Johnson (2004). `error_ellipse` ([https://www.mathworks.com/matlabcentral/fileexchange/4705-error\\_ellipse](https://www.mathworks.com/matlabcentral/fileexchange/4705-error_ellipse)), MATLAB Central File Exchange. All rights reserved



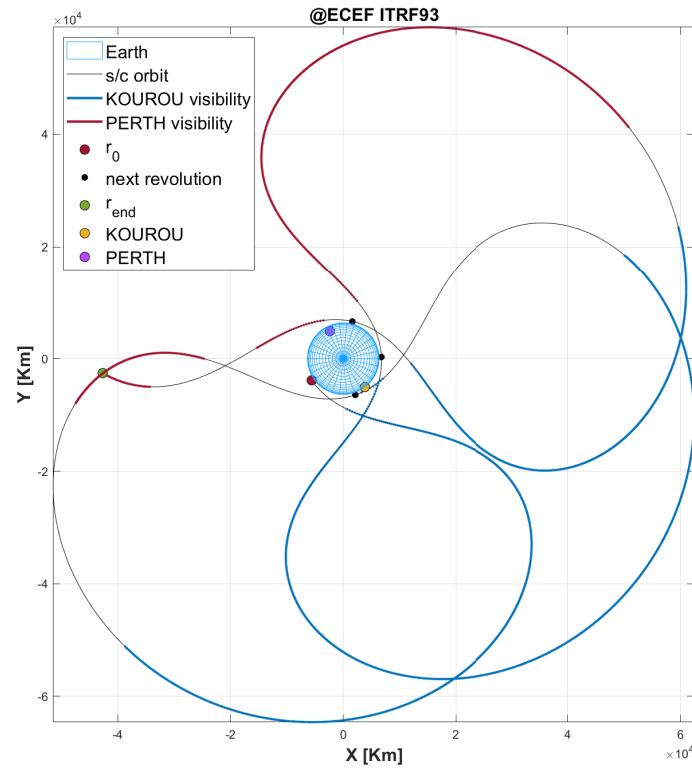
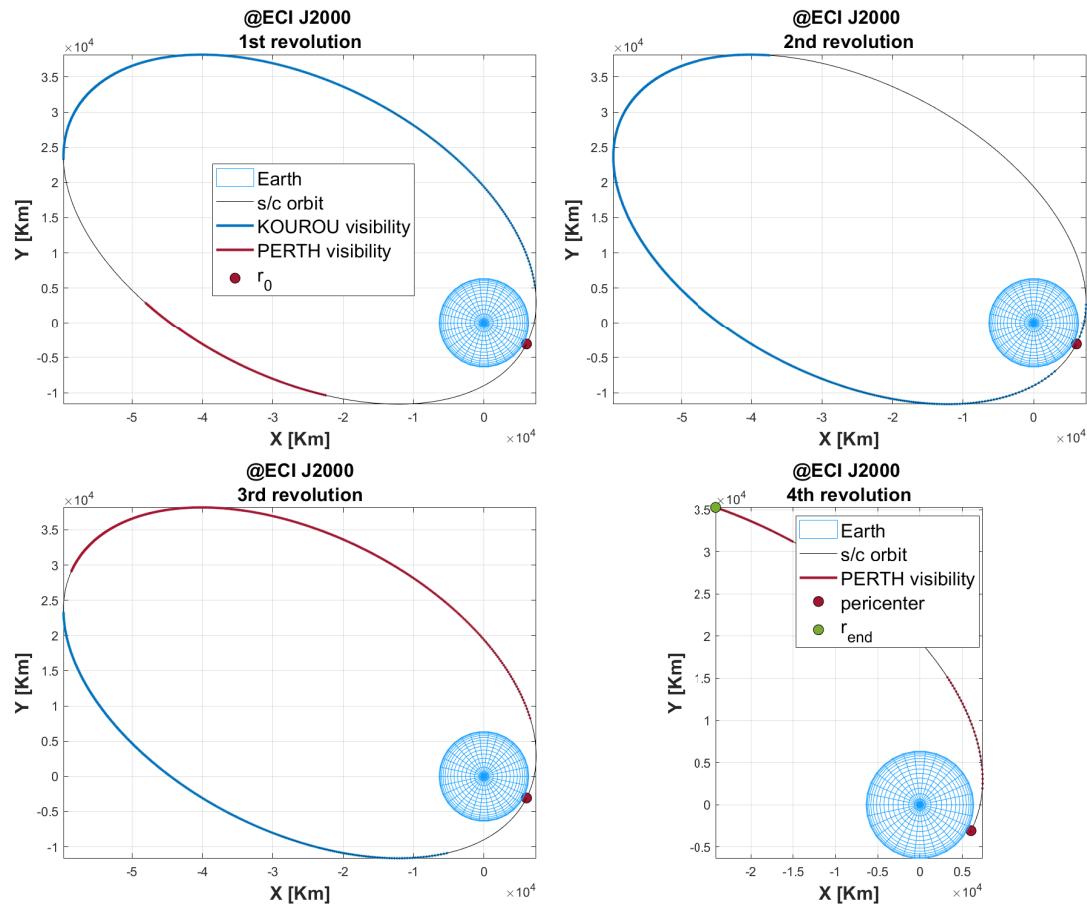
### A 2.1 Extra plots for ex 2.1

S/C orbit propagation in different reference frames:





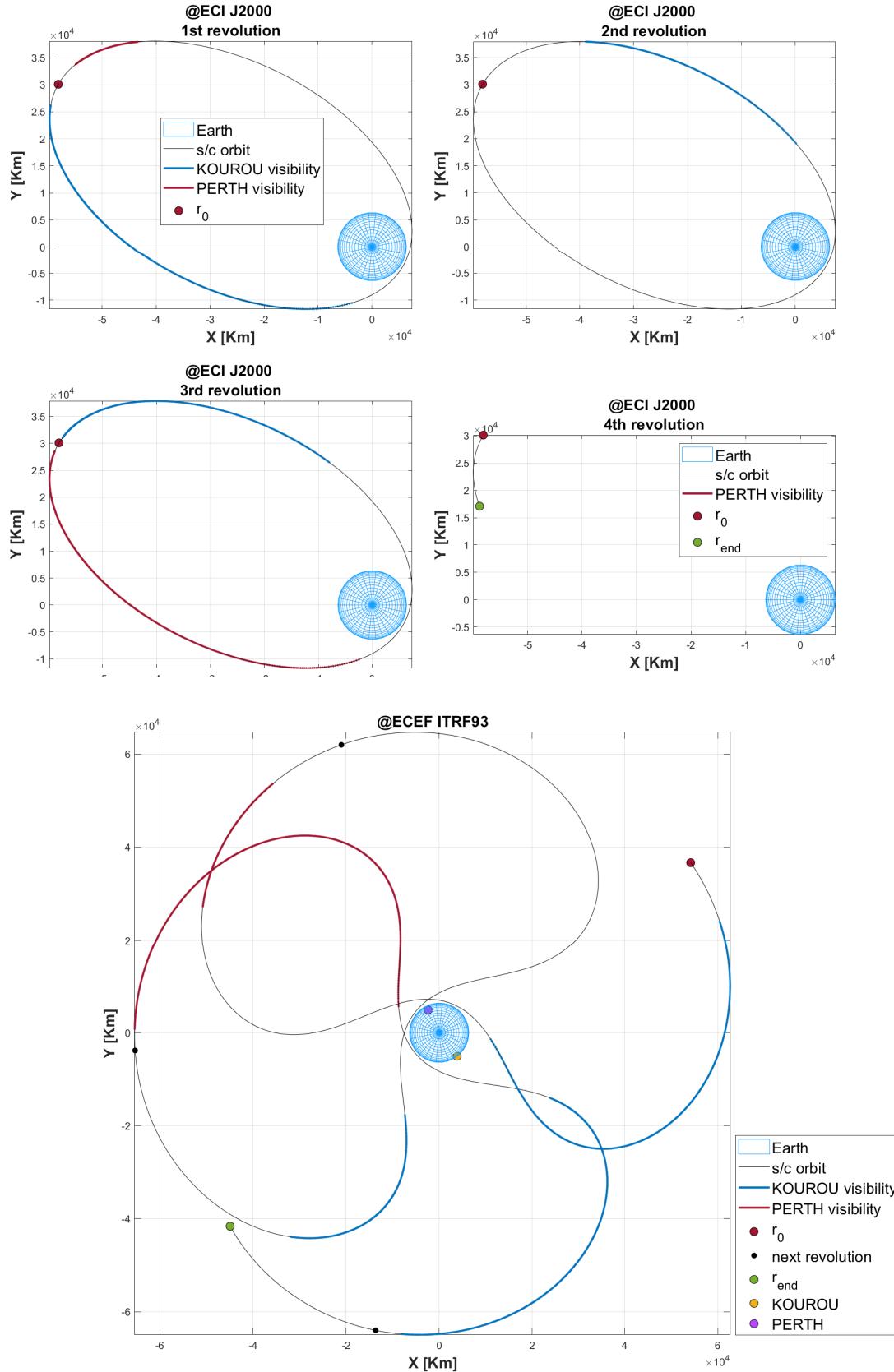
Expected visibility from ground stations:





## A 2.2 Extra plots for ex 2.2

Real visibility:





### A 2.3 Extra plot for ex 2.3

Example zoom-in on fitting plot for each measurement:

