SGN – Assignment #2

Davide Zamblera, 944071

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 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]	$\begin{bmatrix} +5.6e-3 & +3.5e-3 & -7.1e-4 & 0 & 0 & 0 \\ +3.5e-3 & +9.7e-3 & +7.6e-4 & 0 & 0 & 0 \\ -7.1e-4 & +7.6e-4 & +8.1e-4 & 0 & 0 & 0 \\ 0 & 0 & 0 & +2.8e-7 & 0 & 0 \\ 0 & 0 & 0 & 0 & +2.7e-7 & 0 \\ 0 & 0 & 0 & 0 & 0 & +9.6e-8 \end{bmatrix}$

As a starting point the dynamics of the problem is implemented, both in its complete, expressed in Eq. (1) and linearized form, in Eq. (2).

$$\begin{cases} \dfrac{d\mathbf{r}}{dt} = \mathbf{v} \\ \dfrac{d\mathbf{v}}{dt} = -\dfrac{\mu}{r^3}\mathbf{r} \end{cases} \tag{1}$$

$$\begin{cases} \dot{\mathbf{\Phi}} = \mathbf{A}\mathbf{\Phi} \\ \mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ \dfrac{3\mu}{r^5}\mathbf{r}\cdot\mathbf{r}^T - \dfrac{\mu}{r^3}\mathbf{I} & \mathbf{0} \end{bmatrix} \end{cases} \tag{2}$$

The propagation of the initial condition to all pericenters and apocenters is obtained by integration with ode113 and the use of an event function which records all epochs at which $\mathbf{r}\cdot\mathbf{v} = 0$, Fig. 1 show all the events that were found. The time of integration is chosen as slightly more than 4 periods T, where T was calculated using $2\pi\sqrt{a^3/\mu}$ and a was found with the energy formula from the initial state.

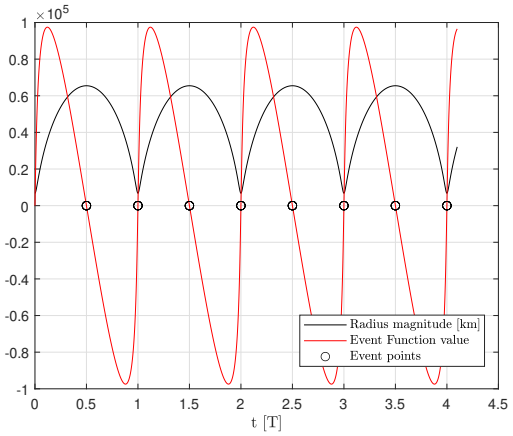


Figure 1: Determination of Apocenter and Pericenter epochs

To propagate the mean and covariance using a linearized approach the equations in Eq. (3) are used. While for the unscented transform the generalized σ points are computed and then propagated to the relevant epochs, then mean and covariance are obtained as a weighted sample mean and sample covariance.

$$\boldsymbol{\mu} = \mathbf{\Phi}\boldsymbol{\mu_0} \quad \mathbf{P} = \mathbf{\Phi}\mathbf{P_0}\mathbf{\Phi}^T \tag{3}$$

The results and differences between the methods are visible in Fig. 2. For the apocenter they give similar results, for the pericenters both methods have a much higher uncertainty in the direction of the movement, but for the linearized approach this is more accentuated. As velocities are much greater at pericenter this make sense, higher tangential velocities make our knowledge about the exact position of the spacecraft along that direction more uncertain.

A Monte Carlo analysis is then produced by generating a population of points from the initial mean and covariance and by propagating them to the various epochs and then estimating the relevant statistical quantities.

The final results in the equatorial plane are represented in Fig. 2, while all the numerical results are contained from Table 2 to Table 5 for the mean values and from Table 6 to Table 7 for the square root traces of $\mathbf{P}$. The plot of the covariances are made using ellipses at 95% or 2σ accuracy.

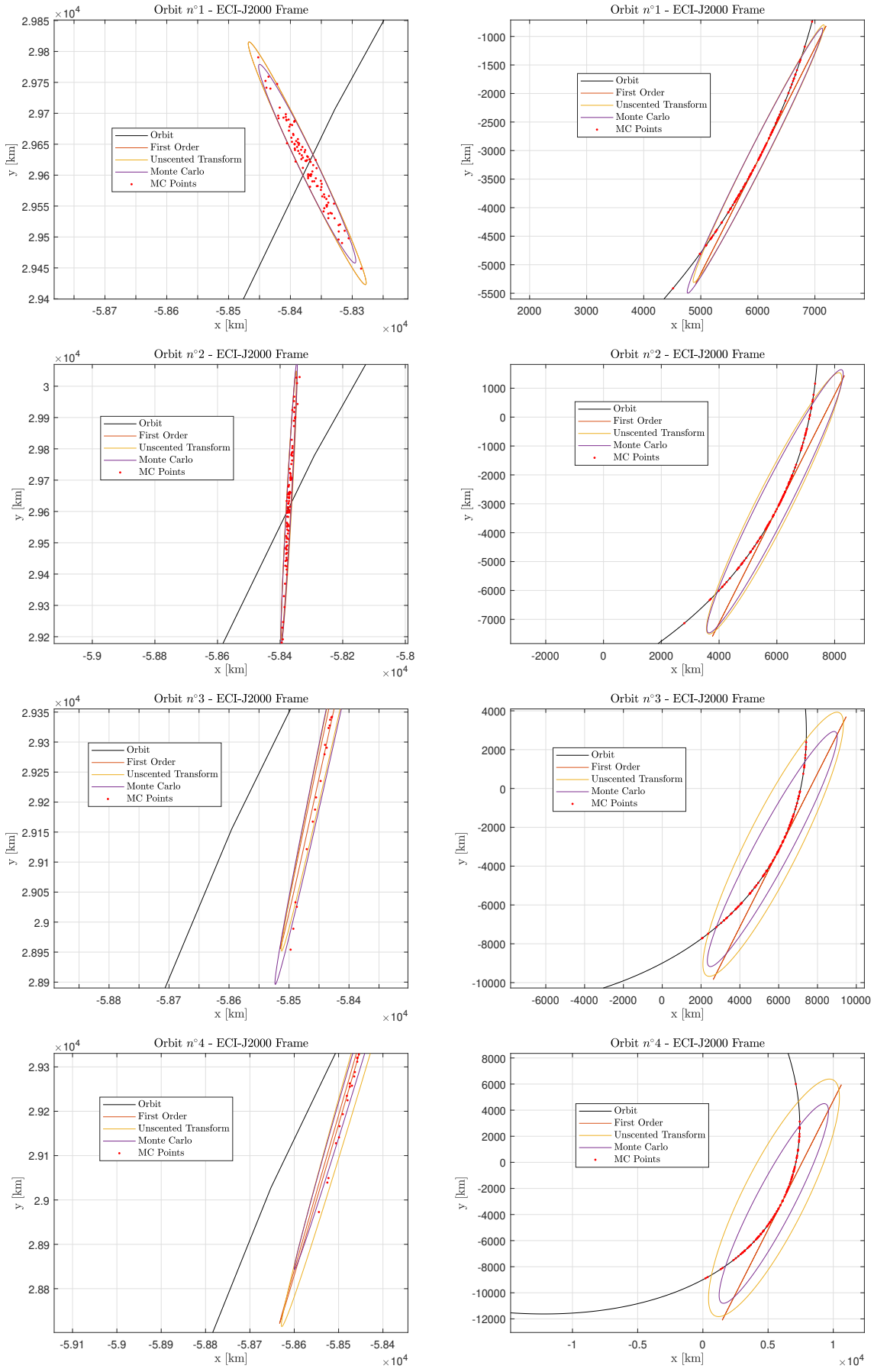


Figure 2: Covariances at 95% confidence, left apocenters, right pericenters

Table 2: Estimate of the mean state at pericenter and apocenter, 1st orbit in ECI J2000.

Orbit	Method	x [km]	y [km]	z [km]	v_x [km/s]	v_y [km/s]	v_z [km/s]
-	LinCov	-58372.8281	29619.1721	1283.4365	-0.4820	-0.9528	0.0644
Apo	UT	-58372.7836	29619.1709	1283.4347	-0.4820	-0.9528	0.0644
-	MC	-58372.8451	29618.4881	1283.5099	-0.4820	-0.9528	0.0644
-	LinCov	6054.3080	-3072.0388	-133.1154	4.6475	9.1861	-0.6206
Per	UT	6014.8616	-3053.6545	-132.1768	4.6188	9.1267	-0.6166
-	MC	5949.1846	-3176.7808	-123.7155	4.7209	9.0710	-0.6187

Table 3: Estimate of the mean state at pericenter and apocenter, 2nd orbit in ECI J2000.

Orbit	Method	x [km]	y [km]	z [km]	v_x [km/s]	v_y [km/s]	v_z [km/s]
-	LinCov	-58372.8281	29619.1721	1283.4365	-0.4820	-0.9528	0.0644
Apo	UT	-58371.5231	29617.6244	1283.4479	-0.4820	-0.9528	0.0644
-	MC	-58371.4913	29624.2130	1283.1909	-0.4824	-0.9525	0.0644
-	LinCov	6054.3080	-3072.0388	-133.1154	4.6475	9.1861	-0.6206
Per	UT	5903.6967	-2984.3328	-130.3147	4.5208	8.9545	-0.6045
-	MC	5936.5969	-2913.2112	-134.9739	4.4669	8.9917	-0.6038

Table 4: Estimate of the mean state at pericenter and apocenter, 3rd orbit in ECI J2000.

Orbit	Method	x [km]	y [km]	z [km]	v_x [km/s]	v_y [km/s]	v_z [km/s]
-	LinCov	-58372.8281	29619.1721	1283.4365	-0.4820	-0.9528	0.0644
Apo	UT	-58370.4015	29615.5185	1283.4925	-0.4818	-0.9528	0.0644
-	MC	-58364.8051	29639.8855	1282.1176	-0.4836	-0.9518	0.0644
-	LinCov	6054.3080	-3072.0388	-133.1154	4.6475	9.1861	-0.6206
Per	UT	5715.3652	-2874.7951	-126.8034	4.3625	8.6650	-0.5844
-	MC	5669.5414	-3119.2628	-113.7806	4.5859	8.7143	-0.5966

Table 5: Estimate of the mean state at pericenter and apocenter, 4th orbit in ECI J2000.

Orbit	Method	x [km]	y [km]	z [km]	v_x [km/s]	v_y [km/s]	v_z [km/s]
-	LinCov	-58372.8281	29619.1721	1283.4365	-0.4820	-0.9528	0.0644
Apo	UT	-58368.3355	29612.3886	1283.5407	-0.4817	-0.9529	0.0644
-	MC	-58368.7499	29618.4311	1283.2944	-0.4821	-0.9527	0.0644
-	LinCov	6054.3079	-3072.0388	-133.1154	4.6475	9.1861	-0.6206
Per	UT	5452.3783	-2720.0679	-121.9789	4.1398	8.2602	-0.5563
-	MC	5439.5595	-3152.5730	-101.9666	4.5740	8.4566	-0.5845

The results show how uncertainty propagation without new real data from the system lead to a rapid growth in the uncertainties.

From these results, by taking the Monte Carlo analysis as ground truth it is clear how the linearized approach is not a good approximation for pericenter values even from the 1st orbit, while for the apocenter has results compatible with the other propagations. The unscented transform, being a second order approximation, does a better job at estimating the state, but tends to slightly overestimate the covariance.

Table 6: $\sqrt{tr(P_{rr})}$ for all events in ECI J2000.

Method	1st Apo.	1st Per.	2nd Apo.	2nd Per.	3rd Apo.	3rd Per.	4th Apo.	4th Per.
LinCov	89.149	1034.745	176.089	2069.490	277.674	3104.235	382.406	4138.980
UT	89.149	1036.574	176.112	2084.119	277.743	3153.397	382.578	4254.853
MC	73.079	1066.866	186.756	2098.834	311.124	2831.186	331.033	3571.450

Table 7: $\sqrt{tr(P_{vv})}$ for all events in ECI J2000.

Method	1st Apo.	1st Per.	2nd Apo.	2nd Per.	3rd Apo.	3rd Per.	4th Apo.	4th Per.
LinCov	0.0048	0.8673	0.0140	1.7346	0.0234	2.602	0.0327	3.4692
UT	0.0048	0.8723	0.0140	1.774	0.023	2.7346	0.0327	3.7781
MC	0.0039	0.8842	0.01489	1.6931	0.0262	2.2224	0.0282	2.7078

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 8.

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 9 (and in WeBeep as 87654.tle). Use SGP4 and the provided TLE to simulate the measurements acquired by the sensor network in Table 8 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 8).

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 9 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 8: 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 9: 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
```

The visibility windows are predicted using a dynamical system with keplerian motion, the elevation angle is used to determine the beginning and the end of a window, considering that for the various stations the minimum elevation needed to observe the spacecraft is different. This requires the transformation of the data from inertial earth centered to a non-inertial topocentric frame. The results are contained in Table 10 and displayed as a polar plot in Fig. 3.

Table 10: Table of visibility windows for Kourou and Perth [MJD2000]

Pass n.	Station	Start Time	End Time
1	KOUROU	8350.7327	8351.0737
2	KOUROU	8351.9175	8352.0807
3	KOUROU	8352.6814	8353.1494
1	PERTH	8351.0959	8351.7321
2	PERTH	8352.2355	8352.6577

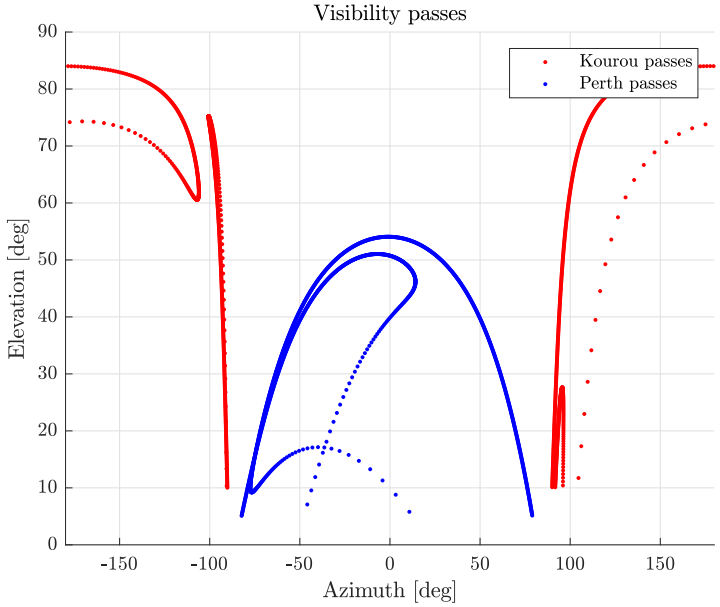


Figure 3: Visibility passes for the various stations

The simulation of all the relevant measurements Az, h, ρ are carried out by propagating the initial conditons defined in the available TLE using the sgp4 dynamical model, which includes in addition to the gravitational attraction force with the main body, other lower order perturbation as periodic zonal harmonics and luni-solar perturbations. An additional Gaussian noise is applied to the values to simulate the measurement process, with standard deviations provided by the instrument characteristics.

After simulation of the relevant measurements in the predicted visibility windows, a series of least square problems were solved, to estimate the initial state of the spacecraft.

The approach used was the maximum likelihood approach, which employs a standard weighted least squares problem reported in Eq. (4), with the weight matrix equivalent to the inverse of the measurement noise. This means that lower accuracy measurements will be considered less in the estimation search.

$$\min \mathrm{J} = \frac{1}{2}(\tilde{\mathbf{y}} - \mathbf{y})\mathbf{R}^{-1}(\tilde{\mathbf{y}} - \mathbf{y}) \tag{4}$$

The results are reported in Table 11 in terms of estimated initial state, while the uncertainty

on the estimate of the initial state is reported as covariance matrices from Fig. 5 to Fig. 7. In Table 12 are also reported the square roots of the traces of the covariance matrix, as expected they have diminishing values with each new trial. In particular from the first trial which uses only the measurements from Perth, the second trial improves by using all the available measurements, the total residual will grow since there are more points, but the estimate will be more precise. The third trial instead adds an important piece of dynamics, the J2 perturbation, this lowers the residual and improves the covariance of the estimate. A even better model would include other secondary dynamics that are present in the real measurement but this current model cannot represent.

Table 11: Estimated states at reference epoch for the various trials

Case	x	y	z	v_x	v_y	v_z
PERTH	-58270.881	29812.707	1224.170	-0.519	-0.933	0.064
ALL	-58253.397	29827.197	1284.407	-0.519	-0.934	0.071
ALL-J2	-58120.116	30099.208	1248.222	-0.524	-0.931	0.066

The residual errors are also reported and displayed in Fig. 4, which shows the angular error vectors $[e_{Az}, e_h]$ in magnitude for the various observation times, as it can be seen in the first and second subplot variations upon the usual random noise are visible, especially in the second plot, the third plot generated using also the J2 effect instead suffers less from the modeling errors.

Table 12: Square root of covariance trace

-	PERTH	ALL	ALL and J2
$\sqrt{trace(P_{rr})}$	2.138599	1.971163	0.583216
$\sqrt{trace(P_{vv})}$	0.000173	0.000134	0.000029

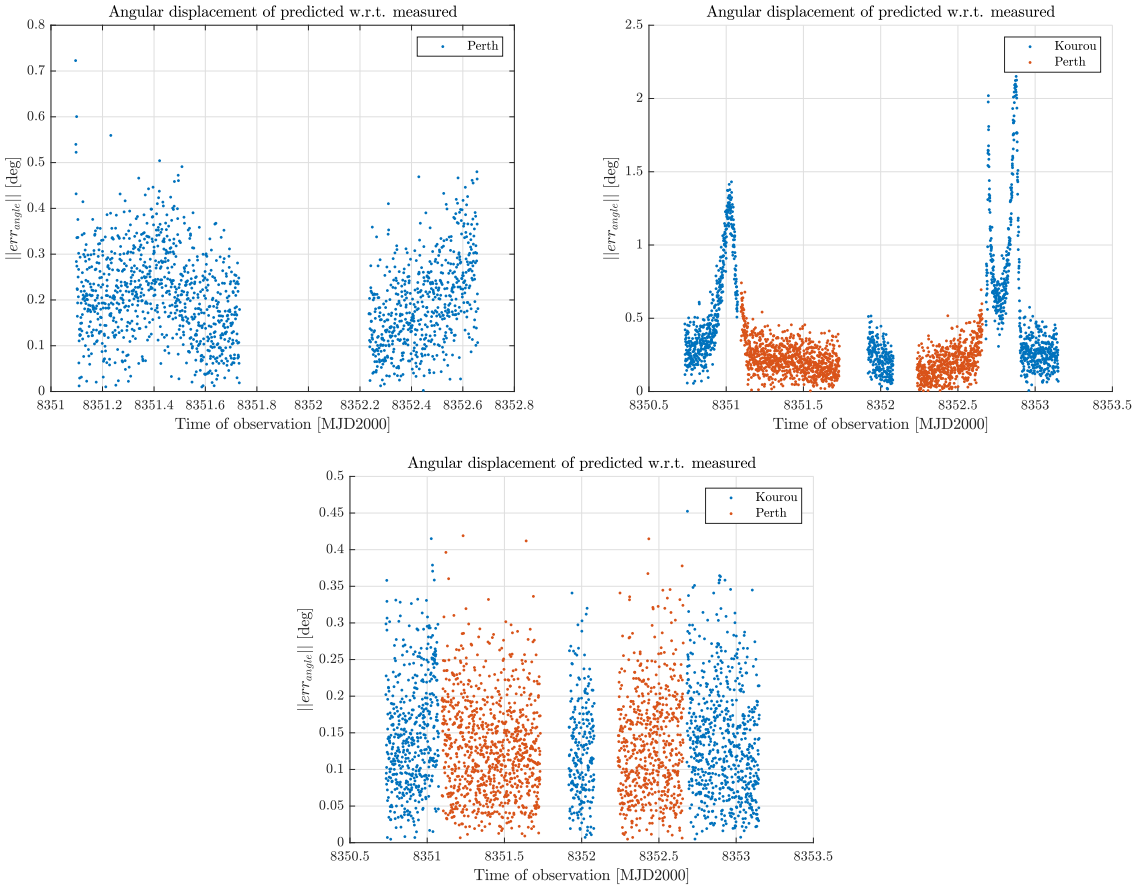


Figure 4: Residual for the various measurement epochs

$$\begin{bmatrix} +8.895E-01 & +1.683E+00 & +4.761E-01 & -2.613E-05 & +2.814E-05 & +4.057E-05 \\ +1.683E+00 & +3.211E+00 & +8.618E-01 & -4.916E-05 & +5.309E-05 & +7.467E-05 \\ +4.761E-01 & +8.618E-01 & +4.731E-01 & -1.532E-05 & +2.159E-05 & +3.776E-05 \\ -2.613E-05 & -4.916E-05 & -1.532E-05 & +9.456E-10 & -8.012E-10 & +7.081E-10 \\ +2.814E-05 & +5.309E-05 & +2.159E-05 & -8.012E-10 & +1.187E-09 & +2.717E-09 \\ +4.057E-05 & +7.467E-05 & +3.776E-05 & +7.081E-10 & +2.717E-09 & +2.793E-08 \end{bmatrix}$$

Figure 5: Estimated covariance matrix, case PERTH

$$\begin{bmatrix} +6.677E-01 & +1.277E+00 & +3.372E-01 & -2.399E-05 & +2.542E-06 & -5.092E-05 \\ +1.277E+00 & +2.499E+00 & +5.882E-01 & -4.554E-05 & +8.200E-06 & -9.645E-05 \\ +3.372E-01 & +5.882E-01 & +7.192E-01 & -1.547E-05 & -1.221E-05 & -8.076E-05 \\ -2.399E-05 & -4.554E-05 & -1.547E-05 & +9.432E-10 & -1.861E-11 & +2.664E-09 \\ +2.542E-06 & +8.200E-06 & -1.221E-05 & -1.861E-11 & +4.495E-10 & +1.193E-09 \\ -5.092E-05 & -9.645E-05 & -8.076E-05 & +2.664E-09 & +1.193E-09 & +1.664E-08 \end{bmatrix}$$

Figure 6: Estimated covariance matrix, case PERTH and KOUROU

$$\begin{bmatrix} +5.703E-02 & +1.097E-01 & -3.687E-02 & -2.196E-06 & +1.388E-06 & -2.189E-06 \\ +1.097E-01 & +2.145E-01 & -6.854E-02 & -4.199E-06 & +2.781E-06 & -4.388E-06 \\ -3.687E-02 & -6.854E-02 & +6.860E-02 & +1.696E-06 & -1.376E-06 & +7.366E-07 \\ -2.196E-06 & -4.199E-06 & +1.696E-06 & +9.108E-11 & -5.654E-11 & +1.218E-10 \\ +1.388E-06 & +2.781E-06 & -1.376E-06 & -5.654E-11 & +4.678E-11 & -3.579E-11 \\ -2.189E-06 & -4.388E-06 & +7.366E-07 & +1.218E-10 & -3.579E-11 & +6.812E-10 \end{bmatrix}$$

Figure 7: Estimated covariance matrix, case PERTH and KOUROU, J2 model

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 13.

Table 13: Parameters of SGN-I.

Parameter	Value
Size [m]	$[l, h, d] = [10, 5, 3]$
Density [kg/m^3]	ρ =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:

$$\mathbf{J}\dot{\boldsymbol{\omega}} = -\boldsymbol{\omega} \wedge \mathbf{J}\boldsymbol{\omega} \tag{5}$$

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:

$$\dot{\mathbf{q}} = \frac{1}{2}\begin{bmatrix} 0 & -\boldsymbol{\omega}(1) & -\boldsymbol{\omega}(2) & -\boldsymbol{\omega}(3) \\ \boldsymbol{\omega}(1) & 0 & \boldsymbol{\omega}(3) & -\boldsymbol{\omega}(2) \\ \boldsymbol{\omega}(2) & -\boldsymbol{\omega}(3) & 0 & \boldsymbol{\omega}(1) \\ \boldsymbol{\omega}(3) & \boldsymbol{\omega}(2) & -\boldsymbol{\omega}(1) & 0 \end{bmatrix}\mathbf{q} \tag{6}$$

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*):

$$\mathbf{C}_{T,I} = \begin{bmatrix} \mathbf{q}(1)^2 + \mathbf{q}(2)^2 - \mathbf{q}(3)^2 - \mathbf{q}(4)^2 & 2(\mathbf{q}(2)\mathbf{q}(3) + \mathbf{q}(1)\mathbf{q}(4)) & 2(\mathbf{q}(2)\mathbf{q}(4) - \mathbf{q}(1)\mathbf{q}(3)) \\ 2(\mathbf{q}(2)\mathbf{q}(3) - \mathbf{q}(1)\mathbf{q}(4)) & \mathbf{q}(1)^2 - \mathbf{q}(2)^2 + \mathbf{q}(3)^2 - \mathbf{q}(4)^2 & 2(\mathbf{q}(3)\mathbf{q}(4) + \mathbf{q}(1)\mathbf{q}(2)) \\ 2(\mathbf{q}(2)\mathbf{q}(4) + \mathbf{q}(1)\mathbf{q}(3)) & 2(\mathbf{q}(3)\mathbf{q}(4) - \mathbf{q}(1)\mathbf{q}(2)) & \mathbf{q}(1)^2 - \mathbf{q}(2)^2 - \mathbf{q}(3)^2 + \mathbf{q}(4)^2 \end{bmatrix} \tag{7}$$

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 14.

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 14: 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 15).

Table 15: 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]	$\text{diag}([10, 10, 10, 0.1, 0.1, 0.1])$

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

$$\begin{aligned} \ddot{x} &= 3n^2 x + 2n\dot{y} \\ \ddot{y} &= -2n\dot{x} \\ \ddot{z} &= -n^2 z \end{aligned} \tag{8}$$

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:

$$\mathbf{C}_{L,I} = \begin{bmatrix} \cos(nt) & \sin(nt) & 0 \\ -\sin(nt) & \cos(nt) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{9}$$

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 16. 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 $\boldsymbol{\Phi}$

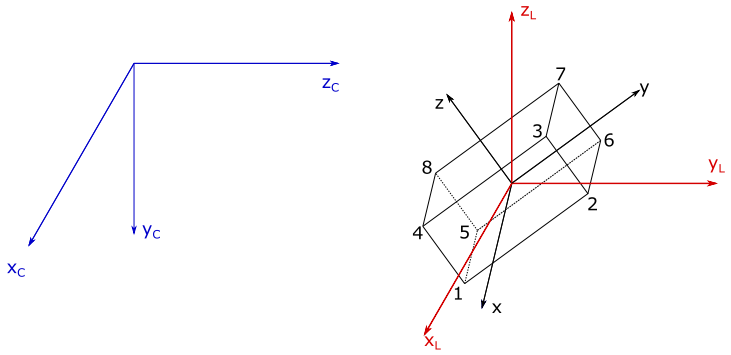


Figure 8: Pay attention to the three different reference frames, in blue the camera frame (C pedices), in red the LVLH frame (L pedices), and in black the target body frame.

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

$$\mathbf{C}_{\text{cam},L} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \tag{10}$$

The overall geometry of the observation is described in Figure 8.

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

$$\mathbf{y} = \left[u_0 - Dfoc\frac{Y}{Z}, v_0 + Dfoc\frac{X}{Z}, \frac{bDfoc}{Z} \right] \tag{11}$$

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 13. 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 16: Parameters of the stereo camera.

Parameter	Value
Focal length [mm]	$foc = 3$
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.

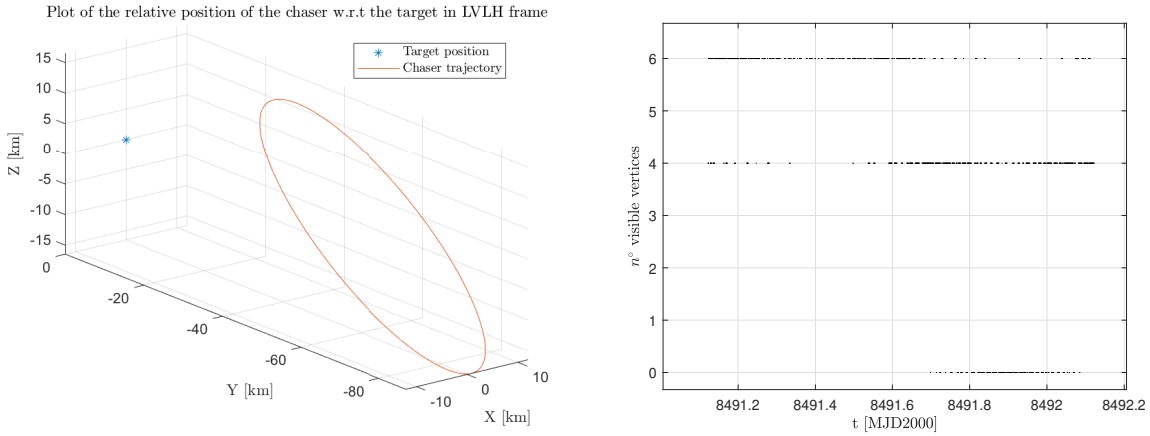
2. Verify that the visible features of the target SGN-I lie inside the field-of-view of the camera during the entire navigation window.

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.

As no real data are provided there is the need to generate simulation of real measurements, independently from the estimation process, to do this the external function *meas_sim* was used, the measurements were further corrupted by a noise applied directly on the pixel values. The nominal trajectory which acts as ground truth is depicted in Fig. 9a, while the number of visible vertex is visible in Fig. 9b, without any maneuver it can be seen that in the last part of the observation the satellite exits the f.o.v for a non insignificant number of epochs.



(a) Relative trajectory **(b)** Vertex visibility

Figure 9

To validate the measurement function and the visibility condition, the plot of all the measured pixel is shown in Fig. 10, the vertical axis in the real camera points downward. Clearly the simulator performs well, considering the sensor size of 1920x1200, the only pixels outside the camera screen are due to the added measurement noise.

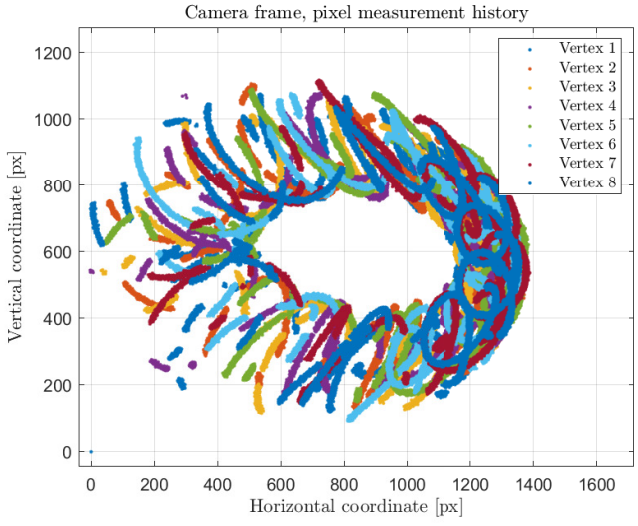


Figure 10

The implementation of the Kalman filter requires the dynamical model for prediction, the analytical solution is found in Eq. (12). The existence of an analytical solution will speed up the estimation effort.

$$\begin{cases} \mathbf{\Phi} = e^{\mathbf{A}(t-t_0)} \\ \mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{A_{21}} & \mathbf{A_{22}} \end{bmatrix} \\ \mathbf{A_{21}} = \begin{bmatrix} 3n^2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -n^2 \end{bmatrix} \quad \mathbf{A_{22}} = \begin{bmatrix} 0 & 2n & 0 \\ -2n & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \end{cases} \tag{12}$$

Then to provide the simulated measurements for comparison with the true measurements a measure function $\mathbf{h}$ was developed, the main points for its construction are:

1. Take the available vertex $\mathbf{p}$ from target body frame T to the LVLH frame by a sequence of rotations ${}^L\mathbf{p} = \mathbf{C_{LI}}\mathbf{C_{TI}^{-1}}\,{}^T\mathbf{p}$. The matrix $\mathbf{C_{TI}}$ is the attitude of the satellite in inertial frame, it is obtained from the quaternion representation, while $\mathbf{C_{LI}}$ can be obtained by the mean motion of the satellite.

2. Take the target centered vertex vectors and translate them in chaser centered frame, then transport them in the camera frame: $\mathbf{X} = \mathbf{C_{CL}}({}^L\mathbf{p} - \mathbf{r})$ where $\mathbf{r}$ is the position subvector of the state of our dynamics.

3. Finally the pinhole camera model is applied to the vector $\mathbf{X}$ and the pixel coordinates and the disparity is found.

The Jacobian of the measurement function is obtained from a chain rule as reported in Eq. (13).

$$\begin{cases} \mathbf{H} = \dfrac{\partial \mathbf{y}}{\partial \mathbf{X}} \dfrac{\partial \mathbf{X}}{\partial \mathbf{x}} \\ \dfrac{\partial \mathbf{y}}{\partial \mathbf{X}} = \begin{bmatrix} 0 & -D\ foc\ \dfrac{1}{Z} & D\ foc\ \dfrac{Y}{Z^2} \\ D\ foc\ \dfrac{1}{Z} & 0 & -D\ foc\ \dfrac{X}{Z^2} \\ 0 & 0 & -D\ foc\ \dfrac{b}{Z^2} \end{bmatrix} \\ \dfrac{\partial \mathbf{X}}{\partial \mathbf{x}} = -\mathbf{C_{CL}} \end{cases} \tag{13}$$

Since there are multiple measurements, the true final vector of measurements will be a concatenation of the single visible vertex measurements with dimension $[3n_{vis}, 1]$, while the jacobian will be a concatenation too, of the jacobian evaluated for the visible points, with dimension $[3n_{vis}, 6]$.

The results of both the methods are visible in Fig. 11 and Fig. 12. The most important features to recognize in these figures are:

1. The 3σ boundaries correctly encapsulate the estimation errors for all the epochs, this is a useful information that contributes to the assessment of the validity of the results.

2. Both methods rapidly converge, eliminating the errors of the last available estimate, the power of the sequential algorithm is that these estimates could be used in real time, instead of waiting for the whole dynamics to take place, this is particularly important for applications such as rendez-vous and close approaches.

3. From these results it is difficult to see much of a difference in the rate of convergence and final errors between the two estimation methods, this is due to the linearity of the problem. For a problem with strong nonlinearities the UKF will tend to have smaller final error, at the expense of a slightly higher computational cost.

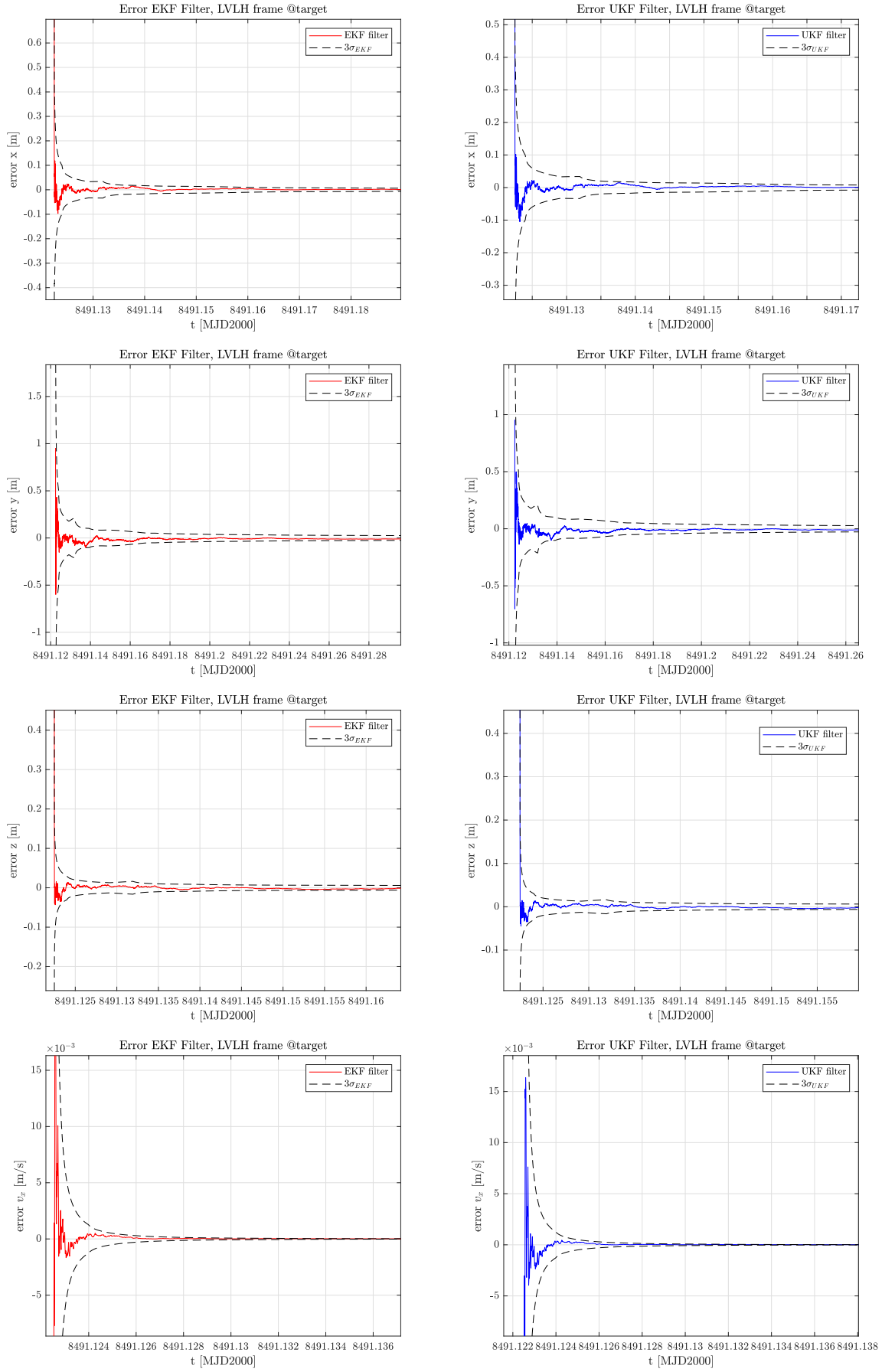


Figure 11: Comparison results of Extended (left) and Unscented (right) Kalman filters

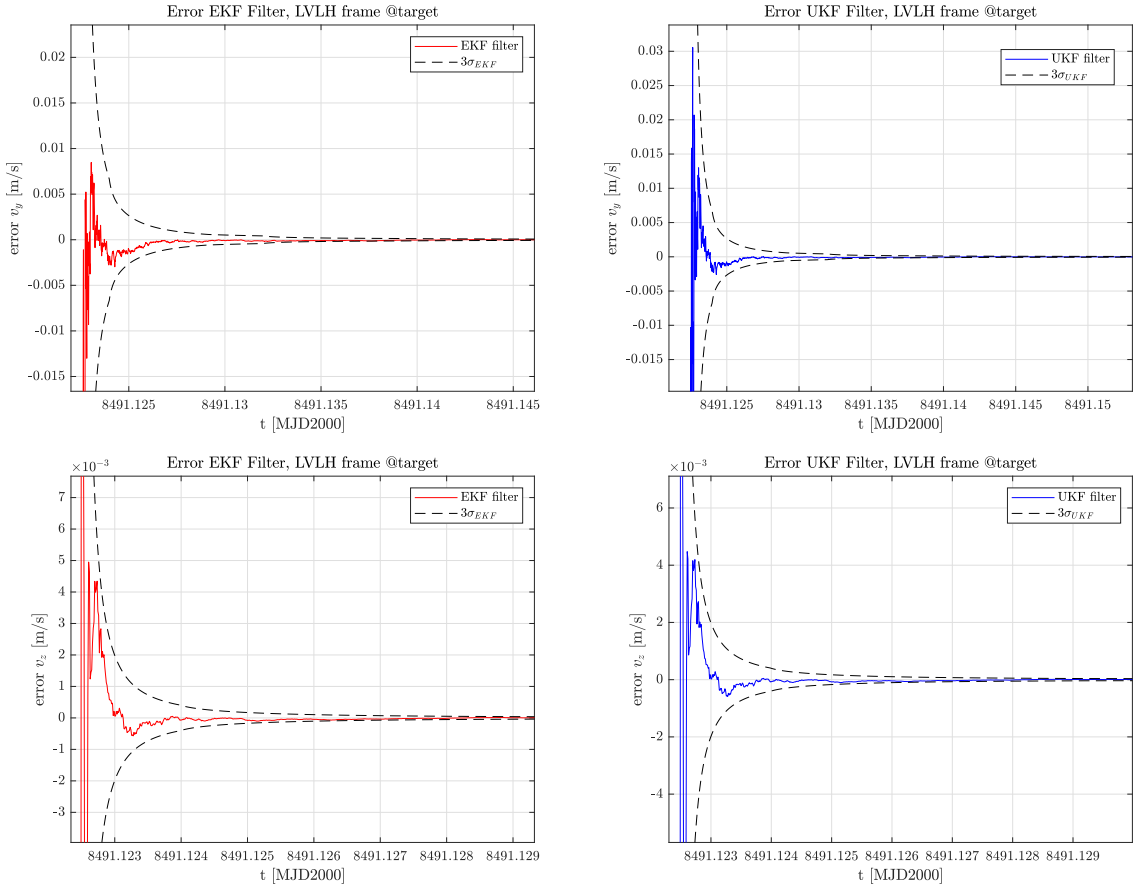


Figure 12: Comparison results of Extended (left) and Unscented (right) Kalman filters