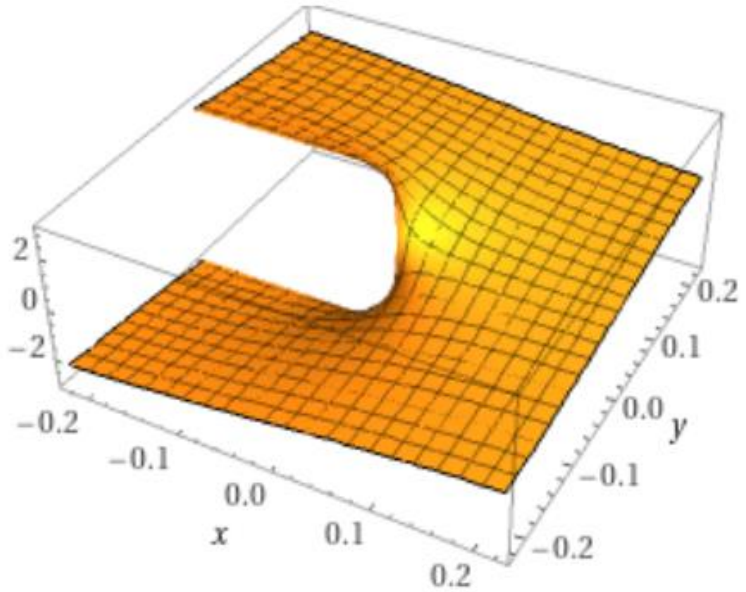
Solution Discussion, HA3



Victor Pettersson, 3rd of May 2024

Grading

- Task 1:
  a) 1 p
  b) 1 p
  c) 1 p
  d) 1 p
- Task 2:
  a) 1 p
  b) 2 p
  c) 2 p
- Task 3:
  abc) 3.5 p
  d) 1.5 p

Task 1

- Key questions:
  - What approximations are made in EKF and Sigma-point filters, respectively?
  - Where/at what stage do we apply these approximations?
  - How do we benchmark our approximations? Given that we seek a Gaussian approximation, what is the best such approximation in general?

1 Approximations of mean and covariance

In this task you will study how the sample mean and covariance are approximated in EKF, UKF, and CKF in a scenario with the dual bearing measurement model. However, you only need to implement EKF and one sigma points based KF.

The non-linear Kalman filters all use the same type of update for the state estimate, with a Kalman gain $\mathbf{K} = \mathbf{P}_{xy}\mathbf{P}_{yy}^{-1}$; here x denotes the state and y denotes the measurement. Depending on the type of non-linear Kalman filter, different approximations are used to compute the cross covariance $\mathbf{P}_{xy}$ and the covariance $\mathbf{P}_{yy} = \mathbf{S}$.

Consider a 2D state vector $\mathbf{x}$ that consists of x-position and y-position. We will consider three different state densities:

$$p_1(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}_1, \mathbf{P}_1) = \mathcal{N}\left(\mathbf{x}; \begin{bmatrix}125\\125\end{bmatrix}, \begin{bmatrix}10^2 & 0\\ 0 & 5^2\end{bmatrix}\right), \tag{1}$$

$$p_2(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}_2, \mathbf{P}_2) = \mathcal{N}\left(\mathbf{x}; \begin{bmatrix}-25\\125\end{bmatrix}, \begin{bmatrix}10^2 & 0\\ 0 & 5^2\end{bmatrix}\right), \tag{2}$$

$$p_3(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}_3, \mathbf{P}_3) = \mathcal{N}\left(\mathbf{x}; \begin{bmatrix}60\\60\end{bmatrix}, \begin{bmatrix}10^2 & 0\\ 0 & 5^2\end{bmatrix}\right), \tag{3}$$

and the dual bearing measurement model from the implementation part of HA3, with the bearing sensors located in $\mathbf{s}_1 = [0, 100]^T$ and $\mathbf{s}_2 = [100, 0]^T$, each with Gaussian measurement noise with standard deviation $\sigma_\varphi = 0.1\pi/180$ rad. In this task, we will focus on the approximation of the mean $E[\mathbf{y}]$ and the covariance $\text{Cov}(\mathbf{y}) = \mathbf{P}_{yy}$.

Task 1

- Key questions:
  - What approximations are made in EKF and Sigma-point filters, respectively?
  - How do we benchmark our approximations? Given that we seek a Gaussian approximation, what is the best such approximation in general?

First-order Taylor approximation

Approximating integral by weighted sum

Monte-Carlo approximation for N going to infinity

Task 1

$$p(\mathbf{y}) = \int p(\mathbf{x}, \mathbf{y}) d\mathbf{x} \approx \int \mathcal{N}\left(\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix}; \begin{bmatrix} \hat{\mathbf{x}} \\ \hat{\mathbf{y}} \end{bmatrix}, \begin{bmatrix} \mathbf{P}_{xx} & \mathbf{P}_{xy} \\ \mathbf{P}_{xy}^T & \mathbf{P}_{yy} \end{bmatrix} \right) d\mathbf{x} = \cdots = \mathcal{N}(\mathbf{y}; \hat{\mathbf{y}}, \mathbf{P}_{yy})$$

$$\mathbf{y} = \mathbf{h}(\mathbf{x}) + \mathbf{r}$$

EKF

$$p(\mathbf{x}_k, \mathbf{y}_k | \mathbf{y}_{1:k-1}) \approx \mathcal{N}\left(\begin{bmatrix} \mathbf{x}_k \\ \mathbf{y}_k \end{bmatrix} | \mathbf{y}_{1:k-1}; \begin{bmatrix} \hat{\mathbf{x}}_{k|k-1} \\ \mathbf{h}(\hat{\mathbf{x}}_{k|k-1}) \end{bmatrix}, \begin{bmatrix} \mathbf{P}_{k|k-1} & \mathbf{P}_{k|k-1}\tilde{\mathbf{H}}^T \\ \tilde{\mathbf{H}}\mathbf{P}_{k|k-1} & \tilde{\mathbf{H}}\mathbf{P}_{k|k-1}\tilde{\mathbf{H}}^T + \mathbf{R}_k \end{bmatrix} \right)$$

CKF/UKF

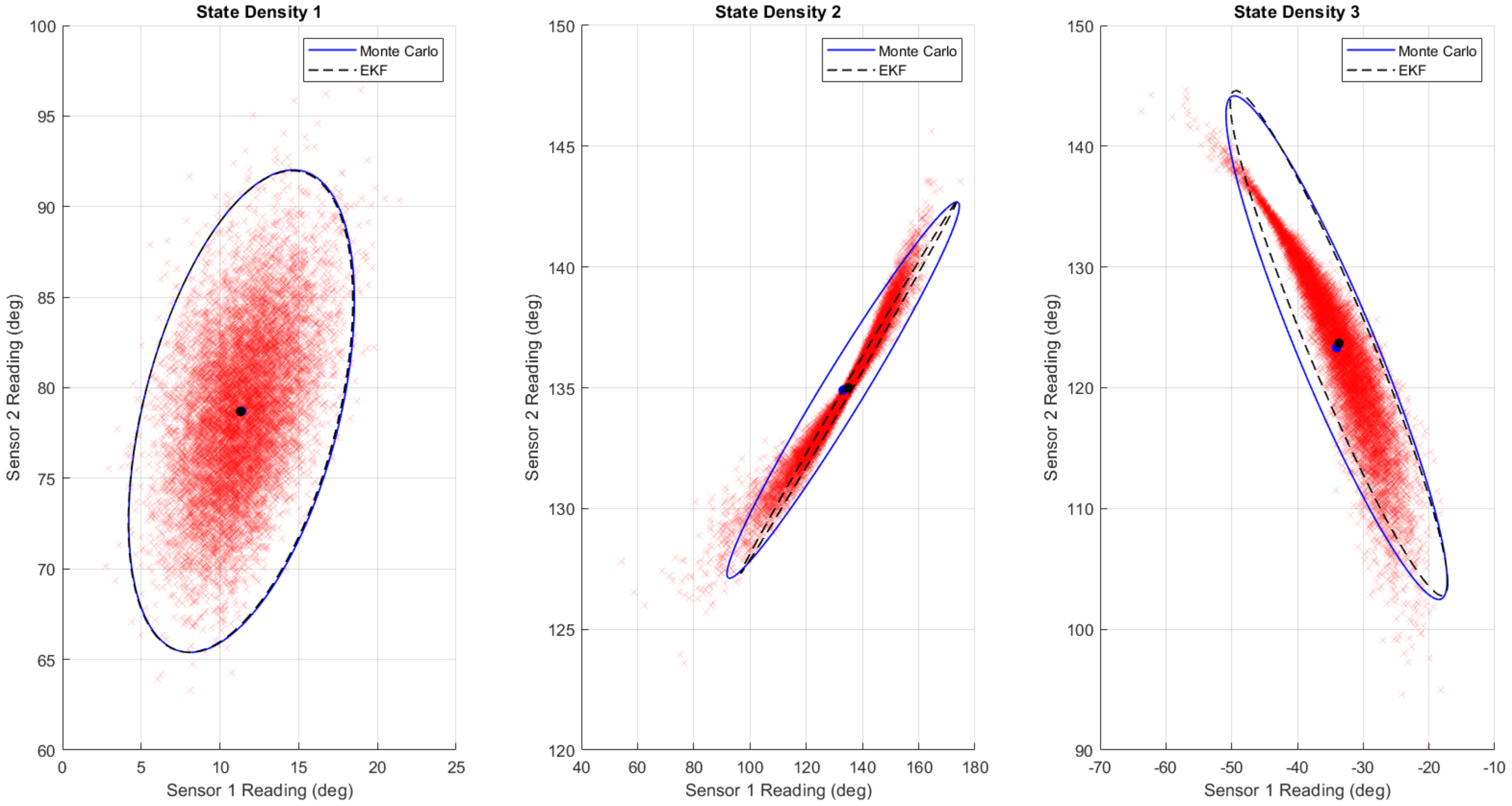
$$\hat{\mathbf{y}} \approx \sum_i w_i \mathbf{h}(\mathcal{X}^{(i)})$$

$$\mathbf{P}_{yy} \approx \sum_i w_i \left(\mathbf{h}(\mathcal{X}^{(i)}) - \hat{\mathbf{y}} \right) \left(\mathbf{h}(\mathcal{X}^{(i)}) - \hat{\mathbf{y}} \right)^T$$

Task 1

EKF

$$p(\mathbf{x}_k, \mathbf{y}_k | \mathbf{y}_{1:k-1}) \approx \mathcal{N}\left(\begin{bmatrix}\mathbf{x}_k \\ \mathbf{y}_k\end{bmatrix} \middle| \mathbf{y}_{1:k-1}; \begin{bmatrix}\hat{\mathbf{x}}_{k|k-1} \\ \mathbf{h}(\hat{\mathbf{x}}_{k|k-1})\end{bmatrix}, \begin{bmatrix}\mathbf{P}_{k|k-1} & \mathbf{P}_{k|k-1}\tilde{\mathbf{H}}^T \\ \tilde{\mathbf{H}}\mathbf{P}_{k|k-1} & \tilde{\mathbf{H}}\mathbf{P}_{k|k-1}\tilde{\mathbf{H}}^T + \mathbf{R}_k\end{bmatrix}\right)$$



Task 1

CKF/UKF

$$\hat{\mathbf{y}} \approx \sum_i w_i \mathbf{h}(\mathcal{X}^{(i)})$$

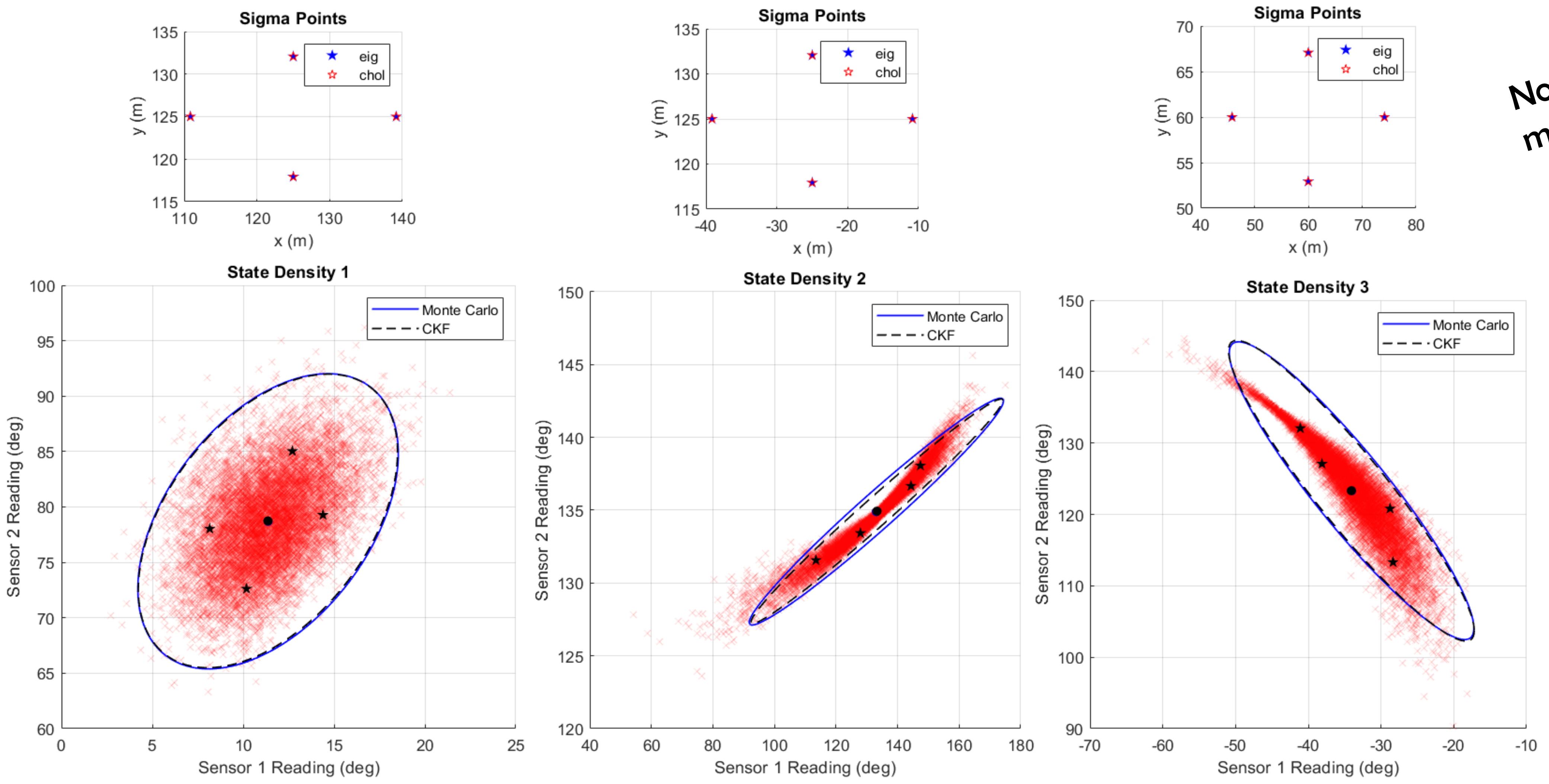
$$\mathbf{P}_{yy} \approx \sum_i w_i \left(\mathbf{h}(\mathcal{X}^{(i)}) - \hat{\mathbf{y}}\right)\left(\mathbf{h}(\mathcal{X}^{(i)}) - \hat{\mathbf{y}}\right)^T$$

Sigma-points are chosen using a transformation of standard normal variables. Will different transformations yield different sigma points?

$$\left.\begin{array}{c}
\mathbf{P} = \mathbf{V}\mathbf{D}\mathbf{V}^T \implies \mathbf{P}^{1/2} = \mathbf{V}\mathbf{D}^{1/2}\mathbf{V}^T \\[2ex]
\mathbf{V}\mathbf{V}^T = \mathbf{V}^T\mathbf{V} = \mathbb{I}_n \\[2ex]
\mathbf{D} = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n) \\[2ex]
\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbb{I}_n) \implies \hat{\mathbf{x}} + \mathbf{V}\mathbf{D}^{1/2}\mathbf{z} \sim \mathcal{N}(\hat{\mathbf{x}}, \mathbf{P})
\end{array}\right\} \text{EIGEN VALUE DECOM}$$

$$\left.\begin{array}{c}
\mathbf{P} = \mathbf{L}\mathbf{L}^T \\[2ex]
\implies \hat{\mathbf{x}} + \mathbf{L}\mathbf{z} \sim \mathcal{N}(\hat{\mathbf{x}}, \mathbf{P})
\end{array}\right\} \text{CHOLESKY DECOMP}$$

Task 1

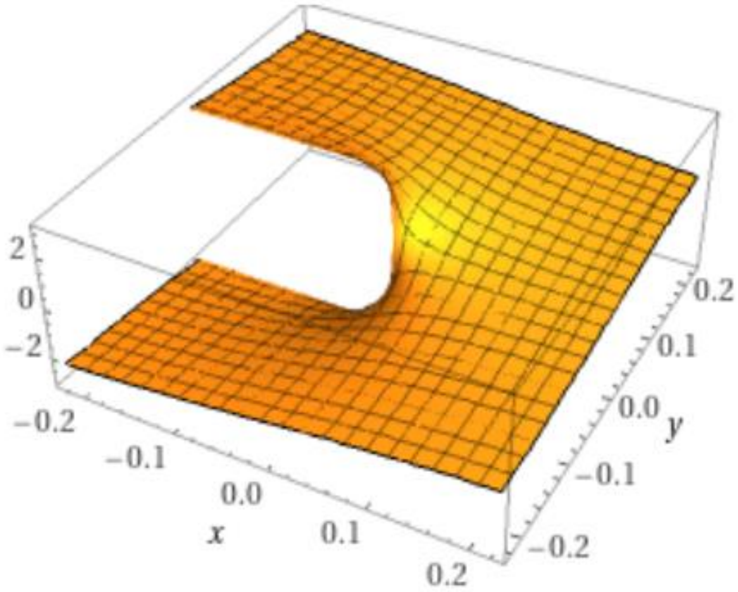


Not in my case!

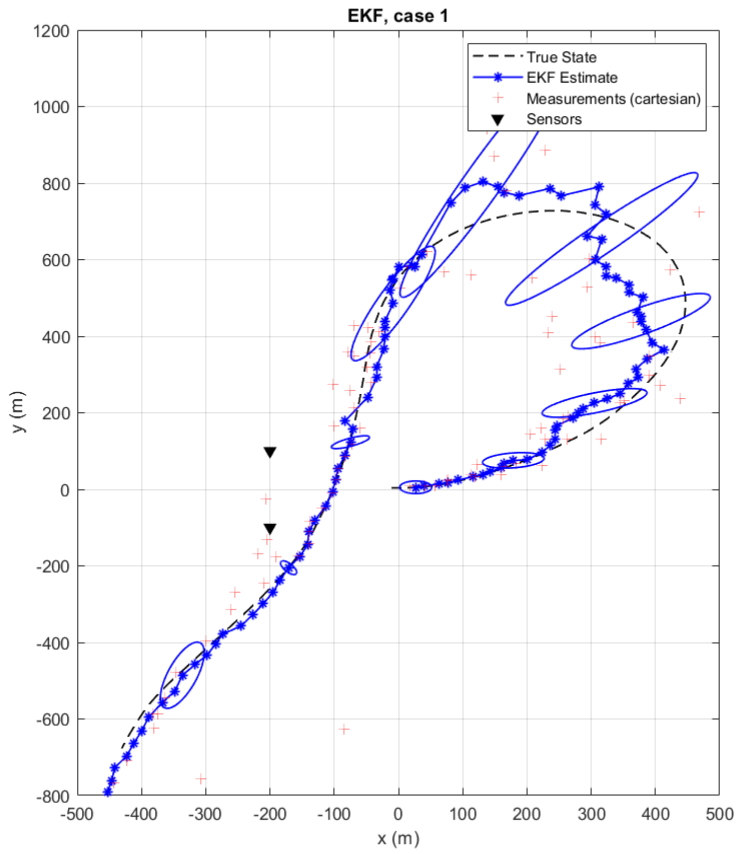
Task 1

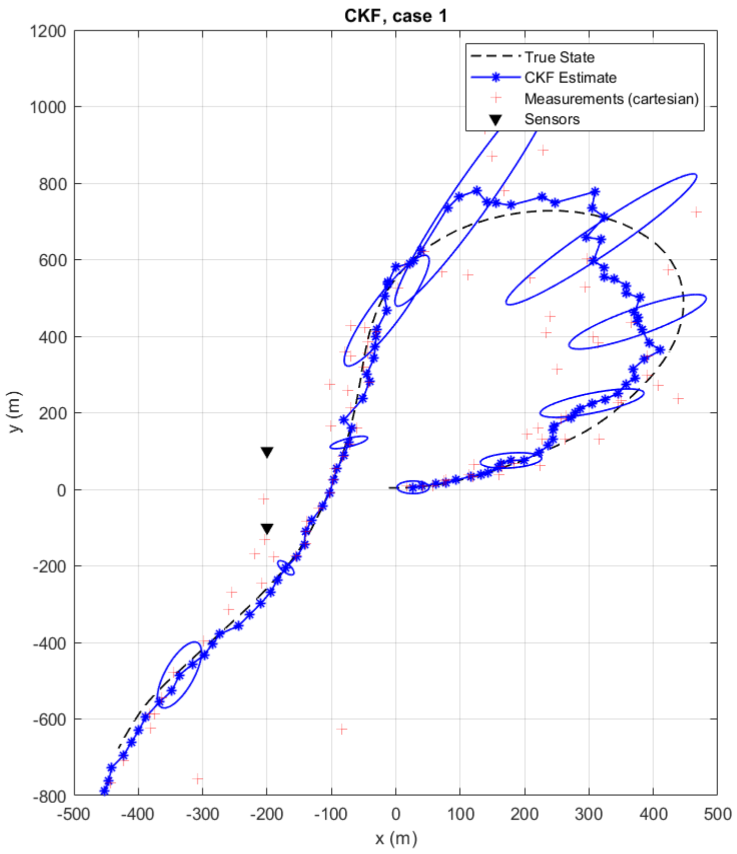
The EKF fares bad when h(x) is highly non-linear!

- Distribution 2 especially has its mean close to one of the sensors. In this region, atan2(y, x) becoming more non-linear and its derivative starts having large amplitudes.



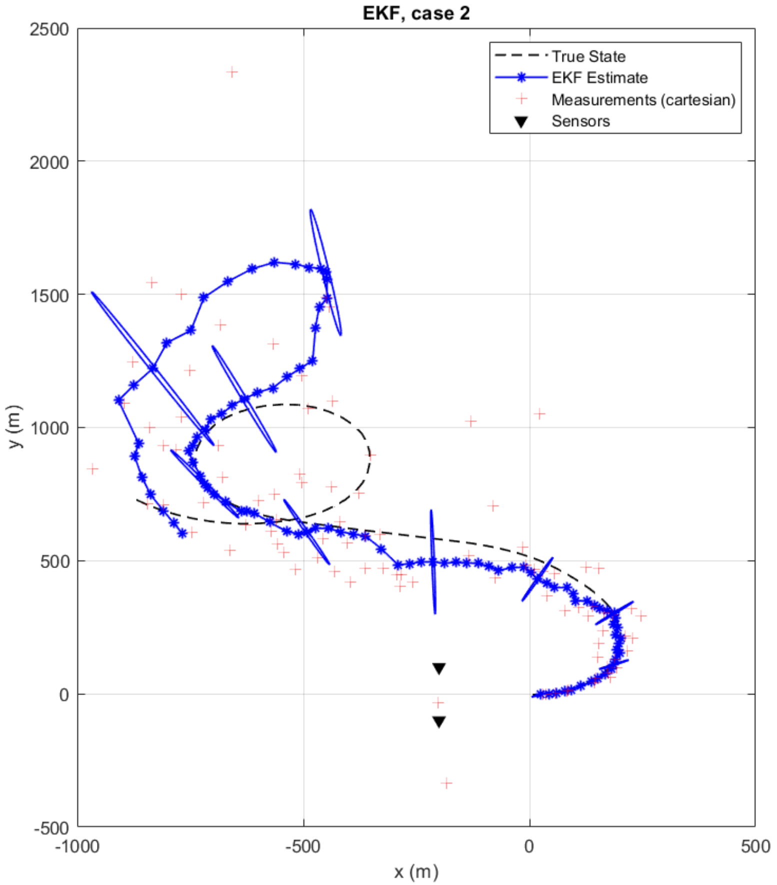
Task 2 – Case 1

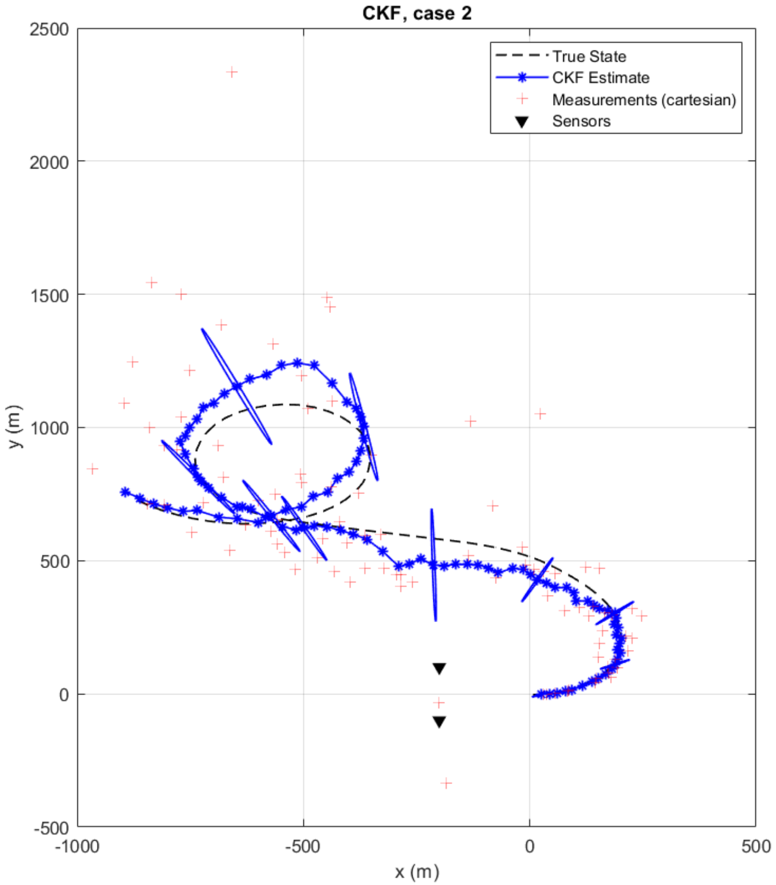




In general, covariances are larger with distance and angle from sensor center-line.

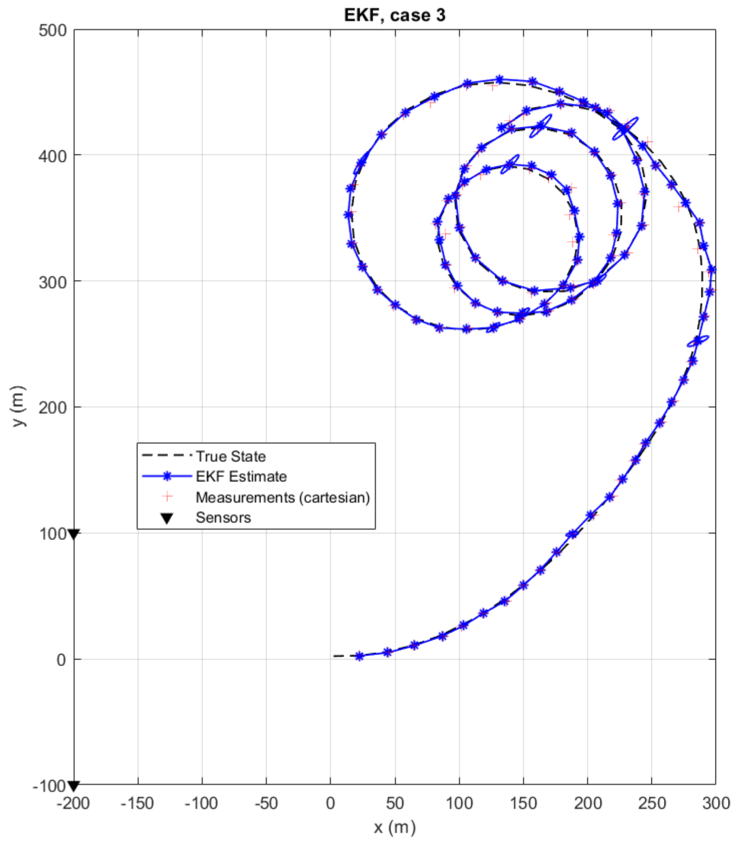
Task 2 – Case 2



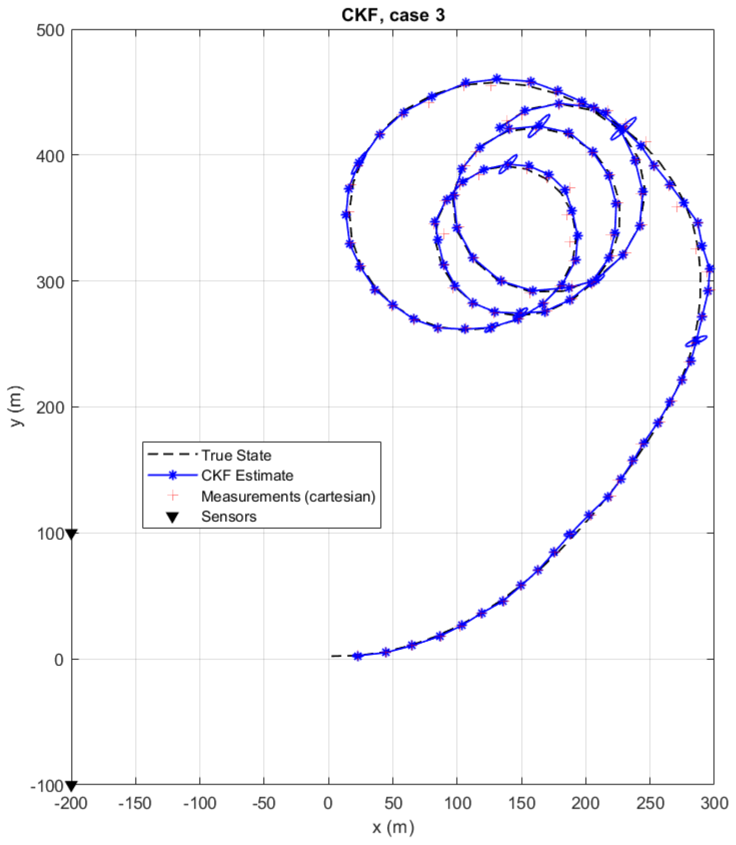


Less measurement noise for one of the sensors, improved angular accuracy and slight improvement in distance.

Task 2 – Case 3

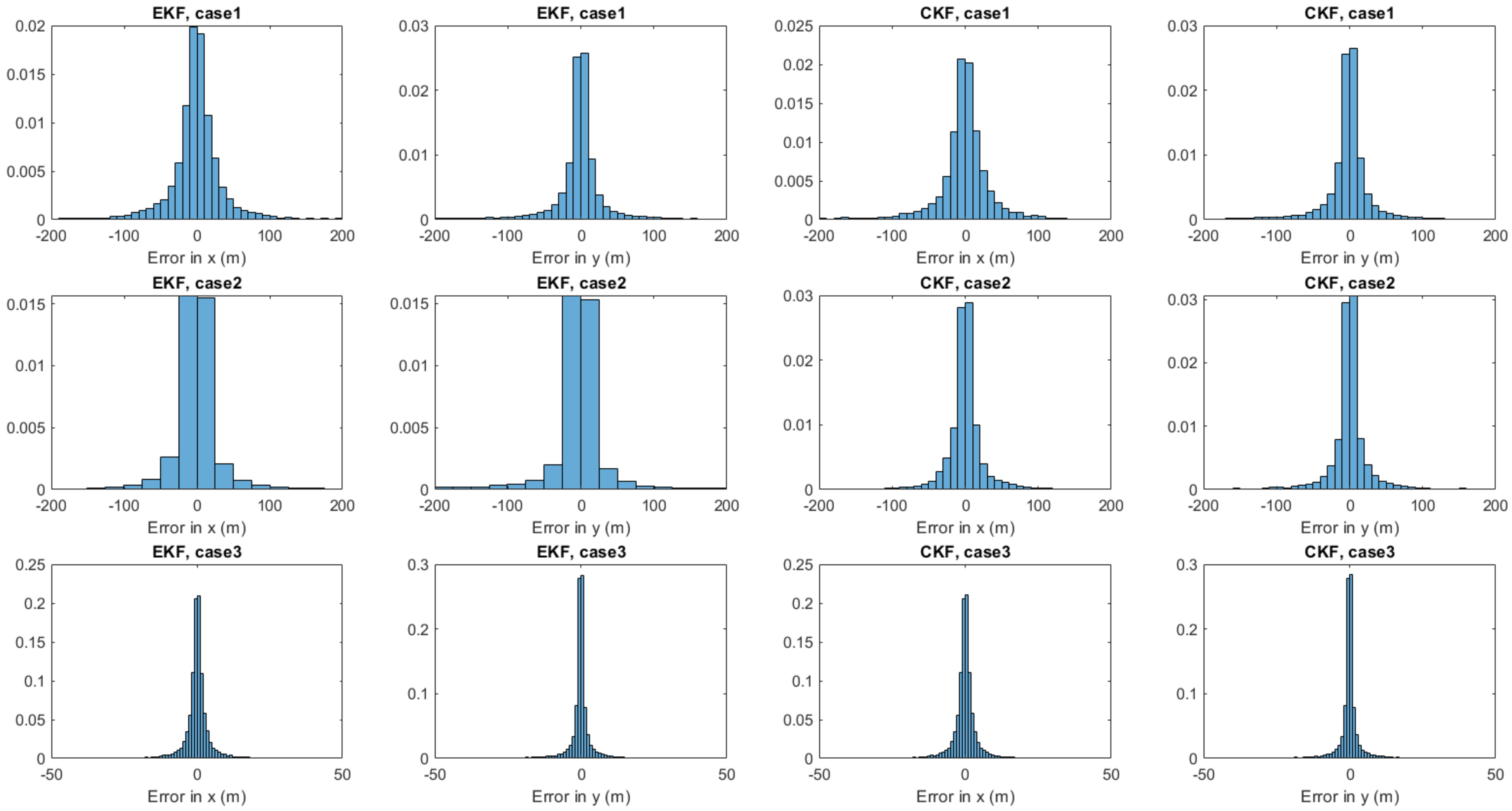


EKF, case 3
True State
EKF Estimate
Measurements (cartesian)
Sensors
x (m)
y (m)



CKF, case 3
True State
CKF Estimate
Measurements (cartesian)
Sensors
x (m)
y (m)

Task 2 – Histograms

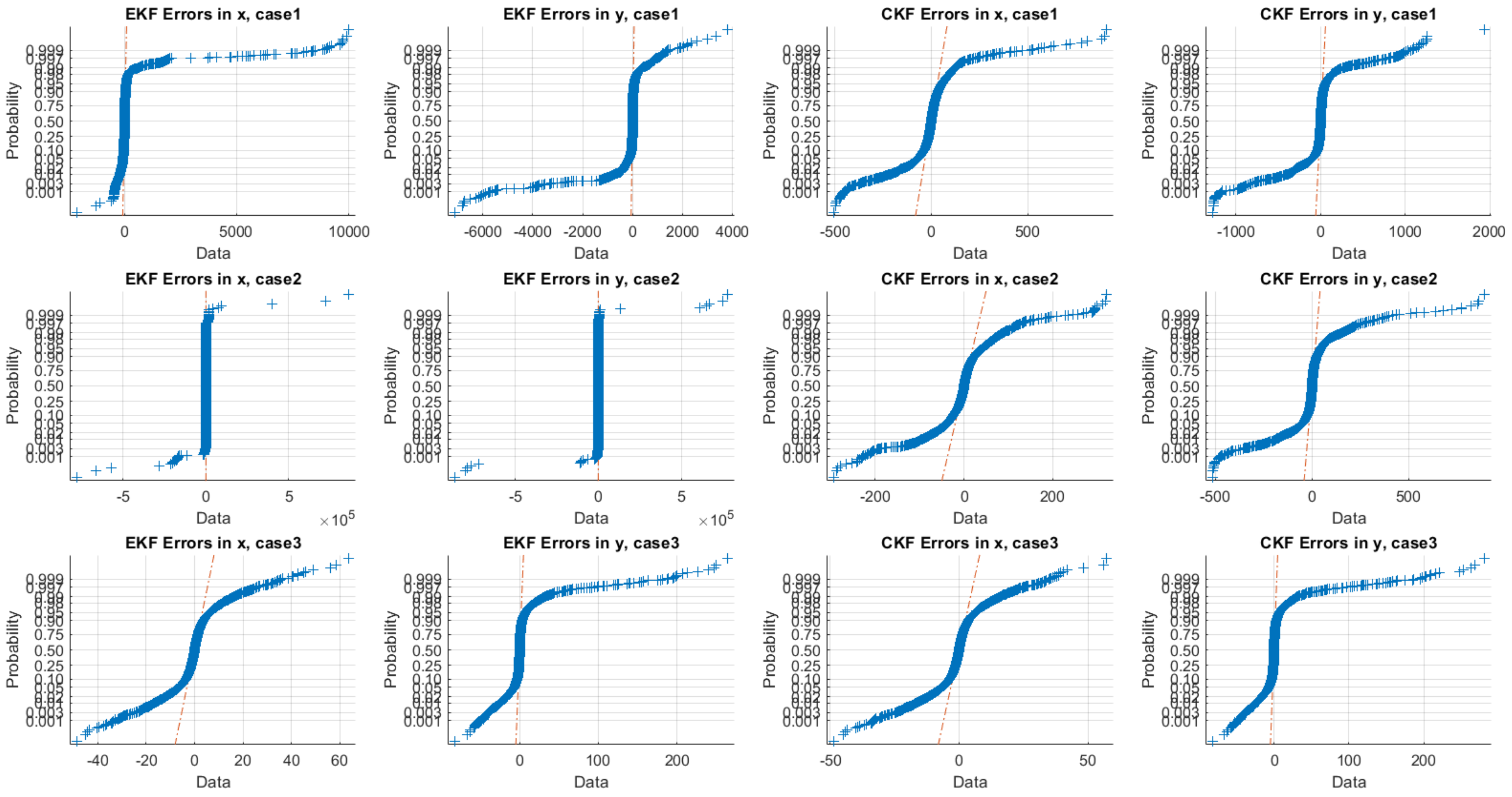


Histograms don't look Gaussian and there seems to be more spread along!

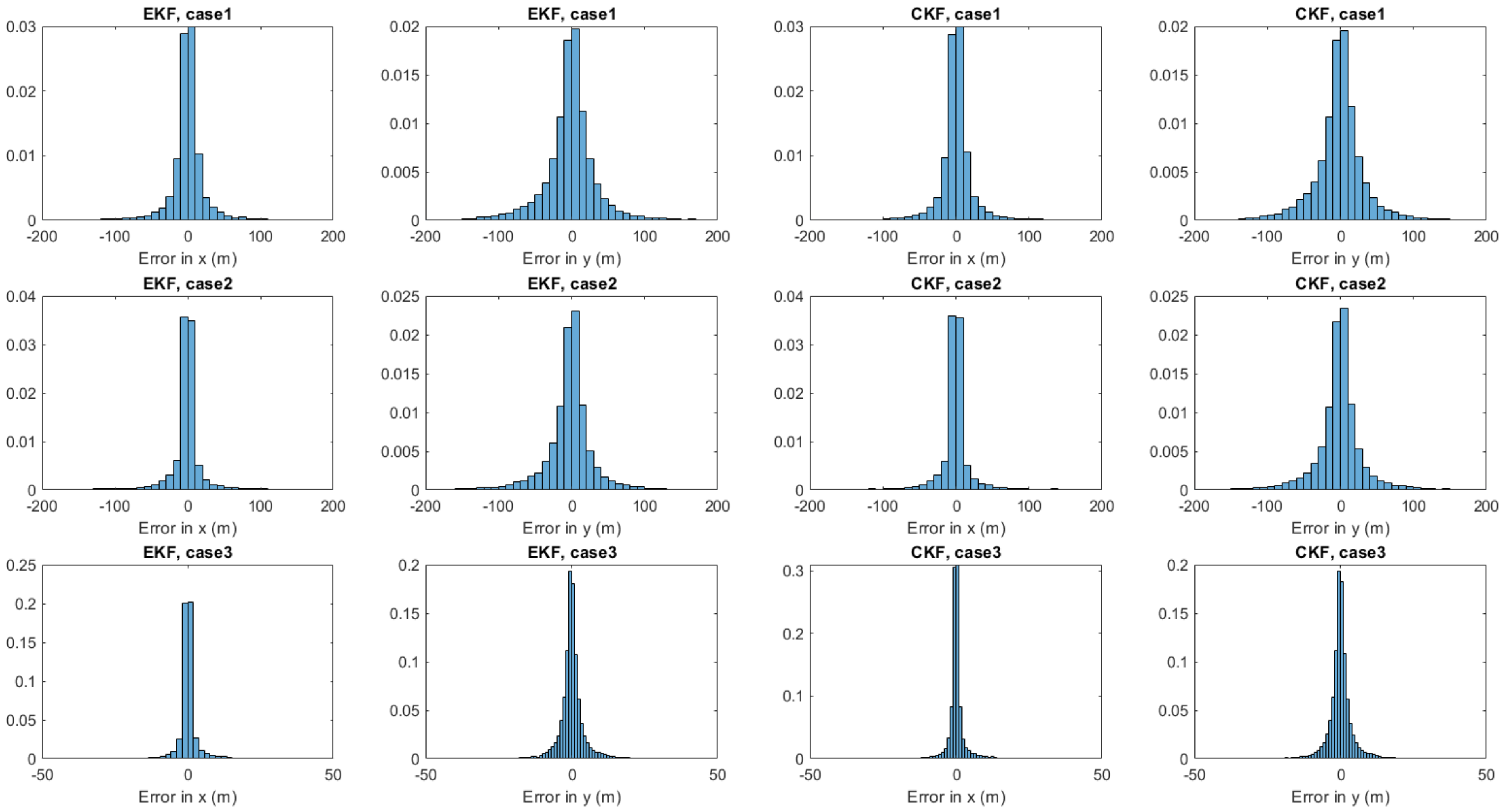
- Sensor array is along y which gives improved angular accuracy in that direction.
- What happens if we put the sensor array along x?

Task 2 – Normal plots

normplot(data)



Task 2 – Histograms, array along x!



More spread along y!

Task 3 (thanks Xixi)

$$\sigma_v = 100 \text{ and } \sigma_\omega = \frac{100\pi}{180}.$$

Tuning nonlinear filters

- Task a-larger process noise
- the filter takes very little consideration to the process model and trusts mostly on the measurements.
- non-smooth and un-physical trajectory
- dependant on the accuracy of the measurements
- unsure in its prediction which is shown with large covariance contours.

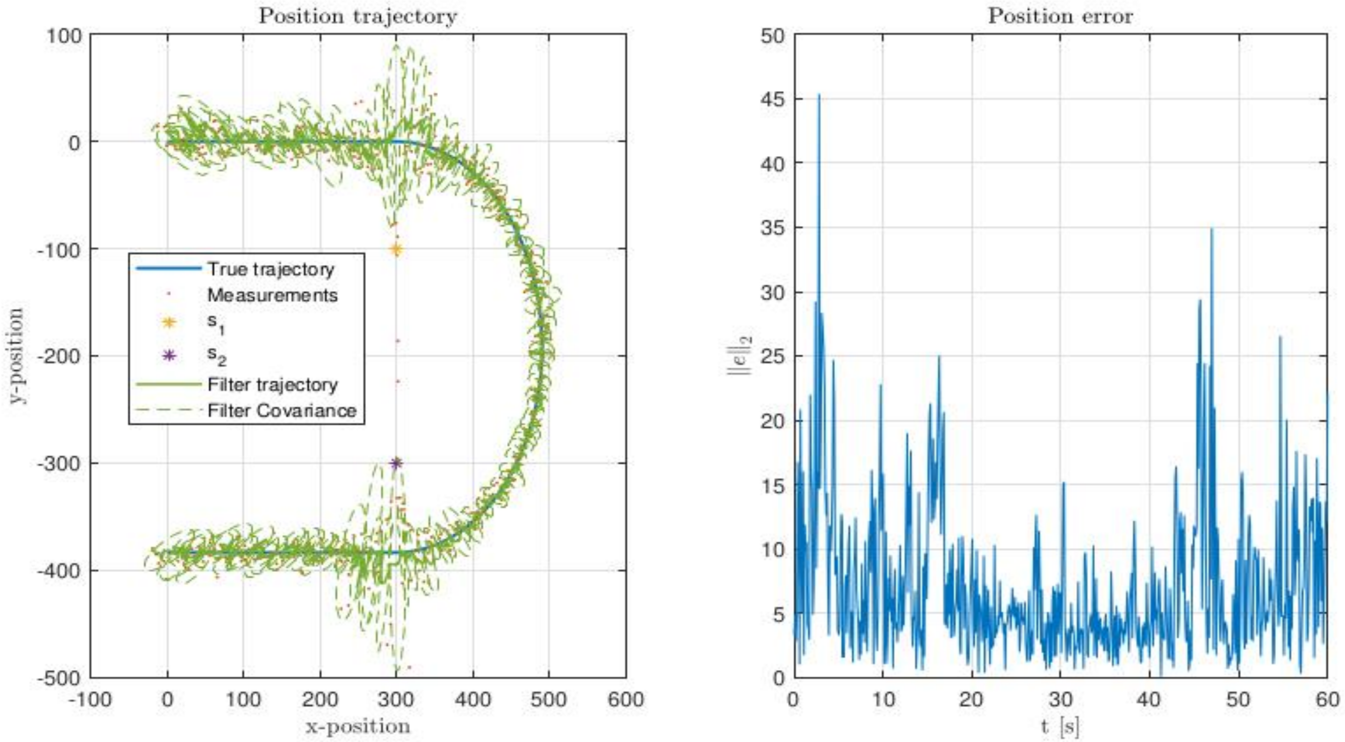


Figure 3.1: Using too large process noise variance

Tuning nonlinear filters

- Smaller process noise
- trusts the process model and thus gives a more smooth trajectory.
- does not use the measurements enough leading to a drift and systematic errors.
- very small covariance contours
- systematic errors starting after the trajectory goes from straight to turning (t = 15) and from turning to straight (t = 45).

$$\sigma_v = 0.01 \quad \sigma_\omega = \frac{\pi}{100 \cdot 180}.$$

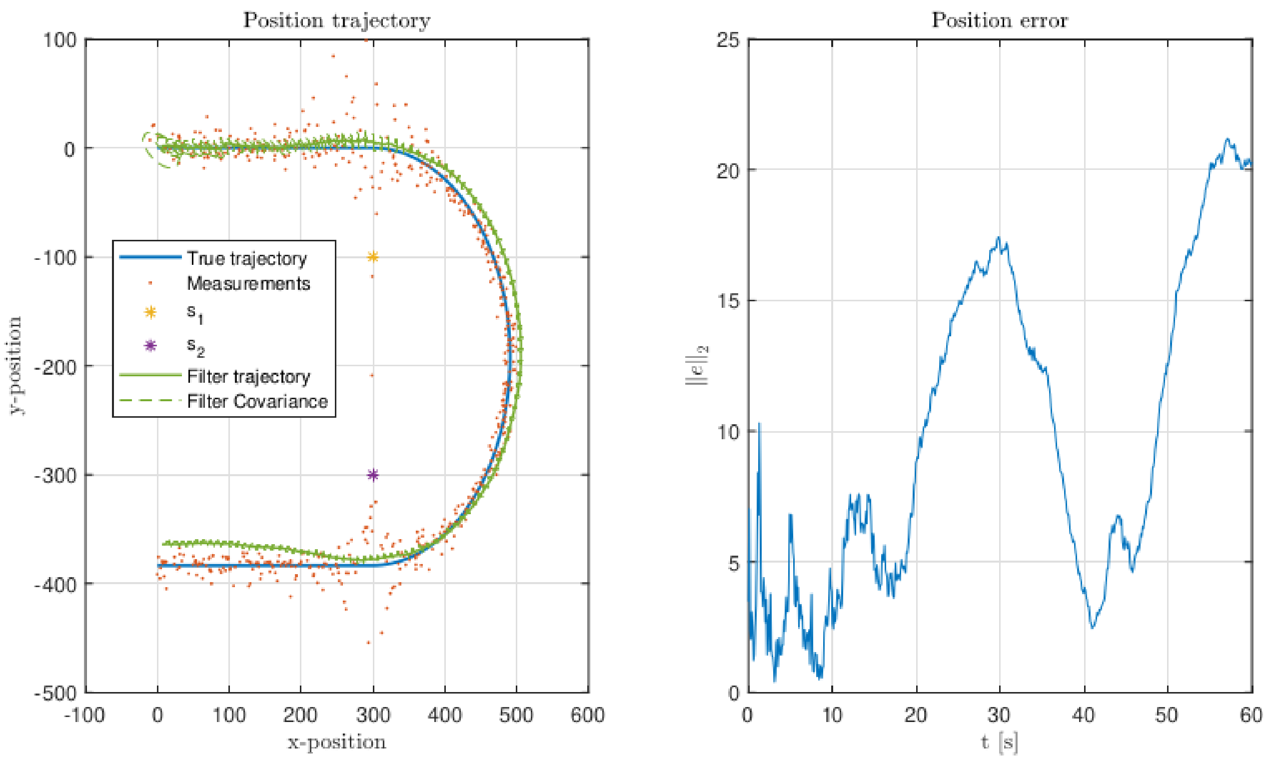


Figure 3.2: Using too small process noise variance

Tuning nonlinear filters

- well tuned process noise
- Constant velocity → std for velocity can be small
- measurement noise is more significant because of the trajectory passing the line between the two sensors.

$$\sigma_\omega = \frac{0.6 \cdot \pi}{180}.$$

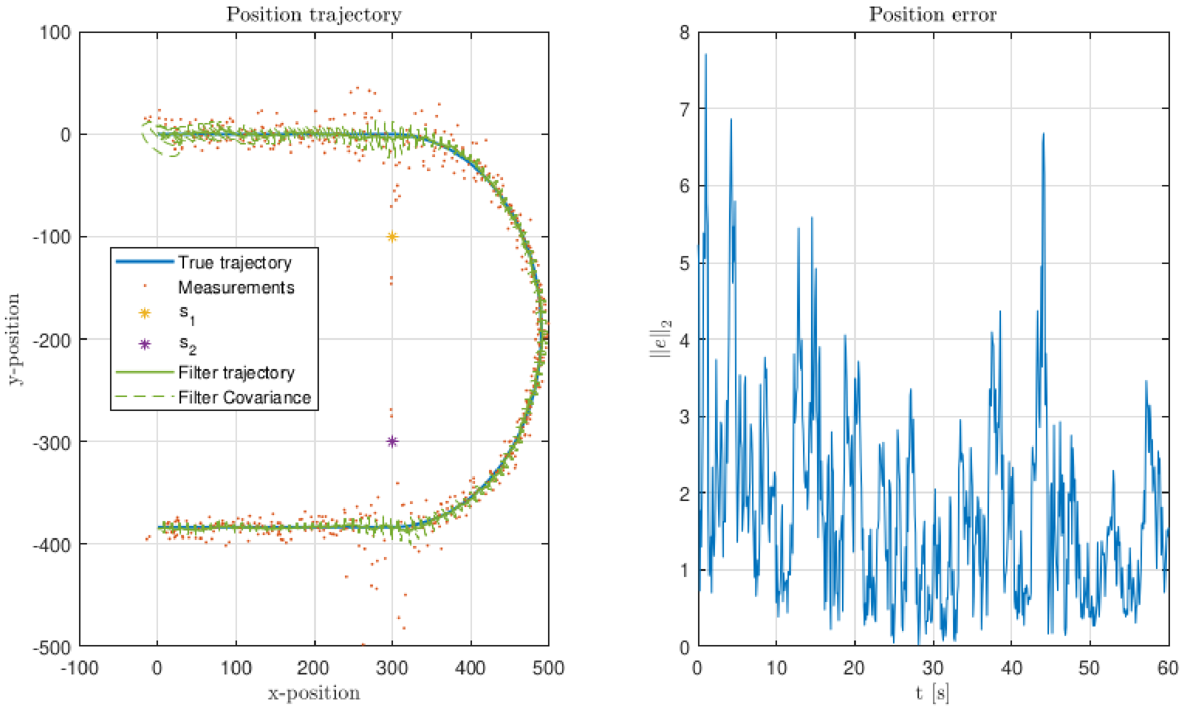


Figure 3.3: Using well-tuned process noise variance

Tuning nonlinear filters

- Velocity is good everywhere except the beginning
- It takes some time for filters to converge
- Heading mismatch while it changes from straight to turning
- Inaccurate measurements
- Reduce the noise on the turn-rate. However, reducing the noise will lead to a delay in the changing of heading and turn rate
- Low process noise for the straight line
- Higer process noise during the turn

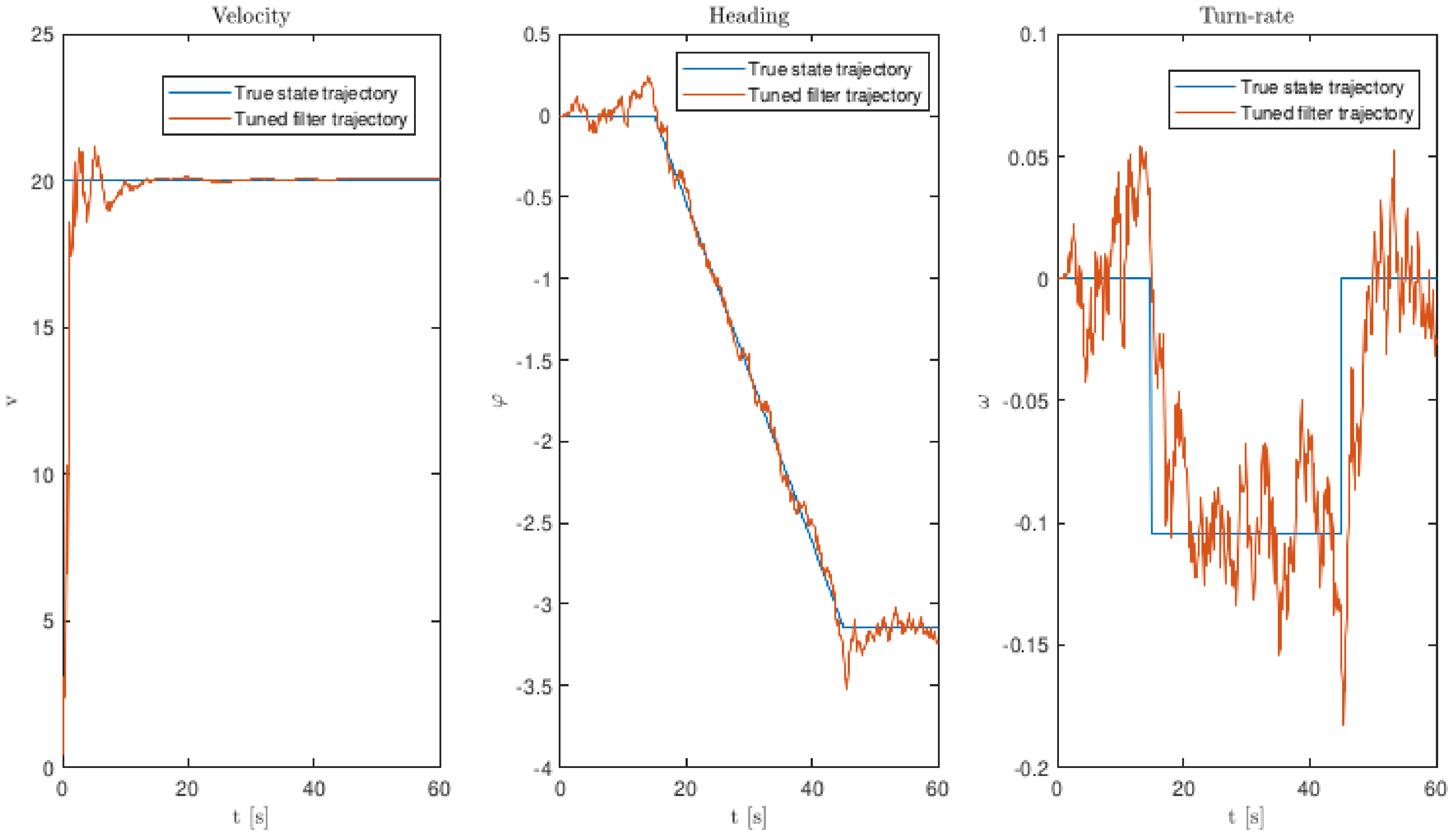


Figure 3.4: Trajectories for velocity, heading and turn-rate.