Solution to analysis in Home Assignment 3

Alexander Boström + alebostr

Analysis

In this report I will present my independent analysis of the questions related to home assignment 3. I have discussed the solution with Jakkapan Kurusakdapong, Kanatip Anuchit and Polychronis Kanellos but I swear that the analysis written here are my own.

1 Approximations of mean and covariance

A)

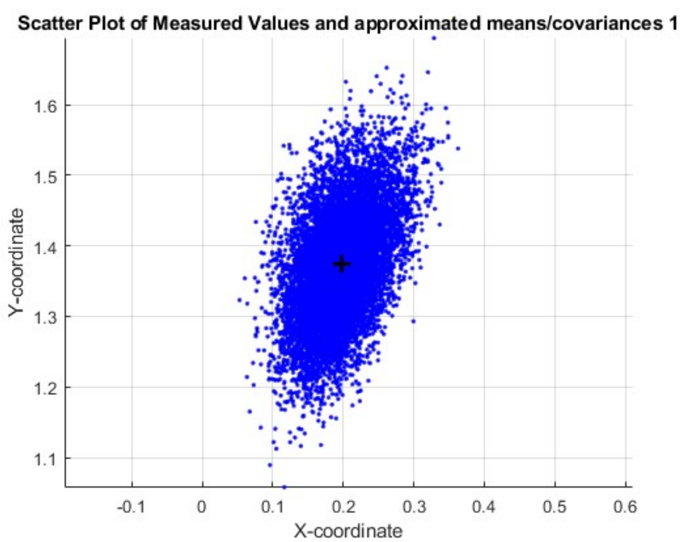


Figure 1.1: Measurement samples of Case 1

Mean of *measurments*:

$$\begin{bmatrix} 0.1985 \\ 1.3746 \end{bmatrix}$$

Covariance of *measurments*:

$$\begin{bmatrix} 0.0018 & 0.0015 \\ 0.0015 & 0.0060 \end{bmatrix}$$

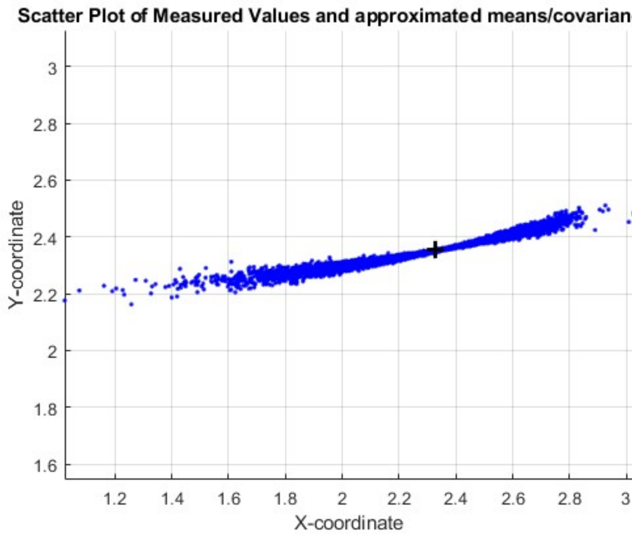


Figure 1.2: Measurement samples of Case 2

Mean of *measurments*:

$$\begin{bmatrix} 2.3248 \\ 2.3548 \end{bmatrix}$$

Covariance of *measurement*:

$$\begin{bmatrix} 0.0568 & 0.0105 \\ 0.0105 & 0.0020 \end{bmatrix}$$

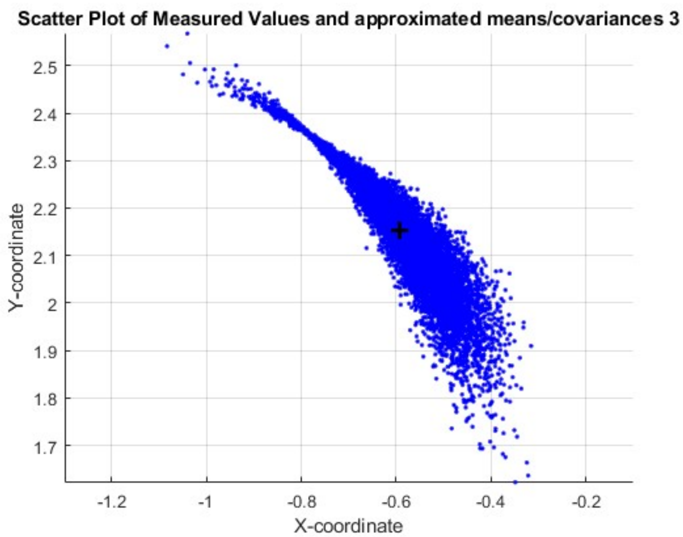


Figure 1.3: Measurement samples of Case 3

Mean of *measurement*:

$$\begin{bmatrix} -0.5942 \\ 2.1517 \end{bmatrix}$$

Covariance of *measurement*:

$$\begin{bmatrix} 0.0097 & -0.0111 \\ -0.0111 & 0.0149 \end{bmatrix}$$

B)

EKF

Case	Approximated Mean	Approximated Covariance
Case 1 (EKF)	$\begin{bmatrix} 0.1974 \\ 1.3734 \end{bmatrix}$	$\begin{bmatrix} 0.0017 & 0.0015 \\ 0.0015 & 0.0060 \end{bmatrix}$
Case 2 (EKF)	$\begin{bmatrix} 2.3562 \\ 2.3562 \end{bmatrix}$	$\begin{bmatrix} 0.0500 & 0.0100 \\ 0.0100 & 0.0020 \end{bmatrix}$
Case 3 (EKF)	$\begin{bmatrix} -0.5880 \\ 2.1588 \end{bmatrix}$	$\begin{bmatrix} 0.0092 & -0.0111 \\ -0.0111 & 0.0148 \end{bmatrix}$

Table 1: Approximated Mean and Covariance with EKF for Different test Cases

The EKF linearizes the measurement model around the predicted state density and approximates the mean and covariance according to:

$$\text{Mean}(y)_{\text{EKF}} = h(\bar{x})$$

where $h(\cdot)$ is the measurement model evaluated at the predicted state estimate $\bar{x}$.

$$\text{Cov}(y)_{\text{EKF}} = HPH^T + R$$

where H is the Jacobian matrix of the measurement model with respect to the state, P is the predicted state covariance, and R is the measurement noise covariance.

UKF

Case	**Approximated Mean**	**Approximated Covariance**
Case 1 (UKF)	$\begin{bmatrix} 0.1983 \\ 1.3743 \end{bmatrix}$	$\begin{bmatrix} 0.0017 & 0.0015 \\ 0.0015 & 0.0059 \end{bmatrix}$
Case 2 (UKF)	$\begin{bmatrix} 2.3269 \\ 2.3550 \end{bmatrix}$	$\begin{bmatrix} 0.0566 & 0.0105 \\ 0.0105 & 0.0020 \end{bmatrix}$
Case 3 (UKF)	$\begin{bmatrix} -0.5949 \\ 2.1524 \end{bmatrix}$	$\begin{bmatrix} 0.0097 & -0.0112 \\ -0.0112 & 0.0150 \end{bmatrix}$

Table 2: Approximated Mean and Covariance with UKF for Different test Cases

The UKF method computes a set of sigma points ($x_k^{(i)}$), $2n+1$, around the predicted state $\bar{x}$ using the Cholesky decomposition of the covariance matrix. Then, it applies the nonlinear measurement model to each sigma point to obtain predicted measurement points. Lastly, it computes the weighted average of the predicted measurement points to get the approximated mean and covariance.

$$\text{Mean}(y)_{\text{UKF}} = \sum_{i=0}^{2n} h(x_k^{(i)})W_i$$

$$\text{Cov}(y)_{\text{UKF}} \approx R + \sum_{i=0}^{2n} (h(x_k^{(i)}) - \hat{y}_{kk-1})(h(x_k^{(i)}) - \hat{y}_{kk-1})^T W_i;$$

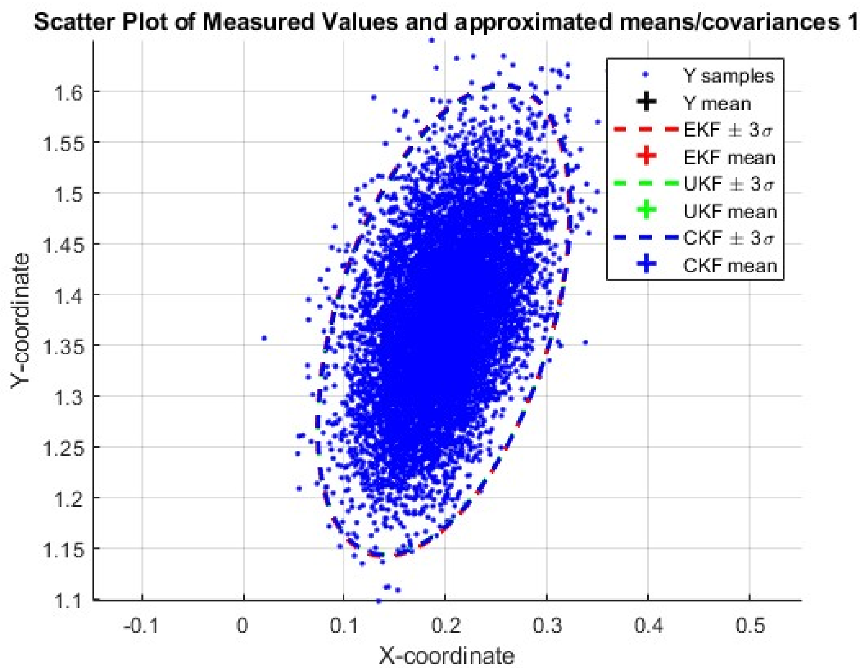
CKF

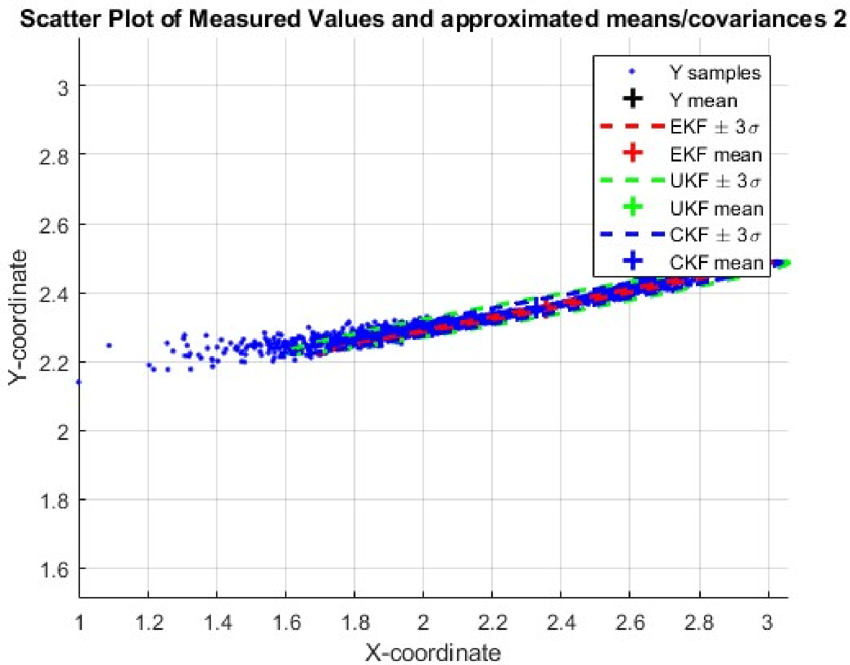
Case	**Approximated Mean**	**Approximated Covariance**
Case 1 (CKF)	$\begin{bmatrix} 0.1983 \\ 1.3743 \end{bmatrix}$	$\begin{bmatrix} 0.0017 & 0.0015 \\ 0.0015 & 0.0059 \end{bmatrix}$
Case 2 (CKF)	$\begin{bmatrix} 2.3265 \\ 2.3550 \end{bmatrix}$	$\begin{bmatrix} 0.0566 & 0.0105 \\ 0.0105 & 0.0020 \end{bmatrix}$
Case 3 (CKF)	$\begin{bmatrix} -0.5948 \\ 2.1523 \end{bmatrix}$	$\begin{bmatrix} 0.0097 & -0.0112 \\ -0.0112 & 0.0150 \end{bmatrix}$

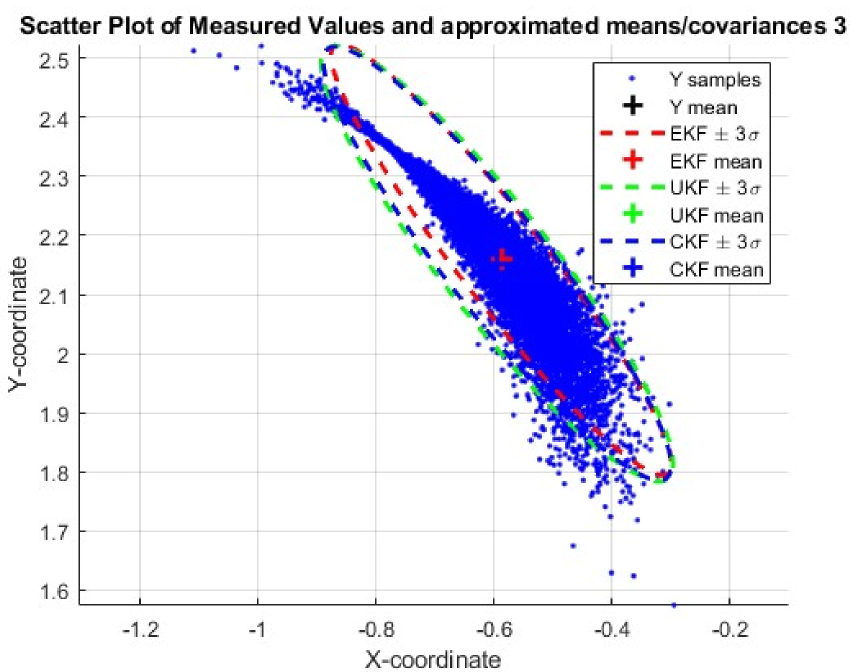
Table 3: Approximated Mean and Covariance with CKF for Different test Cases

Similar to UKF, CKF computes a set of sigma points, but this time $2n$ points. Similarly, CKF then applies the nonlinear measurement model to each sigma point to obtain predicted measurement points.

C)




Scatter Plot of Measured Values and approximated means/covariances 2
Y samples
Y mean
EKF ± 3σ
EKF mean
UKF ± 3σ
UKF mean
CKF ± 3σ
CKF mean
Y-coordinate
X-coordinate


Scatter Plot of Measured Values and approximated means/covariances 3
Y samples
Y mean
EKF ± 3σ
EKF mean
UKF ± 3σ
UKF mean
CKF ± 3σ
CKF mean
Y-coordinate
X-coordinate

D)

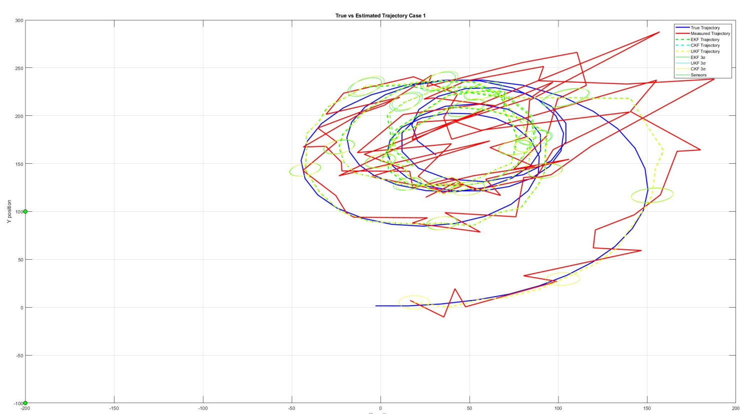
Overall, the three different methods are very similar in the approximation for both the mean and the covariance. However, in the cases 2, the EKF method may not work as well in capturing the covariance of the nonlinear measurements. Since EKF linearizes the system model around the current state estimate, it can introduce errors if the linearization point is far from the true state.

EKF involves the computation of Jacobian matrices for the state transition and observation functions, which can be computationally expensive, especially for high-dimensional systems. In contrast, UKF and CKF typically require fewer computations since they do not need to compute Jacobians.

If the system exhibits strong nonlinearities, I would prefer UKF or CKF due to their ability to accurately capture nonlinearities without the need for costly linearizations.

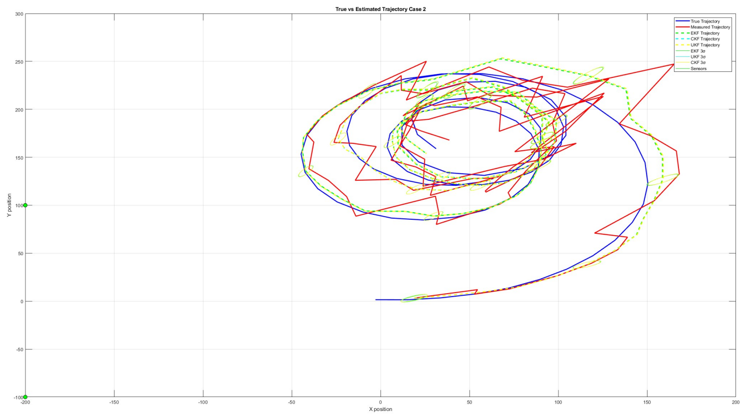
2 Non-linear Kalman filtering

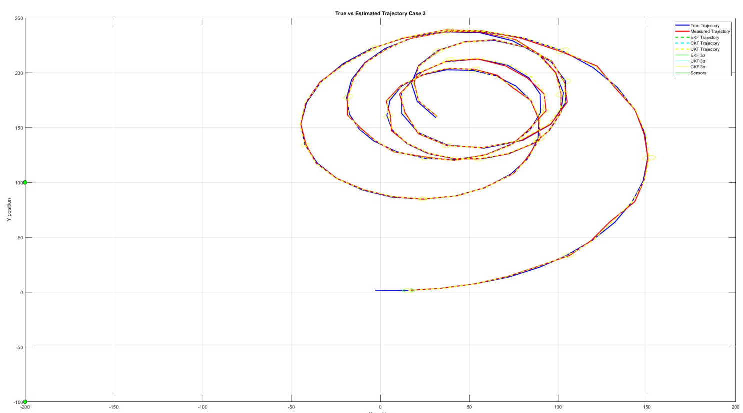
A)



From the first case, as seen from the figure, one can observe that the measurement sequence is very noisy, causing the filter estimates to deviate significantly from the true trajectory. However, despite the noisy measurements, the filter estimates still manage to roughly follow the general shape of the true trajectory.

B)





Case 2 is quite similar to Case 1, where the measurement sequence is very noisy. However, as the noise is less than in Case 1, the estimates fit the true state better.

For Case 3, a significant difference can be observed, as the noise in the sensor is decreased, resulting in the measurement sequence being much closer to the true trajectory. This also makes the filter's prediction very close to the true state.

I believe that the error covariance represents the uncertainty well because it is the largest in Case 1 and decreases in Case 2 and Case 3. This makes sense as the measurement noise gets lower.

C)

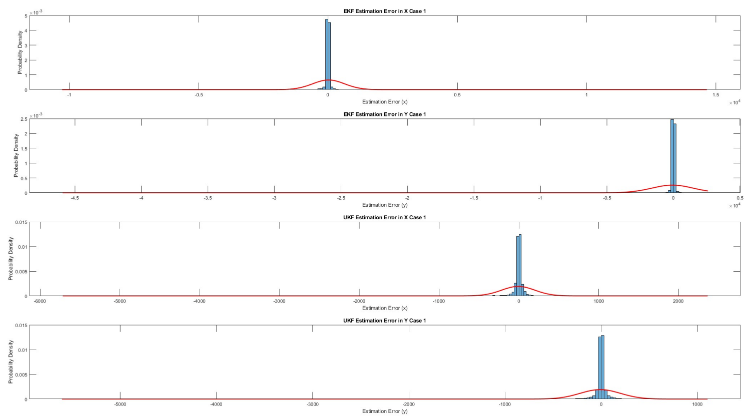


Figure 2.1: Case 1

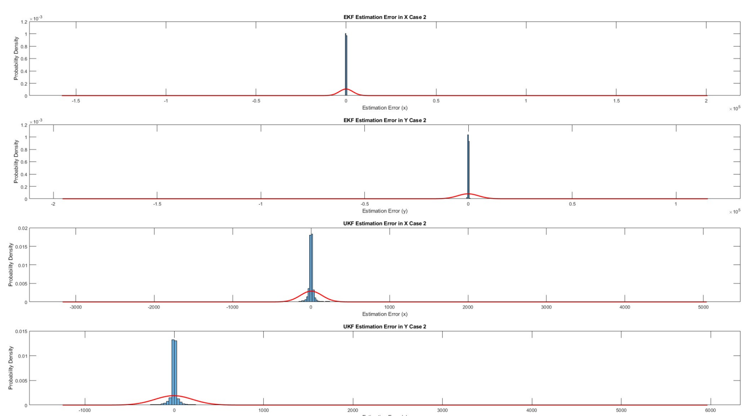


Figure 2.2: Case 2

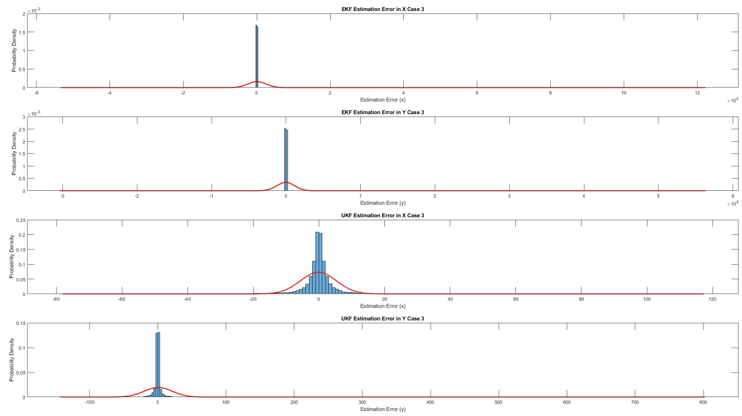


Figure 2.3: Case 3

I suspect that for part C, I encountered some issues when simulating the sequence Y, as it was very noisy in some simulations or showed significant deviations in its measurements compared to the true state. This made the covariance standard deviation very large, as seen in the fitted Gaussian distributions.

However, from what I can conclude from the figures above, after evaluating the errors from multiple simulations of the EKF and UKF filters, it seems like the EKF performs better on average than the UKF for all three cases. This could be because the degree of nonlinearity of the models used favors the EKF in approximating the nonlinearities more efficiently.

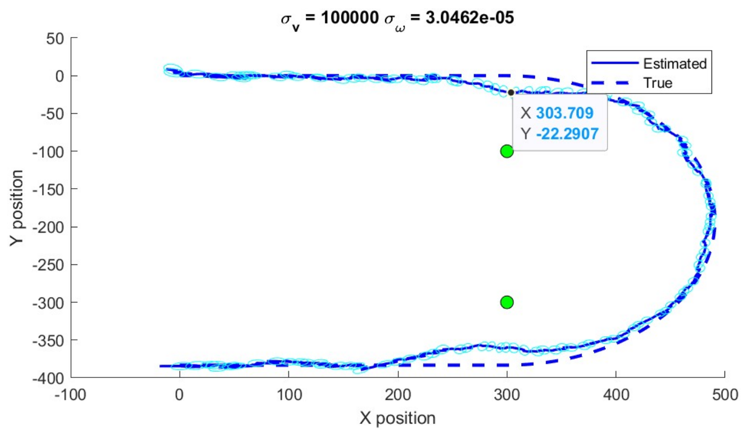
For most of the plots in all the cases above, the fitted Gaussian distributions do not match the real distribution of the histogram.

3 Tuning non-linear filters

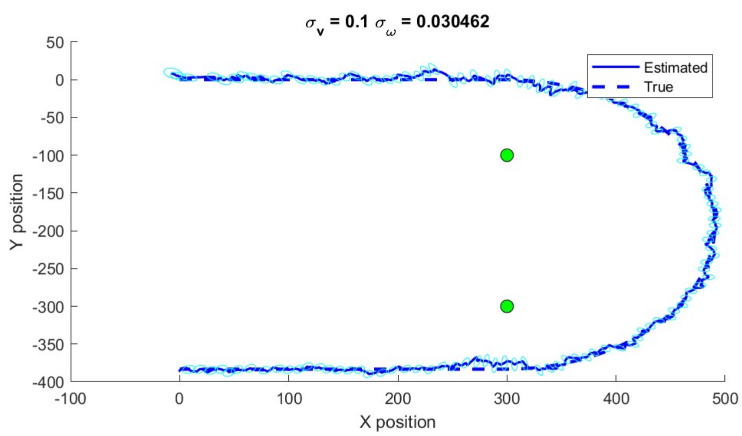
A)

When increasing the process noise:

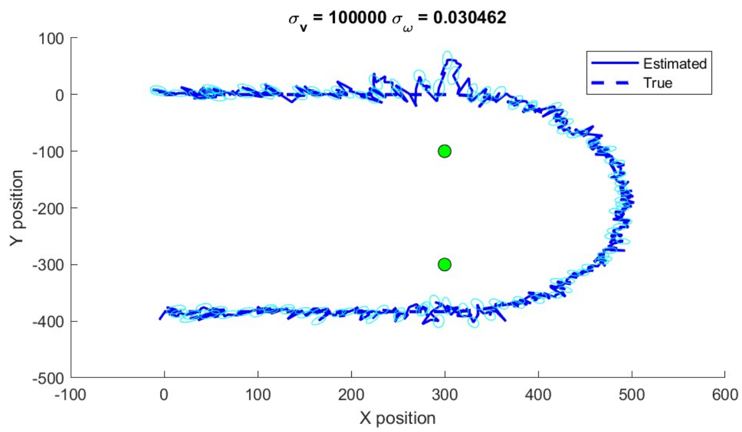
- Increasing σ_v:



- Increasing σ_ω:

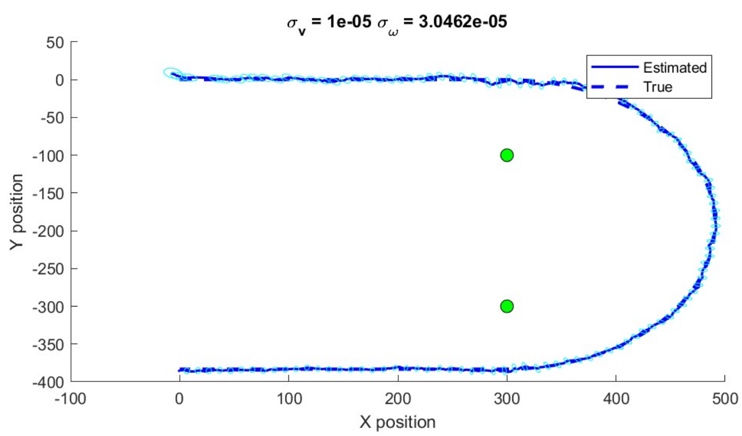


- Increasing both parameters:

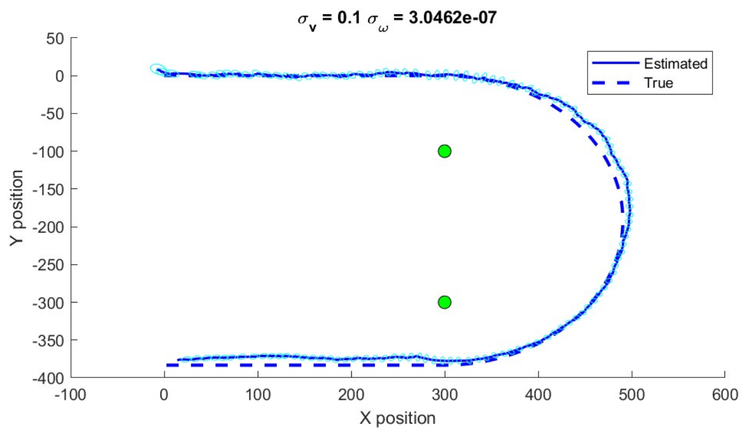


When decreasing the process noise:

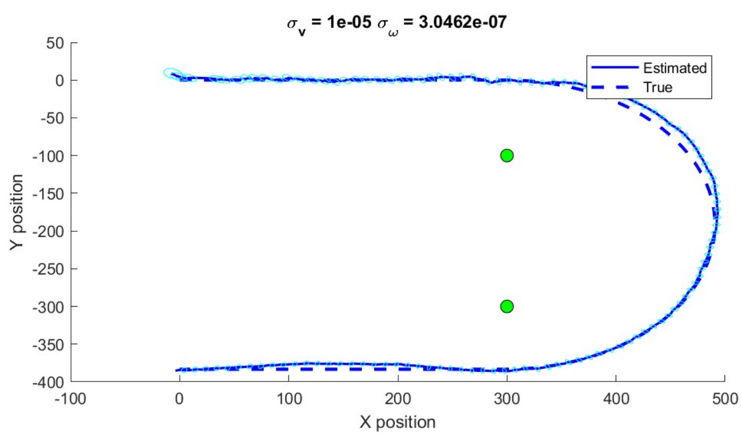
- Decreasing σ_v:



- Decreasing σ_ω:



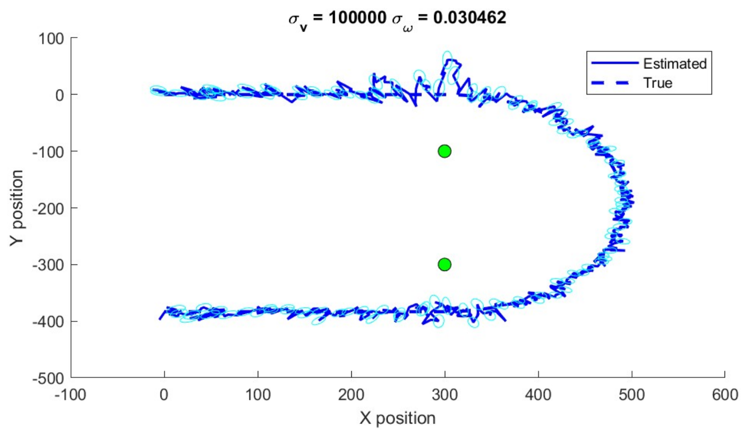
- Decreasing both parameters:



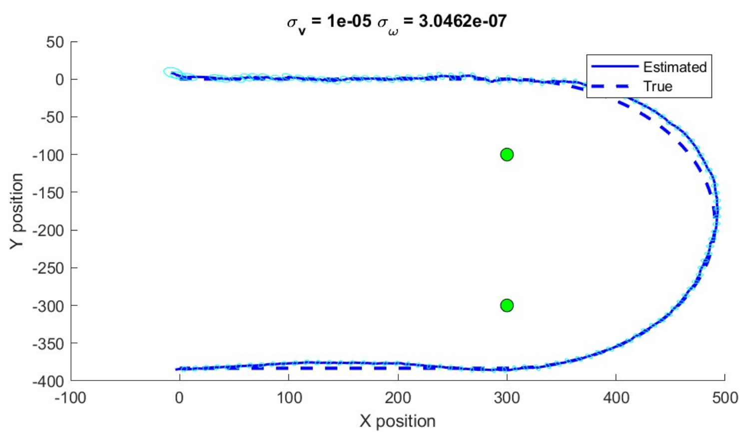
C)

To obtain good position estimates for the entire sequence, I needed to consider how the noise affects both the velocity and turn rate. I generate a measurement sequence and filter it with three different process noise settings:

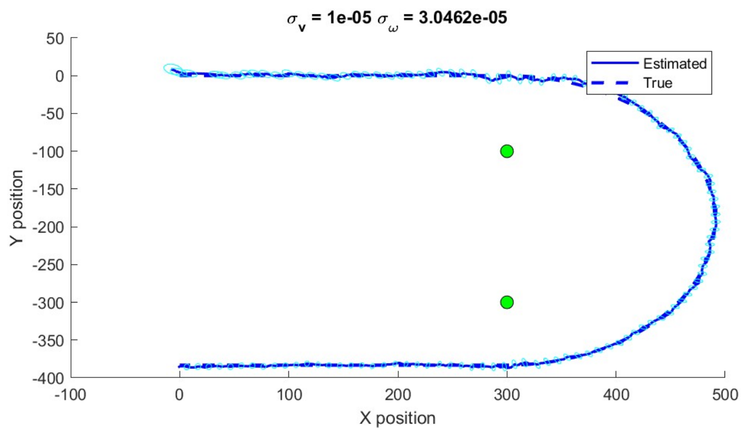
- Too large



- Too small

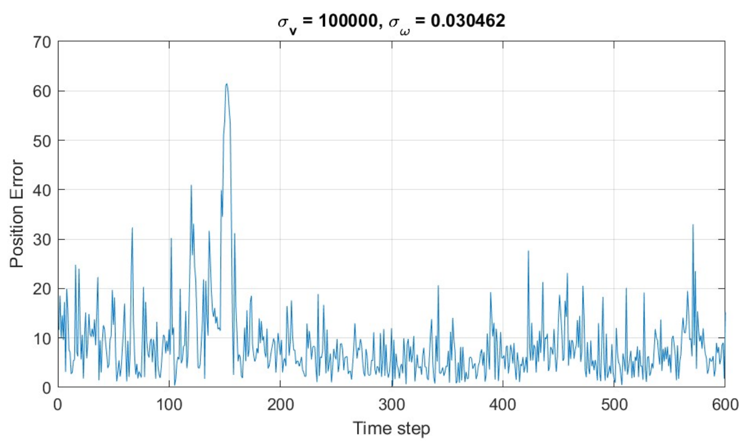


- Well-tuned

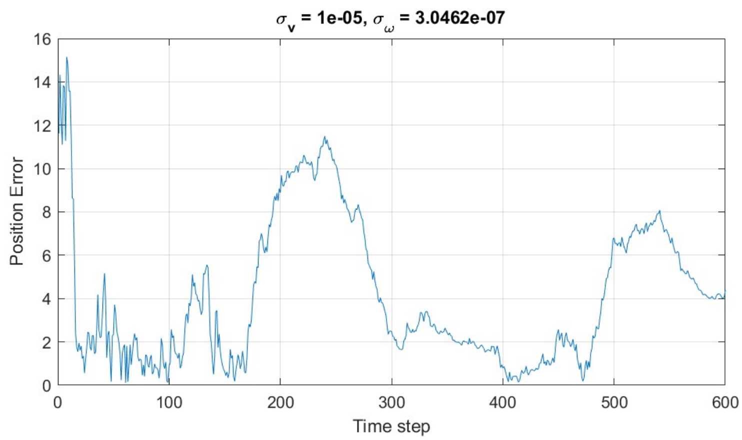


I compute the position error $\|p_k - \hat{p}_{k|k}\|^2$ and plot it vs time for all three noise settings.

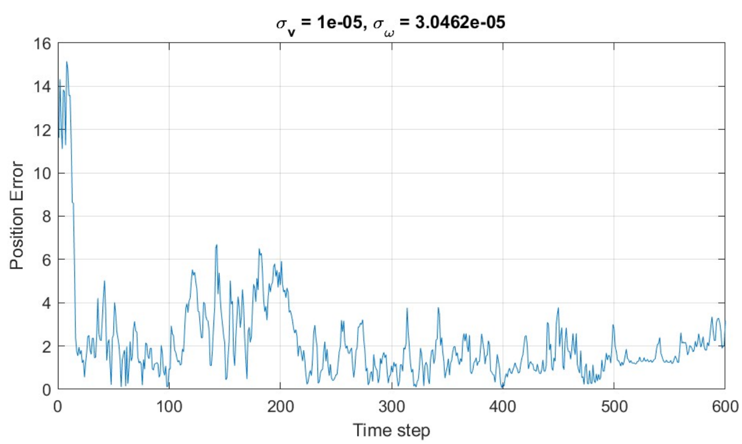
- Too large



- Too small



- Well-tuned



D)

Yes, it's possible to tune the filter to improve the accuracy of estimates for velocity, heading, and turn-rate. That is what techniques like EKF or UKF are commonly used for. There can be conflicts in tuning the filter for different parts of the true trajectory. Parameters that work well for estimating states during straight-line motion might not perform optimally during turns.

In my case, it seems like when the vehicle approaches the sensors, the measurement noise impacts the filter estimates more. Ideally, one would like to trust the motion model more than the measurement model when approaching the sensor positions.

Ideally, the filter parameters should be adaptable to different parts of the trajectory to achieve optimal performance throughout. However, this may not always be feasible due to the complexity of the motion model/measurement model and the limitations of the filter.