Solution to analysis in Home Assignment 3

Yuhong Zhou + yuhong

Analysis

In this report I will present my independent analysis of the questions related to home assignment 1. I have discussed the solution with Xinying Wang but I swear that the analysis written here are my own.

1 Approximations of mean and covariance

(a) We use a sufficient number of samples : N = 1000, such that our approximated sample mean and covariance are accurate and reliable. The result show as in table 1.

Table 1: Approximated Mean and Covariance Matrix

Scenario	Mean	Covariance Matrix
Scenario 1	$\begin{matrix} 0.1978 \\ 1.3745 \end{matrix}$	$\begin{matrix} 0.0018 & 0.0015 \\ 0.0015 & 0.0061 \end{matrix}$
Scenario 2	$\begin{matrix} 2.3267 \\ 2.3551 \end{matrix}$	$\begin{matrix} 0.0562 & 0.0104 \\ 0.0104 & 0.0020 \end{matrix}$
Scenario 3	$\begin{matrix} -0.5943 \\ 2.1513 \end{matrix}$	$\begin{matrix} 0.0097 & -0.0111 \\ -0.0111 & 0.0148 \end{matrix}$

(b) For each state density, we compute the approximated mean and covariance analytically using the type of density approximations that are used in EKF, UKF, and CKF, respectively. The computations are very similar to the computation of the prediction step in these three types of non-linear Kalman filter. The result show as in table 2.

Table 2: Approximated Mean and Covariance by EKF, UKF and CKF

Method	Mean	Covariance Matrix	
Scenario 1			
EKF	0.1974	0.0017	0.0015
	1.3734	0.0015	0.0060
UKF	0.1983	0.0017	0.0015
	1.3743	0.0015	0.0059
CKF	0.1983	0.0017	0.0015
	1.3743	0.0015	0.0059
Scenario 2			
EKF	2.3562	0.0500	0.0100
	2.3562	0.0100	0.0020
UKF	2.3269	0.0600	0.0108
	2.3550	0.0108	0.0020
CKF	2.3265	0.0566	0.0105
	2.3550	0.0105	0.0020
Scenario 3			
EKF	−0.5880	0.0092	−0.0111
	2.1588	−0.0111	0.0148
UKF	−0.5949	0.0099	−0.0112
	2.1524	−0.0112	0.0151
CKF	−0.5948	0.0097	−0.0112
	2.1523	−0.0112	0.0150

(c) The Figure 1.1 shows the result for the state density of scenario 1. The Figure 1.2 shows the result for the state density of scenario 2. The Figure 1.3 shows the result for the state density of scenario 3.

In Figure 1.1, we can find that all three types of density approximation perform well. The approximated means and covariance are very close to the sample value. In Figure 1.2 and Figure 1.3, the UKF and CKF method still perform well. But the EKF method performs much worse than UKF and CKF. The mean of EKF is away from the sample mean. The EKF covariance is much closer to the sample than the sample covariance.

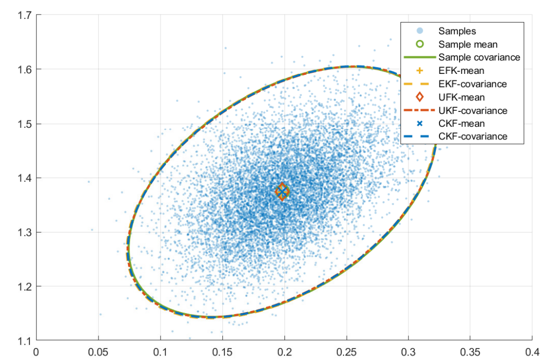

Figure 1.1: Scenario 1.

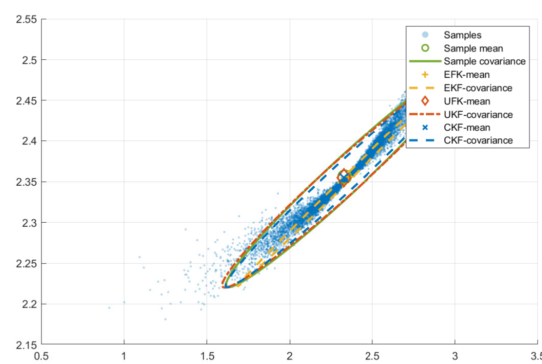

Figure 1.2: Scenario 2.

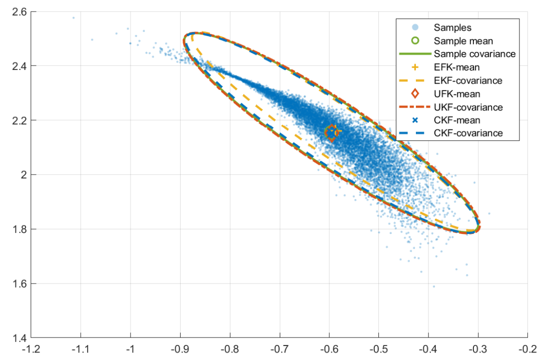

Figure 1.3: Scenario 3.

(d) In Figure 1.1, we can find that all three types of density approximation perform well. The approximated means and covariance are very close to the sample value. In Figure 1.2 and Figure 1.3, the UKF and CKF method still perform well. But the EKF method performs much worse than UKF and CKF. The mean of EKF is away from the sample mean. The EKF covariance is much closer to the sample than the sample covariance.

For EKF method, it handles the non-linear model by linearizing the system model around the current state estimate. But if the model is highly non-linear such as in Figure 1.2 and Figure 1.3, the linearized model can not accurately represent the true system dynamics, leading to degraded filter performance. The UKF and CKF method perform well for highly non-linear models because they do not need to linearize the model instead of using sigma point method.

The trade-off when using UKF (or CKF) compared with EKF lies in the computational complexity versus accuracy. UKF and CKF method often provide more accurate estimations compared to EKF, especially for highly nonlinear systems. But they are more computationally intensive.

As an engineer, the choice of the filter depends on the specific application and the characteristics of the system being modeled. I need to balance the trade-off between computational complexity and accuracy. If the we need it to be highly accuracy, the UKF or CKF would be the first choice. If the model is not highly non-linear and we do not need very high accuracy, the EKF would be a good choice. Since it is much faster and not so computationally intensive.

2 Non-linear Kalman filtering

(a) The result of the Case 1 with three different non-linear Kalman filters shown as bellow.

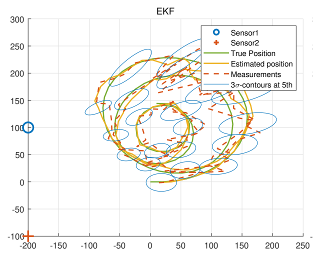

Figure 2.1: Case 1-EKF.

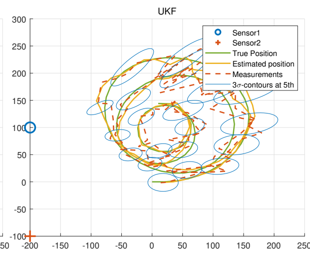

Figure 2.2: Case 1-UKF.

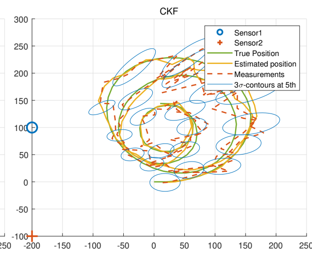

Figure 2.3: Case 1-CKF.

(b) The result of the Case 2 with three different non-linear Kalman filters shown as in Figure 2.4, Figure 2.5 and Figure 2.6.

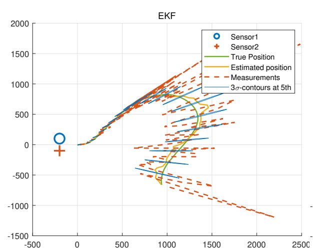

Figure 2.4: Case 2-EKF.

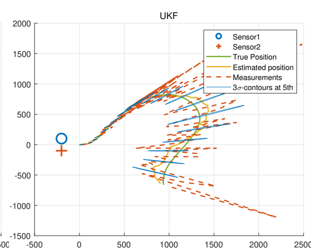

Figure 2.5: Case 2-UKF.

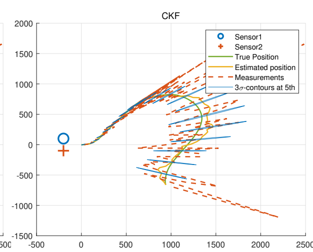

Figure 2.6: Case 2-CKF.

The result of the Case 3 with three different non-linear Kalman filters shown as in Figure 2.7, Figure 2.8 and Figure 2.9.

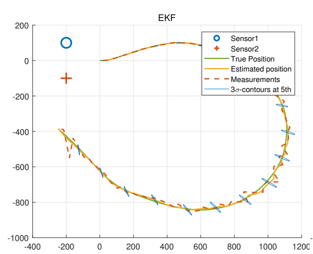

Figure 2.7: Case 3-EKF.

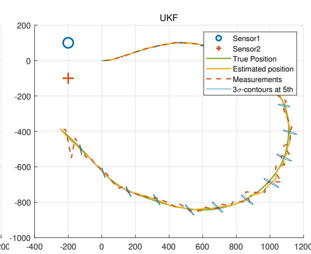

Figure 2.8: Case 3-UKF.

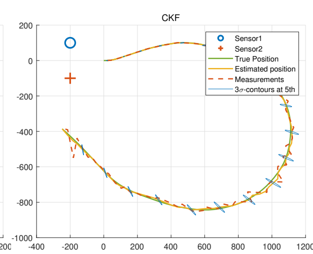

Figure 2.9: Case 3-CKF.

In Case 1 and Case 2, the measurement noise is large which makes the measurements fit bad with the true positions. However, for Case 3, since the noise is much smaller, the measurements perform better than both Case 1 and Case 2. In all three cases, the estimated positions product by three different non-linear Kalman filters fit the real trajectories well. Since they all within the 3σ region. While the result in Case 3 is much better than both Case 1 and Case 2.

Since the estimated positions of all three cases are all in the 3σ region, we can say that the error covariances represent the uncertainty well. However, it is very hard to say which type of non-linear Kalman fiter behaves better in the single case. Since their performances are very close in Figure of the single case.

(c) The result of the three cases are illustrated in Figure 2.10, Figure 2.11 and 2.12. The mean and covariance derived from the estimation error of the Extended Kalman Filter (EKF) vastly exceed those obtained from the Unscented Kalman Filter (UKF) and Cubature Kalman Filter (CKF). This stark contrast implies that the EKF underperforms significantly when compared to the UKF and CKF.

In all figures, we can find that the histogram can not fit with the Gaussian distributions. Since we are handling the non-linear model, the Gaussion distribution will not be a Gaussion distribution after the transformation.

For the EKF, there is not any obvious difference between the histograms of the errors of the x position and their counterparts with respect to the y position. While for UKF and CKF, we can find that the histogram of the errors of the x position is much closer to the Gaussian distribution than the the histogram of the errors of the y position. And the the error of the x position have much smaller mean and covariance than the one of the y position. This could be due to the sensors sharing the same x-position but having varying y-positions. Such an arrangement might amplify the measurement error in the y-position.

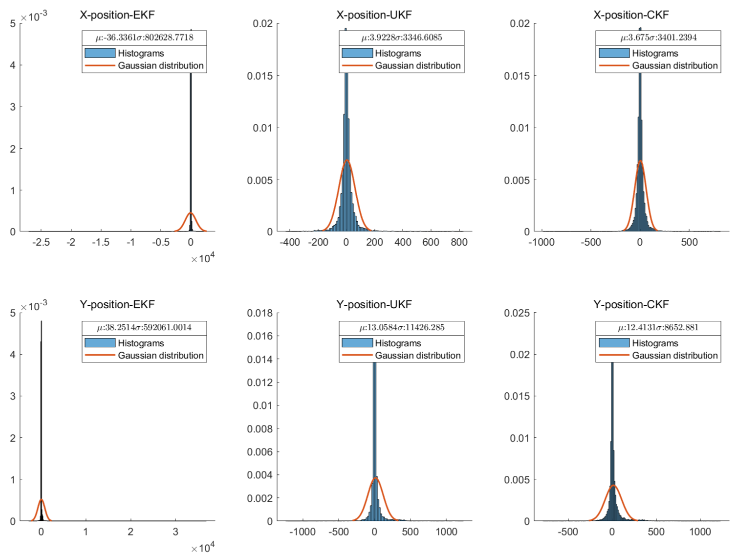

Figure 2.10: Scenario 1.

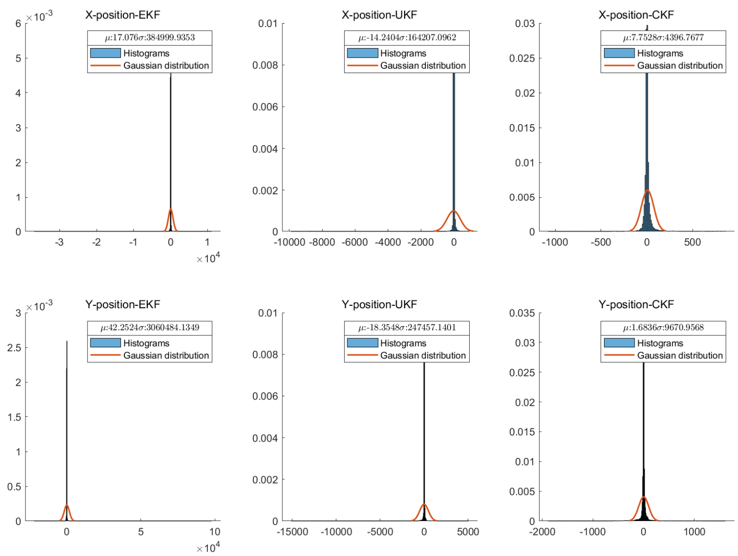

Figure 2.11: Scenario 2.

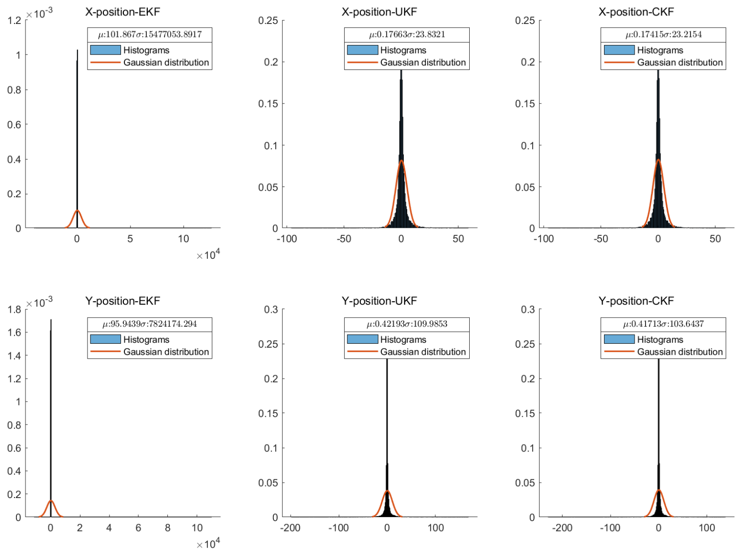

Figure 2.12: Scenario 3.

3 Tuning non-linear filters

(a) **i) Making the process noise very large**

Only increase σ_v

After tuning $\sigma_v = \sigma_v * 1000$, we can find that the result becomes very noisy. The estimate result becomes really bad. The result shown as bellow.

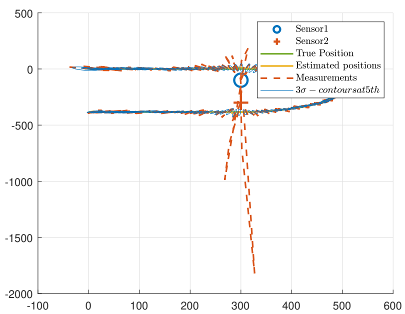

Figure 3.1: Unchanged.

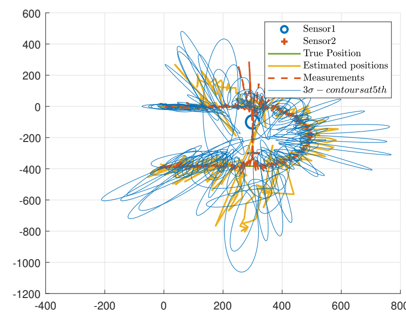

Figure 3.2: $\sigma_v * 1000$.

Only increase σ_w

After tuning $\sigma_w = \sigma_w * 1000$, we can find that we can still get a well performed result. The result shown as bellow.

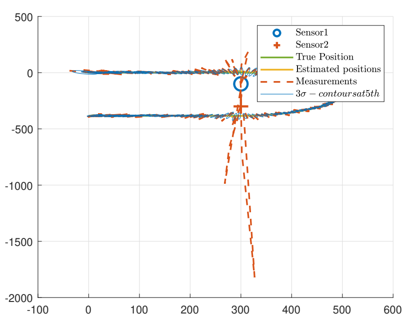

Figure 3.3: Unchanged.

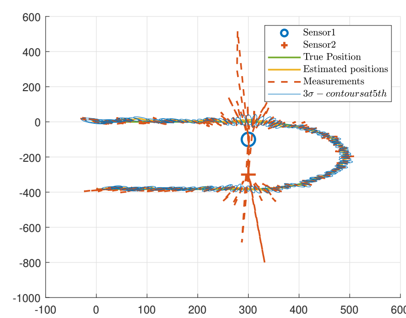

Figure 3.4: $\sigma_w * 1000$.

Increase both σ_v & σ_w

After tuning $\sigma_v = \sigma_v * 1000$, $\sigma_w = \sigma_w * 1000$, we will only get an anomalous, unrealistic and unreliable result. The result shown as bellow.

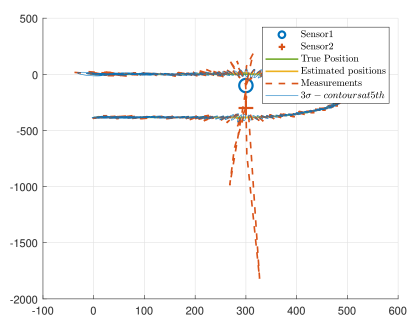

Figure 3.5: Unchanged.

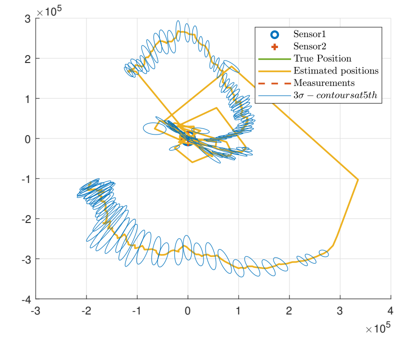

Figure 3.6: $\sigma_v = \sigma_v * 1000$, $\sigma_w = \sigma_w * 1000$.

i) Making the process noise very small

Only decrease σ_v

After tuning $\sigma_v = \sigma_v * 0.001$, we will get a well performed result. The result shown as bellow.

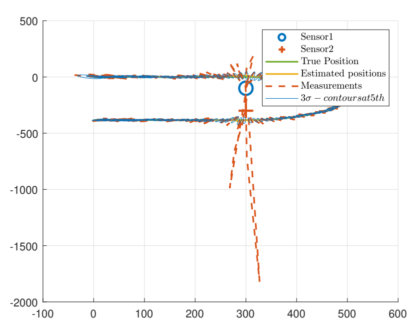

Figure 3.7: Unchanged.

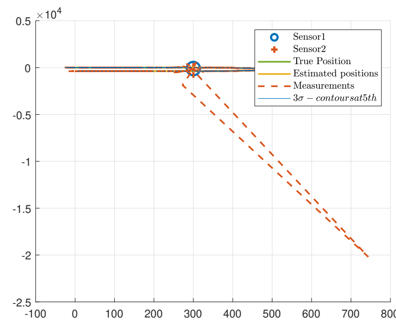

Figure 3.8: $\sigma_v = \sigma_v * 0.001$.

Only decrease σ_w

After tuning $\sigma_w = \sigma_w * 0.001$, the estimated result is good for the straight line before the turn. Then we can find that the estimated turning radius is much larger than it actually is. The result shown as bellow.

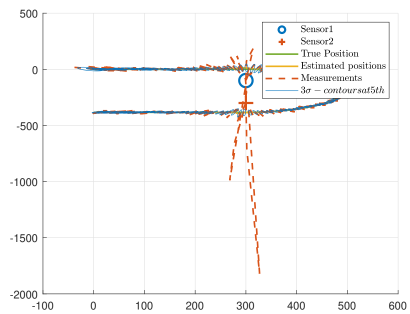

Figure 3.9: Unchanged.

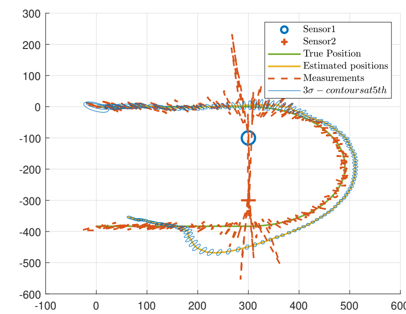

Figure 3.10: $\sigma_w = \sigma_w * 0.001$.

Decrease σ_v & σ_w

After tuning $\sigma_v = \sigma_v * 0.001, \quad \sigma_w = \sigma_w * 0.001$, the result show that the filter estimate the straight line well. But it can not handle the turn, the estimated result of the turn is really bad. The estimated turn is sharper that it actually is. The result shown as bellow.

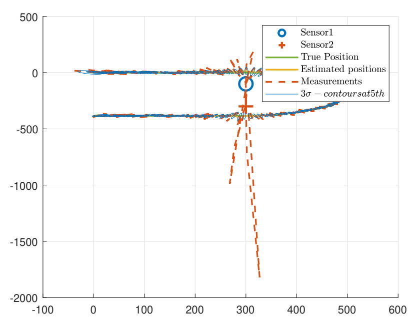

Figure 3.11: Unchanged.

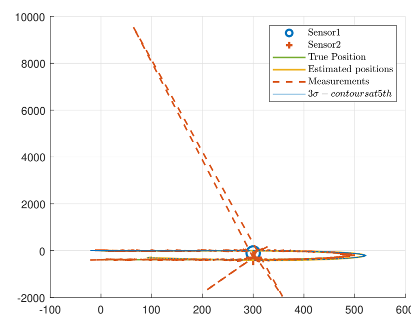

Figure 3.12: $\sigma_v = \sigma_v * 0.001, \quad \sigma_w = \sigma_w * 0.001$.

(b) Tuning $\sigma_v = 1 * 0.1, \quad \sigma_w = \frac{\pi}{180} * 0.16$, we can get good position estimates for the whole sequence. The result shown as bellow.

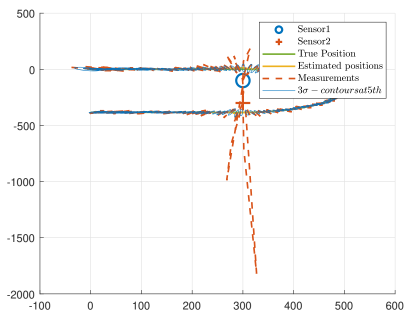

Figure 3.13: Unchanged.

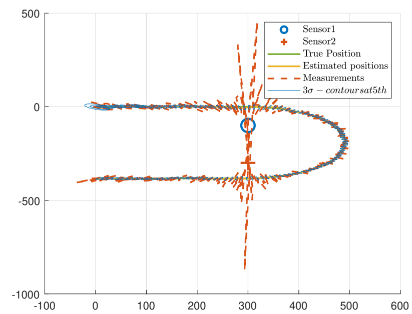

Figure 3.14: $\sigma_v = 1 * 0.1, \quad \sigma_w = \frac{\pi}{180} * 0.16$.

(c) The results and the position of three different process noise settings: too large, too small, and well-tuned shown as the figures bellow. It is obvious that the well-tuned filter performs best and has the minimum position errors. It means the well-tuned filter has a fairly good performance.

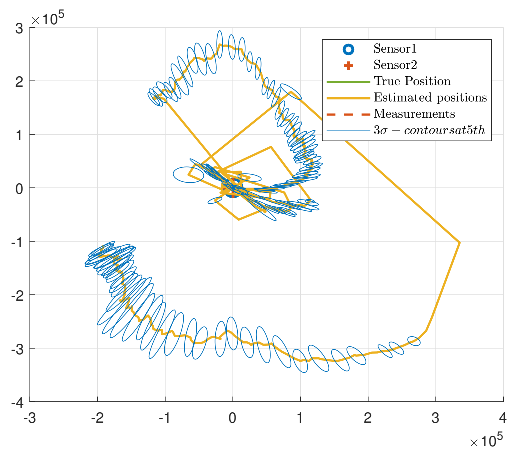

Figure 3.15: Process noise settings: too large.

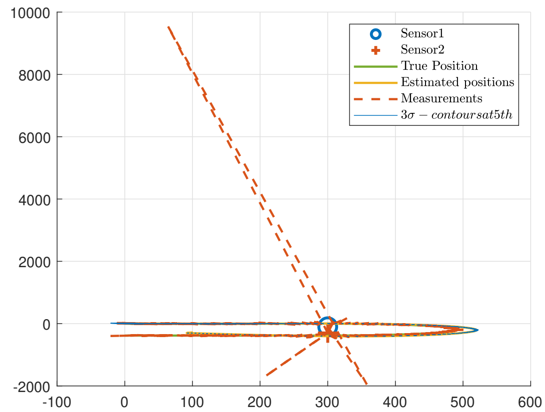

Figure 3.16: Process noise settings: too small.

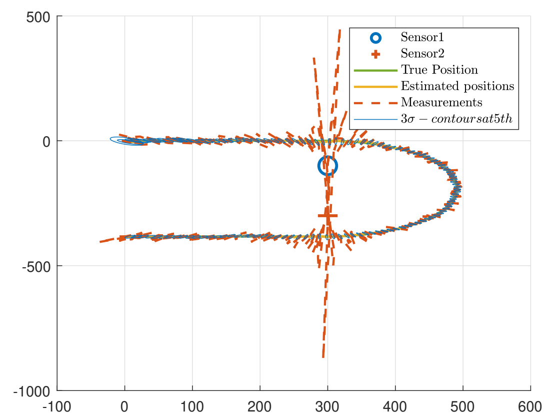

Figure 3.17: Process noise settings: well-tuned.

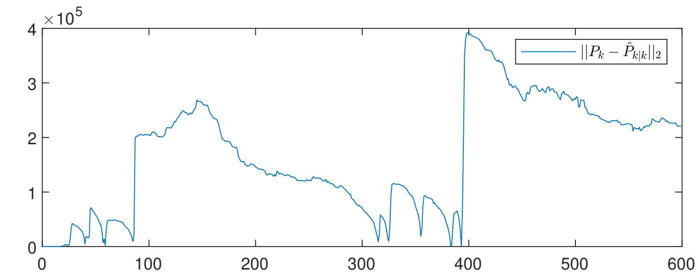

Figure 3.18: $||P_k - \hat{P}_{K|K}||_2$*: too large.*

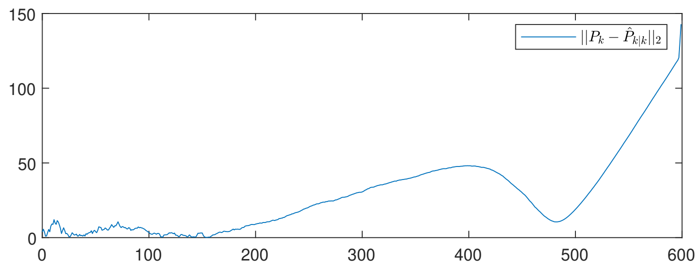

Figure 3.19: $||P_k - \hat{P}_{K|K}||_2$*: too small.*

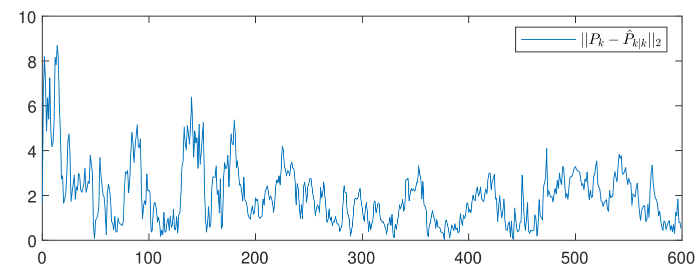

Figure 3.20: $||P_k - \hat{P}_{K|K}||_2$*: well-tuned.*

(d) It seems to be impossible to tune the filter so that you have accurate estimates of those three states for the whole sequence. Since the true trajectory may exhibit different behaviors, such as straight-line and turn, each requiring different parameter settings for accurate estimation.

Yes, there exits a conflict. Optimal parameter settings for estimating velocity, heading, and turn rate in straight-line motion conflict with the requirements for turning. For example, we need the small σ_v and σ_w for the straight-line motion, while we will need a large σ_w to capture the changes and uncertainties associated with the turning.
For the transitions from straight to turning and from turning to straight, I think we will need to set a large value for the parameters. Since we need to capture more changes and uncertainties for the transitions.

In summary, obtaining accurate estimates of velocity, heading, and turn-rate for the entire sequence is desirable but very challenging to achieve. Because of the dynamic nature of the system and the diverse characteristics of the true trajectory. Therefore, we need to balance the parameter tuning to accommodate different trajectory segments and transitions.