Solution to analysis in Home Assignment 3

Fengyuan Liu
CID:liufe

Analysis

In this report I will present my independent analysis of the questions related to home assignment X. I have discussed the solution with NAME1, NAME2 and NAME3 but I swear that the analysis written here are my own.

1 Approximations of mean and covariance

a)

Mean and covariance for state density p1(x): Mean: [0.1981; 1.3742]
Covariance: [0.0017 0.0014; 0.0014 0.0058]
Mean and covariance for state density p2(x): Mean: [2.3271; 2.3552]
Covariance: [0.0554 0.0102 ; 0.0102 0.0020]
Mean and covariance for state density p3(x): Mean: [-0.5962; 2.1536]
Covariance: [0.0100 -0.0114; -0.0114 0.0154]

b)

Firstly I generate 10,000 samples for each of the three state densities with given mean and covariance then compute measurements using the dual bearing measurement function with added random noise. After that, use the approximate mean and covariance function to calculate the mean and covariance of the generated measurements.

The EKF uses the linearized measurement model to calculate the predicted measurements and their covariance using the Jacobian of the measurement function. It then compares the predicted measurement with the actual measurement to update the state estimate.

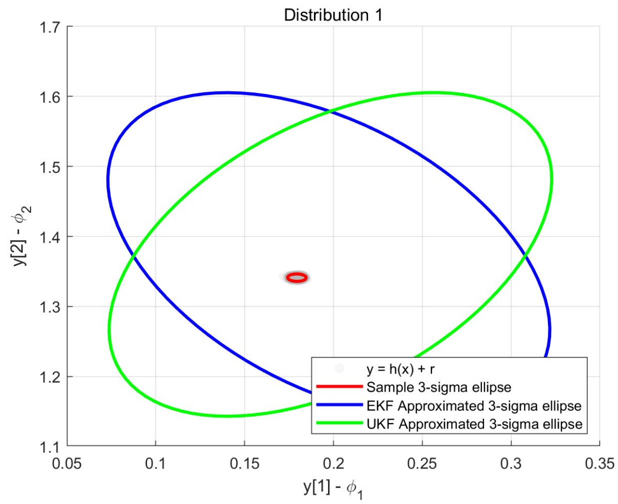
The UKF generates sigma points using the mean and covariance of the state, applies the measurement function to transform these points, calculates the predicted measurement mean and covariance, and then compares with the actual measurement to update the state estimate.

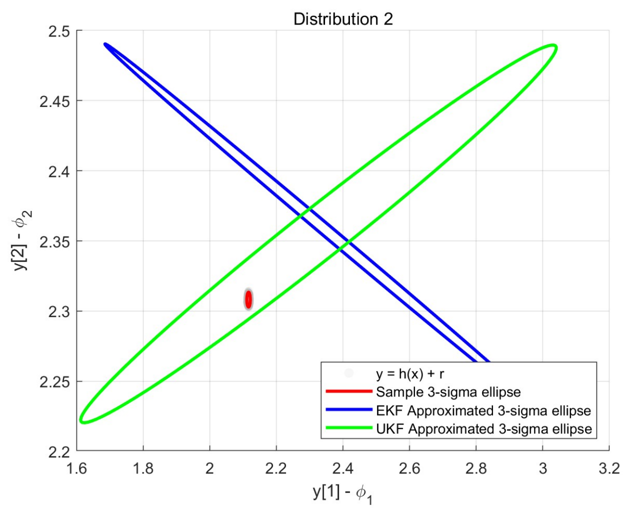
The CKF generates sigma points using the mean and covariance of the state, applies the measurement function to transform these points, calculates the predicted measurement mean and covariance, and then compares with the actual measurement to update the state estimate.

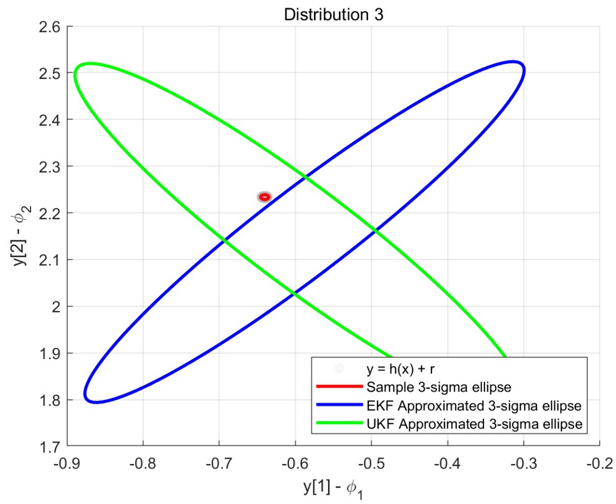
The results are presented with mean and covariance for each state density, filter method, and approximation technique. The approximate mean and covariance are computed for the generated measurements, while the filter techniques estimate the state densities using the measurements.

EKF: Mean and covariance for state density p1(x): Mean: [0.1974; 1.3734]
Covariance: [0.0060 0.0015; 0.0015 0.0017]
Mean and covariance for state density p2(x): Mean: [2.3562; 2.3562]
Covariance: [0.0404 -0.0404; -0.0404 0.0404]
Mean and covariance for state density p3(x): Mean: [-0.5880; 2.1588]
Covariance: [0.0148 -0.0111; -0.0111 0.0092]
UKF: Mean and covariance for state density p1(x): Mean: [0.1983 1.3743]
Covariance: [4.1565 0.0029; 0.0029 4.1565]
Mean and covariance for state density p2(x): Mean: [2.3265 2.3550]
Covariance: [0.1158 0.0228; 0.0228 0.0066]
Mean and covariance for state density p3(x): Mean: [-0.5948 2.1523]
Covariance: [22.5842 -0.0207; -0.0207 22.7411]
CKF: Mean and covariance for state density p1(x): Mean: [0.6532; 1.4200]
Covariance: [0.7856 0.4685; 0.4685 0.3569]
Mean and covariance for state density p2(x): Mean: [0.6047; 2.2239]
Covariance: [4.7373 0.1183; 0.1183 0.2328]
Mean and covariance for state density p3(x): Mean: [-0.9642; 1.9785]
Covariance: [1.0855 -0.6326; -0.6326 0.7928]

c)d)


Distribution 1
y[2] - ϕ_2
y[1] - ϕ_1
1.7
1.6
1.5
1.4
1.3
1.2
1.1
0.05
0.1
0.15
0.2
0.25
0.3
0.35
y = h(x) + r
Sample 3-sigma ellipse
EKF Approximated 3-sigma ellipse
UKF Approximated 3-sigma ellipse


Distribution 2
y[2] - ϕ_2
y[1] - ϕ_1
2.5
2.45
2.4
2.35
2.3
2.25
2.2
1.6
1.8
2
2.2
2.4
2.6
2.8
3
3.2
y = h(x) + r
Sample 3-sigma ellipse
EKF Approximated 3-sigma ellipse
UKF Approximated 3-sigma ellipse



The center of the sample 3-sigma ellipse (red) coincide with the centers of the EKF (blue) and UKF (green) 3-sigma ellipses. The centers of the UKF ellipse is closer to the center of the sample ellipse, with the better approximations of the sample mean.

The ellipses represent the 3-sigma confidence regions computed using the sample covariance (red), EKF covariance (blue), and UKF covariance (green). The size and shape of the UKF ellipse is closer to the sample ellipse, with the better approximations of the sample covariance.

EKF performance: The EKF can perform worse than the UKF, especially in cases where the system is highly nonlinear. This is because EKF linearizes the nonlinear function using a first-order Taylor series expansion, which may result in inaccurate approximations for highly nonlinear systems. In such cases, the EKF's estimated means and covariances may deviate significantly from the true distribution.

UKF performance: The UKF, on the other hand, provides a more accurate approximation for nonlinear systems by using a set of deterministic sigma points to represent the probability distribution. This allows the UKF to capture the true mean and covariance of the nonlinearly transformed Gaussian distribution more accurately than the EKF.

Trade-offs: The main trade-off when using the UKF compared to the EKF is the increased computational complexity. The UKF requires more computations to generate sigma points, propagate them through the nonlinear function, and compute the resulting means and covariances. This can be a concern in real-time applications or when working with limited computational resources.

As an engineer, the choice between EKF and UKF depends on the specific application and its requirements. Some factors to consider include:

1.Linearity of the system: If the system is nearly linear or has mild nonlinearities, the EKF might provide a satisfactory approximation with lower computational complexity. In cases where the system is highly nonlinear, the UKF is likely to provide better results.

2.Computational resources: If computational resources are limited or real-time processing is crucial, the EKF might be a more suitable choice due to its lower complexity. However, if accuracy is more important and there are sufficient computational resources, the UKF would be a better choice.

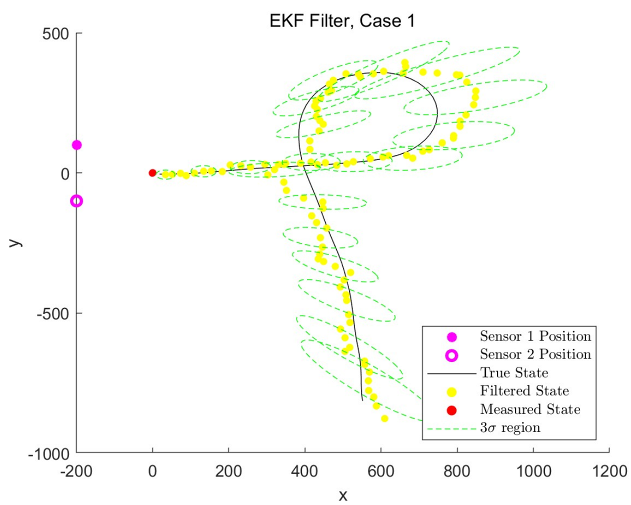
3.Robustness: The UKF is generally more robust to modeling inaccuracies and uncertainties, making it a preferable choice when dealing with complex systems or when the system model is not well known.

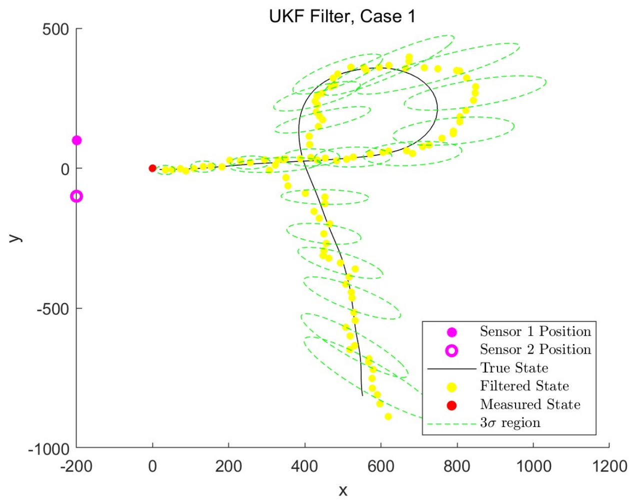
Ultimately, the choice between EKF and UKF depends on the specific application, the nature of the system, and the available resources. In some cases, it may be beneficial to implement both filters and compare their performance in the context of the specific application to make an informed decision.

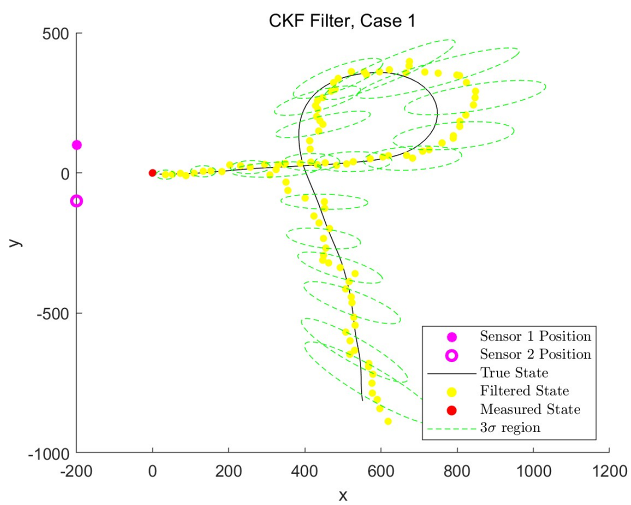
2 Non-linear Kalman filtering

a)b)

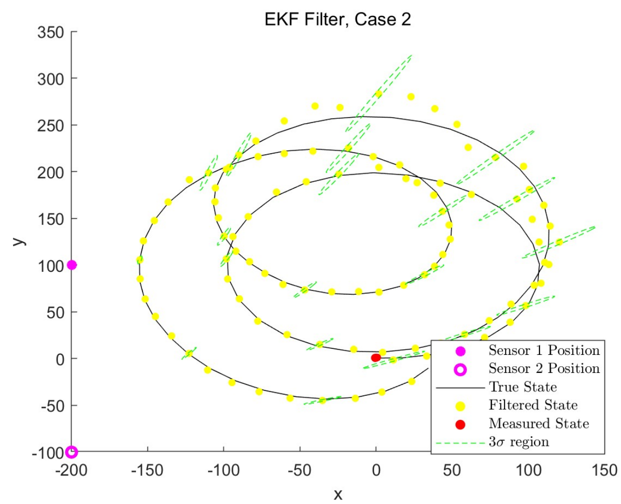
case 1



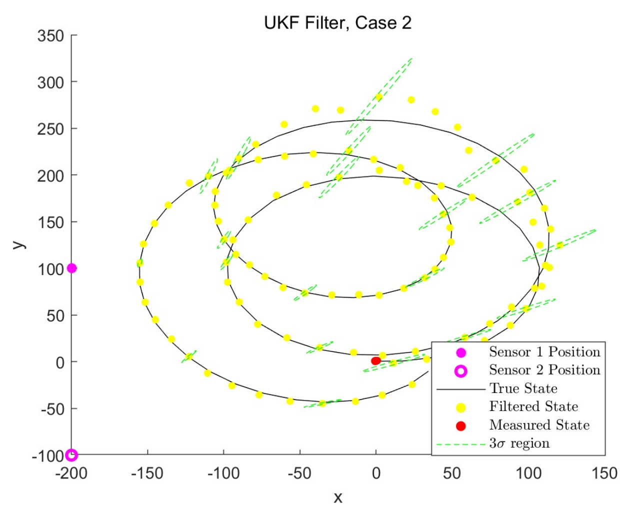

UKF Filter, Case 1
500
0
-500
-1000
y
-200
0
200
400
600
800
1000
1200
x
Sensor 1 Position
Sensor 2 Position
True State
Filtered State
Measured State
3σ region


CKF Filter, Case 1
500
0
-500
-1000
y
-200
0
200
400
600
800
1000
1200
x
Sensor 1 Position
Sensor 2 Position
True State
Filtered State
Measured State
3σ region

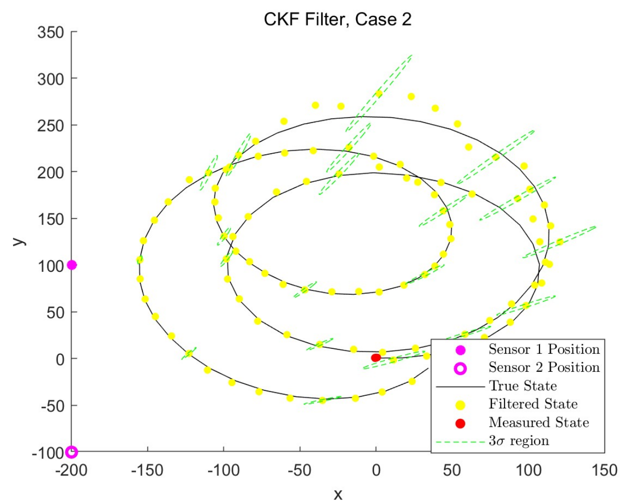
case 2



EKF Filter, Case 2
350
300
250
200
150
100
50
0
-50
-100
-200
-150
-100
-50
0
50
100
150
x
y
Sensor 1 Position
Sensor 2 Position
True State
Filtered State
Measured State
3σ region

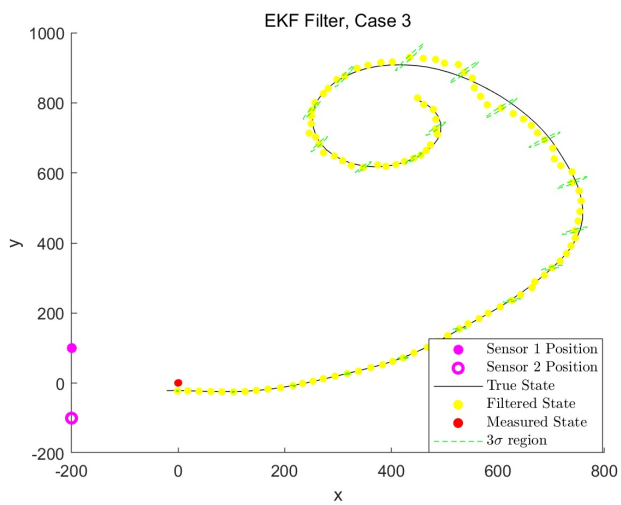


UKF Filter, Case 2
350
300
250
200
150
100
50
0
-50
-100
-200
-150
-100
-50
0
50
100
150
x
y
Sensor 1 Position
Sensor 2 Position
True State
Filtered State
Measured State
3σ region

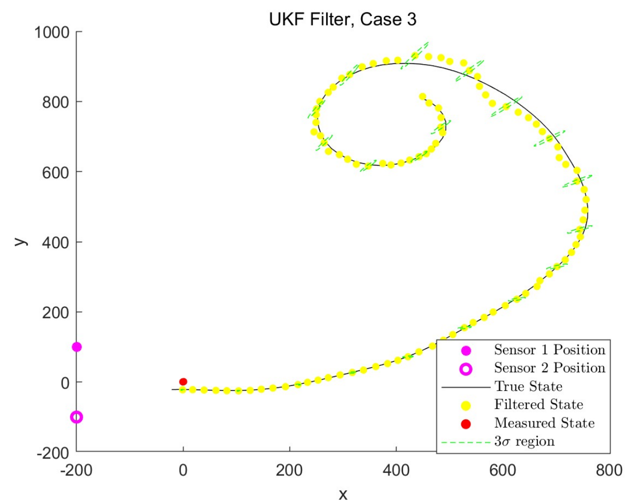


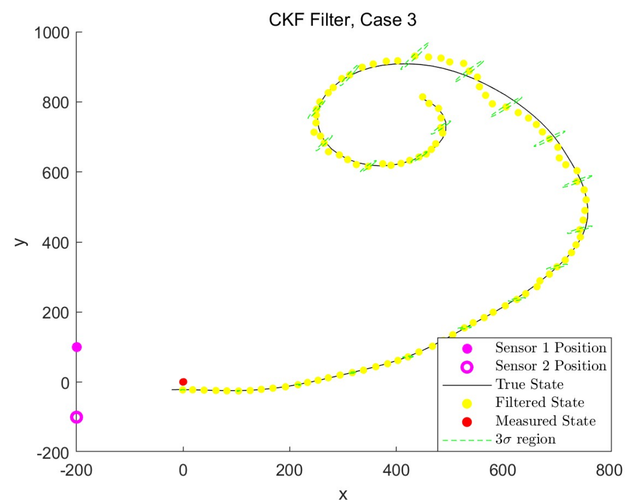
CKF Filter, Case 2
350
300
250
200
150
100
50
0
-50
-100
-200
-150
-100
-50
0
50
100
150
x
y
Sensor 1 Position
Sensor 2 Position
True State
Filtered State
Measured State
3σ region

case 3



EKF Filter, Case 3
1000
800
600
400
200
0
-200
-200
0
200
400
600
800
x
y
Sensor 1 Position
Sensor 2 Position
True State
Filtered State
Measured State
3σ region





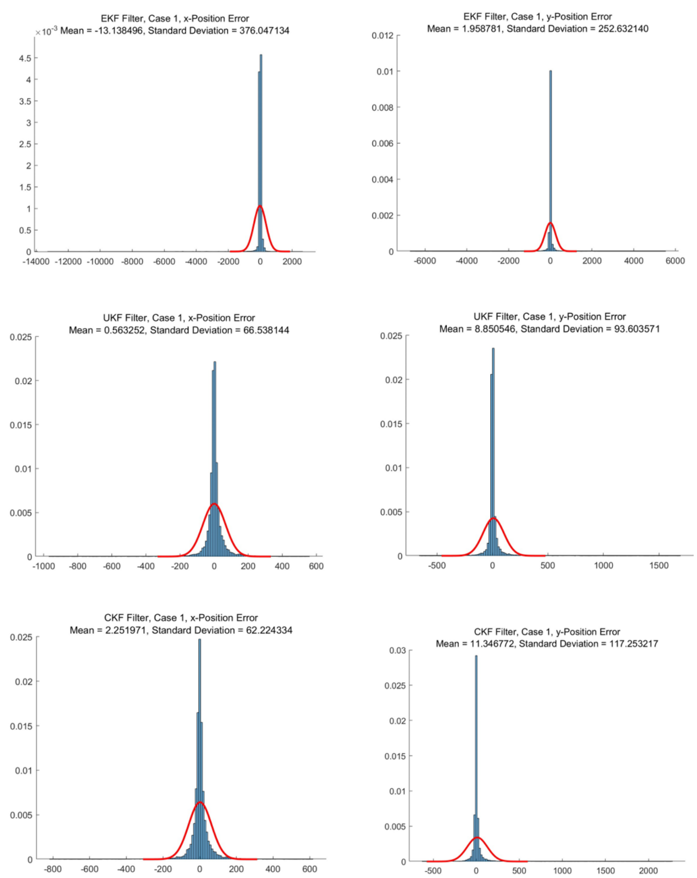
Case 1: Perfect initial condition and correct model dynamics. In this case, both the initial conditions and model dynamics are accurate, which means the filter performance will be optimal. The error covariances are minimal and accurately represent the uncertainty in the system. The filter converge quickly and provide precise estimates of the state variables.

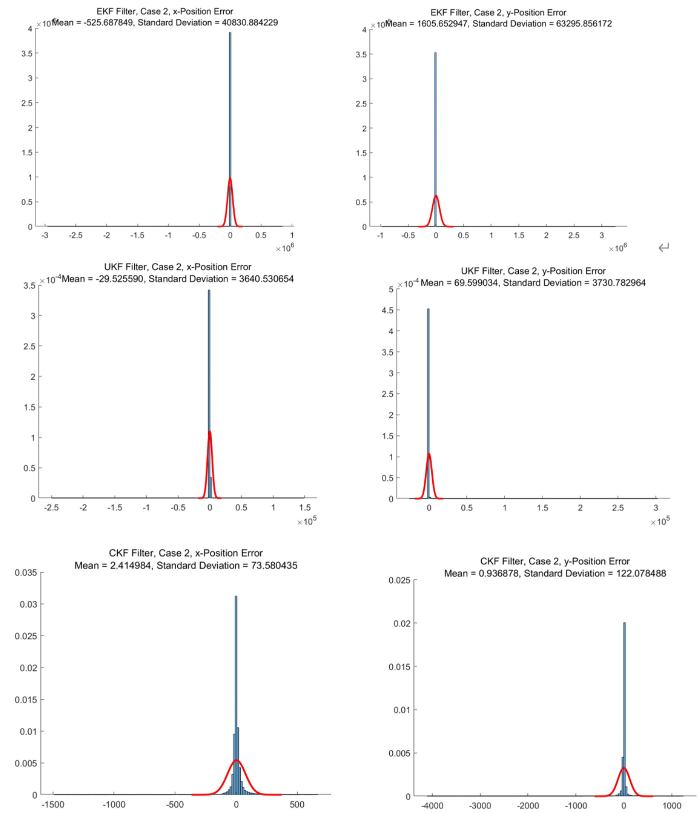
Case 2: Imperfect initial condition but correct model dynamics. In this scenario, the initial conditions are not accurate, but the model dynamics are correct. The filter will initially have larger error covariances due to the uncertainty in the initial conditions. However, as the filter progresses, it adapt and converge to the true state by leveraging the accurate model dynamics. The error covariances represent the uncertainty well, and they decrease as the filter converges, reflecting the reduced uncertainty in the state estimates.

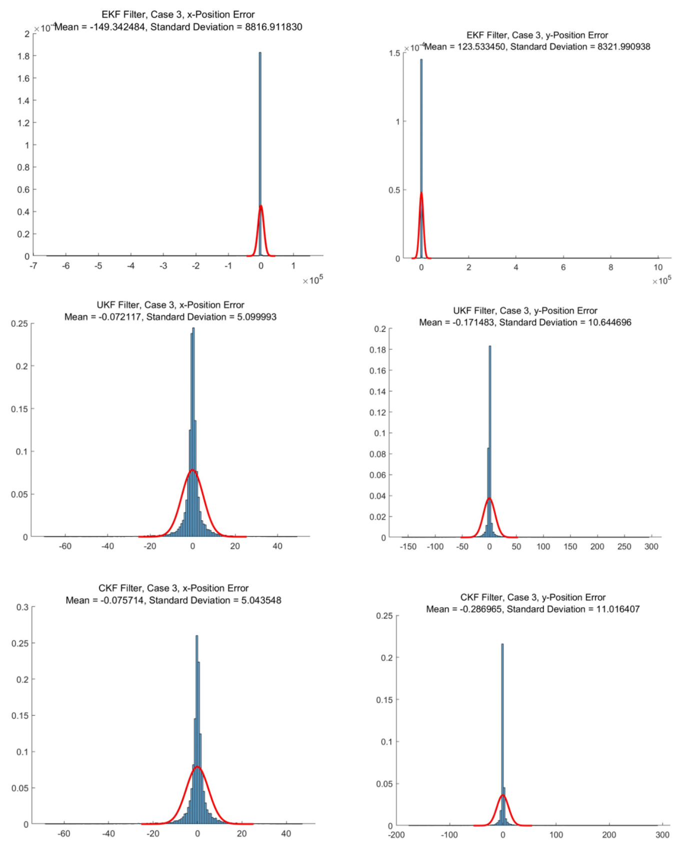
Case 3: Perfect initial condition but incorrect model dynamics. In this case, the initial conditions are perfect, but the model dynamics are not accurate. The filter performance is compromised as it relies on incorrect model dynamics to estimate the state variables. The error covariances increase over time as the filter struggles to provide accurate state estimates. The uncertainty in the system is not represented well, and the error covariances will not be reliable indicators of the true uncertainty.

In summary, the filter performance and error covariance representation of uncertainty are greatly influenced by the accuracy of initial conditions and model dynamics. In a scenario with perfect initial conditions and correct model dynamics, the filter performs optimally, and error covariances represent the uncertainty well. However, when either of these factors is compromised, the filter performance degrades, and the error covariances may not accurately represent the true uncertainty in the system.

c)




EKF Filter, Case 2, x-Position Error
Mean = -525.687849, Standard Deviation = 40830.884229
EKF Filter, Case 2, y-Position Error
Mean = 1605.652947, Standard Deviation = 63295.856172
UKF Filter, Case 2, x-Position Error
Mean = -29.525590, Standard Deviation = 3640.530654
UKF Filter, Case 2, y-Position Error
Mean = 69.599034, Standard Deviation = 3730.782964
CKF Filter, Case 2, x-Position Error
Mean = 2.414984, Standard Deviation = 73.580435
CKF Filter, Case 2, y-Position Error
Mean = 0.936878, Standard Deviation = 122.078488



EKF is based on the first-order linearization of the system dynamics and measurement models. Due to this linear approximation, EKF may not perform well when dealing with highly nonlinear systems. In contrast, UKF and CKF use a deterministic sampling approach to approximate the nonlinear dynamics and measurements, making them more suitable for highly nonlinear systems.

UKF uses a set of sigma points to represent the mean and covariance of the state distribution. These sigma points are transformed through the nonlinear functions, and the posterior mean and covariance are computed based on the transformed sigma points. CKF, on the other hand, uses a cubature rule to compute the integral of the nonlinear functions over the state distribution, which can provide better accuracy than UKF for some systems.

In terms of computational complexity, EKF is generally the least computationally expensive of the three, while UKF and CKF require more computations due to their sampling-based approach. However, the increased computational complexity of UKF and CKF can lead to improved performance in nonlinear systems compared to EKF.

The histograms compare with the fitted Gaussian distribution mainly with the mean and Standard deviation:

Mean: The mean of the histograms should be close to zero, indicating unbiased filters. If the mean is not close to zero, it might suggest that the filter has a systematic bias in the estimation.

Standard deviation: The standard deviation of the histograms should match the standard deviation of the fitted Gaussian distribution, which is derived from the filter's error covariance matrix. This would indicate that the filter's uncertainty representation is accurate.

There are differences between the histograms of the errors for the x position and their counterparts for the y position. These differences may arise due to various factors:

Sensor noise characteristics: If the sensors used for position measurements have different noise characteristics in the x and y directions, this may lead to differences in the estimation errors and their histograms.

Dynamics of the system: The underlying dynamics of the system can influence the errors in the x and y positions differently. If the system has more complex dynamics in one direction, it may be harder for the filter to accurately estimate the position in that direction, leading to larger errors and a different histogram shape.

Initial state uncertainty: If the initial state uncertainty is different for the x and y positions, it can result in different estimation errors and histogram shapes. This can be due to different initial covariance values, or the underlying dynamics affecting one direction more than the other.

Nonlinearity in the system: The presence of nonlinearity in the system dynamics or observation model can lead to different estimation errors in the x and y directions. Some filters, like EKF, may struggle more with highly nonlinear systems, leading to larger errors in one direction compared to the other.

Tuning of the filter: If the filters are not optimally tuned or the parameters are set differently for the x and y directions, this can lead to differences in the estimation errors and their histograms.

3 Tuning non-linear filters

a)b)

If σ_v is very large, while keeping σ_w: the filter will assume there is more uncertainty in the linear velocity. Consequently, it will rely more on the measurements and less on the process model when updating the state estimates. This may result in smoother state estimates if the actual process noise is low, but if the actual process noise is high, it may lead to better tracking of the true state.

If $\sigma_w$is very large, while keeping σ_v : the filter will assume there is more uncertainty in the angular velocity. Similar to the case above, the filter will rely more on the measurements and less on the process model when updating the state estimates. This can affect the filter's ability to track the true state during turns, but the impact on the straight-line portions may not be as significant.

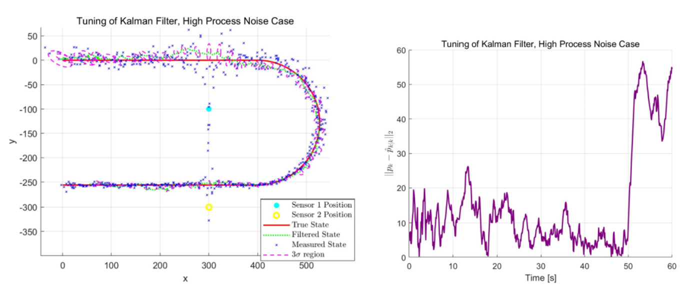
If increase both σ_v and σ_w, the filter will assume there is more uncertainty in both linear and angular velocities. As a result, it will rely more on the measurements and less on the process model when updating the state estimates. This can lead to smoother state estimates if the actual process noise is low, but if the actual process noise is high, it may help the filter better track the true state.

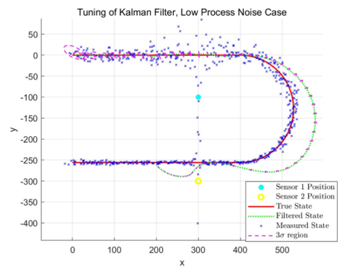
If σ_v is very small, while keeping σ_w: the filter will assume there is less uncertainty in the linear velocity. Consequently, it will rely more on the process model and less on the measurements when updating the state estimates. This may result in more accurate state estimates if the actual process noise is low. However, if the actual process noise is high, the filter might not be able to accurately track the true state, especially during periods of acceleration or deceleration.

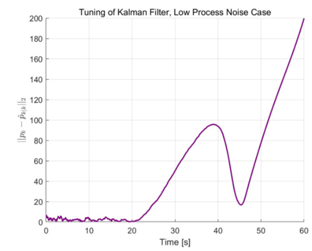
If σ_w is very small, while keeping σ_v: the filter will assume there is less uncertainty in the angular velocity. Similar to the case above, the filter will rely more on the process model and less on the measurements when updating the state estimates. This can lead to more accurate state estimates during turns if the actual process noise is low. However, if the actual process noise is high, the filter might not be able to accurately track the true state during turns.

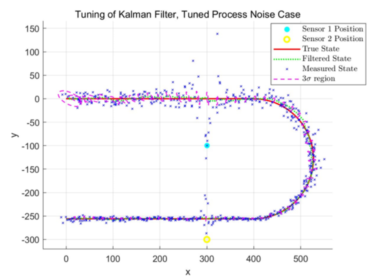
If decrease both σ_v and σ_w, the filter will assume there is less uncertainty in both linear and angular velocities. As a result, it will rely more on the process model and less on the measurements when updating the state estimates. This can lead to more accurate state estimates if the actual process noise is low. However, if the actual process noise is high, the filter might not be able to accurately track the true state, especially during periods of acceleration, deceleration, or turns.

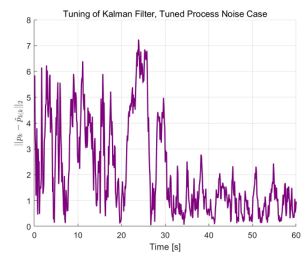
c)











In summary, the first figure shows that higher noise covariances lead to the filter relying more on measurements, resulting in noisy and unrealistic outputs. Errors in position are particularly high during straight trajectories. The second figure demonstrates that with low noise covariances, the model struggles to adapt to velocity and yaw rate changes between straight and curved trajectories, leading to abrupt error increases. In contrast, the third figure reveals that well-tuned noise covariance allows the model to maintain consistent, low errors throughout the entire trajectory, resulting in better filter performance for both straight and curved paths.

d)

It can be challenging to tune the filter to provide accurate estimates of velocity, heading, and turn-rate for the entire sequence. The primary reason is that the system dynamics vary between different parts of the true trajectory, such as straight lines, turns, and transitions between them. The process noise covariance needs to be tuned differently for each scenario, making it difficult to find a single optimal setting that works for all cases.

During straight-line motion, a lower process noise covariance is preferable, as it allows the filter to trust the model more and results in smoother estimates. However, during turns and transitions, the model needs to adapt quickly to the changing dynamics, requiring a higher process noise covariance to trust the measurements more.

There is an inherent conflict between tuning the filter for different parts of the trajectory. The ideal parameter setting for straight-line motion may not be suitable for turns and vice versa. For transitions between straight and turning, or turning to straight, it's even more complex, as the filter needs to balance between trusting the model and the measurements.

In summary, tuning the filter for accurate estimates of velocity, heading, and turn-rate throughout the entire sequence is challenging due to the varying system dynamics. Finding a single optimal process noise covariance setting for all scenarios is difficult, and there may be trade-offs between accuracy in different parts of the trajectory. However, by carefully analyzing the system's behavior and experimenting with different tuning strategies, it may be possible to find a reasonable compromise that works well for most scenarios.