Solution to analysis in Home Assignment 2

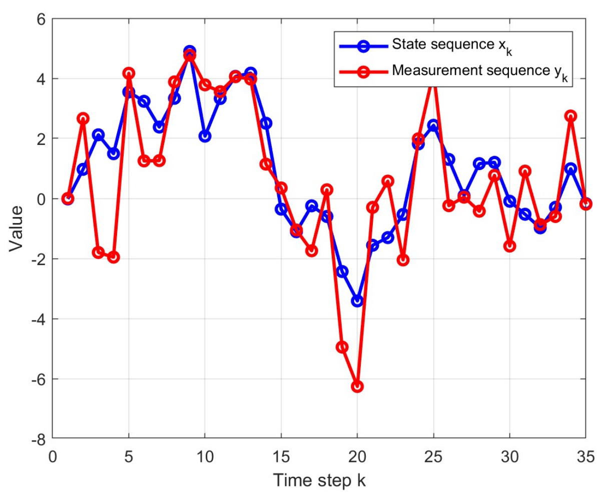
Fengyuan Liu
CID:liufe

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.

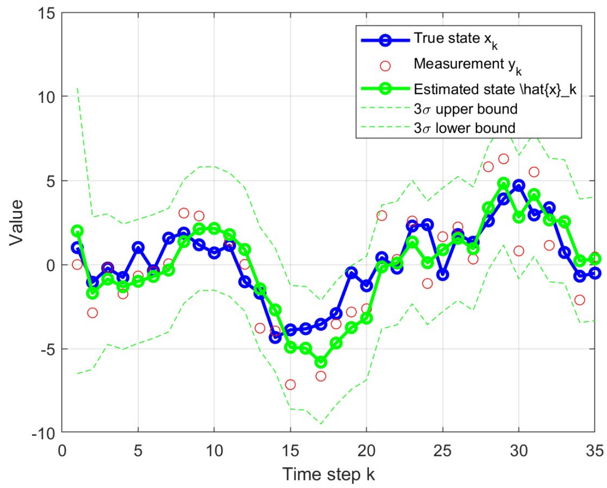
1 A first Kalman filter and its properties

a)



The measurement sequence behaves according to the model. As shown in the figure, the measurement sequence y_k generally follows the state sequence x_k, but it's influenced by the Gaussian noise, making the measurements somewhat dispersed around the true state values.

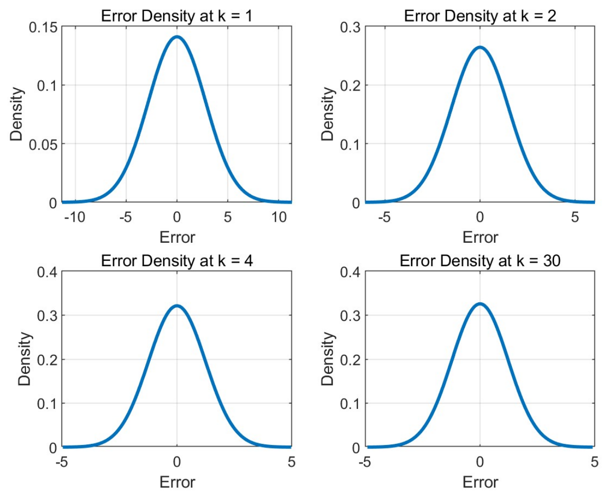
b)



The Kalman filter outputs reasonable state estimates and effectively represents the uncertainty in those estimates through the error covariance.

The estimates provided by the Kalman filter are reasonable because they approximate the true state while taking into account the motion and measurement models and their associated noise variances. The filter is designed to optimally balance the motion model and the measurement model, ensuring that it estimates the true state based on the most recent information available. By comparing the estimated state sequence with the true state sequence in the plot, you can observe that the estimates generally follow the true state closely, despite the noisy measurements.

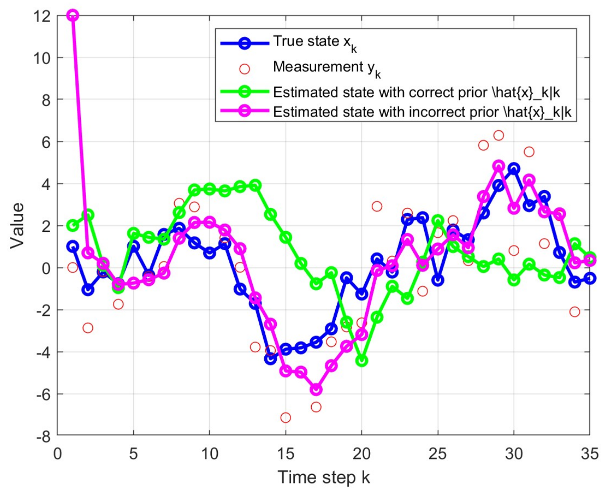
The error covariance in the Kalman filter represents the uncertainty in the state estimates quite well. In the one-dimensional case presented in the previous examples, the covariance reduces to a scalar value, which is the variance $P_est(k)$ at each time step$k$. The square root of the variance gives the standard deviation, which is a measure of the uncertainty in the state estimate. The 3σ bounds plotted in the previous example provide a good visual representation of the filter's uncertainty, indicating that the error covariance represents the uncertainty in the estimates well.



In the second figure, we can see Gaussian densities centered around zero-mean, representing the error distribution of the state estimates at each of the specified time instances. The width of the Gaussian density is determined by the standard deviation (which is derived from the error covariance), and it represents the uncertainty in the state estimates at that time step.

As the Kalman filter processes the measurements, the uncertainty in the state estimates typically decreases, leading to narrower Gaussian densities over time. This indicates that the filter becomes more confident in its estimates as it incorporates more measurements. However, this behavior may vary depending on the specific motion and measurement models, as well as the noise variances.

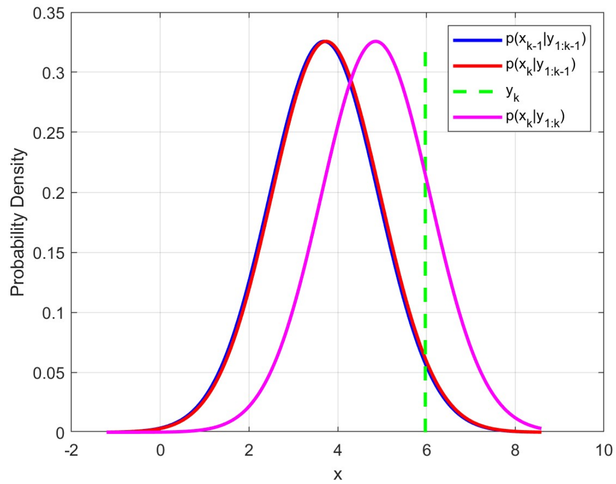
c)



The initial mean for the incorrect prior is set to 12, and the initial covariance is the same as the correct filter (8). In this code it initializes the arrays for the estimates with the incorrect prior and runs the Kalman filter loop. It then plots the true state sequence, measurement sequence, estimated state sequence with the correct prior, and the estimated state sequence with the incorrect prior.

We can observe that the filter with the incorrect prior does not perform as well initially. However, over time, the filter's estimates gradually converge toward the true state sequence as it processes more measurements. This happens because the Kalman filter is designed to adaptively update its estimates based on the measurements and the motion model. As a result, even with an incorrect initial prior, the filter's performance improves over time as it incorporates more information from the measurements.

d)

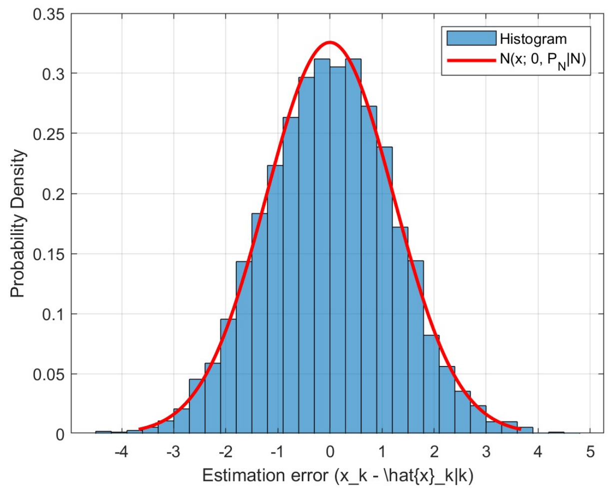


Prediction step: The prediction step advances the state estimate from $k-1$ to k, which is represented by $p(x_k|y_{1:k-1})$. We can see that the Gaussian density for $p(x_k|y_{1:k-1})$ (in red) is shifted from $p(x_{k-1}|y_{1:k-1})$ (in blue) according to the motion model. The uncertainty in the prediction increases due to the process noise (Q).

Update step: The update step incorporates the measurement y_k (green dashed line) to refine the state estimate. The Gaussian density for $p(x_k|y_{1:k})$ (in magenta) represents the updated estimate after taking into account the measurement. The uncertainty in the updated estimate decreases as the filter combines information from both the prediction and the measurement.

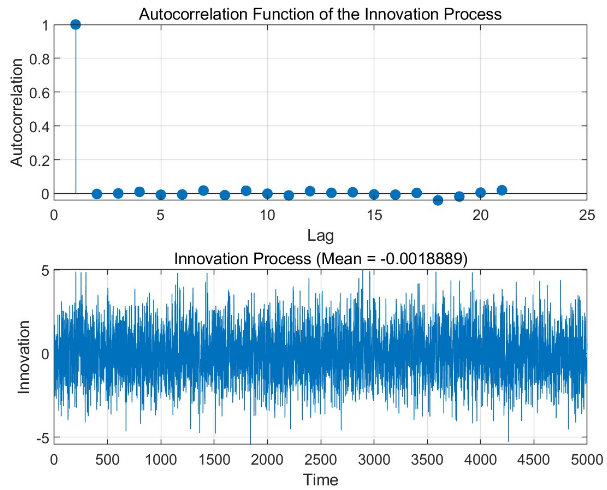
The behavior of the Kalman filter is reasonable because it follows the expected progression of prediction and update steps. In the prediction step, the filter advances the state estimate according to the motion model, and the uncertainty increases due to the process noise. In the update step, the filter refines the state estimate using

e)



In the first task I generate a long true state sequence and the corresponding measurement sequence, filter the long measurement sequence using the Kalman filter, and calculate the estimated mean of the sequence. Then plot a histogram of the estimation error $x_k - \hat{x}_{k|k}$, and compare it to the pdf $N(x; 0, P_{N|N})$, where N is the length of the sequence.

From the results, we can observe that the histogram of the estimation error is well-approximated by the Gaussian density $N(x; 0, P_{N|N})$. This implies that the error covariance $P_{k|k}$ is a good representation of the uncertainty in the state estimates provided by the Kalman filter. The consistency of the filter can be concluded from this result, as the error covariance converges to a constant value and the estimation error follows the expected Gaussian distribution.



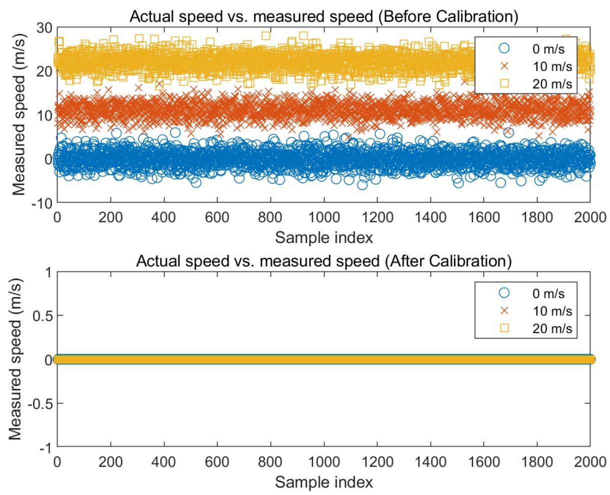
From the plotted autocorrelation function, we can observe that the autocorrelation is close to zero for all lags except for the zero-lag (lag $= 0$), which is equal to 1. This indicates that the innovation process is white noise, as its autocorrelation is nonzero only for the zero-lag. This is consistent with the assumptions of the Kalman filter, which expects the measurement noise to be uncorrelated in time.

The mean of the innovation process is close to zero, which indicates that the filter is unbiased, as there is no systematic error in the state estimates.

In conclusion, the analysis of the innovation process suggests that the Kalman filter is working as expected, as the filter's assumptions of uncorrelated measurement noise and unbiased estimates are satisfied.

2 Tuning a Kalman filter

a)



For each trial (stationary, 10 m/s, and 20 m/s), calculate the mean of the measured speed values (y_k^v).

Since we know the true speeds (0, 10, and 20 m/s) for each trial, we can find the scaling constant C using the relationship:

y_k^v = C * (v_k + r_k^v)

By rearranging the equation, we get:

C = y_k^v / (v_k + r_k^v)

For each trial, calculate the scaling constant C using the mean of the measured speed values (y_k^v) and the known true speeds (v_k). Then, average these scaling constants to get a final estimate of C.

To estimate the variance of the velocity sensor noise Var[r_k^v], first calculate the noise r_k^v for each trial using:

r_k^v = (y_k^v / C) - v_k

Then, calculate the variance of the noise for each trial and average the variances to get the final estimate of Var[r_k^v].

To illustrate that the calibrated model is behaving appropriately, I plot the actual speed (v_k) vs. the measured speed (y_k^v) for each trial, both before and after calibration.

Before Calibration: The first subplot shows the measured speed values for each trial before calibration. The data points might be scattered and may not align perfectly with the true speeds (0 m/s, 10 m/s, and 20 m/s). This indicates that the raw sensor measurements have some noise and inaccuracies.

After Calibration: The second subplot shows the measured speed values for each trial after applying the calibration using the estimated scaling constant

(C). The data points should be closer to the true speeds (0 m/s, 10 m/s, and 20 m/s) than before calibration. This suggests that the calibration process has improved the accuracy of the sensor measurements by compensating for the unknown constant and noise.

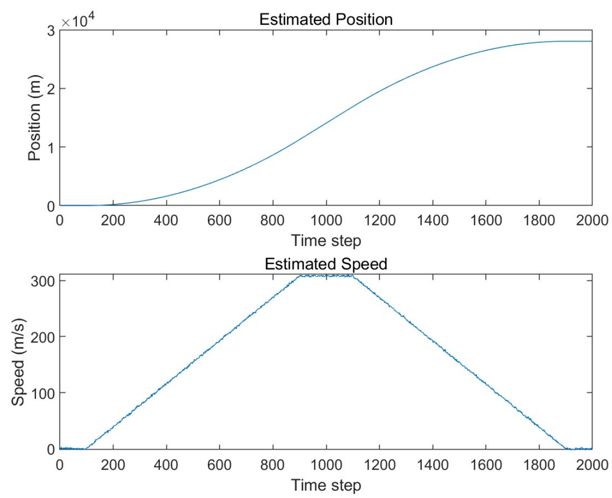
By comparing the two plots, we can see the effectiveness of the calibration process. The calibrated model show a better agreement between the actual speed and the measured speed, indicating that the sensor measurements are more reliable after calibration.

b)

To adapt the Kalman filter equations to handle sensors with different update rates, there are two steps as follows:

Prediction Step: Perform the prediction step as usual, regardless of the availability of the position measurement. This means that you always use the speed measurements to update the state estimate.

Update Step: If a position measurement is available, perform the update step with both position and speed measurements. If a position measurement is not available, perform the update step using only the speed measurement.



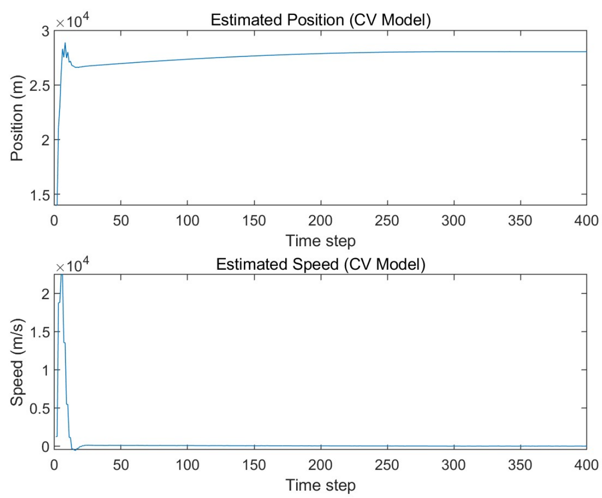
In the plot, we can observe the estimated position and speed of the train over time using the modified Kalman filter, which fuses sensors with different update rates.

Estimated Position: The first subplot shows the estimated position of the train over time. This plot demonstrates how well the Kalman filter is able to track the position of the train despite the noisy position measurements and the fact that position measurements are only available at a lower rate compared to the speed measurements.

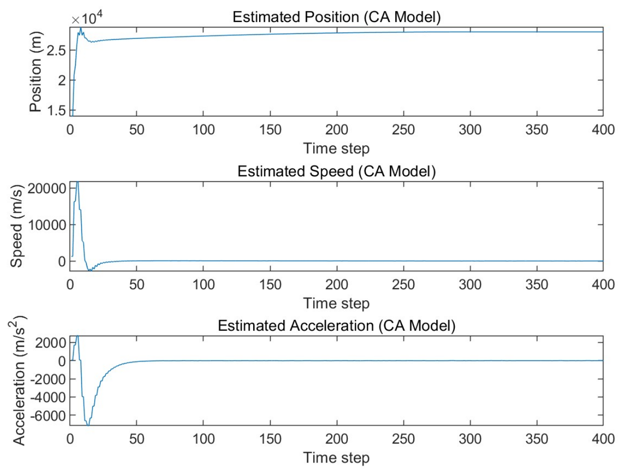
Estimated Speed: The second subplot shows the estimated speed of the train over time. This plot demonstrates how well the Kalman filter is able to track the speed of the train using the speed sensor measurements, which are available at a higher rate compared to the position measurements.

By examining these plots, we can assess the performance of the modified Kalman filter in fusing the data from sensors with different update rates. The plots should display relatively smooth estimates of the position and speed of the train, indicating that the Kalman filter is able to effectively combine the information from both sensors to provide a more accurate estimate of the train's trajectory.

c)



best position measurement noise variance=1
best process noise covariance matrix elements=[0.1 10]
minmum rmse =1.3877e+04



best position measurement noise variance=1
best process noise covariance matrix elements=[0.1 7.8 10]
minmum rmse =1.3877e+04

To tune the model, firstly I split the available sensor data into a training set and a validation set(80;20).Then for each model (CV and CA):

a. Initialize the model with different values of process noise covariance matrix (Q) and position measurement noise variance (R$_p$)*withinthegivenrange*.

b. Run the Kalman filter on the training set and evaluate its performance based on a suitable metric (e.g., root mean square error or RMSE) by comparing the estimated states with the true states.

Test the performance of both models with the selected parameters on the validation set. Compare the performance of the CV and CA models and choose the model that provides better results.

In the plot, we can see how well the CV or CA model is able to track the position and speed of the train. In this case CA model is the better model because it provides smoother and more accurate estimates of the position and speed.

d)

There are two motion models: the constant velocity (CV) model and the constant acceleration (CA) model.

Constant Velocity (CV) Model:

Advantages: Simpler and computationally less demanding. Sufficient for scenarios where the train's acceleration is minimal or negligible. Disadvantages:Cannot account for changes in acceleration, which can lead to less accurate estimates in cases where acceleration is significant.

Constant Acceleration (CA) Model:

Advantages: More flexible and can account for changes in acceleration. Provides more accurate estimates in scenarios where the train's acceleration is significant or non-constant. Disadvantages: Computationally more demanding. May require more tuning of the process and measurement noise covariance matrices.

For different scenarios, if the train's motion mostly consists of constant or near-constant velocities, the CV model should be sufficient and would be preferred due to its simplicity. However, if the train's motion includes significant changes in acceleration, the CA model would likely provide more accurate estimates, despite being more computationally demanding.