Solution to analysis in Home Assignment 2

Nicholas Granlund, nicgra@student.chalmers.se

Analysis

In this report I will present my independent analysis of the questions related to home assignment 1. I have discussed the solution with Martin Lamm, Louise Olsson and NONE and I swear that the analysis written here are my own.

1 Scenario 1 - A first Kalman filter and it's properties

a) Generate a state sequence and a measurements sequence for $N = 35$, do the measurements behave according to model?.

The following results has been generated with the fixed random number generator rng(1). This was done to be able to recreate the same figures over again. The generated state sequence and corresponding measurements can be seen in Figure 1.1 below. As can be seen by the figure, the measurements somewhat track the actual states whilst beeing noise corrupted. This indicates that the model behaves accordingly and that the measurements reads the states as intended.

Figure 1.1: State strajectory of the state space model and measurements

b) Filter the measurements in the Kalman filter. Plot the sequence of estimates together with the $\hat{x}_k \pm 3\sigma$ level. In the same figure, also plot the correct states and the measurements. Are the estimates that the filter outputs reasonable, if so, in what way? Does the error covariance represent the uncertainty in the estimates well?

The results can be seen in Figure 1.2 below. The outputs from the Kalman filtare are reasonable. They are reasonable because the outputs mimic the measurements and is in the same time similar to the actual state trajectorie. The filter estimate doesnt diverge or behave unreasonable and therefor the result is valid. The error covariance represents the uncertainty rather well. It does that because both the measurements and actual states are within the 3σ-levels as shown by the black dotted lines in the Figure.

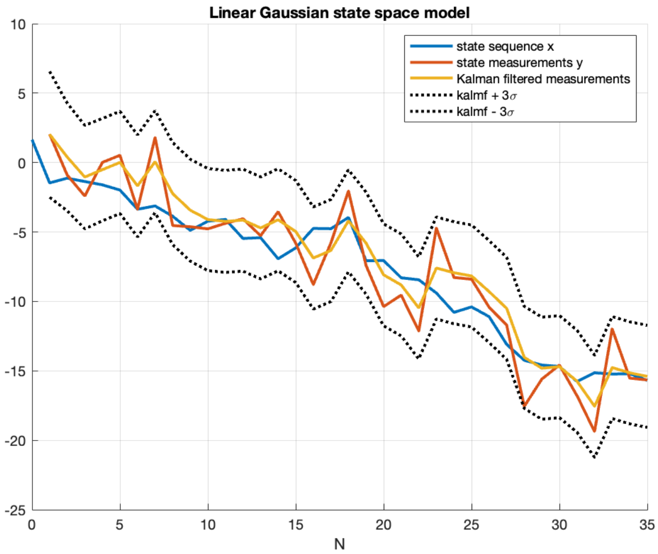

Figure 1.2: State strajectory of the state space model and measurements

c) Study what happens if you use an incorrect prior at time 0. Tell the Kalman filter that the initial mean is 12. Motivate

By telling the Kalman filter that the initial mean is 12, it will have a larger error in the earlier states of the state development. However, as k increases the Kalman filter obtains new information and converges to the right path. As new information is obtained, older information is not as usefull/viable and therefore the effect of the fault assumption of mean = 12 is neglected. This is illustrated by figure 1.3 below.

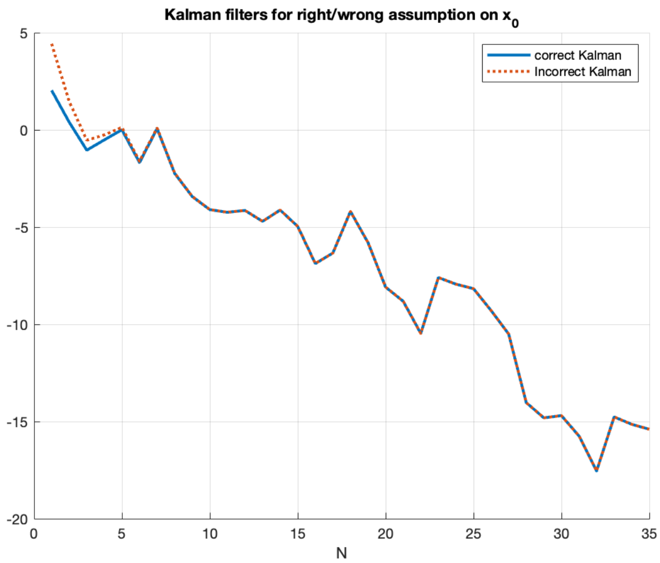

Figure 1.3: Correct and Incorrect Kalman filters. They converge to the same path as k increases.

d) Plot $P(x_{k-1}|y_{1:k-1})$, $P(x_k|y_{1:k-1})$, y and $P(x_k|y_{1:k})$ for a choice if k you deem fitting.

I choose time instant $k = 23$ because it illustrates well what is occuring in the different steps of the kalman filter. As expected, when a prediction is done, our uncertainty increases since the filter cannot be sure wether the state value will increase or decrease. therefor the PDF for the prediction has the same mean as the prior, but has a larger variance due to larger uncertainties. Then, for the measurements we also get greater uncertainties. We have measured a new state value around the mean of the measurement, but the noise in the sensors cannot make us any more certain than this. By then combining our prediction with out measurement, the Kalman filter is able to calculate the most probabilistic state value, yielding in a PDF with higher certainty than both prediction and measurement. The behaviour is therefor deemed reasonable. The plots can be seen in Figure 1.4 below.

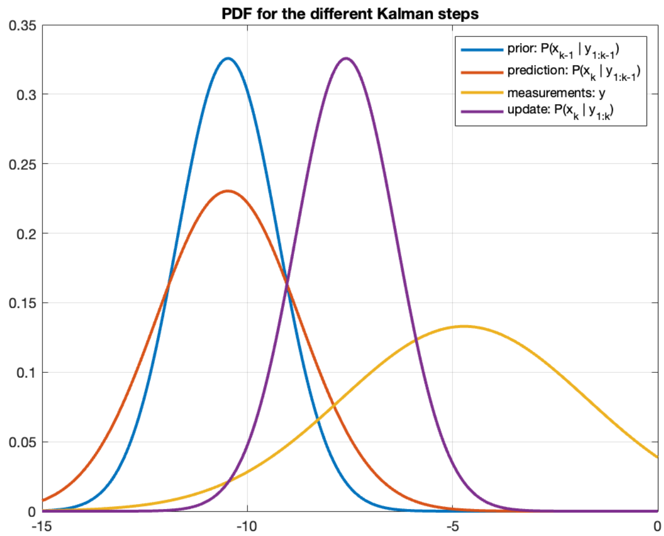

Figure 1.4: Plot for the different steps in the Kalman filter.

e) Consistency: After a while...

2 Scenario 2 - Tuning a Kalman filter

a) Determine the scaling constant C and the variance of the velocity sensor.

By using the measurement when the train is standing still $v_k = 0[m/s]$, then we can obtain information about the expected value of the noise from the velocity sensor. By doing so we get that the expected value $\mathbb{E}[Cr_k^v] = C\mathbb{E}[r_k^v] = -0.0734$, which is close to zero.

This will then help us estimate the scaling constant C for when the train is in motion. For when the train travels at 10 [m/s] we get:

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

$$y_k^v = Cv_k + Cr_k^v$$

$$\mathbb{E}[y_k^v] = \mathbb{E}[Cv_k] + \mathbb{E}[Cr_k^v]$$

$$\mathbb{E}[y_k^v] = 10C + (-0.0734)$$

$$C = \frac{\mathbb{E}[y_k^v] - (-0.0734)}{10}$$

$$C_{10} = 1.1100$$

The same logic follows for the measurements of when the train travels at 20 [m/s]. In that case we get the scaling constant as:

$$C_{20} = 1.1059$$

The mean of these two is then the best approximation we can obtain.

$$C = 1.1020$$

Using our knowledge of the scaling constant C we can now begin to calculate the variance in the noise. When the train travels at 10 [m/s], the variance in the noise is calculated to be 2.4976, and for when the train travels at 20 [m/s], the variance in the noise is 2.5064. It is therefor plausible that the variance in the noise from the sensor is ≈ 2.5. To test if this is plausible, lets try to regenerate the measurements with the knowledge we now have about the noise. Doing so and plotting the generated data in tha same figure as the measured data, we can do a visual examination and conclude that the scaling constant C, mean of the noise and variance has been correctly estimated. See Figure 2.1 below.

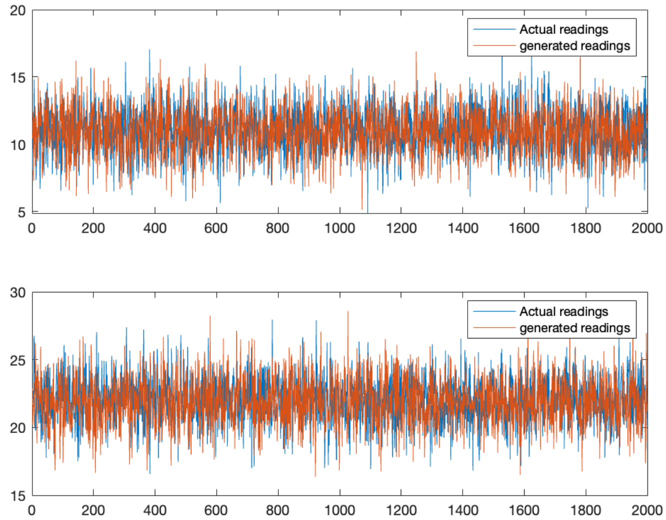

Figure 2.1: Measured data and generated data. Rather similar.

b) Fusing sensors with different rates

To fuse these sensors we want to use only time instances when readings from both sensors is available. This can be done in two ways. The first way is to exchange all the instances of missing data NaN to be the value of the previous measurement. For instance at time k the positional reading would be 2 [m] and at k+1, instead of NaN the reading would still be 2 [m]. The other way would be to discard al the measurements when there is no data from the positional sensor. By doing so we also have to 'double the update cycle' from the velocity sensor from $T_v = 0.1s$ to $T_v = 0.2s$.

By following the second example, we are able to model the position and velocity of the train rather well. And can thus fuse the two sensors with different update speeds. This is done for the upcoming question.

c) Motion model selection and tuning

We can model the system in multiple different ways. But here we will consider CV and CA scenarios.

CV model:

By modeling the system as having constant velocity, we get a poor performing filter. This is illustrated by Figure 2.2 below. Since the velocity is expected to be constant and not change, the Kalman filter lags behind in tracking changes in speed as can be seen by the entire graph being shifted towards the right. $Q = 0.0001$ for the velocity and position.

Better performance can be obtained by increasing Q which allows for larger deviations in the velocity. However, by doing so the estimated velocity may not be as smooth. It is a balance between the filters ability to track fast changes, and smooth filtering. The best performance possible I deem is when $Q = 0.01$ for this scenario.

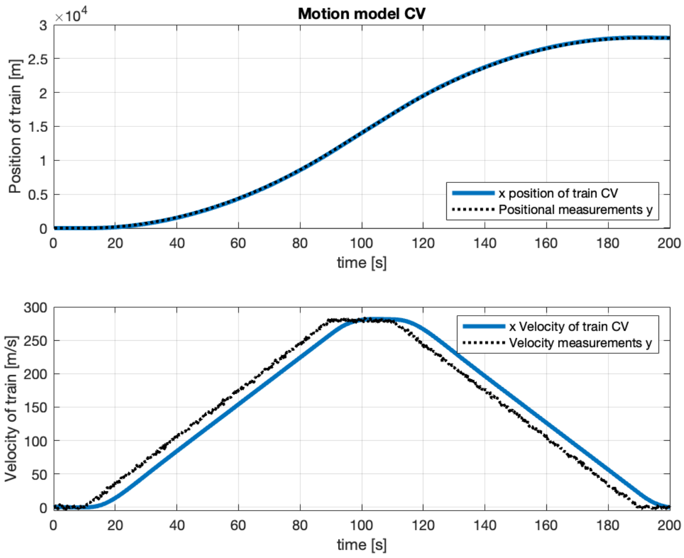

Figure 2.2: CV model of dynamic system. actual speed lags behind.

' **CA model:**

By instead modeling the system as having constant acceleration, we allow for a much more accurate tracking of the changes in speed. And since the acceleration is piece wise constant, this modelling is not that far-fetched. The result of modelling the dynamical system the way can be seen in Figure 2.3 below. Here, instead we can see that the Kalman filter has problem tracking change in acceleration, i.e when the slope of the velocity graph changes. This is rather apparent at time k=90 seconds where the acceleration goes from being positive to 0, and the estimated velocity overshoots in a "bump". Overall this time of modelling allows for a better estimate of the velocity and henceforth a better estimate in the position. $Q = 0.0001$ for the acceleration, velocity and position for this case aswell.

Better performance here can also be obtained by increasing Q. Like the previous scenario, if we increase Q we will allow for larger deviations in the acceleration, and this may be beneficial for the filters performance when there is a change in acceleration. But just as the previous example, this is on the expense of the smoothness of the filtering. The best performing filter here was obtained with $Q = 0.01$ following the same logic as previously.

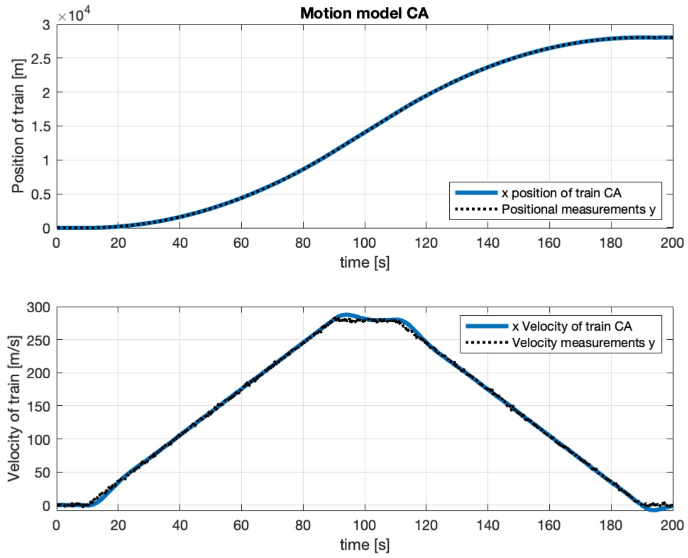

Figure 2.3: CA model of dynamic system. Speed estimate is good but not robust against changes in acceleration.

d) Which model fits better?

I believe it is somewhat answered in the previous question, but I will summarise it here aswell. The CA model fits better, but yet again it depends on how the train is expected to behave. For this given data, the behaviour is such that modelling after CA yields a good estimate of the velocity and speed (since the data is piecewise constant acc). However in reality, the acceleration may vary way more than what we have tuned for. In that case, I believe that the best possible filter to apply for a train in variyng motion, would be a CA filter where the filter is tuned as such that the acceleration is expected to vary. This will probably increase the noise in the velocity measurements, but it will allow for better tracking of the changes in acceleration.

In short, CA is better than CV in this scenario. CV would be better if we model the train between stations where the speed is expected to be more constant. Modelling according to CA would be better when the train leaves/enters the station where change in speed would be expected.

/ Nicholas Granlund