Solution to analysis in Home Assignment 2

Daniel Söderqvist (danisode)

05/04-2023

Analysis

In this report I will present my independent analysis of the questions related to home assignment 2. I swear that the analysis written here are my own.

1 Scenario 1 – A first Kalman filter and its properties

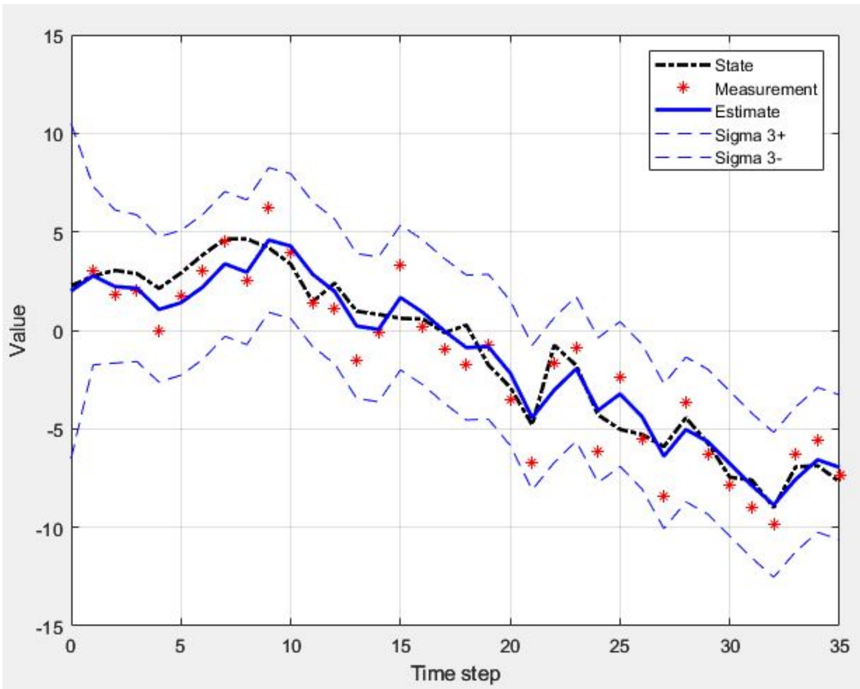
(a) 

Figure 1.1: Actual state, measurements and the estimation from the Kalman filter along with the 3 sigma limits.

If we in this sub-task just look at the true state and the measurements we can see that graph is representing the measurements well. Throughout the hole sequence of time-steps the true state follow the measurements as can be seen in figure (1.1).

(b) If we in this task look at the same figure (1.1) but now let's consider everything. We can see that the estimation from the Kalman filter is behaving somewhat as expected since it follows the true state accordingly throughout all the time steps. We can also see that both the true state and the estimation are represented pretty well by the measurements. It doesn't necessarily have to be like this but since all measurements are within the 3 sigma limit we can see that the estimation agrees well with the measurements which means that to covariance agrees well with the estimation as well.

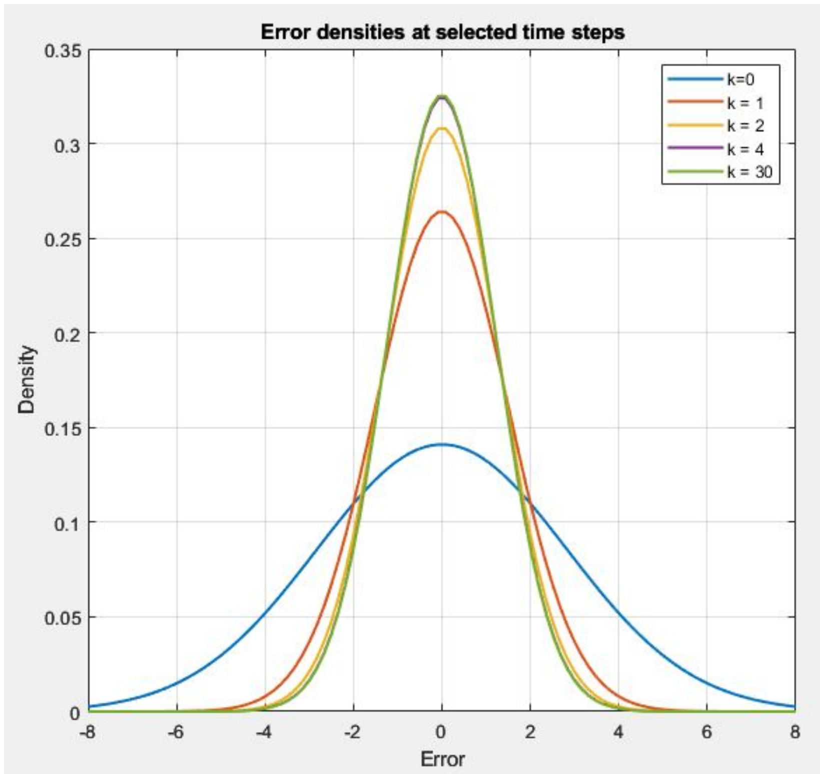


Figure 1.2: Error density of different time-steps.

We can clearly see in figure (1.2) that after more time-steps the Kalman filter gets more and more certain. The error density of the prior is very wide compare to just 1 time-step in. We can therefore also see in the figure that after just a few time-steps the Kalman filter is way more certain than from the beginning. And as can be seen there is not much improvement in the certainty between the fourth and the 30:th time-step which mean that the filter adjust quite quickly.

(c) In this task we will do the same experiment as before but use another that faulty to see how the filter behaves.

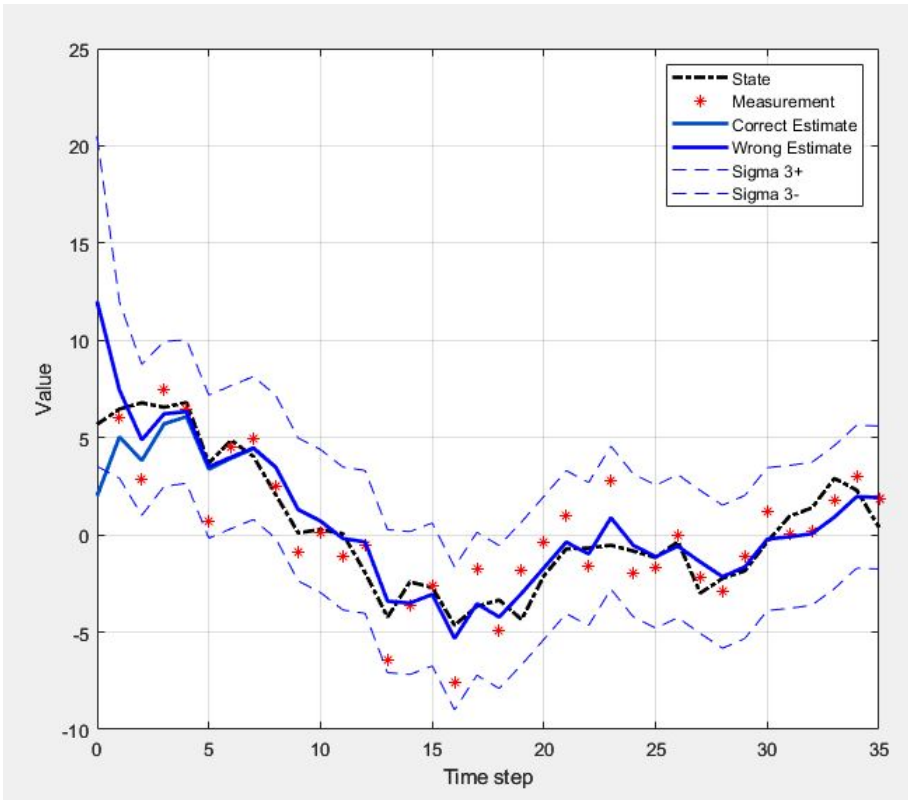


Figure 1.3: Actual state, measurements and the estimation from the Kalman filter for both the correct and wrong estimation along with the 3 sigma limits.

In figure (1.3) can we clearly see the prior of the faulty estimation is wrong, but as expected, the filter is adjusting very fast and we can see that just after a few time-steps the estimation is identical to the correct one. This means that the filter is very powerful in that way meaning that we don't need the correct prior since we are adjusting and adapting to the previous state and the new measurement for each time-step.

(d)

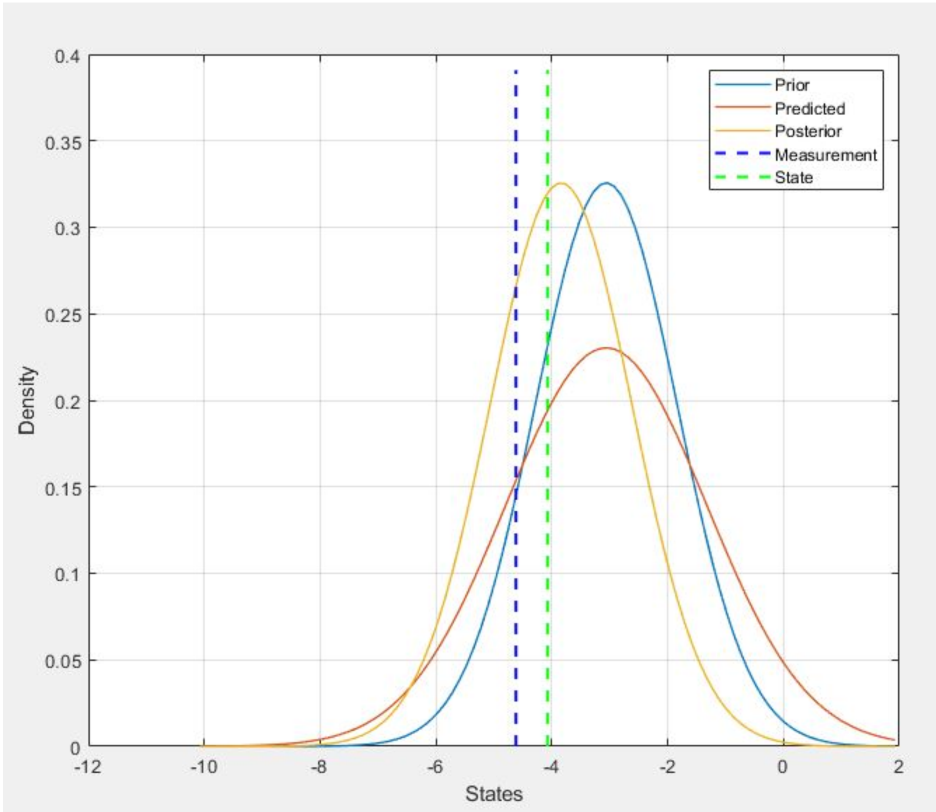
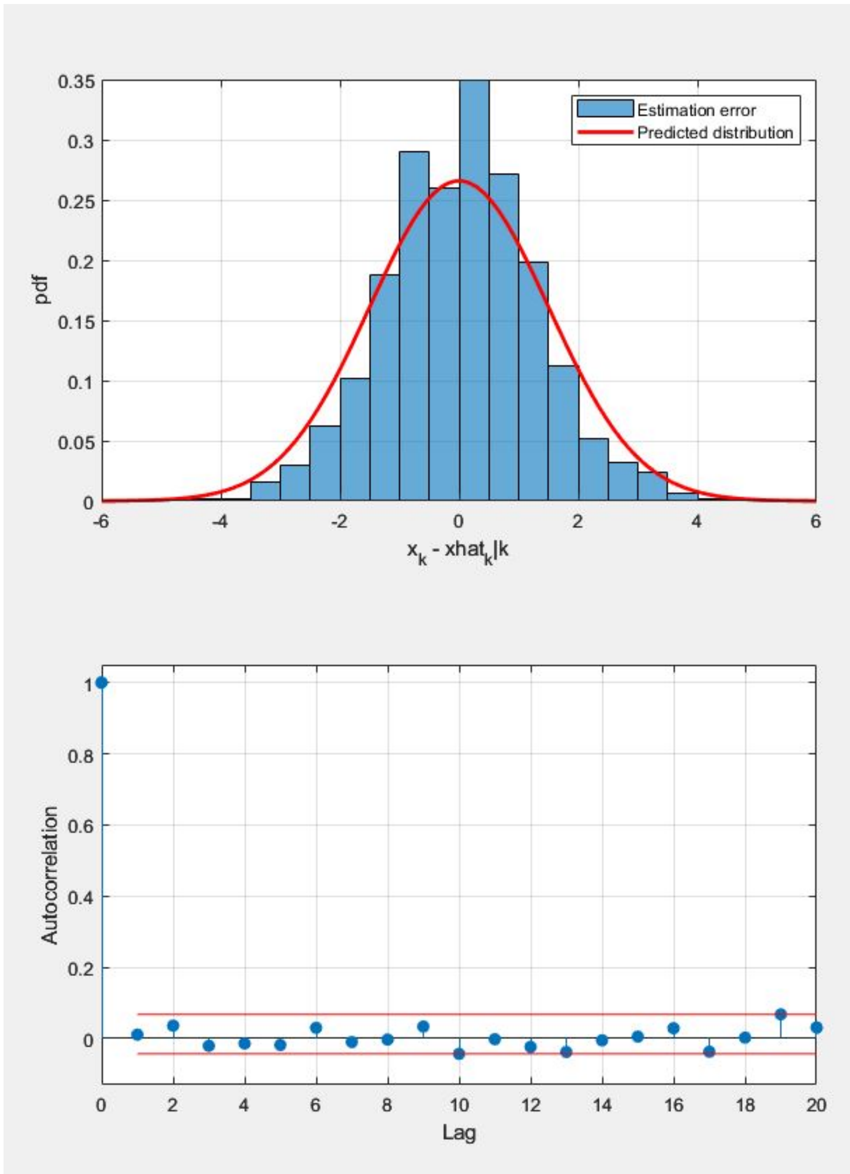


Figure 1.4: The probability density for the prior, predicted and posterior states along with the measurement and the actual state.

In figure (1.4) can we see that if we start by comparing the prior with the predicted probability densities we are more uncertain when we are predicting which is expected. When we then get a measurement data we can see that we are adjusting the mean towards the measurement but not all the way since we still consider the previous probability density. We land somewhere in the middle. We can in this case see that the actual state almost fell in the maximum of the probability density of the posterior. This means that in this case the prediction and the update worked well for estimating the state. This all means that the graphs is behaving like we expect and is therefore reasonable.



(e)

Figure 1.5: Top graph is showing a histogram of the estimated error with the pdf. Bottom graph is showing the auto correlation of the innovation function.

If we look at figure (1.5) we can see in the top graph that the estimated error (histogram) is well represented by the PDF which means that the error is following a Gaussian distribution. This is one of the assumptions we are making in a Kalman filter.

If we now instead look at the bottom graph we can see the auto correlation of the innovation function. We can see that the values very fast decays to around 0. This indicates that the filter is working properly and since the values are small we can draw the conclusion that the filter isn't biased and the values are uncorrelated to each other.

2 Scenario 2 – Tuning a Kalman filter

(a) In this task we are supposed to look at another example. The first task is to find the constant C and the total variance of the 3 different data sets. We start by calculating the variance which simply can be done by taking the variance of each dataset and then take the mean of the 3 variances to get the total one for all of them combined. The total or combined variance is calculated to **variance = 3.02127**. To calculate the constant C we firstly calculate the mean for the 3 datasets. Then we use the given equation to decide C ($y_k = C(v_k + r_k^v)$). By taking the mean of the measurements then divide by the velocity that represent each data set. We do this for the second and third data set which is represented of the velocities 10 and 20 since the first data set represents stationary. We then take the mean of these 2 new values and this represents that combined scalar constant C. By doing this in Matlab we calculated **C to be = 1.10245**.

The following figures (2.1) and (2.2) represents the data sets before and after using the constant.

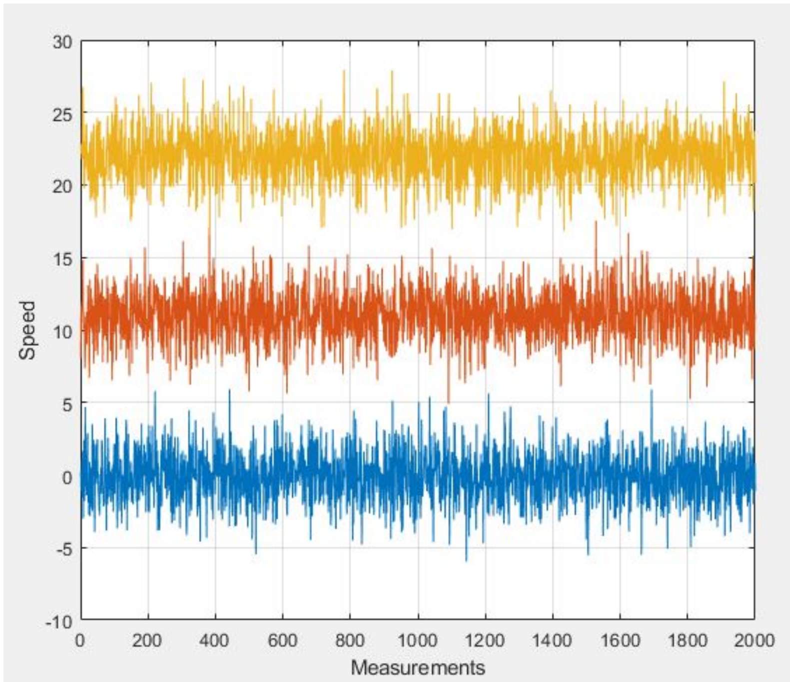


Figure 2.1: Raw data from data sets.

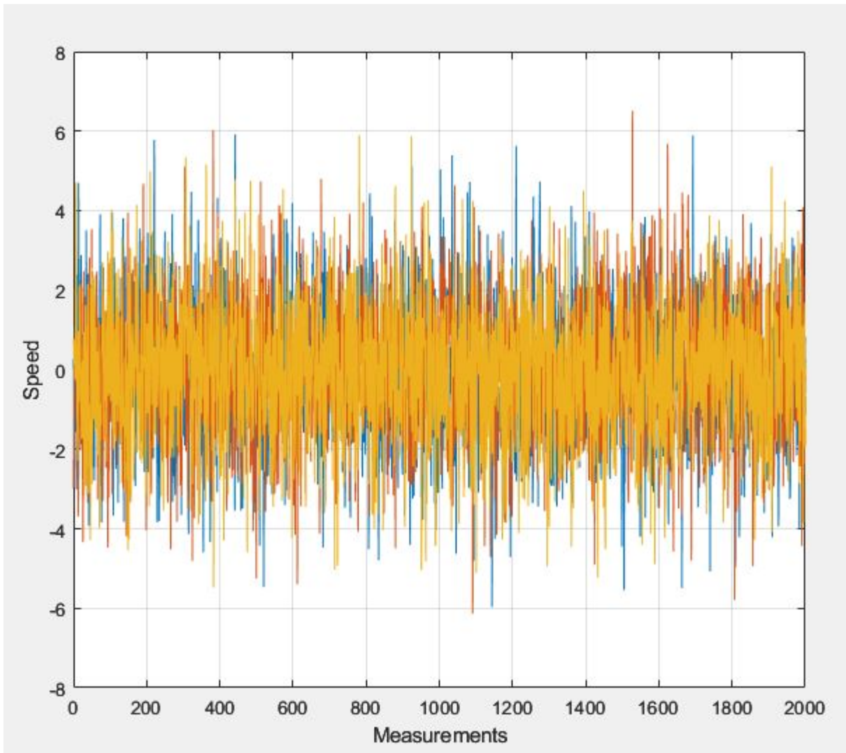


Figure 2.2: Processed data from data sets.

We can verify that the result is accurate since it seems like all the data sets now have the same mean which means that the scalar constant C seems to be correct. We can also see that the variance is around 3 as well which seems to be correct. By calculating the variance after processing the data sets and compare to the total variance we can see that they are identical which is expected.

(b) In this task we are supposed to modify the Kalman filter to be able to handle NaN data in between since it's common that real-life sensors doesn't necessarily provide measurement for all the time-steps. I simply constructed a method to use the previous measurement if there were a NaN measurement. By simply using a for loop with if statements the latest measurement that is nonNan was saved and as fast as we got a NaN measurement we used the previous known one and replaced the NaN with it. We can now use the result as input to the Kalman filter function without any NaN in it.

(c) In this task we are supposed to tune the filter and we do this by both using the constant velocity and constant acceleration method respectively. The formulas for each of the method was taken from lecture notes, lecture videos and quizes. The parameters that affect the results and the tuning are for example the state transition matrix A, the process noise covariance matrix Q, the measurement matrix H and the noise covariance R. To see them exactly look at the attached code. The tuning was with the given formulas straight forward and the results were plotted and can be seen down below.

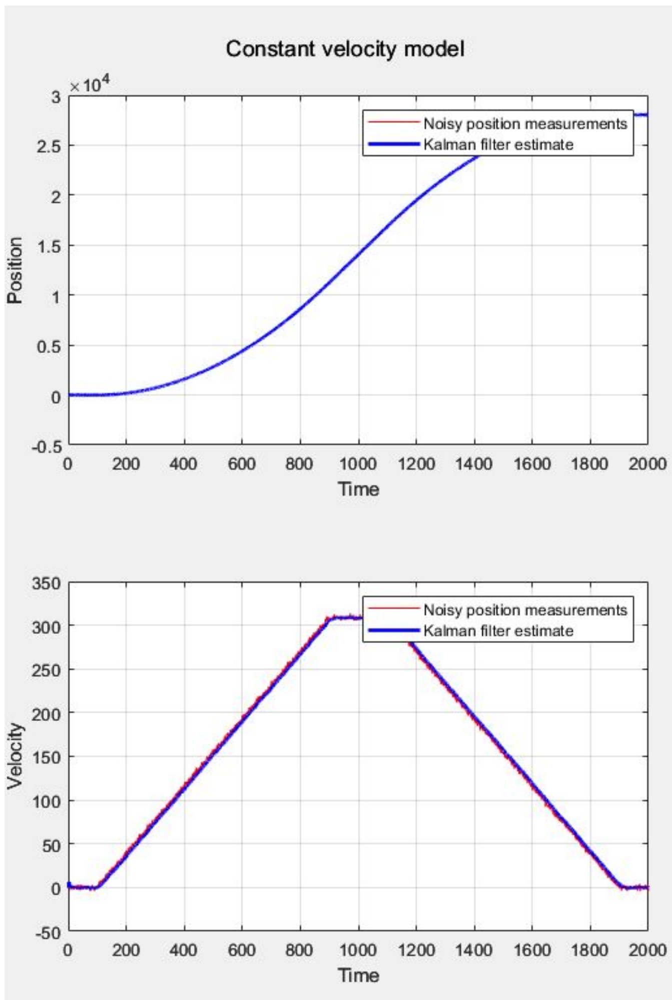


Figure 2.3: Top graph shows the measurements and the estimate of the position and the bottom graph shows the same thing but for the velocity. Both graph represents of the constant velocity model.

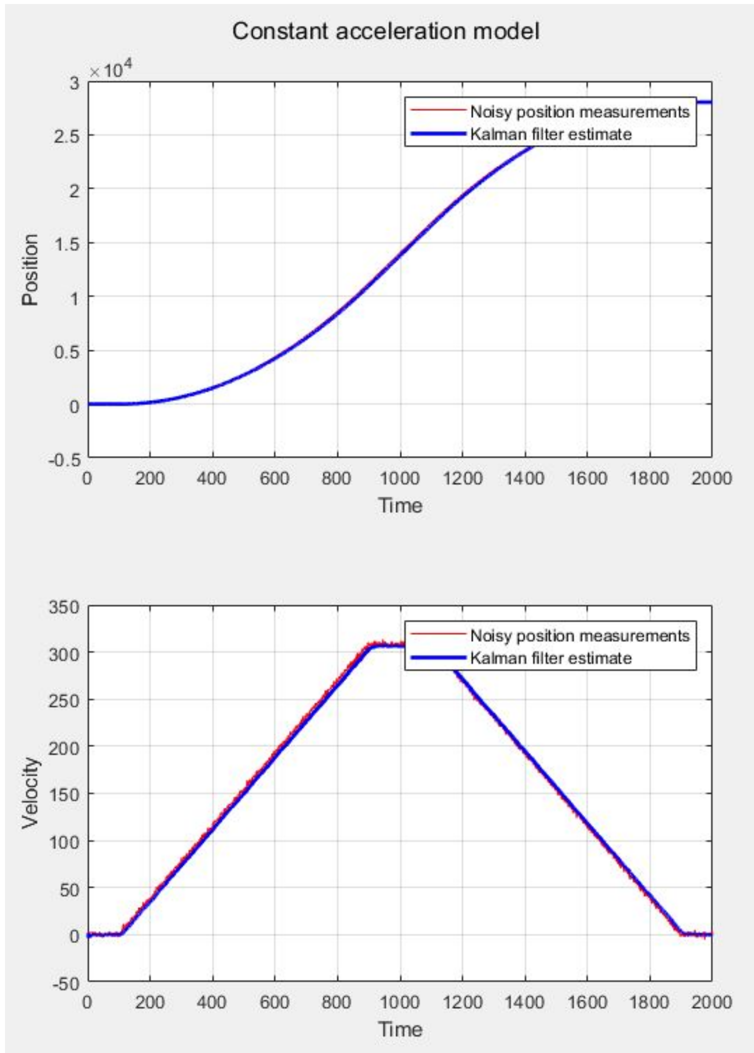


Figure 2.4: Top graph shows the measurements and the estimate of the position and the bottom graph shows the same thing but for the velocity. Both graph represents of the constant acceleration model.

We can see in figures (2.3) and (2.4) that after tuning with the respective model and handling the data with the for loop the filter yield approximately the same results for both methods which is expected. We can also see that the tuning worked fairly okay since the measurements have approximately the same curve as the estimations for both position and velocity regarding both methods.

(d) In this specific case i would have used the constant acceleration method since if we look at the figure (2.4) bottom graph we can see that the amount of time-steps where we have constant acceleration is much greater than the amount of time-steps having constant velocity. This means that the model represents more measurements which in turn leads to more accurately estimations.