ASSIGNMENT #5

Determining and removing drawbacks of exponential and running mean.

Group 2

Ekaterina Karmanova

Timur Chikichev

Iaroslav Okunevich

Nikita Mikhailovskiy

1. We generated a process, which has insignificantly rate change and high measurement noise level.

   Parameters of trajectory:

$$x_i = x_{i-1} + V_{i-1}T + \frac{a_{i-1}T^2}{2} \quad (1)$$

$$V_i = V_{i-1} + a_{i-1}T \quad (2)$$

   Size of trajectory is 200 points;

   Initial condition was $x_1 = 5$; $V_1 = 1$; $T = 1$;

   Variance of noise: $\sigma_a^2 = 0.2^2$;

Plot results for the trajectory and velocity:

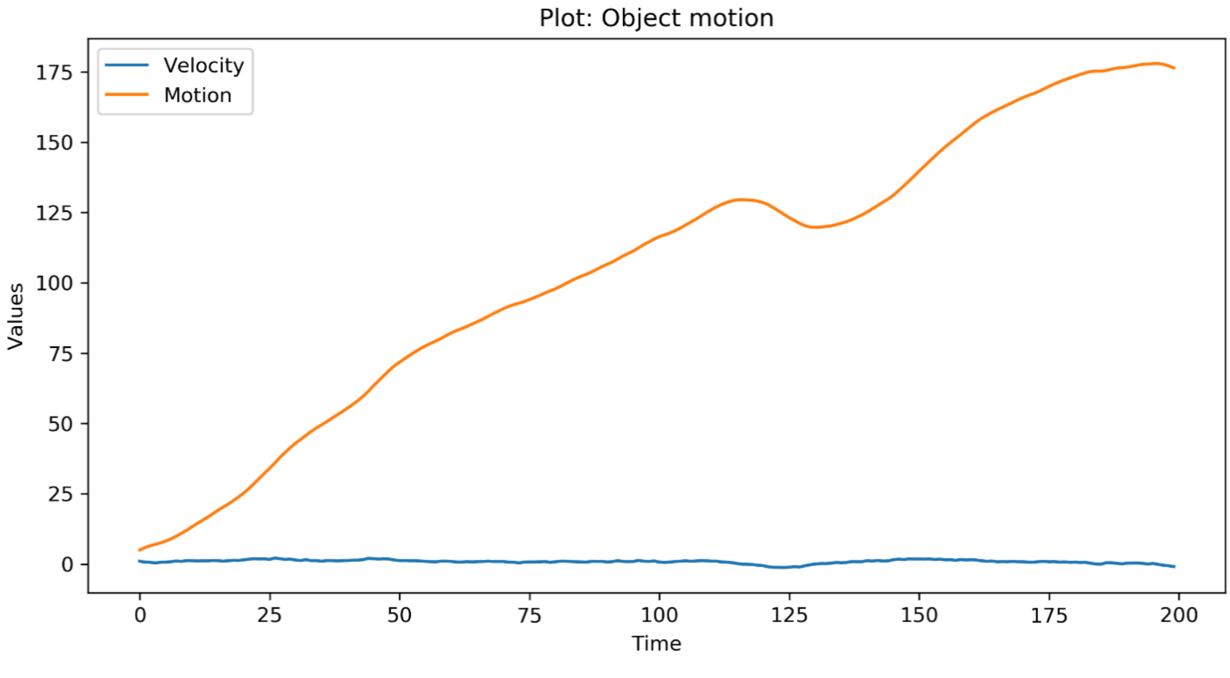

Figure 1 - Plot of result for 200 points for the process

2. Parameters of measurements:

$$z_i = X_i + \eta_i$$

Variance of noise: $\sigma_n^2 = 20^2$

Plot results for the measurements:

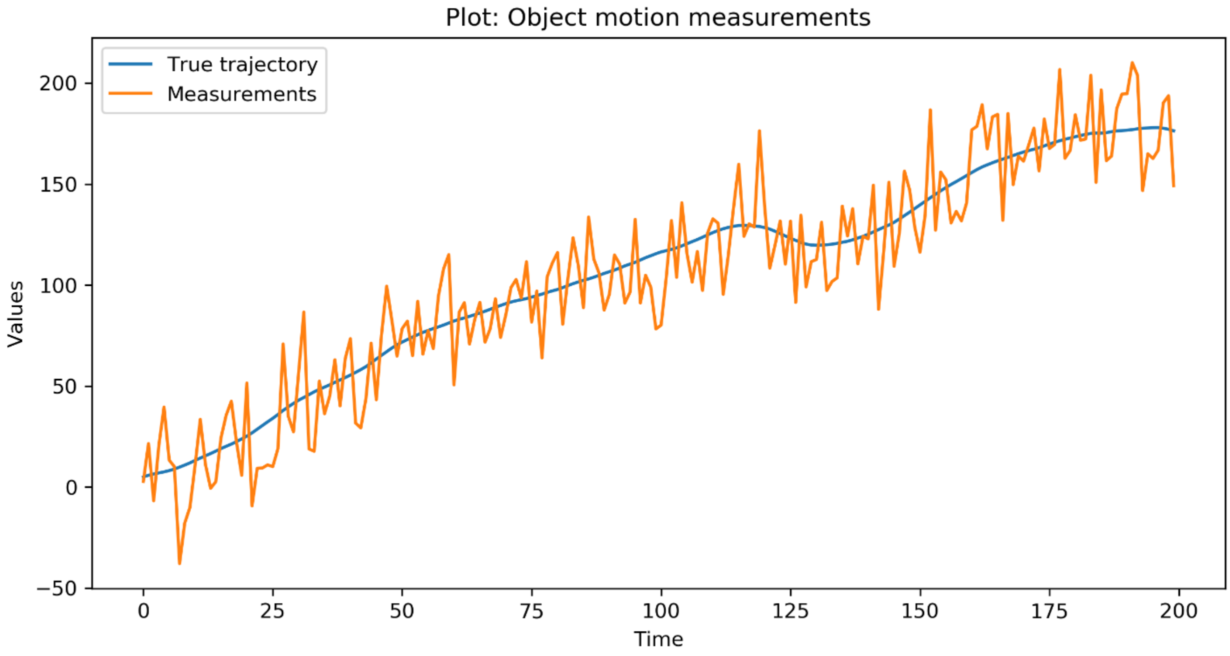

Figure 2 - Plot of result for 200 points for the measurements

3. To present the system at state space with the measurements of coordinate xi the transition matrix and input matrix are set, according to equation of the trajectory (1). In addition, observation matrix (according to the (2) equation) is used to present measurements of coordinate. State vector describes full state of the system:

$$X_i = \Phi X_{i-1} + G a_{i-1}$$

$$Z_i = H X_i + \eta_i$$

Where:

$$\Phi = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}; \; G = \begin{bmatrix} T^2/2 \\ T \end{bmatrix}; \; H = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

The result of presenting the system in state space is equal to the true trajectory and its measurements.

Plot of state vector and measurements:

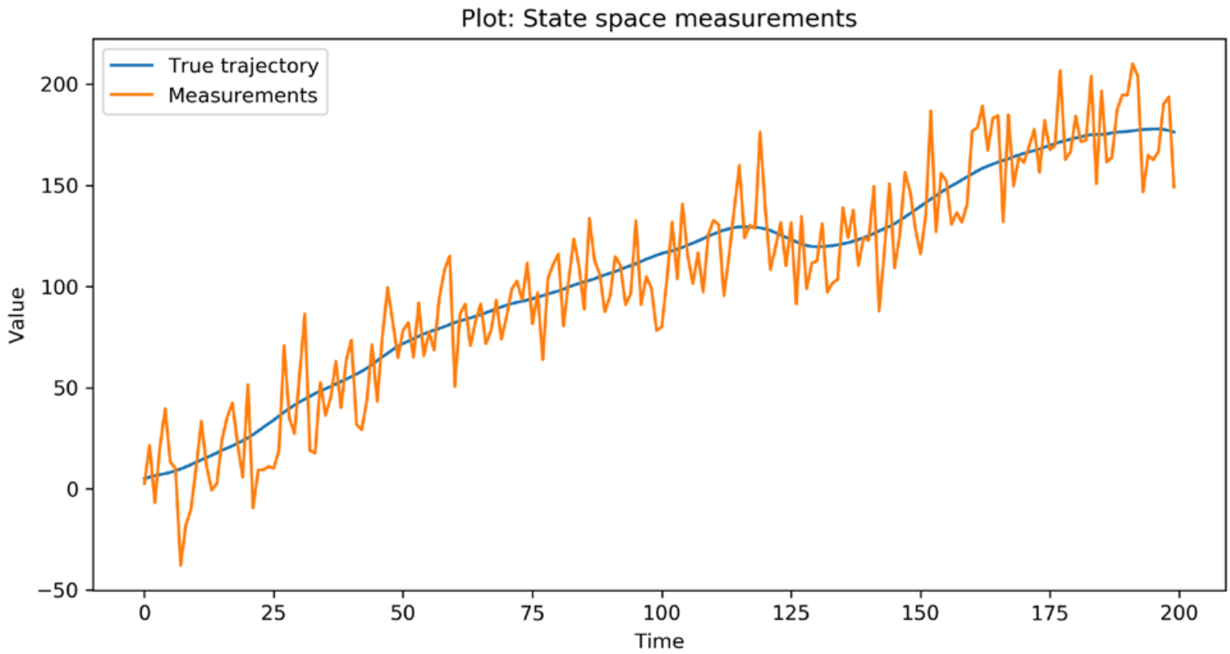

Figure 3 - Plot of result for 200 points for state vector form

4. Kalman filter

Extrapolation and filtration of signal. To estimate state vector, we use Kalman filter instead of Gauss method used in previous assignment.

1) The first step is to make prediction of state vector value using the previous measurements:

$$X_{i,i-1} = \Phi_{i,i-1} X_{i-1,i-1}$$

We also should make a prediction of error covariance matrix:

$$P_{i,i-1} = \Phi_{i,i-1} P_{i-1,i-1} \Phi_{i,i-1}{}^{T} + Q_i \ (3)$$

To determine covariance Q matrix of state noise we use the input matrix G that represents the effect of random acceleration on state vector:

$$Q = GG^T{\sigma_a}^2$$

For matrix operations, we used “numpy” library in Python (like np.dot, np.matrix, Matrix.T etc.)

2) The next step is filtration of signal according to the predicted estimate:

$$X_{i,i} = X_{i,i-1} + K_i(z_i - HX_{i,i-1})$$

K_i – the filter gain - represents the weight of residual between the measurement and prediction of the signal value. The value of filter gain is based on prediction error covariance matrix (P from eq.(3)) and covariance matrix R of measurement noise (R):

$$K_i = P_{i,i-1}H_i^T(H_iP_{i,i-1}H_i^T + R_i)^{-1}$$

Dimension of covariance matrix R of measurement noise is determined by a number of state vector components that are measured.

$$R = {\sigma_n}^2$$

Initial filtered estimate $X_0 = \begin{bmatrix} 2 \\ 0 \end{bmatrix}$

Initial filtration error covariance matrix $P_{0,0} = \begin{bmatrix} 10000 & 0 \\ 0 & 10000 \end{bmatrix}$

5. Plot results with filtered estimates of state vector Xi for two different acceleration values:

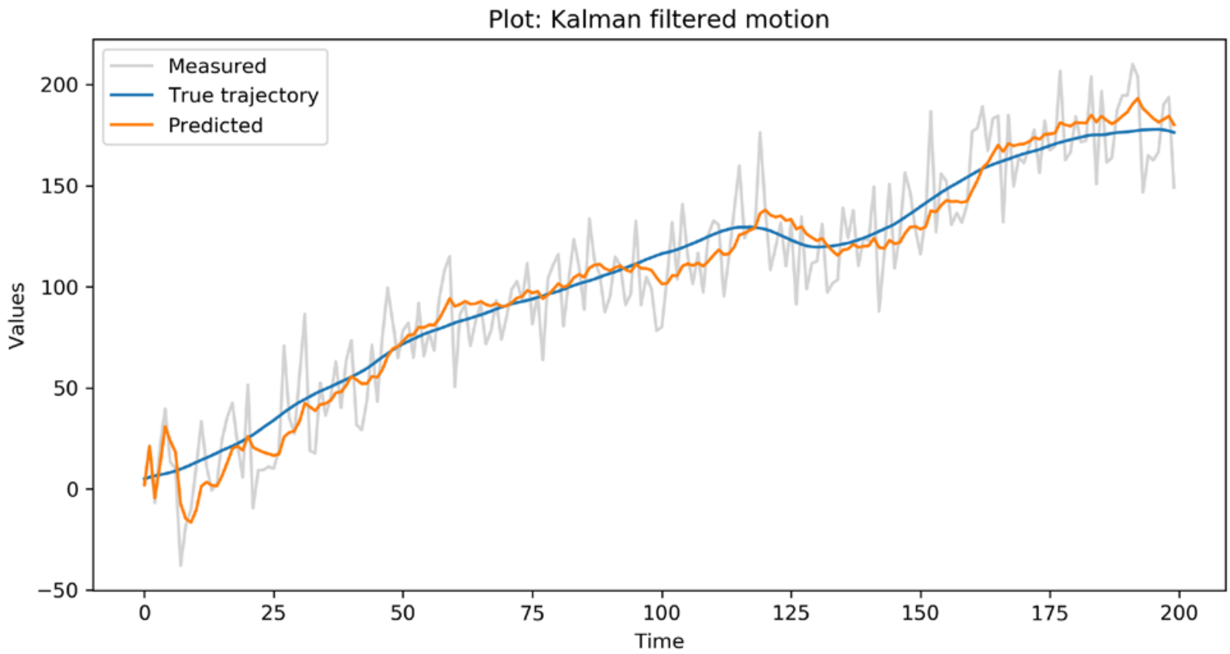

Figure 4 - Plot of result for 200 points for filtered state vector 1

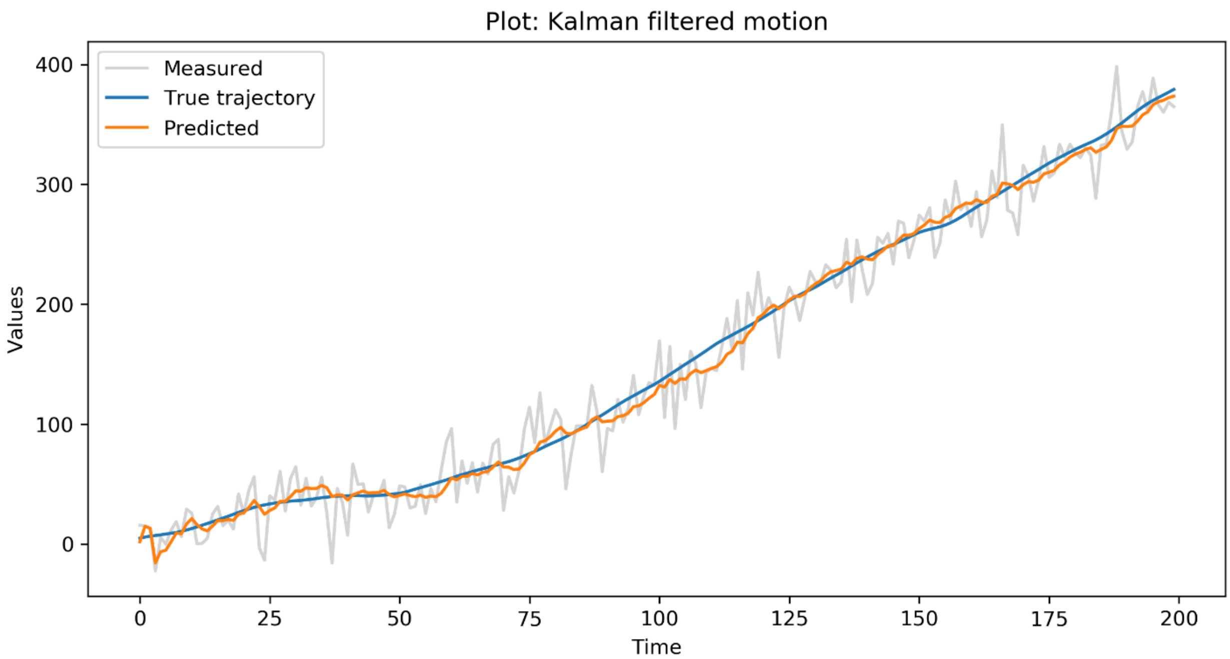

Figure 5 - Plot of result for 200 points for filtered state vector 2

6. Plot filter gain K over the whole filtration interval:

Plot of square root of the first diagonal element corresponding to standard deviation of estimation error of coordinate x_i.

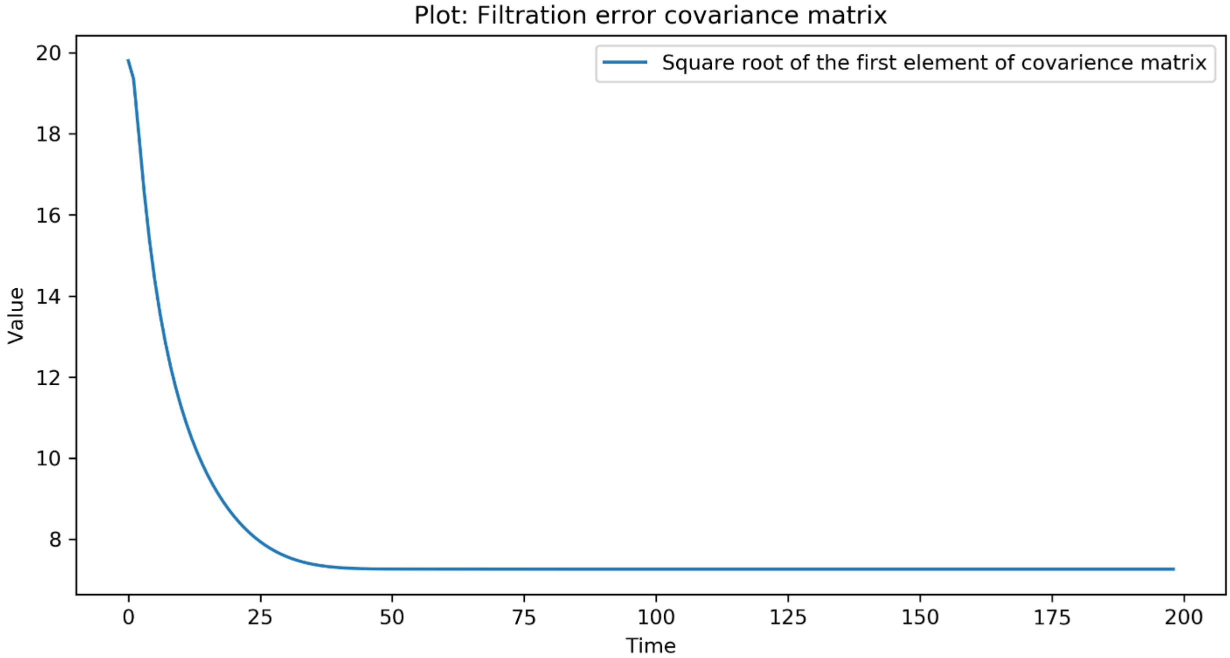

Figure 6 - Plot of result for 200 points for square root of P

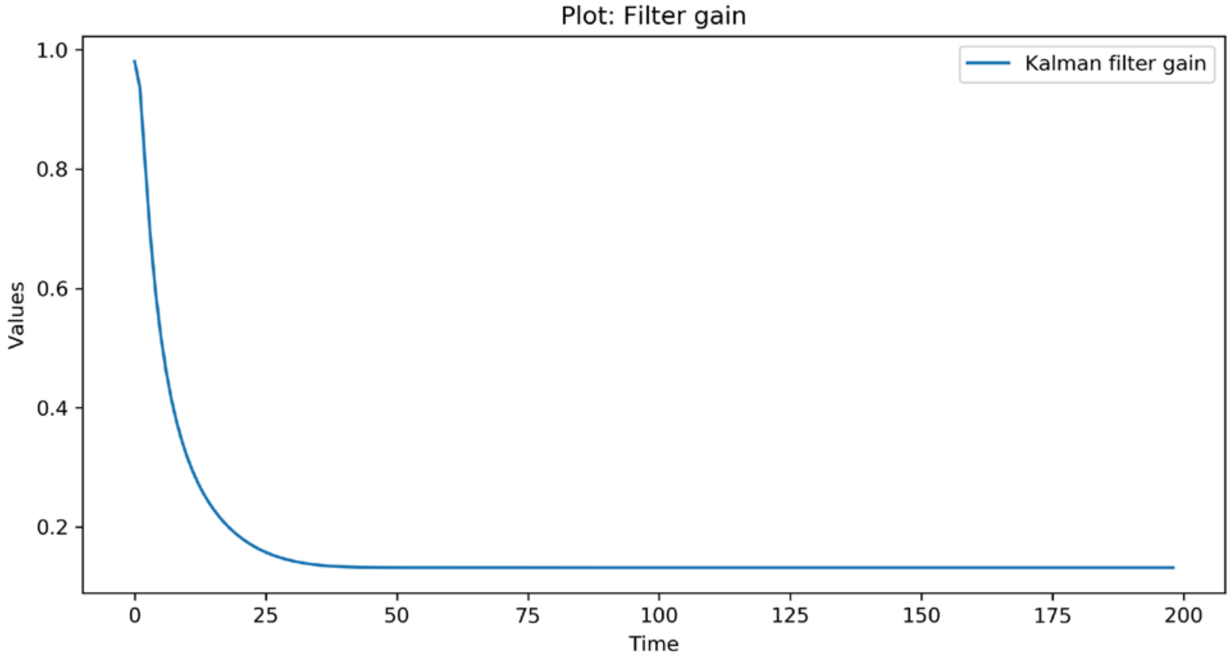

Figure 7 - Plot of result for 200 points for K

The square root of P matrix’s first diagonal element curve and the Kalman gain curve looks similar to each other, however the range of values is different: for P it’s ~ (18-7) and for K it’s ~ (1-0.15). Filter gain K and filtration error covariance matrix become constant very quickly. It means that in conditions of a trajectory disturbed by random noise we cannot estimate more than established limit of accuracy due to uncertainty.

7. When we added extrapolation on 7 steps ahead on every time step, we had slightly different plot:

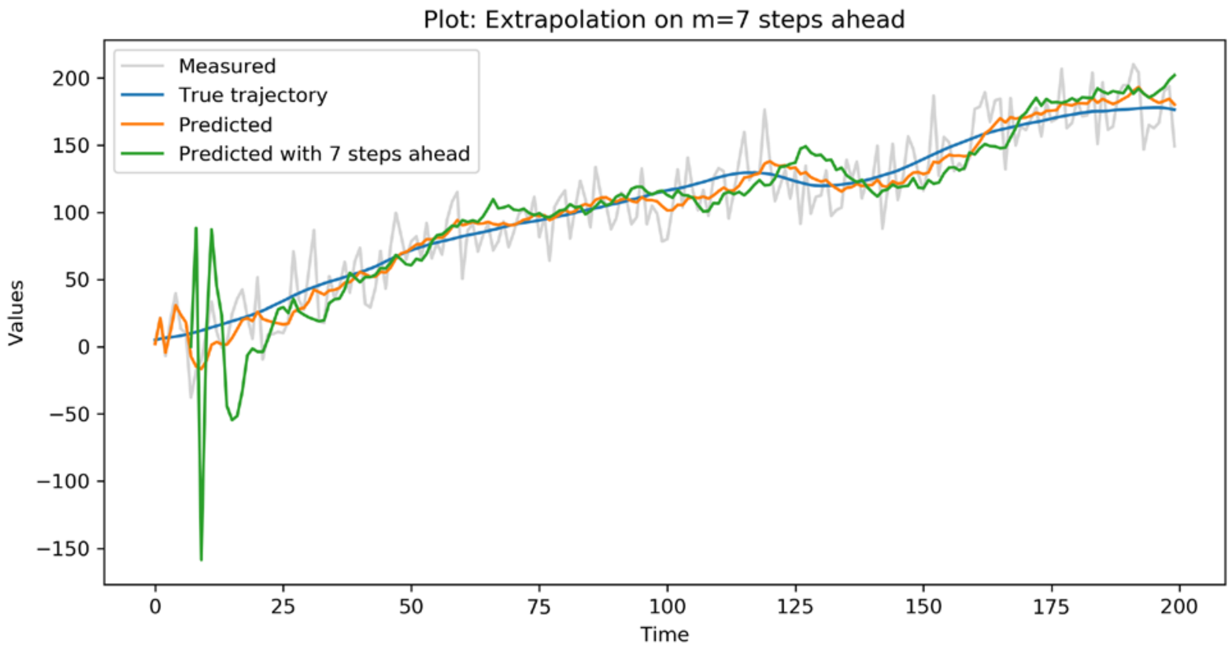

Figure 8 - Plot of result for 200 points for extrapolation (m=7)

8. We have made 500 runs of filter for different measurements and estimate dynamic of its mean-squared error. Result plotted in fig. 8.

To make a different quantity of runs we made an algorithm, which was generating on the new step new curve, filtering it, extrapolating and, finally, finding mean-squared error.

$$Error^{Run}(i) = \left(x_i - \hat{x}_{i,i}\right)^2$$
Run – number of run;
$i = 3, \dots, N$ - observation interval
(please start error calculation from step $i = 3$);
$Run = 1, \dots, M$ - number of runs;

$$Final\_Error(i) = \sqrt{\frac{1}{M-1}\sum_{Run=1}^{M} Error^{Run}(i)}$$

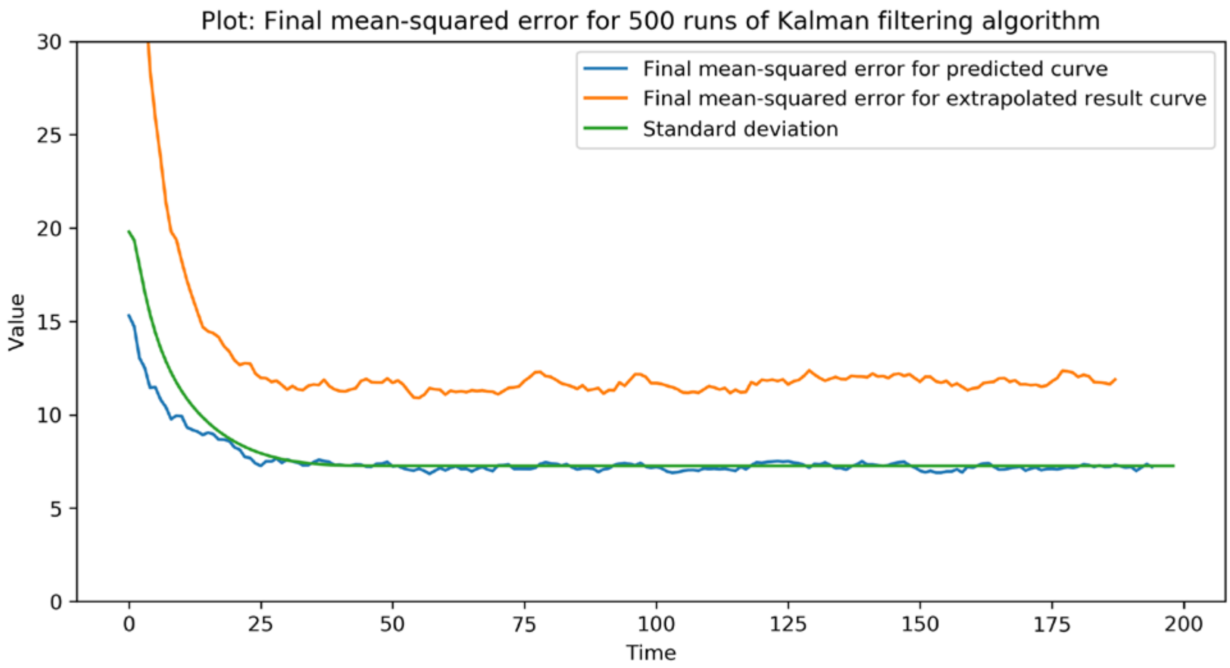

Figure 9 – Plot of final error results for filtered, extrapolated curves in comparison with standard deviation

9. According to the results, a standard deviation is equal to the final mean-squared error if used only Kalman filtering without any extrapolations. Extrapolation sets the offset against the filtered curve. The error has almost doubled, and the deviation of it is increasing.

10. By setting up an initial covariance matrix P to:

$$P_{0,0} = \begin{bmatrix} 100 & 0 \\ 0 & 100 \end{bmatrix}$$

We calculated again mean-squared error. This curve repeats the standard deviation curve and has almost the same value in stationary mode. There is just one difference at the starting points. The first error is smaller with the current initial covariance matrix in comparison with the previous one.

11. In compare with the previous calculations of mean-squared errors, all of them are differ by initial errors. These errors are approaching a single value to the end of filtration, which corresponds with the steady-state value of standart deviation.

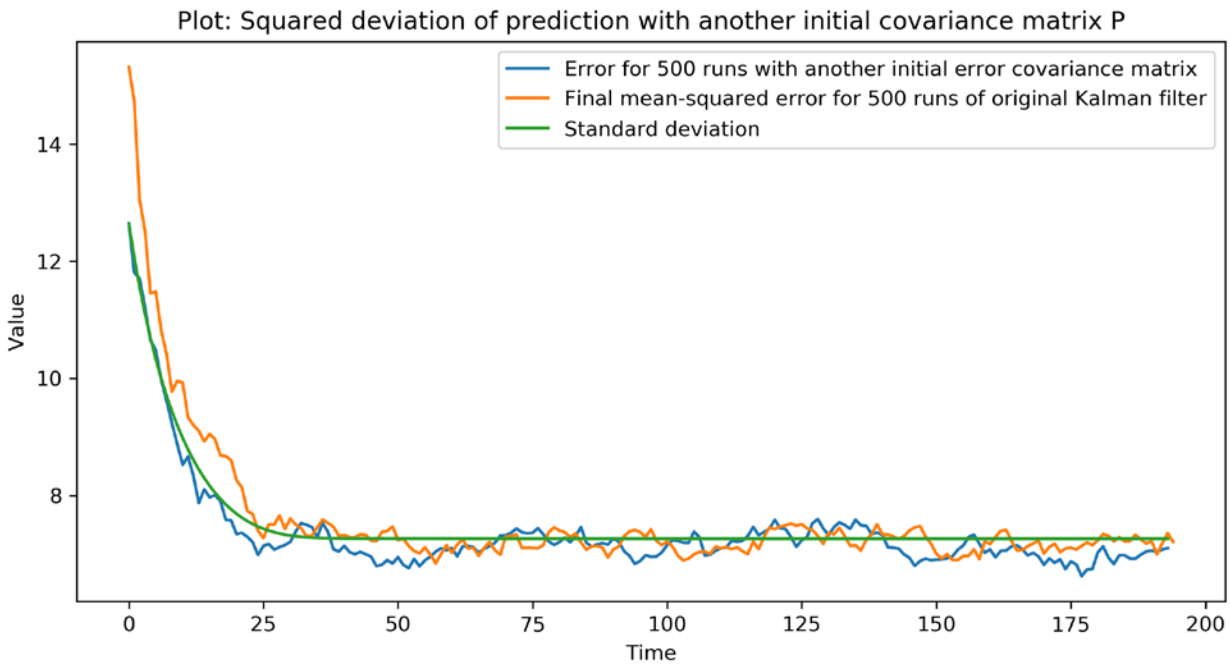

Figure 9

12. When we have determine trajectory (no random disturbance) tru estimation error and errors P tends to zero.

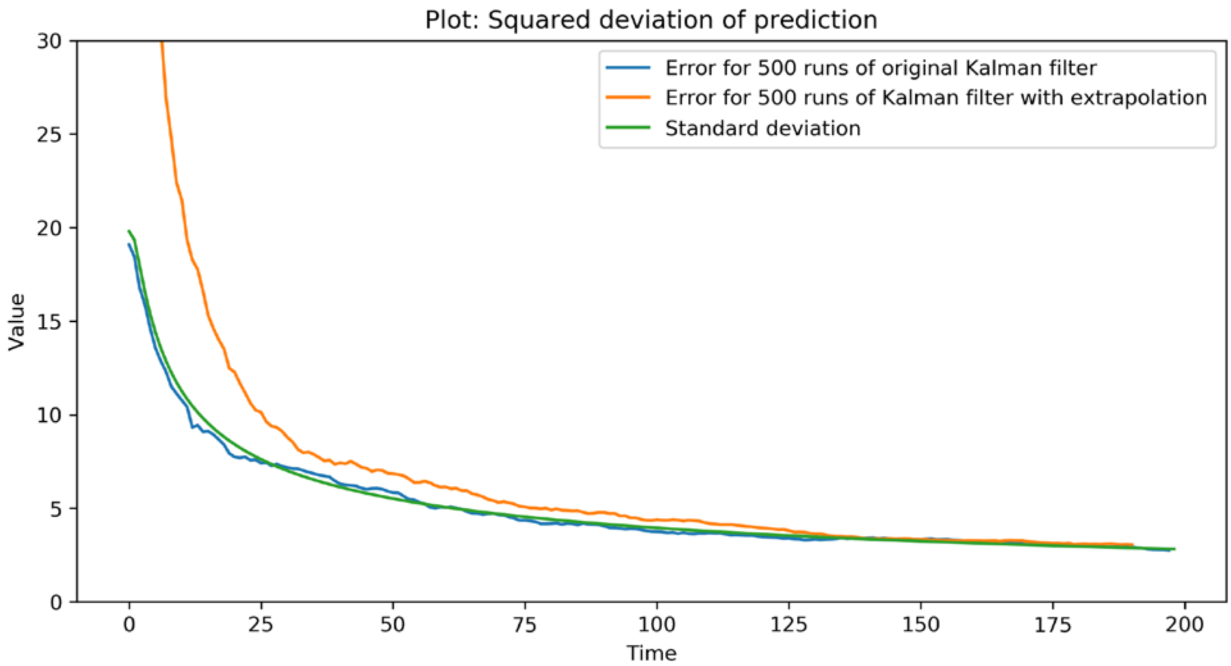

Figure 10

As a result our filter tend to zero too. Deviation of errors are minimal.

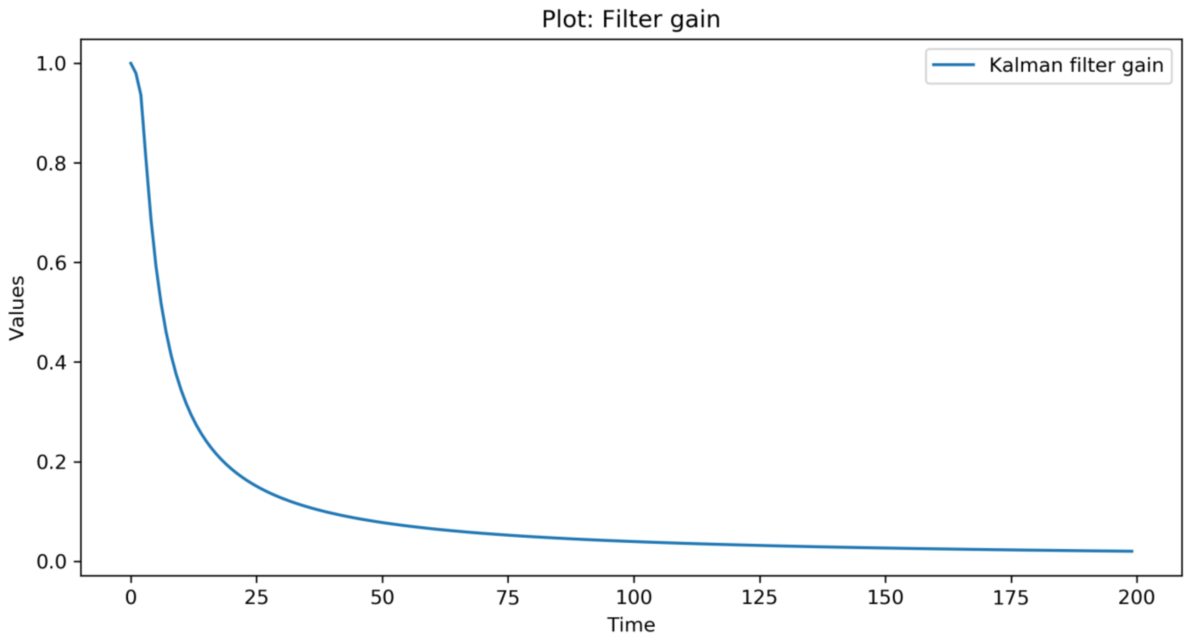

Figure 11

13. On figure 12 we can see, that neglesing state noise in Kalman filter algorithm consequently filter think that it reduce error. In reality true error improves. Filter without any correction won't work properly.

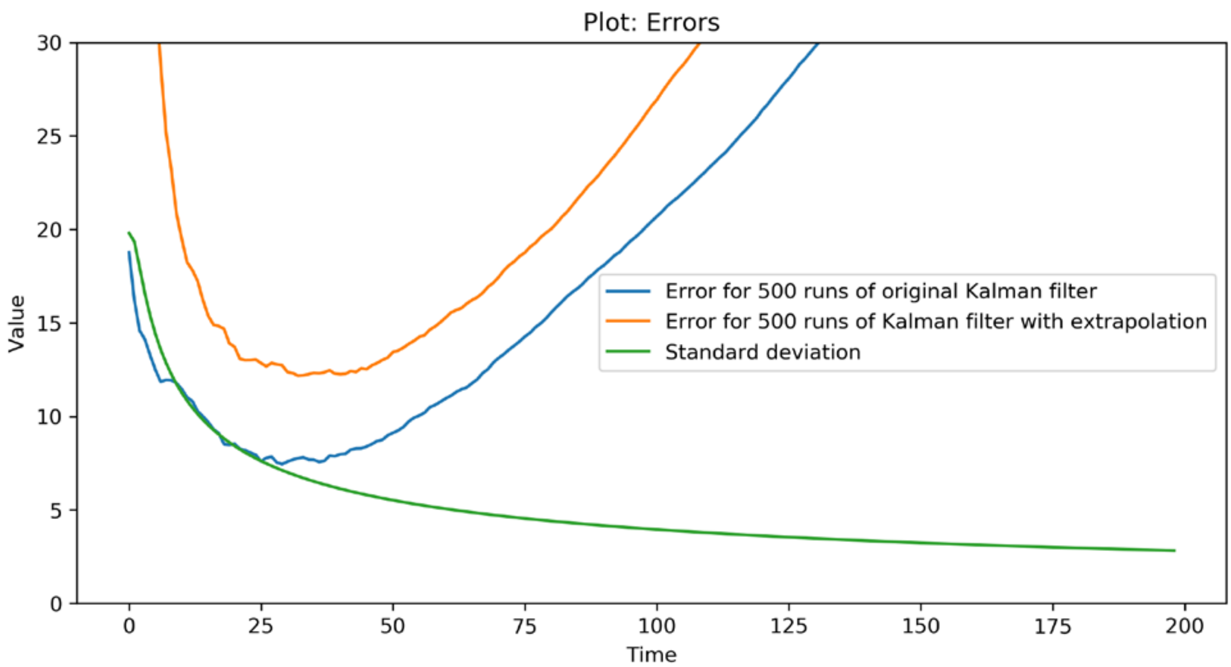

Figure 12

14. We've made two trajectories with different variance of state noise and filtered it to compare how the relation σ_w^2 / σ_n^2 affects on a steady-state filter gain.

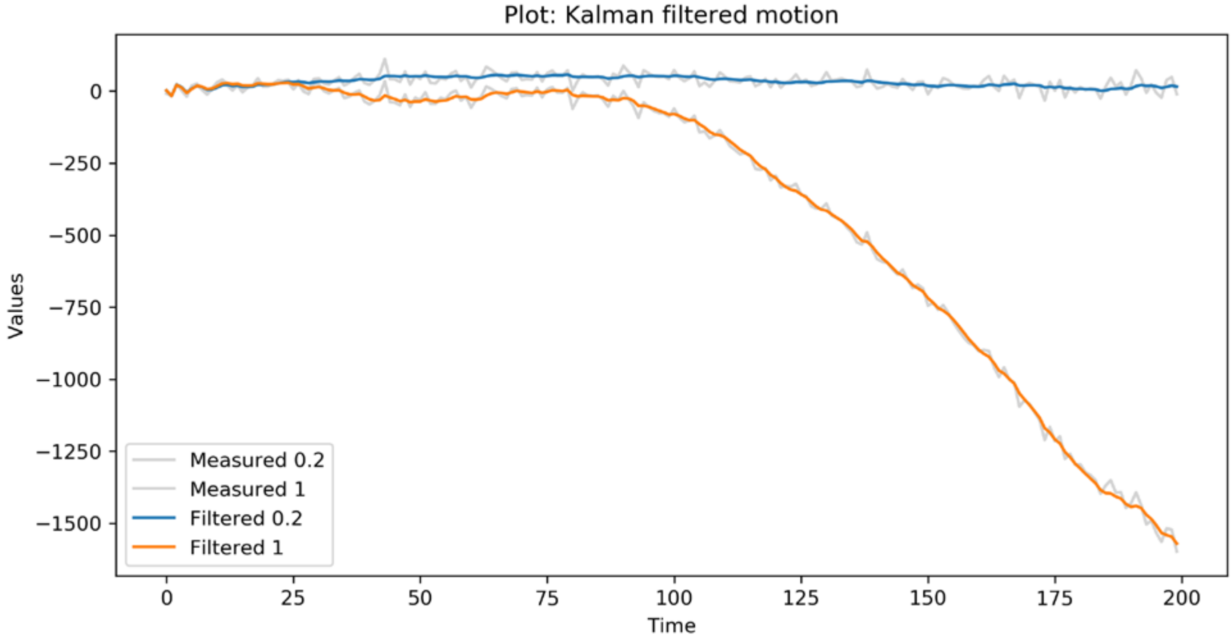

Figure 13 – Generated two trajectories with different variance of acceleration noise

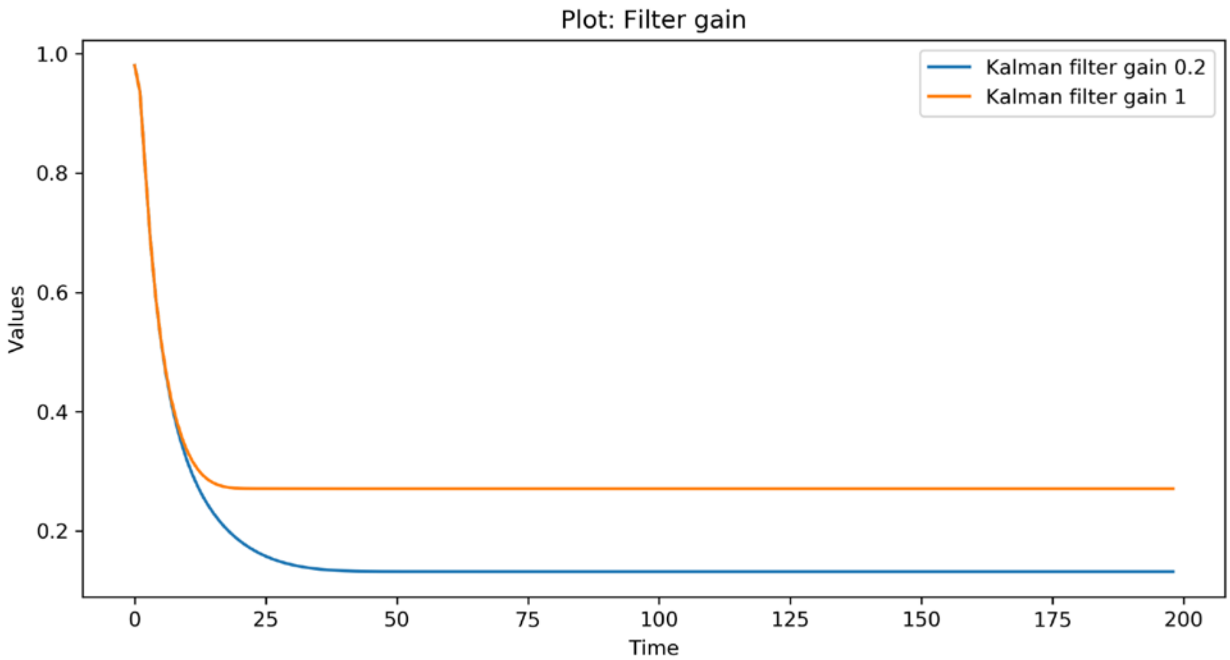

Figure 14 – Filter gain for two trajectories with different variance of acceleration noise

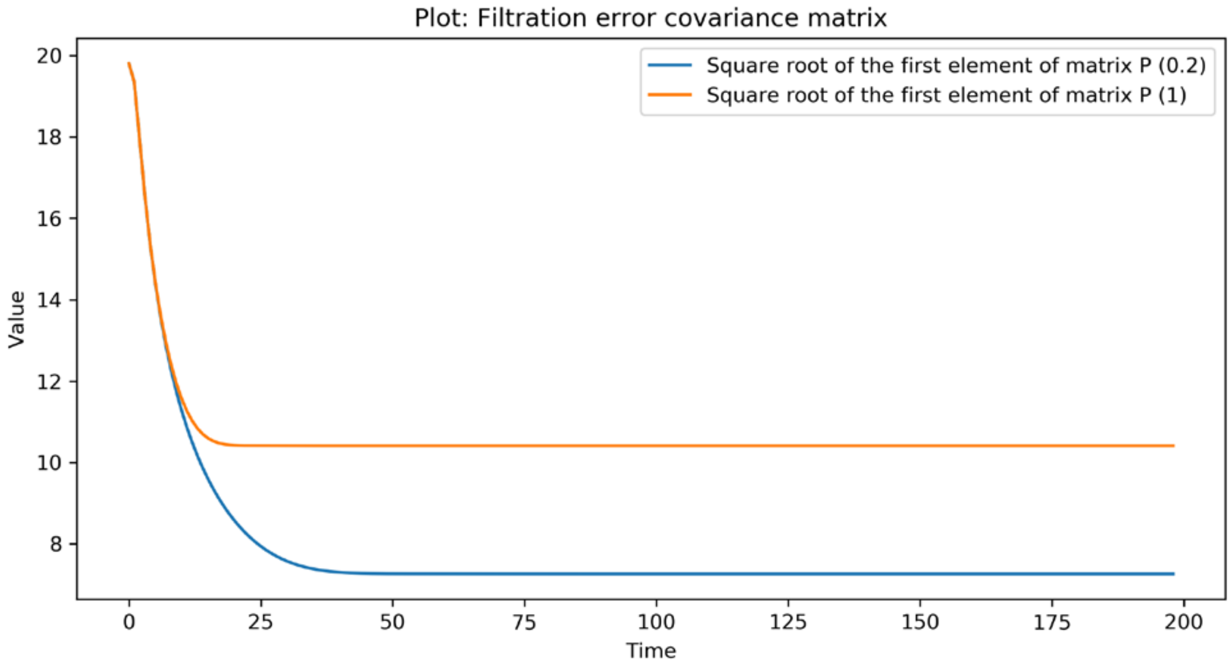

Figure 15 – Standard deviation of two trajectories with different variance of acceleration noise

According to the results and plots, with the higher variance of random noise, the steady-state value of gain and error of filtration will be higher too.

15. We measured sensitivity of filter.

To make a difference we took the underestimated non-optimal filter gain K equaled to $K_{optimal}$ divided by 5. In the figures below, you can see how this gain affects for the filtration.

Value of the initial gain is lower by 5 times in comparison with estimated gain. In addition, time, which was needed for achieving the steady-state value, was increased. The steady-state value decreased too.

By analyzing mean-squared errors, values of underestimated at the start are bigger in comparison with normal gain. Around the 50^{th} point the error values of both filtered results was on the same level. After that the error of curve with underestimated gain increased and got the steady value, that higher compare to curve with normal gain.

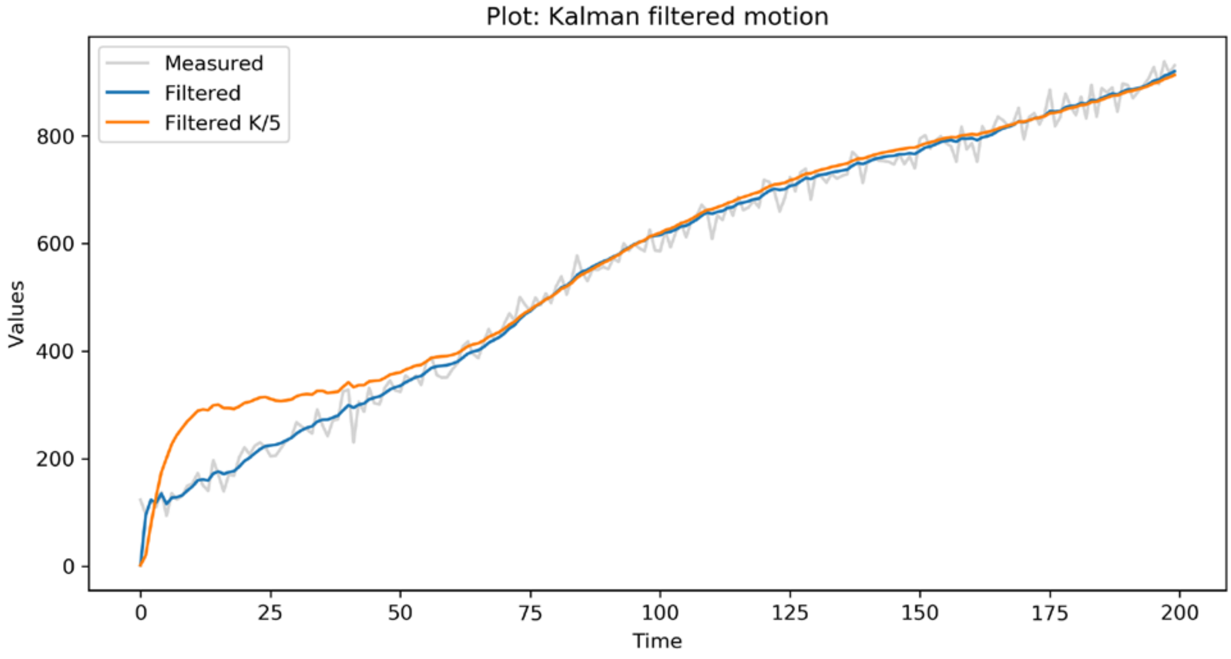

Figure 16 – Filtered measurements with underestimated gain

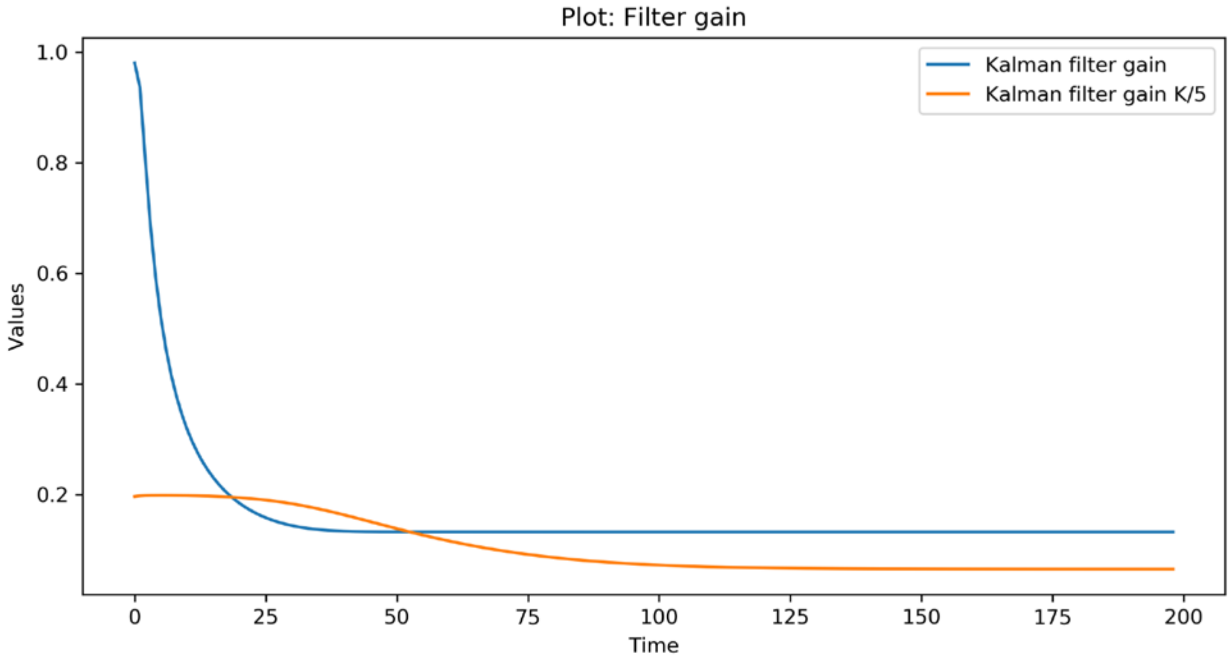

Figure 17

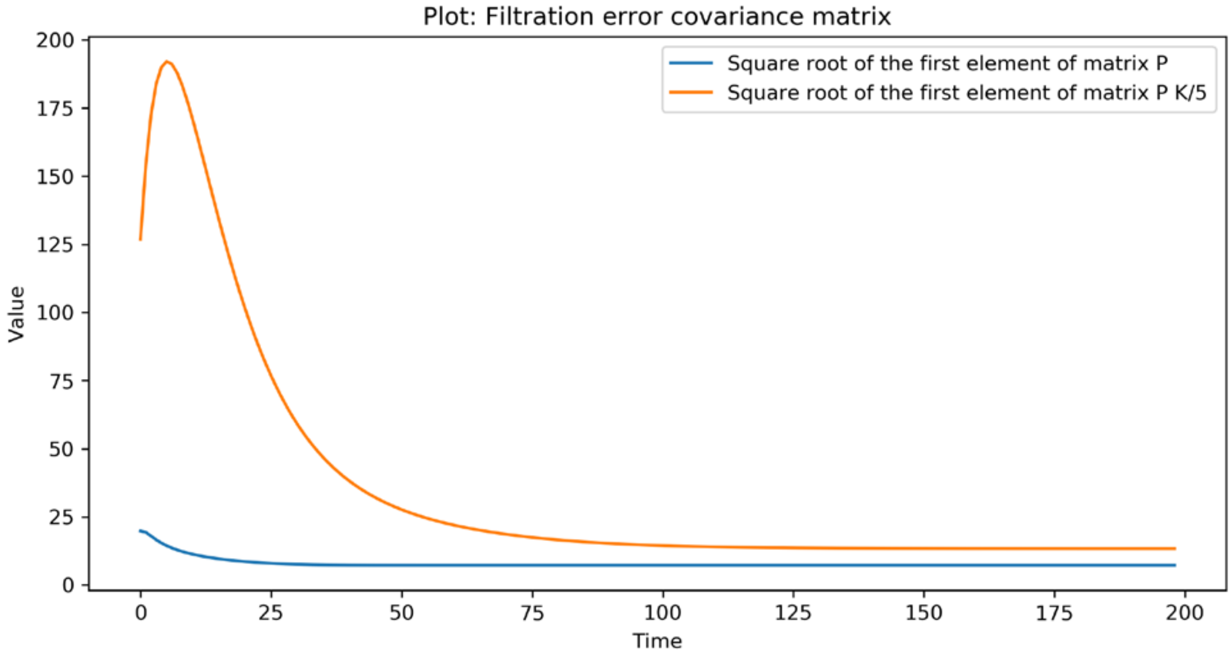

Figure 18

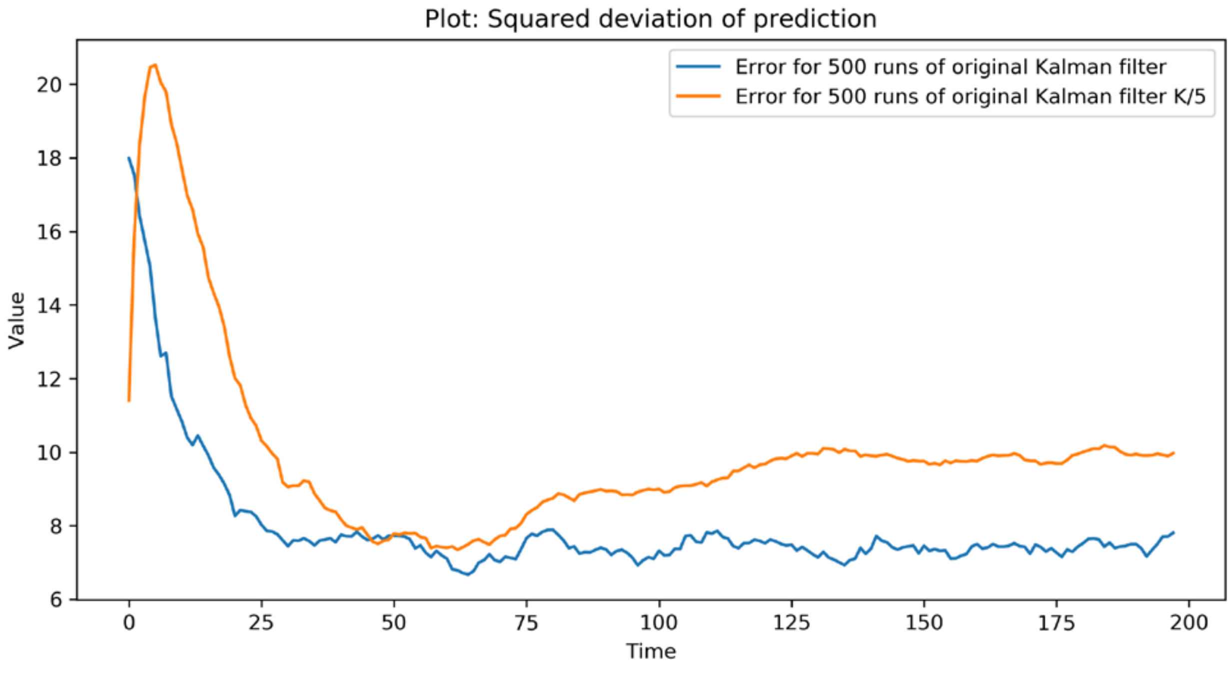

Figure 19

Conclusions

During this assignment, we have learned about Kalman filtering and understood its main principles. We analyzed filter parameters and their relation to the final result and estimation error. Different initial conditions can nullify any profit of using that type of filtering, but with correct initial and model parameters.

In technical, we get a deeper understanding of how to work with matrices in Python.