Assignment 9

Development of optimal smoothing to increase the estimation accuracy

By

Group 19

Ahmed Baza, Skoltech, 2020

Ahmed Elgohary, Skoltech, 2020

Mohammed Sayed, Skoltech, 2020

Ziad Baraka, Skoltech, 2020

1. Generate a true trajectory X_i of an object motion disturbed by normally distributed random acceleration with random measurement gaps

$$
\begin{aligned}
&X_i = X_{i-1} + V_{i-1}T + a_{i-1}\frac{T^2}{2}\\
&V_i = V_{i-1} + a_{i-1}T\\
&\text{If } \zeta > P\\
&\quad Z_i = X_i + \eta\\
&\text{else}\\
&\quad Z_i = \text{NaN}
\end{aligned}
$$

Size of trajectory is 200 points.

Initial conditions: $X_1 = 5;\ V_1 = 0;\ T = 0.1;$

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

Variance of noise Z_i : $\sigma_\eta^2 = 20^2$

```
clc; clear
p=0.2;              %Gap probability
N=200;              %Number of points
normaldist=makedist('Normal',0,0.2);    %acceleration distribution

a=random(normaldist,N,1);             %acceleration

X=zeros(1,N);          %X vector
X(1)=5;                %X initial value
V=zeros(1,N);          %Velocity
V(1)=1;                %Velocity initial value
T=1;                   %Time step

for i=2:N
```

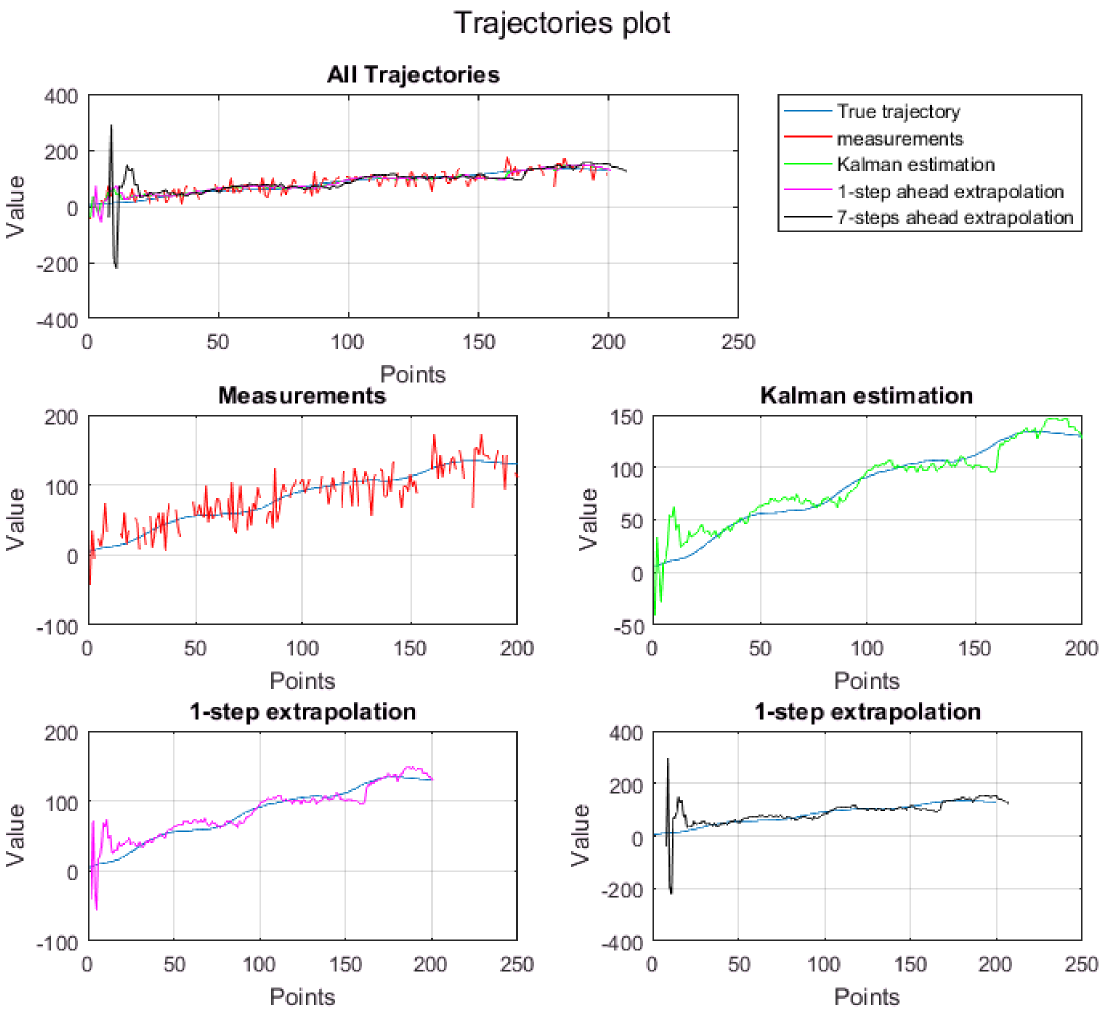
```
    X(i)=X(i-1)+V(i-1)*T+a(i-1)*T^2/2;    %X generation
    V(i)=V(i-1)+a(i-1)*T;       %V generation
end

normaldist2=makedist('Normal',0,20);%Mesurements error distribution
eta=random(normaldist2,200,1);   %Measurements error
Z=zeros(1,200);    %Measurements

Z(1)=X(1)+eta(1);    %measurements initial value
for i=2:N
    zeta=rand;    %gap probability
    if zeta>p
        Z(i)=X(i)+eta(i);    %in case of no gap
    else
        Z(i)=NaN;        %in case of a gap
    end
end
```

2. Present the system at state space taking into account that only measurements of coordinate x_i are availabe

3. Determine filtered and extrapolated errors of estimation (1 step and 7 steps ahead)

```
%State space parameters for Kalman filter
phi=[1 T;0 1];
G=[T^2/2; T];
H=[1 0];
sigma_a=0.2^2;
Q=G*G'*sigma_a;

xi=zeros(1,200);   %Kalman Estimated states vector

xiE1=zeros(1,200);  %Kalman Extrapolated 1-step states vector
xiE7=zeros(1,200);  %Kalman Extrapolated 7-step states vector
%initial conditions
Xi=[2;0];
P=[10000 0; 0 10000];

R=20^2;   %R matrix
K=P*H'/(H*P*H'+R);  %Kalman gain

Pi=zeros(N,1);    %X-position standard deviation vector
Piv=zeros(N,1);   %Velocity standard deviation vector

%Kalman filter
for i=1:N
    if isnan(Z(i))   %Gap check
        Xi=phi*Xi;
        P=phi*P*phi'+Q;
    else          %in case of no gap
        Xi=phi*Xi;
        P=phi*P*phi'+Q;
        Xi=Xi+K*(Z(i)-H*Xi);
        K=P*H'/(H*P*H'+R);
        P=(eye(2)-K*H)*P;
    end

    xi(i)=Xi(1);
    %Extrapolation 1-step
    XiE=phi*Xi;
```

```
    xiE1(i)=XiE(1);

    %Extrapolation 7-step
    XiE=(phi^6)*XiE;
    xiE7(i)=XiE(1);
end

%plotting trajectory for visulization
subplot(3,2,1:2)
    plot(X)
    hold on
    plot(Z,'r')
    plot(xi,'g')
    plot(2:N+1,xiE1,'m')
    plot(8:N+7,xiE7,'k')
    title('All Trajectories')
    legend({'True trajectory','measurements','Kalman estimation', ...
        '1-step ahead extrapolation','7-steps ahead extrapolation'} ...
        ,'location','northeastoutside')
    grid on
    xlabel('Points')
    ylabel('Value')
    set(gcf,'position',[0,0,800,800]);

    subplot(3,2,3)
    plot(X)
    hold on
    plot(Z,'r')
    grid on
    title('Measurements')
    xlabel('Points')
    ylabel('Value')
    subplot(3,2,4)
    plot(X)
    hold on
    plot(xi,'g')
    grid on
    title('Kalman estimation')
    xlabel('Points')
    ylabel('Value')
    subplot(3,2,5)
    plot(X)
    hold on
    plot(2:N+1,xiE1,'m')
    grid on
    title('1-step extrapolation')
    xlabel('Points')
    ylabel('Value')
    subplot(3,2,6)
    plot(X)
    hold on
    plot(8:N+7,xiE7,'k')
    grid on
    title('1-step extrapolation')
    xlabel('Points')
    ylabel('Value')
    suptitle('Trajectories plot')
```

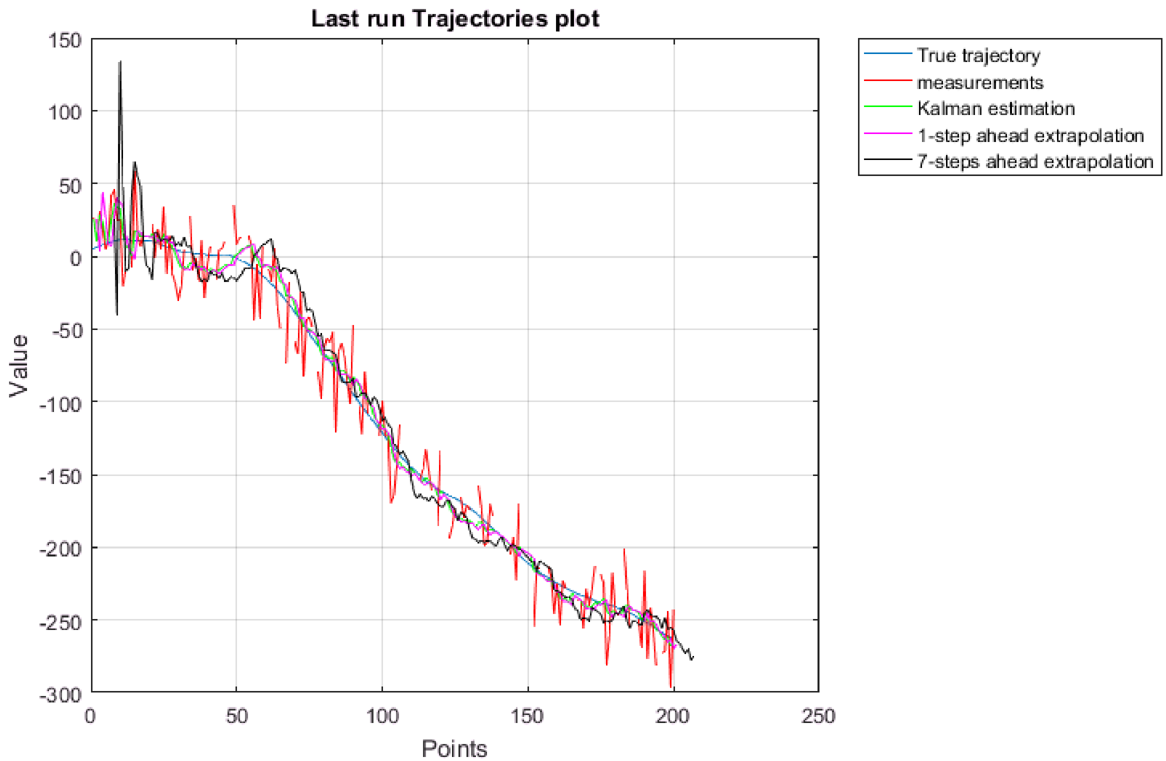


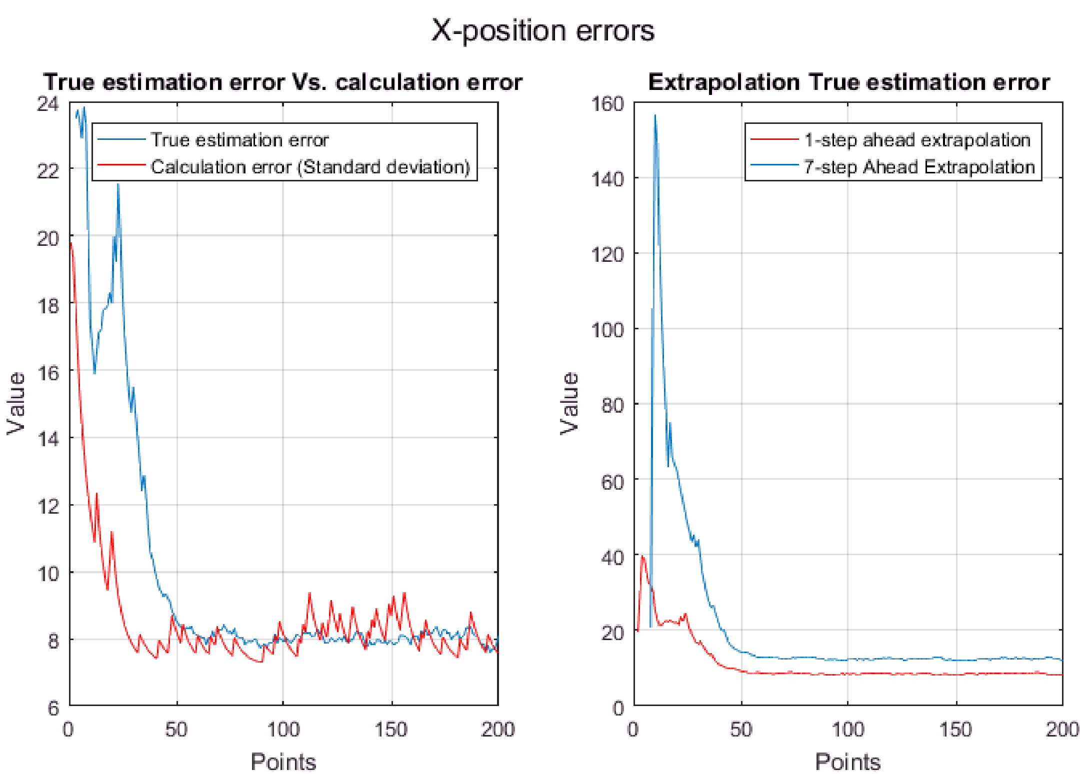
Comment:

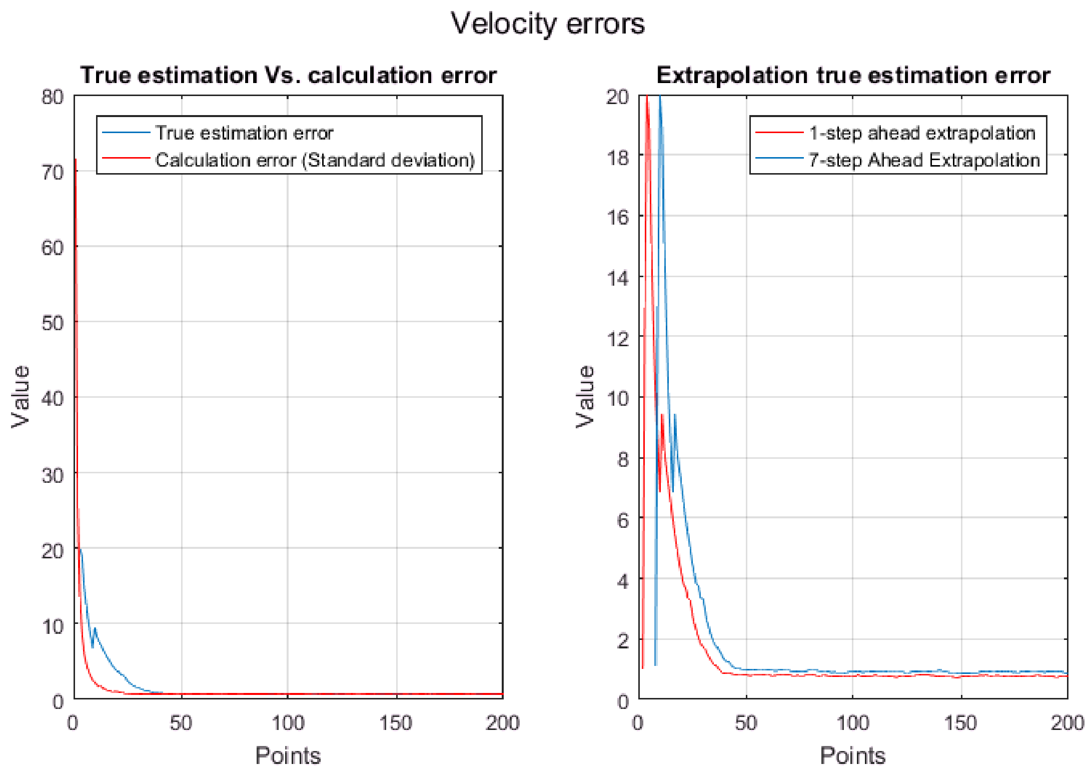
In the trajectories plot above, the true trajectory, the measurements, the Kalman filter estimation and extapolation with 1-step and 7 steps ahead are graphed against each other. It can be seen that the estimation values are kind of following the trajectory with some disturbances. And plotting the filter with different random trajectories leads to the same results. Also, the extrapolation follows the behavior of the estimation more, which makes sense as it's based on it. Moreover, it's more noisy at the beginning of the trajectory so it cannot be trusted at this region as an estimation of the true model, and lastly we can predict that both the error of filter estimation and the extrapolation will have the same behavior with close values, with the 1-step ahead closer to the estimated values, which we will analyze in the following point in terms of error.

3. Determine filtered and extrapolated errors of estimation (1 step and 7 steps ahead) over 500 runs of filter. Compare them with true estimation errors.

```
clc;clear
% code for M=500 runs
figure
Kalman_with_gaps(500,0.2)
```


Last run Trajectories plot
True trajectory
measurements
Kalman estimation
1-step ahead extrapolation
7-steps ahead extrapolation
Value
Points
150
100
50
0
-50
-100
-150
-200
-250
-300
0
50
100
150
200
250


X-position errors
True estimation error Vs. calculation error
True estimation error
Calculation error (Standard deviation)
Value
Points
24
22
20
18
16
14
12
10
8
6
0
50
100
150
200
Extrapolation True estimation error
1-step ahead extrapolation
7-step Ahead Extrapolation
Value
Points
160
140
120
100
80
60
40
20
0
0
50
100
150
200



Comment:

Plotting the true error and the calculated error in the position, we can see that the behaviour is slightly different than what we got before using Kalamn with full position state measurements in previous assignments, after the error reaches a steady state there were some jumps in the error magnitude due to the gaps in measurements. and in our case it's best approximation that we can come up with.

Also, we can see that the calculated error can clearly follow the same trend as the true error, that means even with the lack of measurements at some points our filter can still provide acceptable results.

The second error plot shows the difference between the 1 step ahead extrapolation and the 7 steps ahead extrapolation. You can see the error in the 1 step extrapolation is very close to the true estimation error in the previous graph, however when the extrapolation steps increases the error increase as at some point you will be extrapolating on a predicted value not filtered with measurement so its normal to expect some errors. As overall performance the filter is still able to introduce acceptable results.

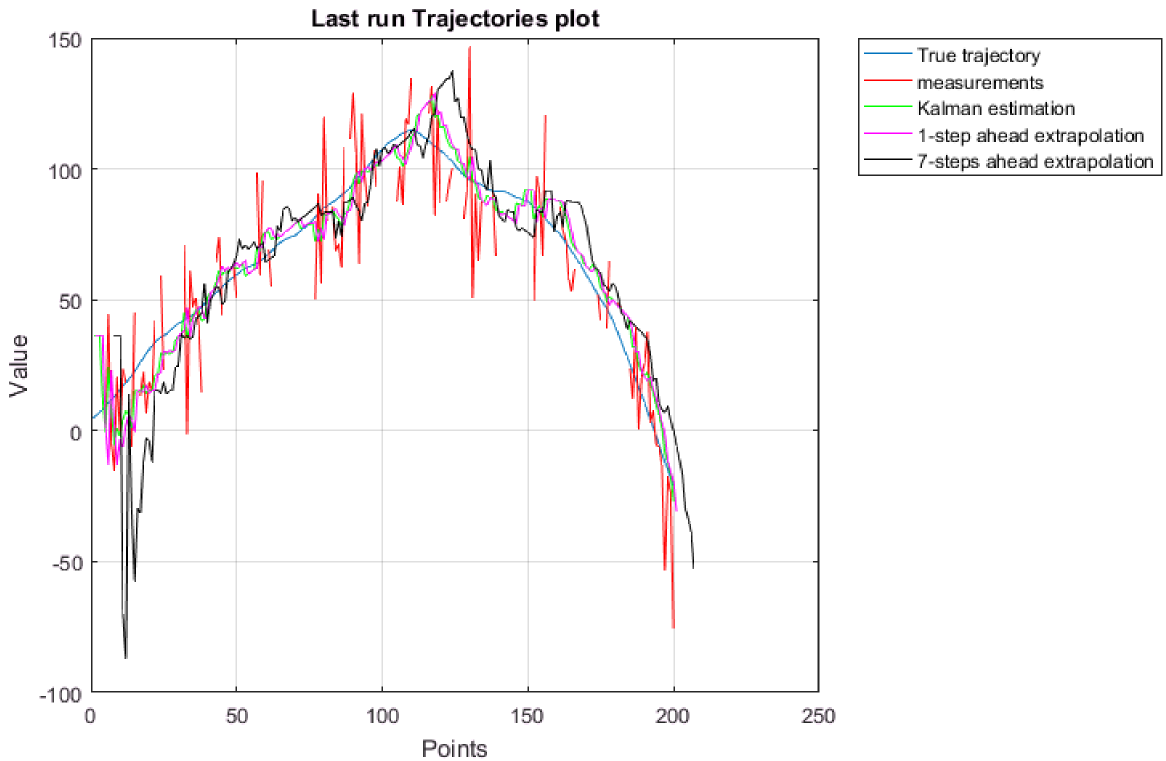
The velocity estimation error has the same traits of the position error, but with the extrapolated velocity error exactly similar in shape to the true estimation error of estimated velocity and close in value.

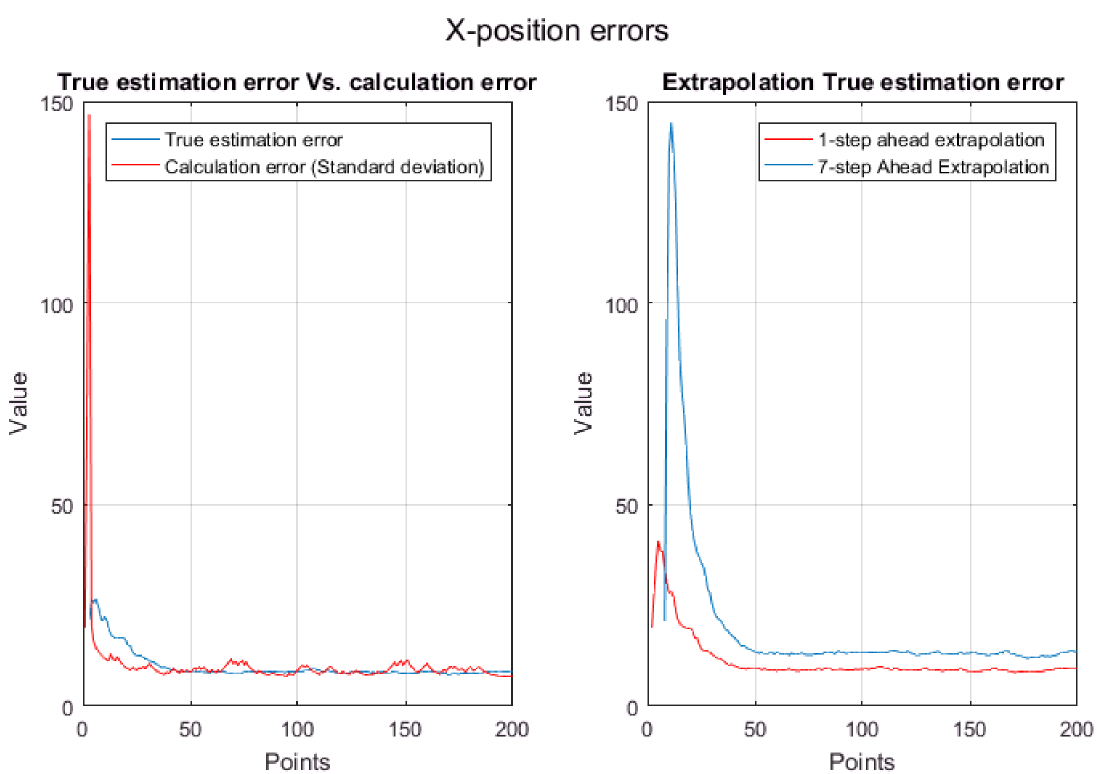
4. Analyze the decrease of estimation accuracy in conditions of measurement gaps.

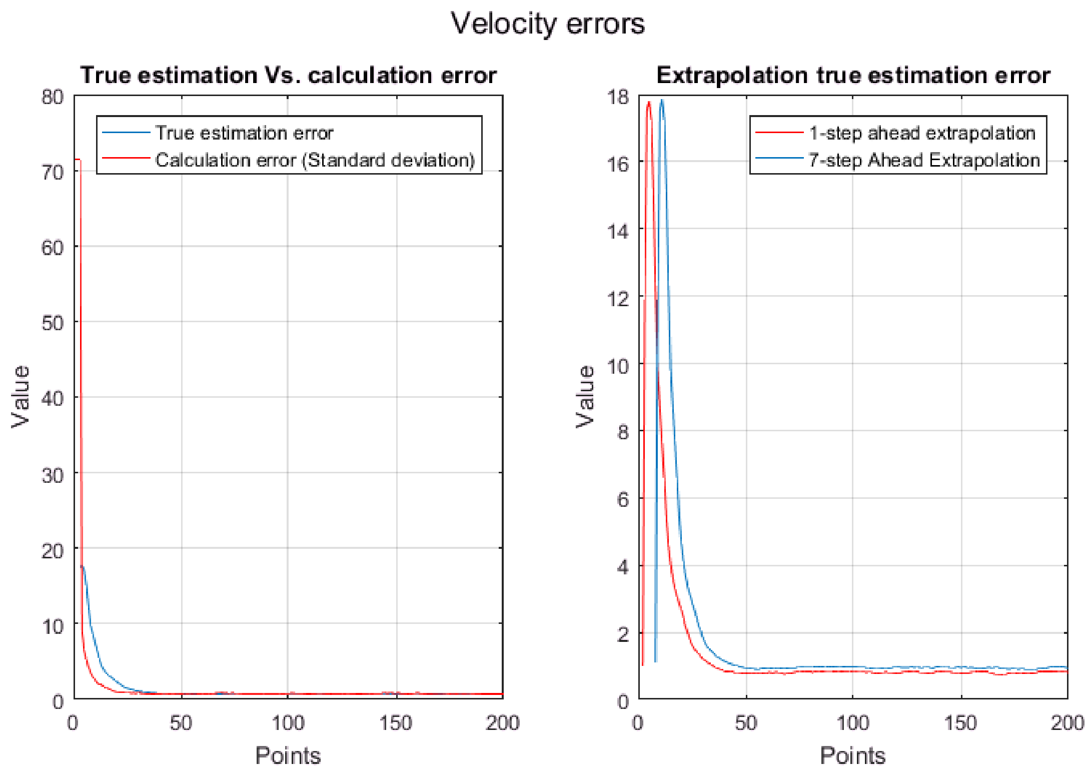
Compare results when of probability of measurement gaps is

a) p=0.3

```
clc;clear
% code for M=500 runs
figure
Kalman_with_gaps(500,0.3)
```


Last run Trajectories plot
True trajectory
measurements
Kalman estimation
1-step ahead extrapolation
7-steps ahead extrapolation
Value
Points


X-position errors
True estimation error Vs. calculation error
True estimation error
Calculation error (Standard deviation)
Extrapolation True estimation error
1-step ahead extrapolation
7-step Ahead Extrapolation
Value
Points



Comment:

As we increase the probability of creating gaps or the number of gaps in data we can notice two things:

1. The error takes a greater number of time steps to get to a steady state.
2. The magnitude of the mean error is greater than what it was with smaller numbers of gaps.

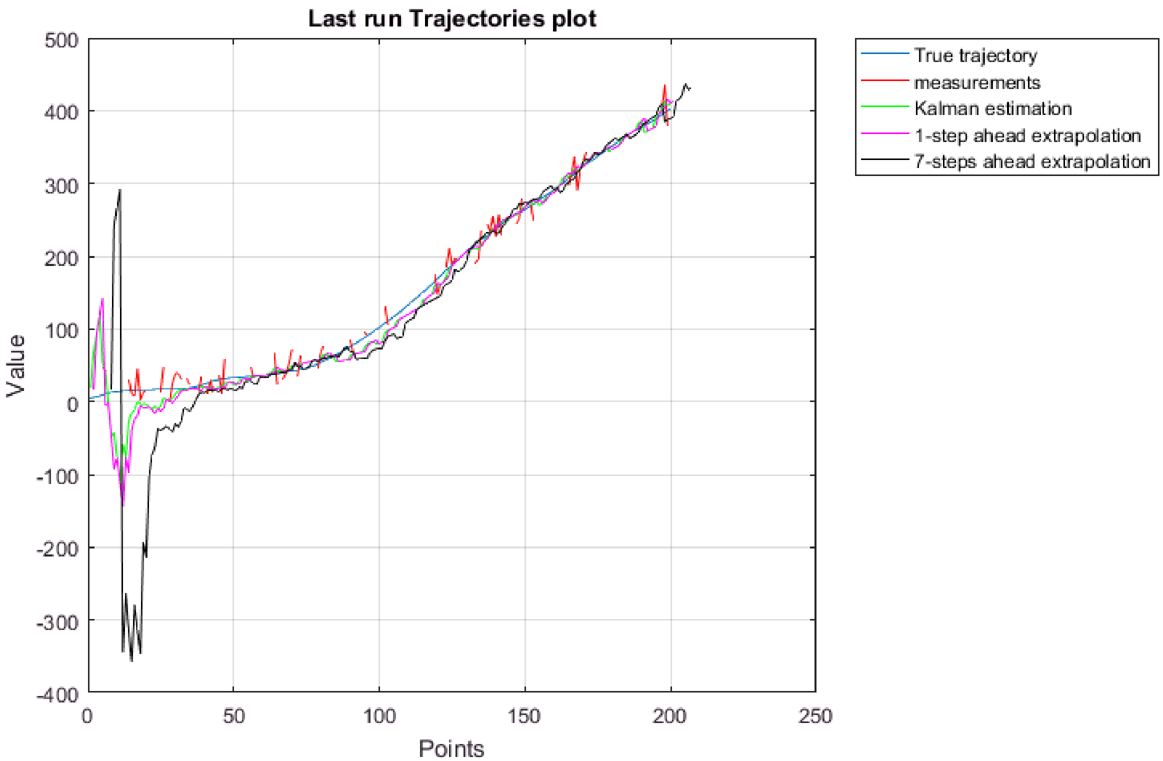
With a greater number of gaps the filter takes more time steps to get enough measurement values that can get the kalman gain tuned, that leads to the delay in achieving the steady state error.

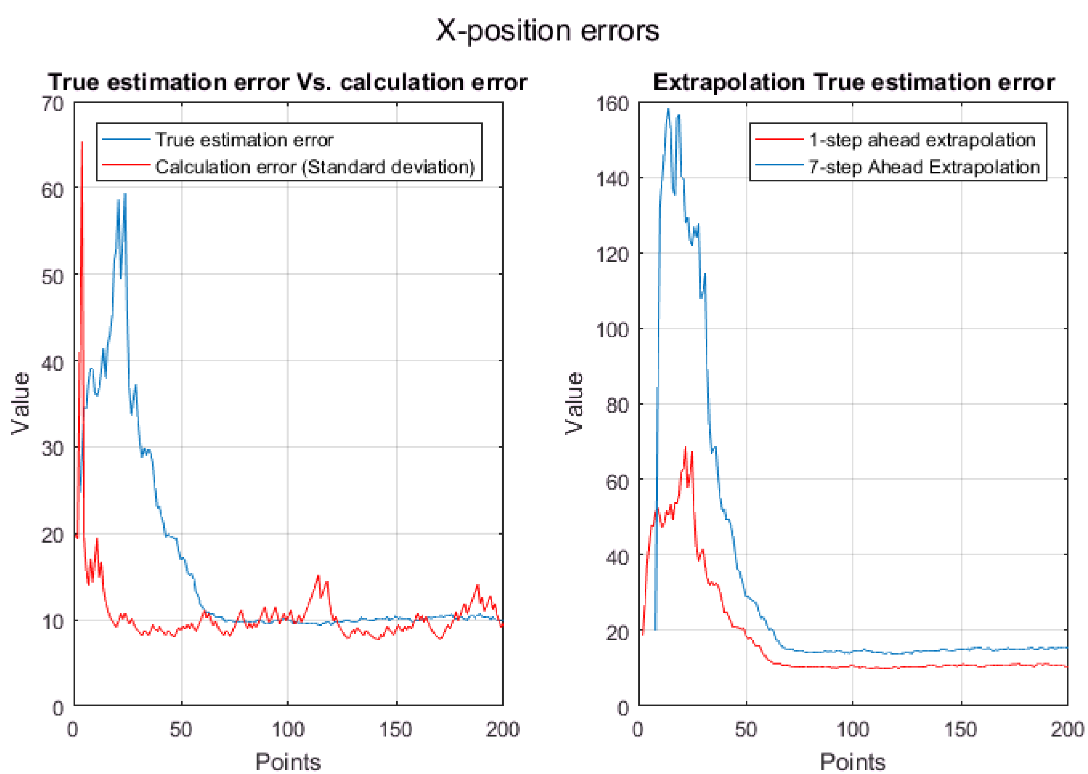
Also in the case of a small number of gaps the filter can quickly improve the predicted gain (using predicted P matrix) estimated by a measurement point in the next iteration, however with the high number gaps the value of the gain don’t change quickly enough to the point of increasing the mean error values.

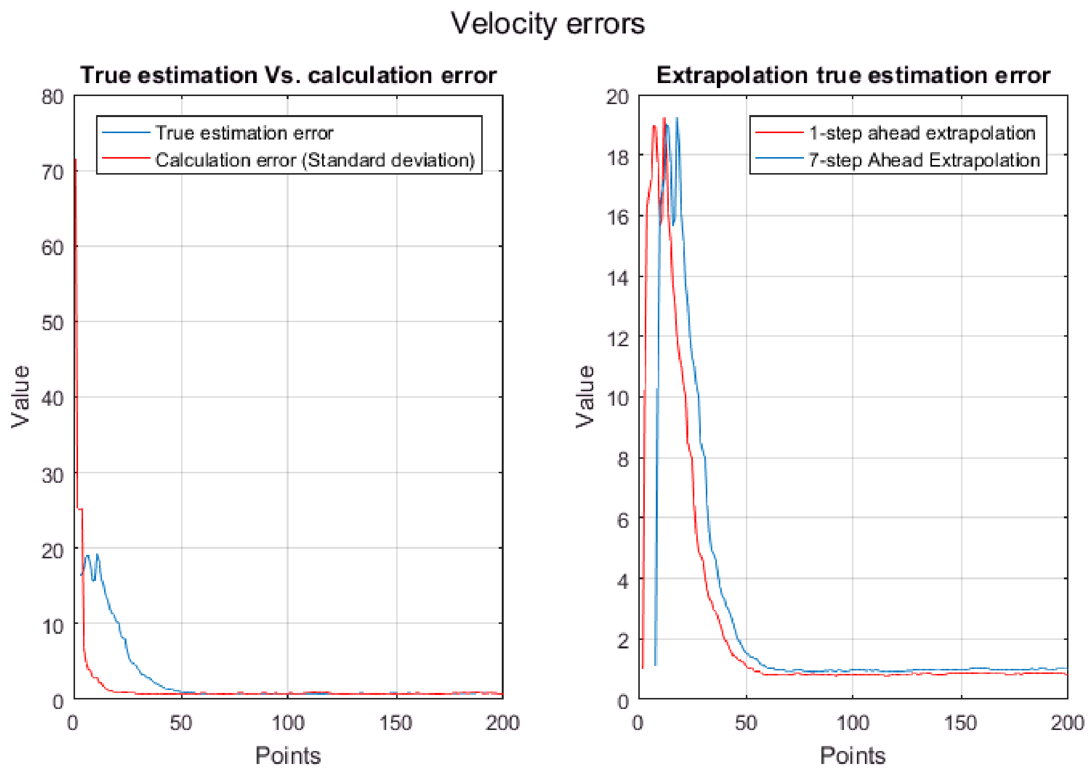
Analyzing the errors for position and velocity we can see that they increased slightly from being around the 8 when p was equal to 0.2 to around 8.5 here which isn't a significant increase in value, and yet the same peaks trend appeared.

b) p=0.5

```
% code for M=500 runs
figure
Kalman_with_gaps(500,0.5)
```


Last run Trajectories plot
True trajectory
measurements
Kalman estimation
1-step ahead extrapolation
7-steps ahead extrapolation
Value
Points
500
400
300
200
100
0
-100
-200
-300
-400
0
50
100
150
200
250


X-position errors
True estimation error Vs. calculation error
True estimation error
Calculation error (Standard deviation)
Value
Points
70
60
50
40
30
20
10
0
0
50
100
150
200
Extrapolation True estimation error
1-step ahead extrapolation
7-step Ahead Extrapolation
Value
Points
160
140
120
100
80
60
40
20
0
0
50
100
150
200



Comment:

P=0.5 this means we lost almost half of our measurements. Which can be visualizied from the

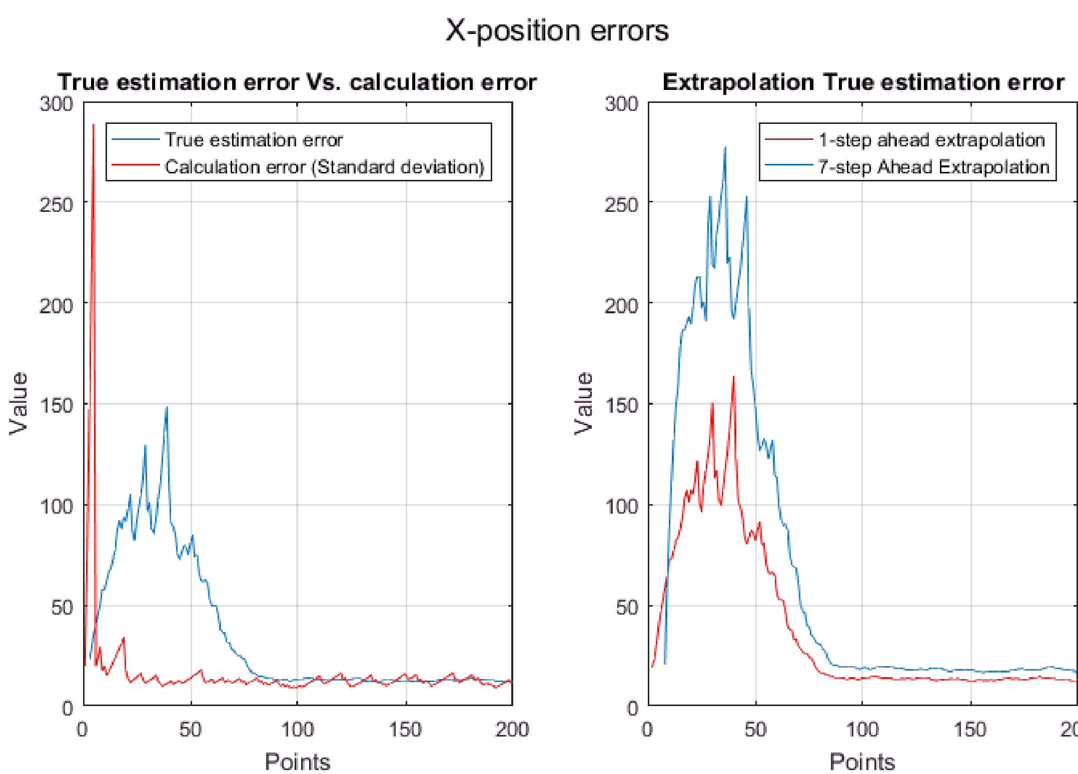
As we can see the calculated error starts to deviate from the true error. As we discussed before losing data points prevents the filter from updating the values of its gain and calculated error making it slower to get tuned gain. Thus increasing the mean error and deviating from the true error and hence the true trajectory. You can also see the same behaviour in the velocity estimate. The values of the extrapolated error are also getting higher due to the same reasons.

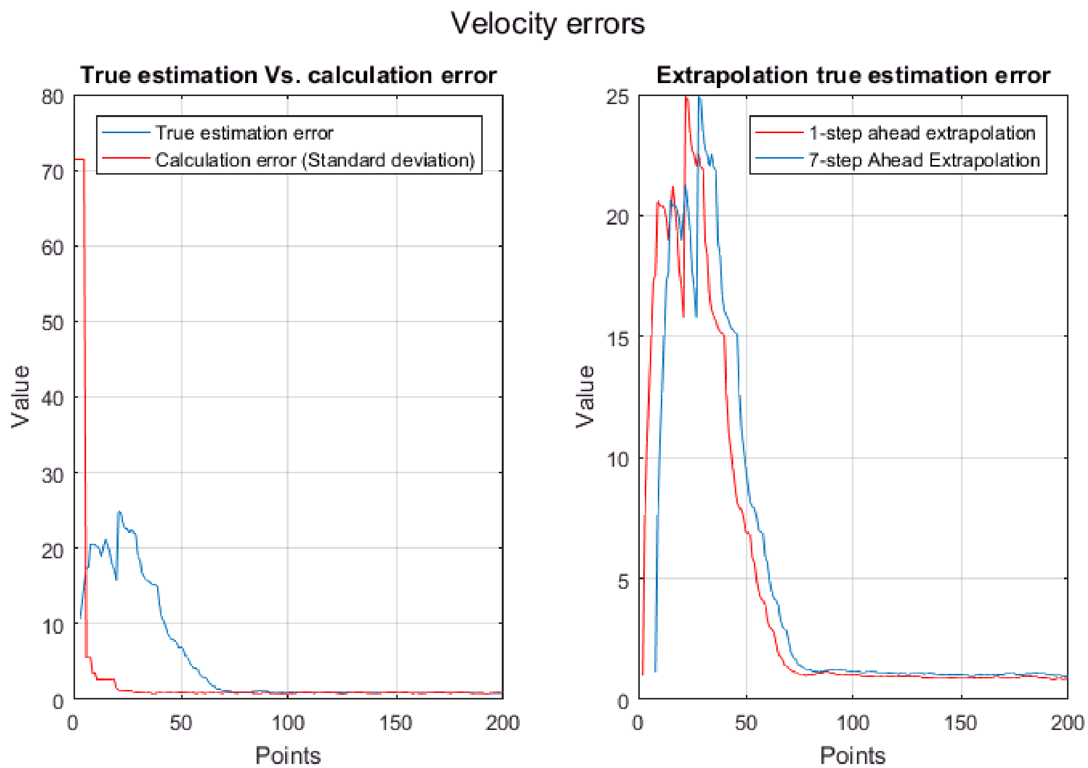
Analyzing the errors for position and velocity we can see that they increased but this time more than the last one, from being around the 8 when p was equal to 0.2 and around 8.5 here to almost 10 which a higher increase in value, and the same peaks trend appeared.

c) p=0.7

```
% code for M=500 runs
figure
Kalman_with_gaps(500,0.7)
```


Last run Trajectories plot
Value
Points
True trajectory
measurements
Kalman estimation
1-step ahead extrapolation
7-steps ahead extrapolation


X-position errors
True estimation error Vs. calculation error
True estimation error
Calculation error (Standard deviation)
Value
Points
Extrapolation True estimation error
1-step ahead extrapolation
7-step Ahead Extrapolation
Value
Points



Comment:

P=0.7. we have only 30% of our data. which from the visualization of the trajectories we can see how the kalman filter plot was kind of discrete than a continous function and that gave us insight on how the error behaved.

The filter takes almost half the time steps to tune its gain, the deviation in error is clearly greater also the steady state error. You can see in the first graph of the late run in 500 runs that the filter in ealy time steps is trying to follow the available parts of measurement (the gain is't tuned yet) that results in bad estimation of the trajectory also very bad extrapolation with error values very high comparing to the standard deviation of the noise.

General conclusion:

Looking for estimation accuracy in our analysis we can declare, and the logic would agree, that the more measurements we had the more accurate the estimation was, and that was clear when we analysed for different gap propability values, and the error was constantly increasing from the less to probabilty to the highest. which is exteremly logical to understand because we have more data to analys and make trajectory of.

Learning Log:

In this assignment we tried to test the effect of measurement gaps on the performance of a Kalman filter. As we reflected upon this case we can put what we learned in the following points:

1- The Kalman filter can get good estimation with the presence of some gaps in the measurement set but you always have to check your results when the gaps number increases as the filter performance starts to deteriorate.

2- In the case of gaps, you can't trust the extrapolated estimates in the early stages of tuning. Thus if you want to extrapolate using Kalman filter you will need to provide some data to tune the gain and then use the extrapolated estimates.

3- Determining the acceptable number of gap points will be hard in a real application as we don’t know the true trajectory, based on what we tried here we can assume that below 20 % would result in good approximation, to really decide we should be familiar with the data in hand, and what percentage of error is acceptable and what isn’t.