Assignment 9: Tracking and forecasting in conditions of measurement gaps

Group 6:

- **Andrei Shumeiko**
- **Ayush Gupta**
- **Olga Klyagina**
- **Simona Nitti**


```
clc
clear
close all
```

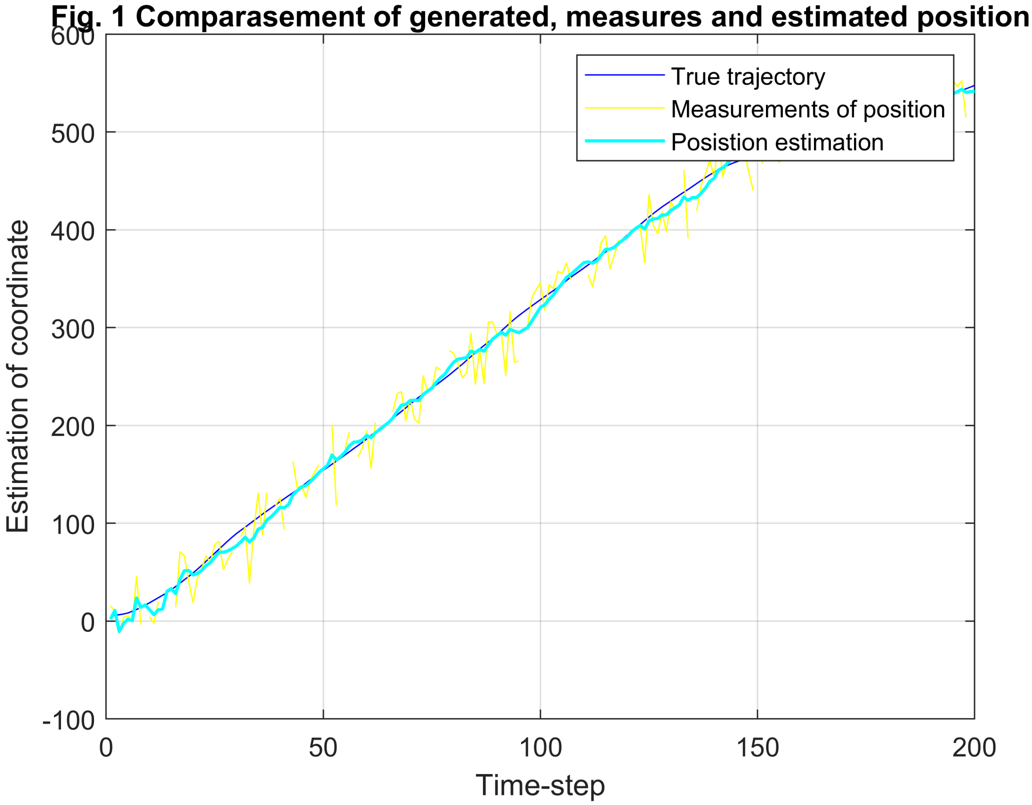
Task 1: Generate a true trajectory and create measurements with gaps

```
n = 200;
x = 5*ones(1,n);
V = ones(1,n);
T = 1;
var_a = 0.2^2;
a = sqrt(var_a)*randn(1,n);

for i = 2:n
    x(i) = x(i-1)+V(i-1)*T+a(i-1)*T^2/2;
    V(i) = V(i-1)+a(i-1)*T;
end

var_eta = 20^2;
eta = sqrt(var_eta)*randn(1,n);
p = 0.2; % Probability of measurement gaps
ksi = rand(1,n);
z = zeros(1,n);

for i = 1:n
    if ksi(i) <= p
        z(i) = NaN;
    else
        z(i) = x(i)+eta(i);
    end
end
```

Task 2: . Develop Kalman filter to track moving object under this conditions.

```
phi = [1 T; 0 1]; % transition matrix that relates X(i) and X(i-1);
G = [T^2/2 T].';  % input matrix, that determines how random acceleration affects state vector;
```

```
H = [1 0];        % observation matrix
R = var_eta;
Q = G*G'*var_a;

X(:,1) = [2 0]';
P = 1e4*eye(2);

K = zeros(2,n);
X_pred = zeros(2,n);
for i = 2:n
    X_pred(:,i) = phi*X(:,i-1);
    P_pred = phi*P*phi'+Q;
    K(:,i) = P_pred*H'/(H*P_pred*H.'+R);

    if isnan(z(i))
        X(:,i) =  X_pred(:,i);
        P = P_pred;
    else
        X(:,i) = X_pred(:,i) + K(:,i)*(z(i)-H*X_pred(:,i));
        P = (eye(2)-K(:,i)*H)*P_pred;
    end
end
K(:,1)= [NaN,NaN];

figure
plot(x,'b')
hold on
plot(z,'y')
plot(X(1,:),'c','linewidth',1.2)
title('Fig. 1 Comparasement of generated, measures and estimated position')
xlabel('Time-step')
ylabel('Estimation of coordinate')
legend('True trajectory','Measurements of position',...
    'Posistion estimation')
grid on
```

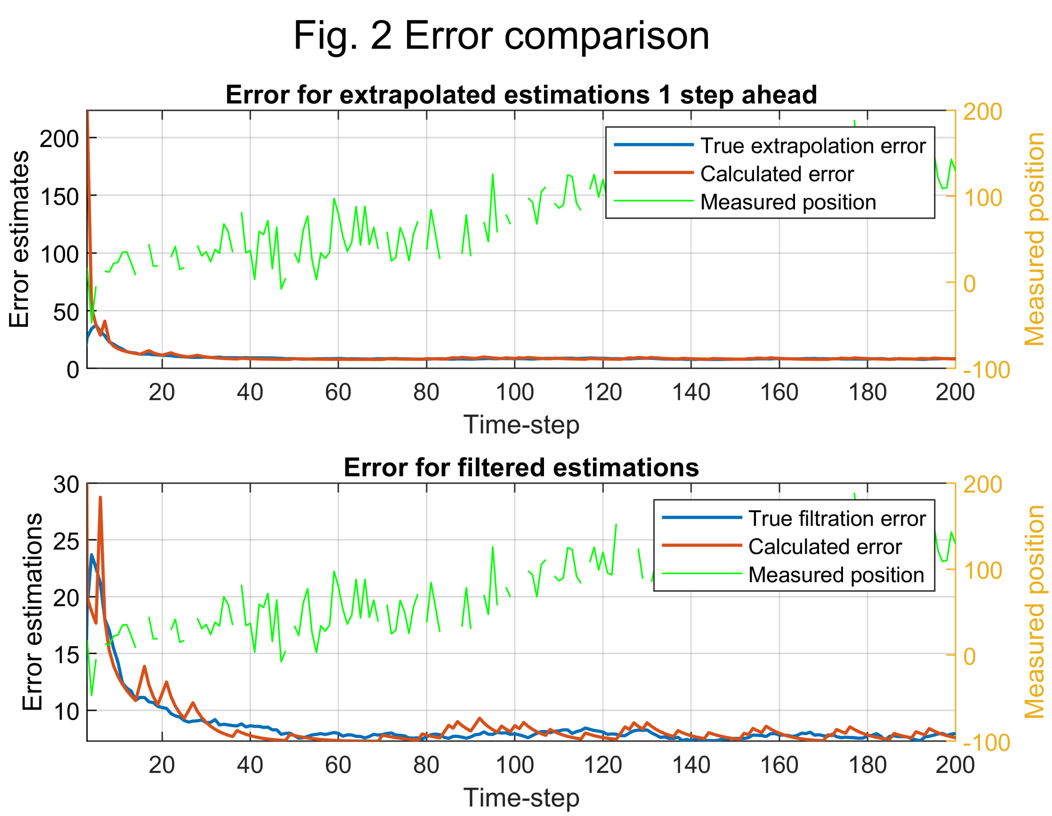


Task 3: Determine filtered and extrapolated errors of estimation (1 step and 7 steps ahead) over 500 runs of filter.

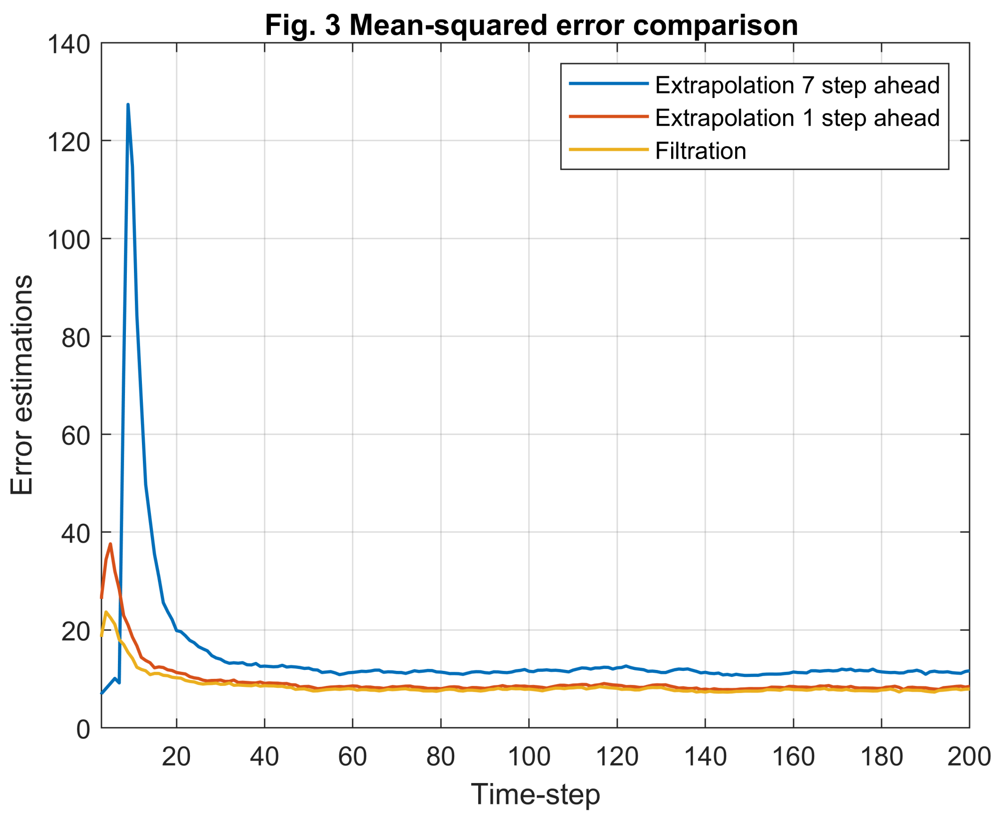
```
M = 500;

Final_Error_estr = zeros(1,n);
error_estr = zeros(1,n);
Final_Error_estr_7 = zeros(1,n);
error_estr_7 = zeros(1,n);
Final_Error_filt = zeros(1,n);
error_filt = zeros(1,n);
for run = 1:M
    x = 5*ones(1,n);
    V = ones(1,n);
    a = sqrt(var_a)*randn(1,n);

    for i = 2:n
        x(i) = x(i-1)+V(i-1)*T+a(i-1)*T^2/2;
        V(i) = V(i-1)+a(i-1)*T;
    end

    eta = sqrt(var_eta)*randn(1,n);
    ksi = rand(1,n);
    z = zeros(1,n);

    for i = 1:n
        if ksi(i) <= p
```

```
            z(i) = NaN;
        else
            z(i) = x(i)+eta(i);
        end
    end

    X(:,1) = [2 0]';
    P = 1e4*eye(2);
    sigma_x = sqrt(P(1,1))*ones(1,n);
    K = zeros(2,n);
    X_pred = zeros(2,n);
    for i = 2:n
        X_pred(:,i) = phi*X(:,i-1);
        P_pred = phi*P*phi'+Q;
        calculated_pred_error(i) = sqrt(P_pred(1,1));

        K(:,i) = P_pred*H'/(H*P_pred*H.'+R);
        if isnan(z(i))
            X(:,i) =  X_pred(:,i);
            P = P_pred;
        else
            X(:,i) = X_pred(:,i) + K(:,i)*(z(i)-H*X_pred(:,i));
            P = (eye(2)-K(:,i)*H)*P_pred;
        end
        sigma_x(i) = sqrt(P(1,1));
    end
    K(:,1)= [NaN,NaN];

    m = 7;
    for i = 1:n-m+1
        X_estr(:,i+m-1) = phi^(m-1)*X(:,i);
    end

    for i = 3:n
        error_estr_7(i) = (x(1,i)-X_estr(1,i))^2;
        Final_Error_estr_7(i)= Final_Error_estr_7(i)+error_estr_7(i);

        error_filt(i) = (x(1,i)-X(1,i))^2;
        Final_Error_filt(i)= Final_Error_filt(i)+error_filt(i);

        error_estr(i) = (x(1,i)-X_pred(1,i))^2;
        Final_Error_estr(i)= Final_Error_estr(i)+error_estr(i);
    end
end
Final_Error_estr = sqrt(Final_Error_estr/(M-1));
Final_Error_filt = sqrt(Final_Error_filt/(M-1));
Final_Error_estr_7 = sqrt(Final_Error_estr_7/(M-1));

figure
subplot(2,1,1)
plot(Final_Error_estr, 'linewidth',1.2)
hold on
plot(calculated_pred_error, 'linewidth',1.2)
grid on
```

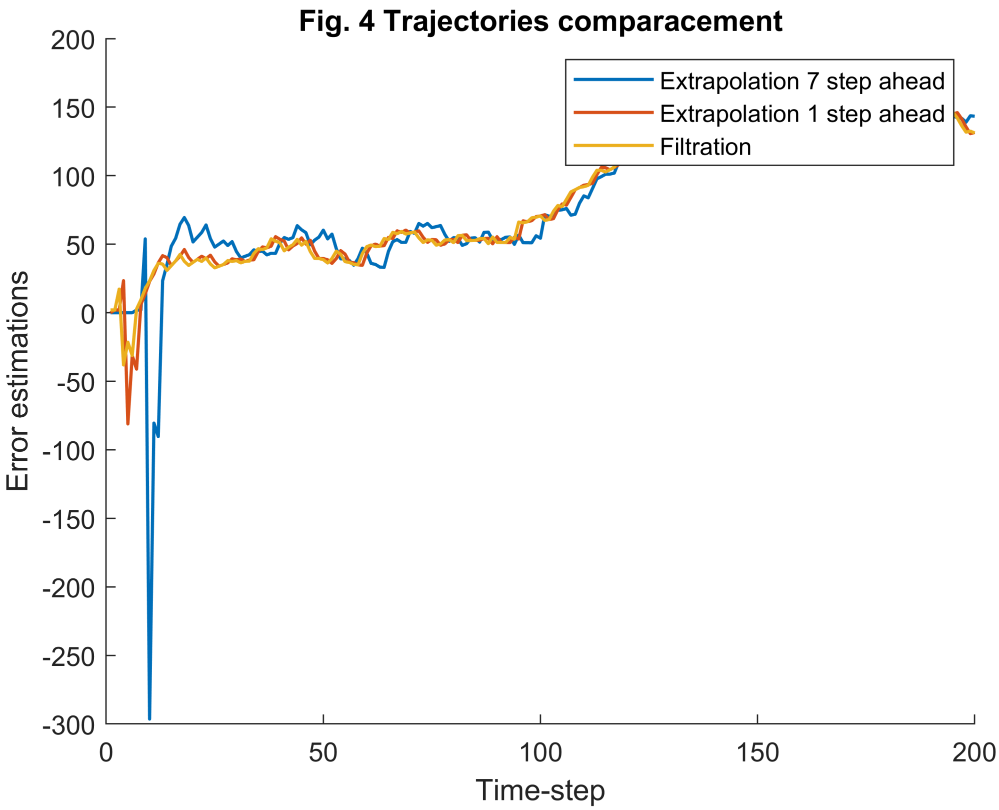
```
title('Error for extrapolated estimations 1 step ahead')
ylabel('Error estimates')
xlabel('Time-step')
yyaxis right
plot(z,'linewidth',0.6,'Color','g')
ylabel('Measured position')
legend('True extrapolation error','Calculated error ', 'Measured position')
xlim([3 inf])

subplot(2,1,2)
plot(Final_Error_filt,'linewidth',1.2)
grid on
hold on
plot(sigma_x,'linewidth',1.2)
yyaxis left
ylabel('Error estimates')
title('Error for filtered estimations')
ylabel('Error estimations')
xlabel('Time-step')
yyaxis right
plot(z,'linewidth',0.6,'Color','g')
ylabel('Measured position')
legend('True filtration error','Calculated error ', 'Measured position')
sgtitle('Fig. 2 Error comparison')
xlim([3 inf])
```



```
figure
plot(Final_Error_estr_7, 'linewidth',1.2)
hold on
plot(Final_Error_estr, 'linewidth',1.2)
plot(Final_Error_filt, 'linewidth',1.2)
grid on
title('Fig. 3 Mean-squared error comparison ')
legend('Extrapolation 7 step ahead ','Extrapolation 1 step ahead ','Filtration')
ylabel('Error estimations')
xlabel('Time-step')
xlim([3 inf])
```



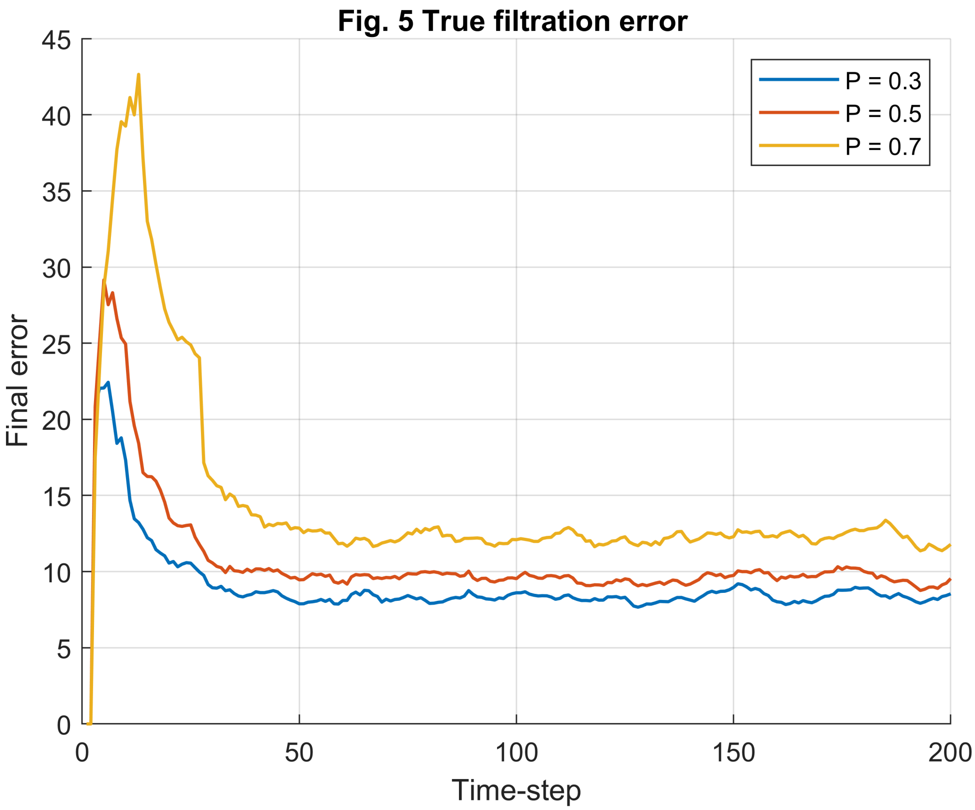

```
figure
hold on
plot(X_estr(1,:), 'linewidth',1.2)
plot(X_pred(1,:), 'linewidth',1.2)
plot(X(1,:), 'linewidth',1.2)
legend('Extrapolation 7 step ahead ','Extrapolation 1 step ahead ','Filtration')
ylabel('Error estimations')
xlabel('Time-step')
title('Fig. 4 Trajectories comparacement')
```



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

```
M = 500;

Final_Error = zeros(3,n);
error_filt = zeros(3,n);
for run = 1:M
    x = 5*ones(1,n);
    V = ones(1,n);
    a = sqrt(var_a)*randn(1,n);

    for i = 2:n
        x(i) = x(i-1)+V(i-1)*T+a(i-1)*T^2/2;
        V(i) = V(i-1)+a(i-1)*T;
    end

    eta = sqrt(var_eta)*randn(1,n);
    ksi = rand(1,n);
    z = zeros(1,n);

    for j = 1:3
        p = 0.3+(j-1)*0.2;
        for i = 1:n
            if ksi(i) <= p
```

```
                z(i) = NaN;
            else
                z(i) = x(i)+eta(i);
            end
        end

        X(:,1) = [2 0]';
        P = 1e4*eye(2);
        K = zeros(2,n);
        X_pred = zeros(2,n);
        for i = 2:n
            X_pred(:,i) = phi*X(:,i-1);
            P_pred = phi*P*phi'+Q;
            K(:,i) = P_pred*H'/(H*P_pred*H'+R);
            if isnan(z(i))
                X(:,i) =  X_pred(:,i);
                P = P_pred;
            else
                X(:,i) = X_pred(:,i) + K(:,i)*(z(i)-H*X_pred(:,i));
                P = (eye(2)-K(:,i)*H)*P_pred;
            end
        end
        K(:,1)= [NaN,NaN];

        for i = 3:n
            error_filt(j,i) = (x(1,i)-X(1,i))^2;
            Final_Error(j,i)= Final_Error(j,i)+error_filt(j,i);
        end
    end
end
Final_Error = sqrt(Final_Error/(M-1));

figure
hold on
plot(Final_Error(1,:),'linewidth',1.2)
plot(Final_Error(2,:),'linewidth',1.2)
plot(Final_Error(3,:),'linewidth',1.2)
grid on
title('Fig. 5 True filtration error ')
legend('P = 0.3','P = 0.5','P = 0.7')
ylabel('Final error')
xlabel('Time-step')
```



Conclusions

- The true trajectory of the moving object was generated and measured. There are measurements gaps which occur with a probability of 0.2 (For tasks 1,2 and 3). To imitate the measurement gaps, a random value is taken for every time step (i). The Kalman filter algorithm is created to account these gaps. When there is no measurement, the filtered step is equal to the extrapolated estimation of position and the filtration covariance matrix is equal to the extrapolated error covariance matrix. From Fig. 1 one can observe the gaps presented in the measurements and the prediction done by the Kalman filter which is quite accurate with the true trajectory.
- For filtered and extrapolated errors of estimation for 1 step ahead the filtration, Fig 2(1) we can oberve that the calculated prediction erros and true extrapolation errors vary from ~8.3 to ~8.7. In Fig 2(2), when the measurement gaps are taken into the account in the filtration step, we can observe some small crests in the calculated filtration error. Peaks of calculated error of filtration match the missing values of measurement.
- As follows from Fig.3, the more steps of extrapolation are calculated, the later peak of mean-squared error is reached. Mean-squared error's steady-state level is higher for more steps of prediction. This can be observed also from Fig. 4, where estimated and extrapolated trajectories are compared. Trajectory with m=7 steps of extrapolation has a huge fluctuation in the early time steps (and also a shift forward comparing to the real trajectory)
- Fig. 5 shows the same relationship between quantity of unavailable data and true filtration error. More available data is in set leads to the faster error decrease with time and to lower steady-state error.