Assignment 6 Analysis of accuracy of tracking in conditions of biased state noise

Group 6:

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


```
clc
clear
close all
```

Task 1: Generating a true trajectory

```
n=200;
q=0.2;
t=1;
var_a=0.2^2;

x=5*ones(1,n);
v=ones(1,n);
a=randn(1,n)*sqrt(var_a);
a_b=a+q; %biased

for i=2:n
    x(1,i)=x(1,i-1)+v(1,i-1)*t+(a_b(1,i-1)*t^2)/2;
    v(1,i)=v(1,i-1)+a_b(1,i-1)*t;
end
```

Task 2: Measurements

```
var_n = 20^2;
eta = sqrt(var_n)*randn(1,n);
z = eta+x; % measurements of coordinate x
```

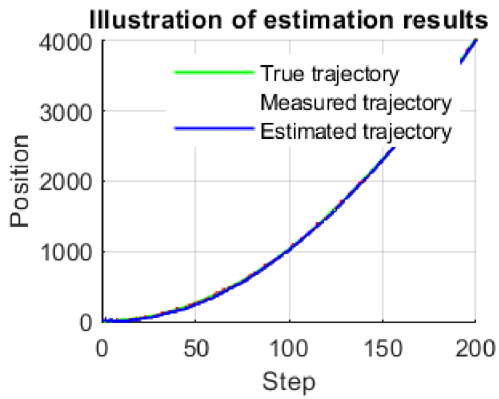
Task 3: Estimation of state vector X in assumption of unbiased acceleration

```
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_n;
Q = G*G'*var_a;
```

```
X(:,1) = [2 0].';
P = [1e4 0; 0 1e4];
sigma_x = sqrt(P(1,1))*ones(1,n);

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

    K(:,i) = P_pred*H.'/(H*P_pred*H.'+R);
    X(:,i) = X(:,i) + K(:,i)*(z(i)-H*X(:,i));
    P = (eye(2)-K(:,i)*H)*P_pred;
    sigma_x(i) = sqrt(P(1,1));
end

x_backup=x;
z_backup=z;
X_backup=X;
```

Task 4: Plot of true trajectory, measurements, filtered estimates of state vector X(i)

```
figure
subplot(2,2,1)
hold on
plot(x,'g','Linewidth',1)
plot(z,'r','Linewidth',0.5)
plot(X(1,:),'b','Linewidth',1)
title('Illustration of estimation results')
legend('True trajectory','Measured trajectory','Estimated trajectory')
xlabel('Step')
ylabel('Position')
grid on
```



Task 5: Estimation of dynamics of of mean-squared error of estimation over observation interval

```
m=500;
K = zeros(2,n);
error_run=zeros(1,n);
final_error=zeros(1,n);
sigma_x = sqrt(P(1,1)*ones(1,n));
x=5*ones(1,n);
v=ones(1,n);

for run=3:m

    a=randn(1,n)*sqrt(var_a);
    a_b=a+q; %biased

    for i=2:n
        x(1,i)=x(1,i-1)+v(1,i-1)*t+(a_b(1,i-1)*t^2)/2;
        v(1,i)=v(1,i-1)+a_b(1,i-1)*t;
    end

    var_eta = 20^2;
    eta = sqrt(var_eta)*randn(1,n);
    z = eta+x;

    X(:,1) = [2 0].';
    P = [1e4 0; 0 1e4];
```

```
    sigma_x = sqrt(P(1,1)*ones(1,n));
    R = var_eta;
    Q = G*G.'*var_a;

    for i = 2:n
        X(:,i) = phi*X(:,i-1);
        P_pred = phi*P*phi.'+ Q;

        K(:,i) = P_pred*H.'/(H*P_pred*H.'+R);
        X(:,i) = X(:,i) + K(:,i)*(z(i)-H*X(:,i));
        P = (eye(2)-K(:,i)*H)*P_pred;
        sigma_x(i) = sqrt(P(1,1));
    end

    for i=3:n
        error_run(1,i)=(x(1,i)-X(1,i))^2;
        final_error(1,i) = final_error(1,i) + error_run(1,i);
    end
end

final_error = sqrt(final_error/(m-1));
```

Task 6: Plots of final error and standard deviation of estimation error

```
subplot(2,2,2)
hold on
plot(final_error)
plot(sigma_x,'g','Linewidth',1)
title('True error compared to errors of estimation')
legend('True estimation error, 500 runs','Kalman estimation error')
xlabel('Step')
ylabel('Error value')
grid on
```

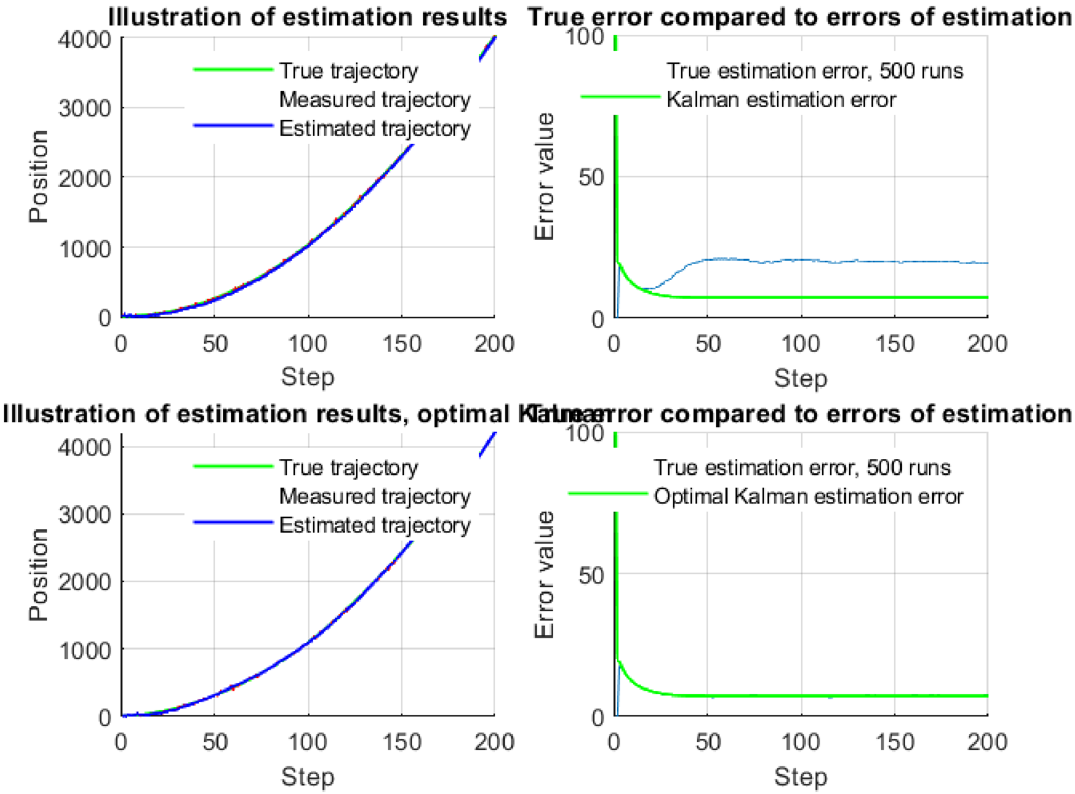
Task 7 and Task 8: Optimal Kalman filter algorithm and True estimation errors

```
m=500;
K = zeros(2,n);
error_run=zeros(1,n);
final_error=zeros(1,n);
sigma_x = sqrt(P(1,1)*ones(1,n));
x=5*ones(1,n);
v=ones(1,n);

for run=3:m

    a=randn(1,n)*sqrt(var_a);
    a_b=a+q; %biased

    for i=2:n
        x(1,i)=x(1,i-1)+v(1,i-1)*t+(a_b(1,i-1)*t^2)/2;
        v(1,i)=v(1,i-1)+a_b(1,i-1)*t;
    end

    var_eta = 20^2;
    eta = sqrt(var_eta)*randn(1,n);
    z = eta+x;

    X(:,1) = [2 0].';
    P = [1e4 0; 0 1e4];
```

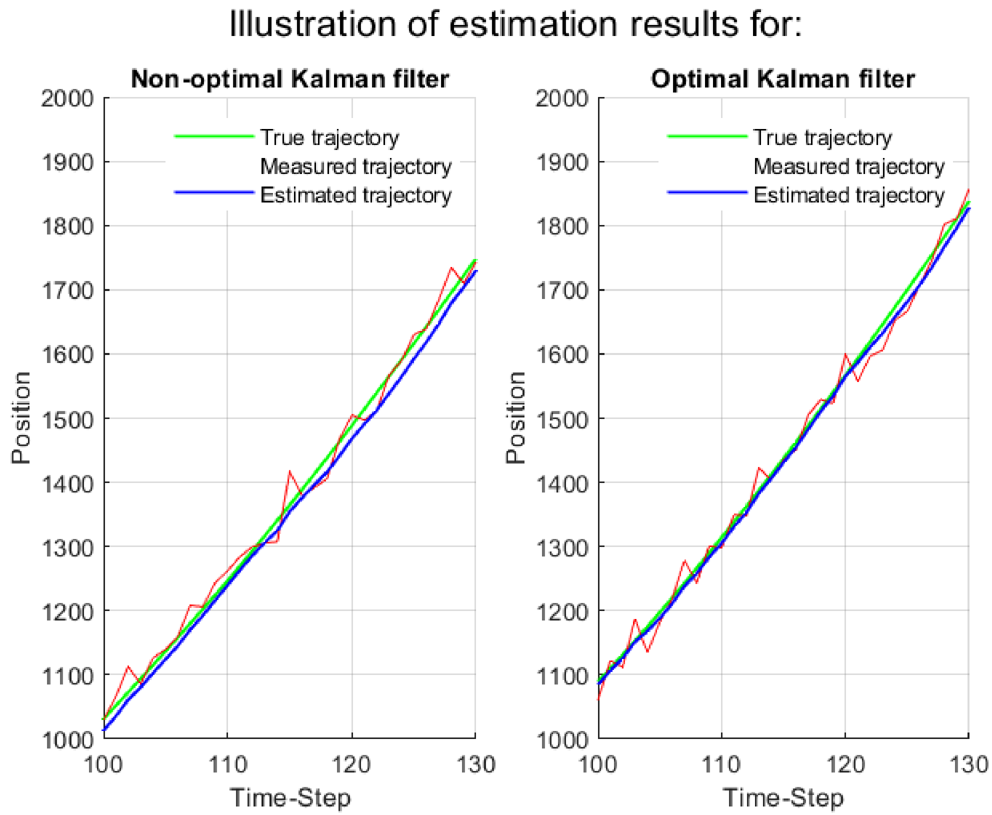
```
    sigma_x = sqrt(P(1,1)*ones(1,n));
    R = var_eta;
    Q = G*G.'*var_a;

    for i = 2:n
        X(:,i) = phi*X(:,i-1)+G*q;
        P_pred = phi*P*phi.'+Q;

        K(:,i) = P_pred*H.'/(H*P_pred*H.'+R);
        X(:,i) = X(:,i) + K(:,i)*(z(i)-H*X(:,i));
        P = (eye(2)-K(:,i)*H)*P_pred;
        sigma_x(i) = sqrt(P(1,1));
    end

    for i=3:n
        error_run(1,i)=(x(1,i)-X(1,i))^2;
        final_error(1,i) = final_error(1,i) + error_run(1,i);
    end
end

final_error = sqrt(final_error/(m-1));
```

#Plots

```
subplot(2,2,4)
hold on
plot(final_error)
plot(sigma_x,'g','Linewidth',1)
title('True error compared to errors of estimation')
legend('True estimation error, 500 runs','Optimal Kalman estimation error')
xlabel('Step')
ylabel('Error value')
grid on



subplot(2,2,3)
hold on
plot(x,'g','Linewidth',1)
plot(z,'r','Linewidth',0.5)
plot(X(1,:),'b','Linewidth',1)
title('Illustration of estimation results, optimal Kalman')
legend('True trajectory','Measured trajectory','Estimated trajectory')
xlabel('Step')
ylabel('Position')
grid on
```




```
figure
subplot(1,2,1)
sgtitle('Illustration of estimation results for:')
hold on
plot(x_backup,'g','Linewidth',1)
plot(z_backup,'r','Linewidth',0.5)
plot(X_backup(1,:),'b','Linewidth',1)
title('Non-optimal Kalman filter')
legend('True trajectory','Measured trajectory','Estimated trajectory')
xlabel('Time-Step')
ylabel('Position')
axis([100 130 1000 2000])
grid on

subplot(1,2,2)
hold on
plot(x,'g','Linewidth',1)
plot(z,'r','Linewidth',0.5)
plot(X(1,:),'b','Linewidth',1)
title('Optimal Kalman filter')
legend('True trajectory','Measured trajectory','Estimated trajectory')
xlabel('Time-Step')
ylabel('Position')
axis([100 130 1000 2000])
grid on
```



Conclusions:

- The true trajectory is generated for an object motion disturbed by normally distributed BIASED random acceleration. This means that to the unbias acceleration, which is a normally distributed random noise with zero mathematical expectation and variance ^ 2, it is added the bias factor of 0.2. Staring with given initial conditions the trajectory and measurements are plotted for a size of 200 time steps.
- When plotting the filtered estimate of coordinate *x*, obtained with Standard Kalman filter which doesn't take into account the bias of state noise, a slight forward shift, compared to the True Trajectory, may be observed. In order to check the accuracy of this estimation, *true estimation error* has been compared with errors of estimation . provided by the Kalman filter algorithm. After an initial drop, the estimate of true error significantly grows and then stabilizes in a constant trend, at the value of approximately 20. It is almost three times higher than the asymptotic value of the standard deviation, and this implies that ordinary Kalman filter can't deal with biased noise
- To minimize these errors caused by neglecting the biased state noise, an optimal Kalman filter is created in which at the prediction (extrapolation) stage there is a biased correction factor of +Gq. The resulting graph shows the true estimation errors curve fluctuating around the values assumed by the standard deviation of estimation error of coordinate x, over the time interval. Hence, the optimal filter provides good results with constant error.
- The difference in ordinary Kalman filter estimation and modified Kalman filter estimation can be observed directly from the graph showing true trajectory and Kalman estimation result. For modified Kalman method estimated trajectory is closer to the true one than it is for ordinary Kalman method.