Lab6: **Analysis of accuracy decrease of filtration in conditions of correlated biased state and measurement noise**

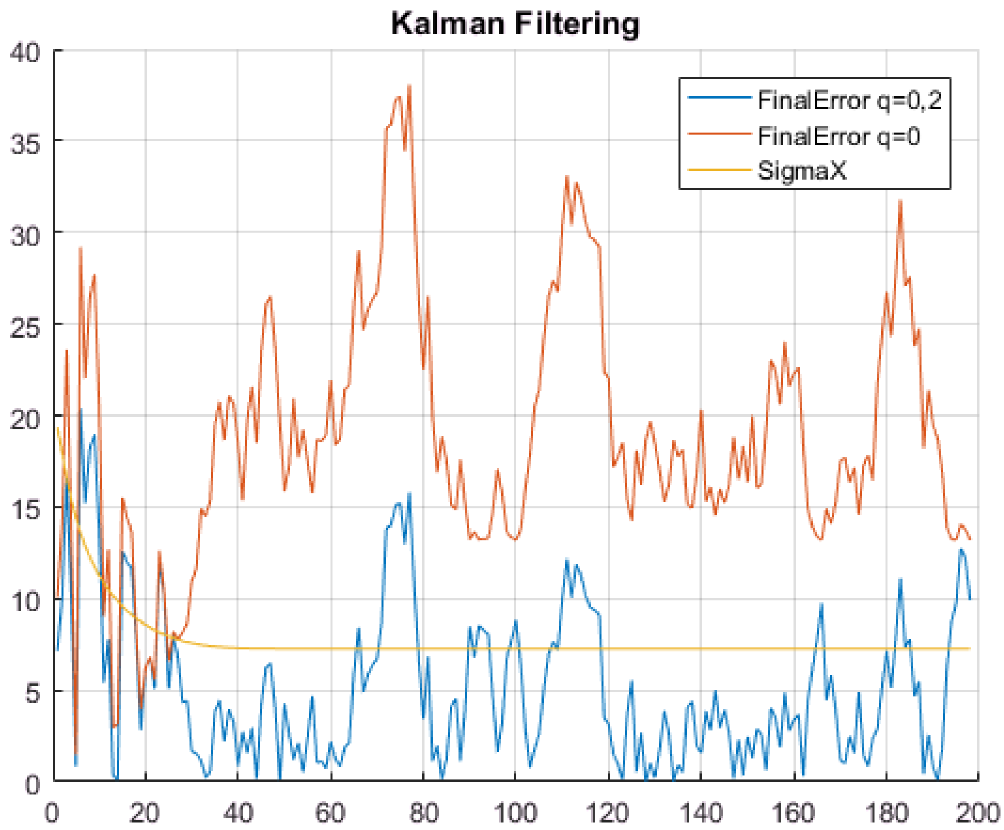
Team1: Dmitry Shadrin and Eugenii Israelit, Skoltech

```
clc; clear; close all;
addpath('functions/');

n = 200;
x1 = 5;
v1 = 1;
t = 1;
sigmaA = 0.2;
sigmaN = 20;
bias = 0.2;
A = normrnd(0, sigmaA, 1, n)+bias;
Noise = normrnd(0, sigmaN, 1, n);

F = [1, t; 0, 1];
G = [(t^2)/2; t];
H = [1, 0];
P = [ 10000, 0;   0, 10000; ];

ErrSum = zeros(1,n);
M = 500;
```

part I:

Divergence of Kalman filter when bias of acceleration (state noise) is neglected in assimilation algorithm. Development of optimal Kalman filter that takes into account bias of acceleration (state noise).

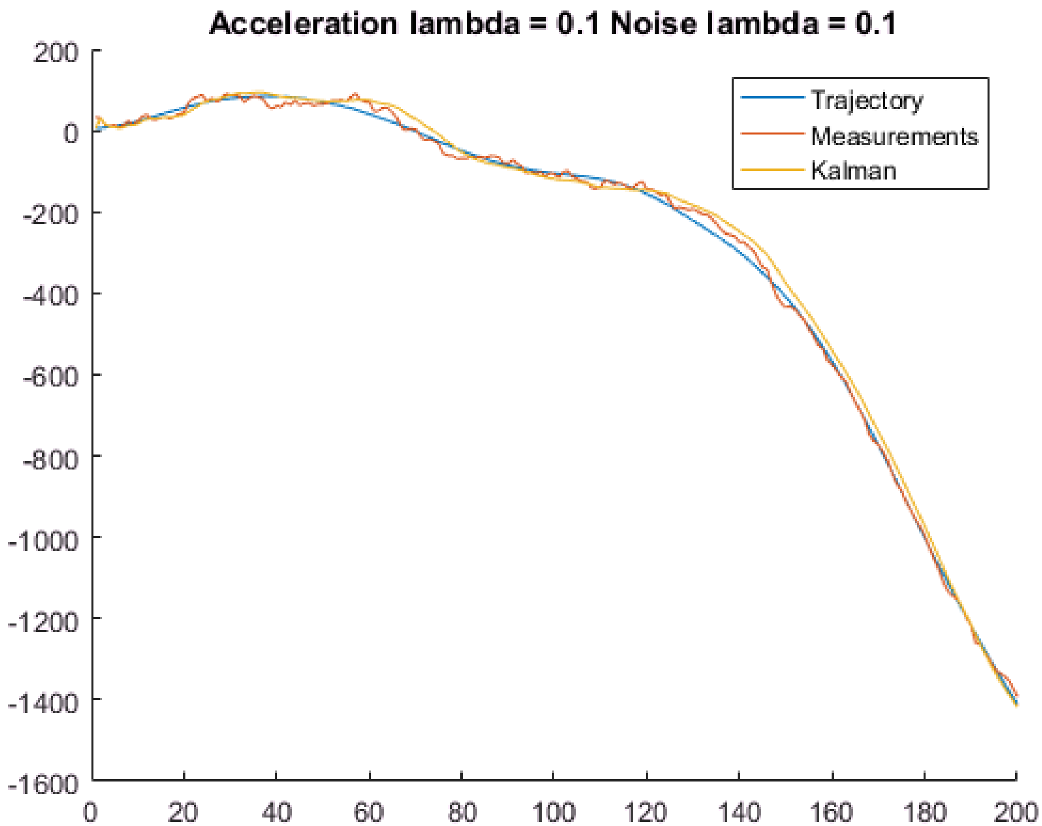
```
BiasCorrection = [bias, 0];
figure; hold on
for i=1:length(BiasCorrection)

    for j=1:M
        [ X, Z ] = calcAccStateSpace( A, Noise, x1, v1, F, G, H );
        [ Xk, SigmaX ] = calcKalman(Z, sigmaA, sigmaN, x1, v1, F, G, H, P, BiasCorrection(i) )
        ErrCur = ( X(1,:) - Xk(1,:) ).^2;
        ErrSum = ErrSum + ErrCur;
    end
    FinalError = ( ErrSum(3:end)./(M-1) ).^0.5;
    plot(FinalError);

end

plot(SigmaX(3:end));
grid
legend('FinalError q=0,2','FinalError q=0','SigmaX');
title('Kalman Filtering');
```

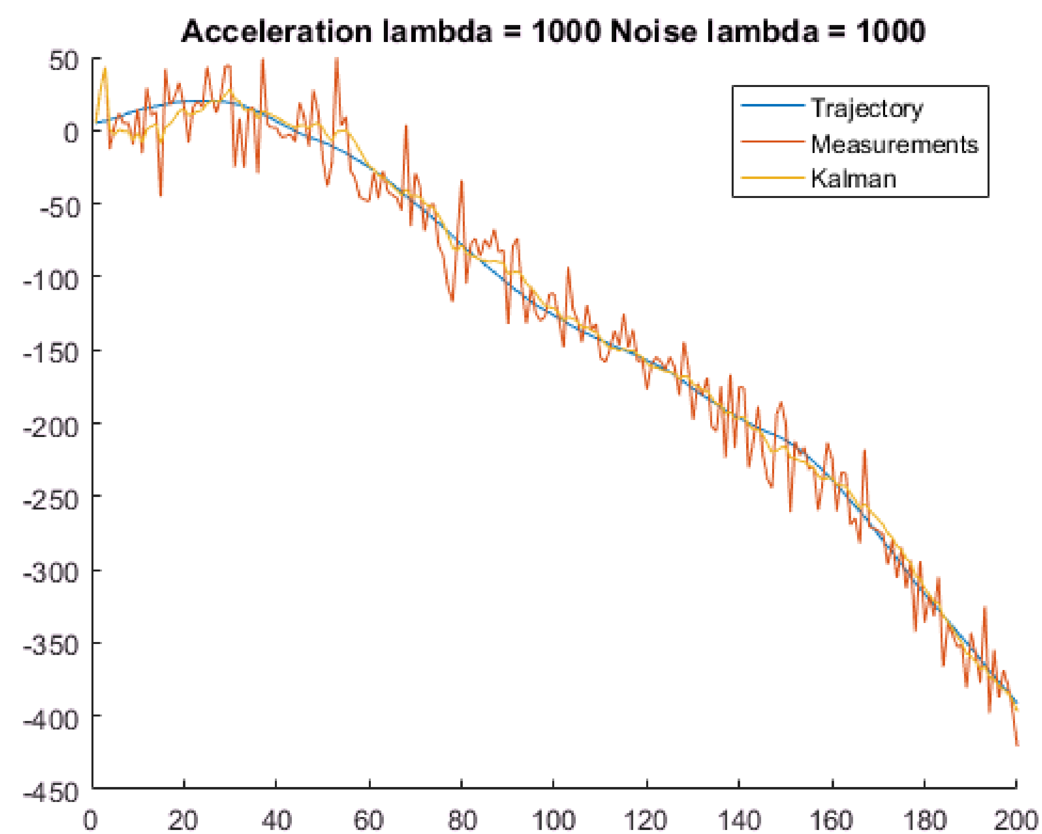


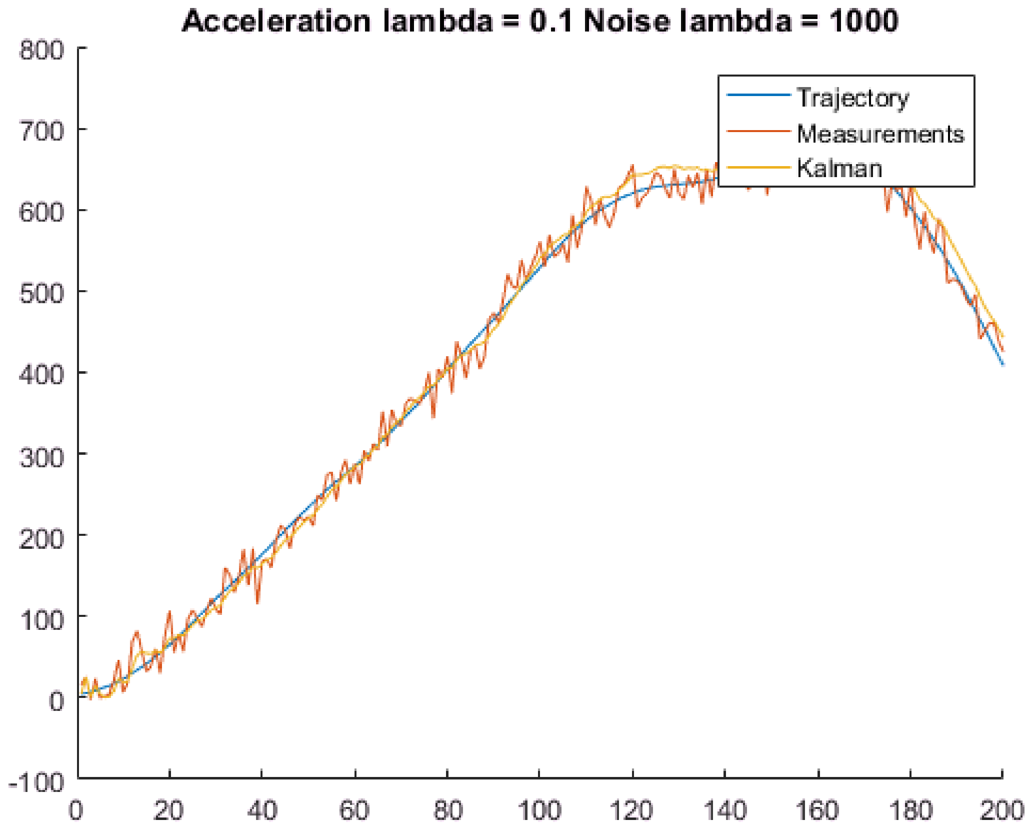
part II:

Sensitivity of estimation results obtained by a Kalman filter that doesn't take into account correlation of state noise (acceleration) and measurement noise.

```
sigmaXi = 1;
Lambda = [ 0.1,  0.1;   1000, 1000;   0.1,  1000 ];

for i=1:length(Lambda)
    A = gaussMarkov( n, sigmaA, sigmaXi, Lambda(i,1), t );
    Noise = gaussMarkov( n, sigmaN, sigmaXi, Lambda(i,2), t );
    [ X, Z ] = calcTrajectory( A, Noise, x1, v1, t);

    for j=1:M
        [ Xk, SigmaX ] = calcKalman(Z, sigmaA, sigmaN, x1, v1, F, G, H, P, 0);
        ErrCur = ( X(1,:) - Xk(1,:) ).^2;
        ErrSum = ErrSum + ErrCur;
    end
    FinalError = ( ErrSum(3:end)./(M-1) ).^0.5;

    figure;
    hold on
    plot(X);
    plot(Z);
    plot(Xk(1,:));
    title(['Acceleration lambda = ', num2str(Lambda(i,1)), ' Noise lambda = ', num2str(Lambda(
    legend('Trajectory', 'Measurements', 'Kalman');
end
```


Acceleration lambda = 0.1 Noise lambda = 0.1
Trajectory
Measurements
Kalman
200
0
-200
-400
-600
-800
-1000
-1200
-1400
-1600
0
20
40
60
80
100
120
140
160
180
200


Acceleration lambda = 1000 Noise lambda = 1000
Trajectory
Measurements
Kalman
50
0
-50
-100
-150
-200
-250
-300
-350
-400
-450
0
20
40
60
80
100
120
140
160
180
200


Acceleration lambda = 0.1 Noise lambda = 1000
Trajectory
Measurements
Kalman
800
700
600
500
400
300
200
100
0
-100
0
20
40
60
80
100
120
140
160
180
200

```
function [ X, Measurments  ] = calcTrajectory( Acc, Noise, x1, v1, t)

    n=length(Acc);
    X(n) = zeros();
    X(1) = x1;
    Vel(n) = zeros();
    Vel(1) = v1;

    for i = 2:n
        Vel(i) = Vel(i-1) + Acc(i-1)*t;
        X(i) = X(i-1) + Vel(i-1)*t + (Acc(i-1)*t^2)/2;
    end

    Measurments = X + Noise;

end
```

```
function [ Xk, SigmaX ] = calcKalman(Z, sigmaA, sigmaN, x1, v1, F, G,
 H, P, bias )

    n=length(Z);

    Xk = zeros(2, n);
    Xk(:, 1) = [x1; v1];

    Q=sigmaA^2 * (G*G');

    SigmaX = zeros(1,n);
    SigmaX(1) = sqrt(P(1,1));

    for i=2:n
        P=F*P*F'+Q;
        K=P*H'/(H*P*H'+ sigmaN^2);
        Xk(:,i) = F*Xk(:, i-1) + G*bias;
        Xk(:,i) = Xk(:,i)+K*(Z(i)-H*Xk(:,i));

        P = (eye(2)-K*H)*P;
        SigmaX(i) = sqrt(P(1,1));
    end

end
```

```
function [ Arr ] = gaussMarkov( n, sigmaA, sigmaXi, lambda, t )

    sigmaZeta = sqrt(1-exp(-2*lambda*t));
    Zeta = sigmaZeta * sigmaA * normrnd(0, sigmaXi, 1, n);

    Arr = zeros(1,n);
    Arr(1)= normrnd(0, sigmaA);

    correlKoef = exp(-lambda*t);

    for i=2:n
        Arr(i) = correlKoef * Arr(i-1) + Zeta(i);
    end

end
```

```
Not enough input arguments.

Error in aaaov (line 3)
    n=length(Acc);
```

Published with MATLAB® R2016a