Assignment 7 Tracking in conditions of correlated biased state and measurement noise

Group 6:

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

PART I: Sensitivity of estimation results obtained by a Kalman filter that does not take into account the correlation of state noise (acceleration) and measurement noise

```
clc
clear
close all
```

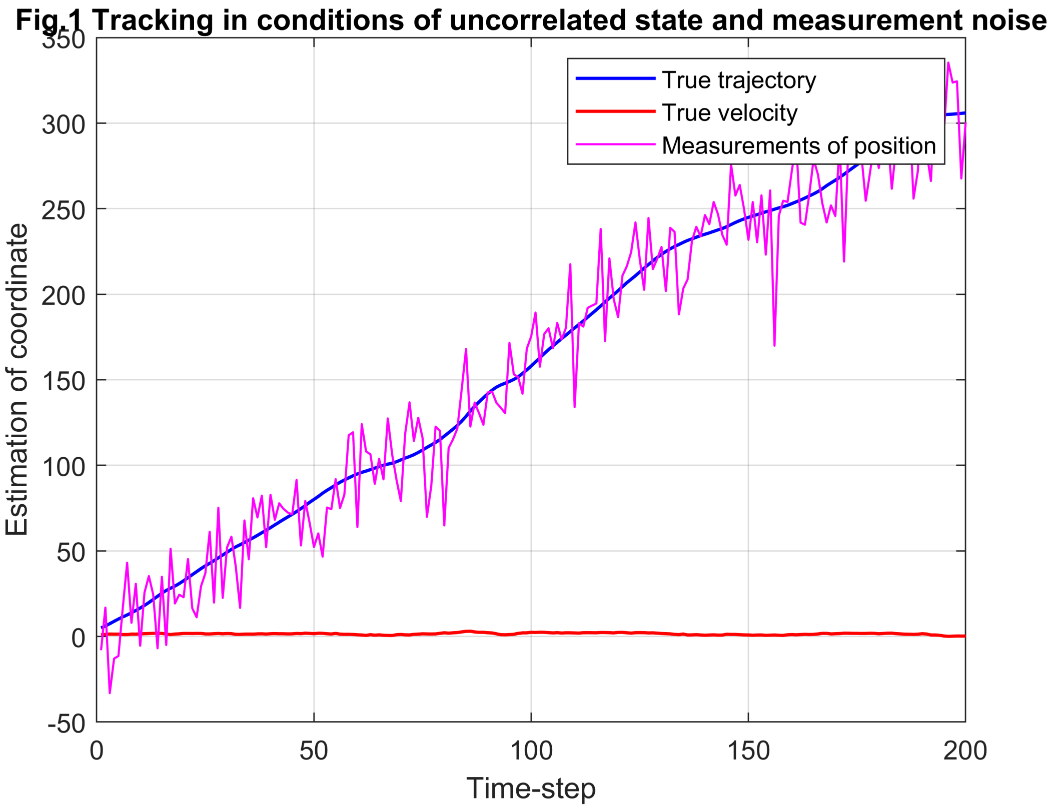
Task 1: Generate a true trajectory (uncorrelated state and measurement noise)

```
n = 200;           % Size of trajectory

T = 1;             % time interval between measurements
var_ksi = 1;       % variance of ksi
ksi = sqrt(var_ksi)*randn(1,n);  % random uncorrelated unbiased noise
sigma_a = 0.2;     % standard deviation of random acceleration.
var_a = sigma_a^2; % variance of random acceleration.

x = 5*ones(1,n);
V = ones(1,n);

%a) lambda = 1000; % => a and eta are uncorrelated noise
a = sigma_a*ksi;

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
```

Task 2: Generate measurements

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

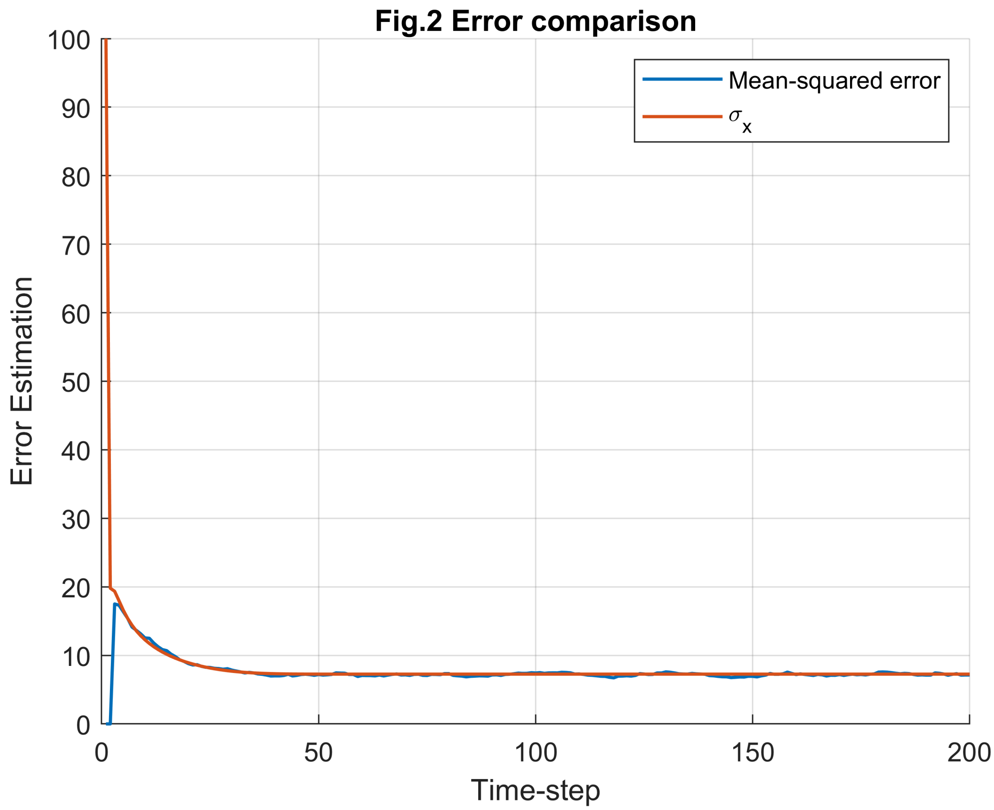
```
figure
plot(x,'b','linewidth',1.2)
hold on
plot(V,'r','linewidth',1.2)
plot(z,'m','linewidth',0.8)
xlabel('Time-step')
ylabel('Estimation of coordinate')
title('Fig.1 Tracking in conditions of uncorrelated state and measurement noise')
legend('True trajectory','True velocity','Measurements of position')
grid on
```



Task 3: Compare true estimation errors with errors of estimation (uncorrelated state and measurement noise)

```
M = 500;

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;      % covariance matrix of measurements noise
Q = G*G'*var_a;   % covariance matrix of state noise

K = zeros(2,n);
Final_Error_filt_1 = zeros(1,n);
error_filt = zeros(1,n);
for run = 1:M
    x = 5*ones(1,n);
```

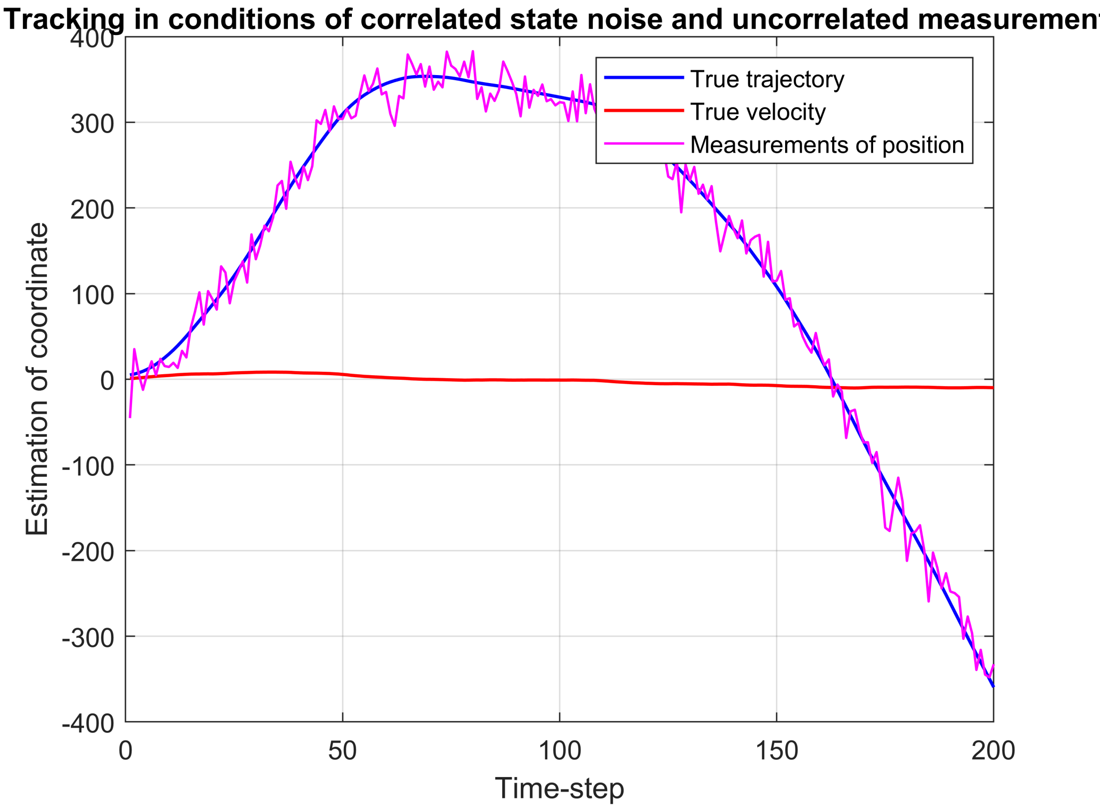
```
    V = ones(1,n);
    ksi = sqrt(var_ksi)*randn(1,n);
    a = sigma_a*ksi;

    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);
    z = eta+x;

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

    for i = 3:n
        error_filt(i) = (x(1,i)-X(1,i))^2;
        Final_Error_filt_1(i)= Final_Error_filt_1(i)+error_filt(i);
    end
end
Final_Error_filt_1 = sqrt(Final_Error_filt_1/(M-1));

figure
hold on
plot(Final_Error_filt_1,'linewidth',1.2)
plot(sigma_x,'linewidth',1.2)
grid on
title('Fig.2 Error comparison')
legend('Mean-squared error','\sigma_x')
ylabel('Error Estimation')
xlabel('Time-step')
```



Task 4: Generate true trajectory and measurements (correlated state noise and uncorrelated measurement noise)

```
n = 200;

T = 1;              % time interval between measurements
var_ksi = 1;        % variance of ksi
ksi = sqrt(var_ksi)*randn(1,n);  % random uncorrelated unbiased noise
sigma_a = 0.2;      % standard deviation of random acceleration.
var_a = sigma_a^2; % variance of random acceleration.

x = 5*ones(1,n);
V = ones(1,n);
a = sigma_a*randn*ones(1,n);
%a)
lambda = 0.1; % => a is correlated noise

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;
    a(i) = exp(-lambda*T)*a(i-1)+sigma_a*ksi(i)*sqrt(1-exp(-2*lambda*T));
end

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

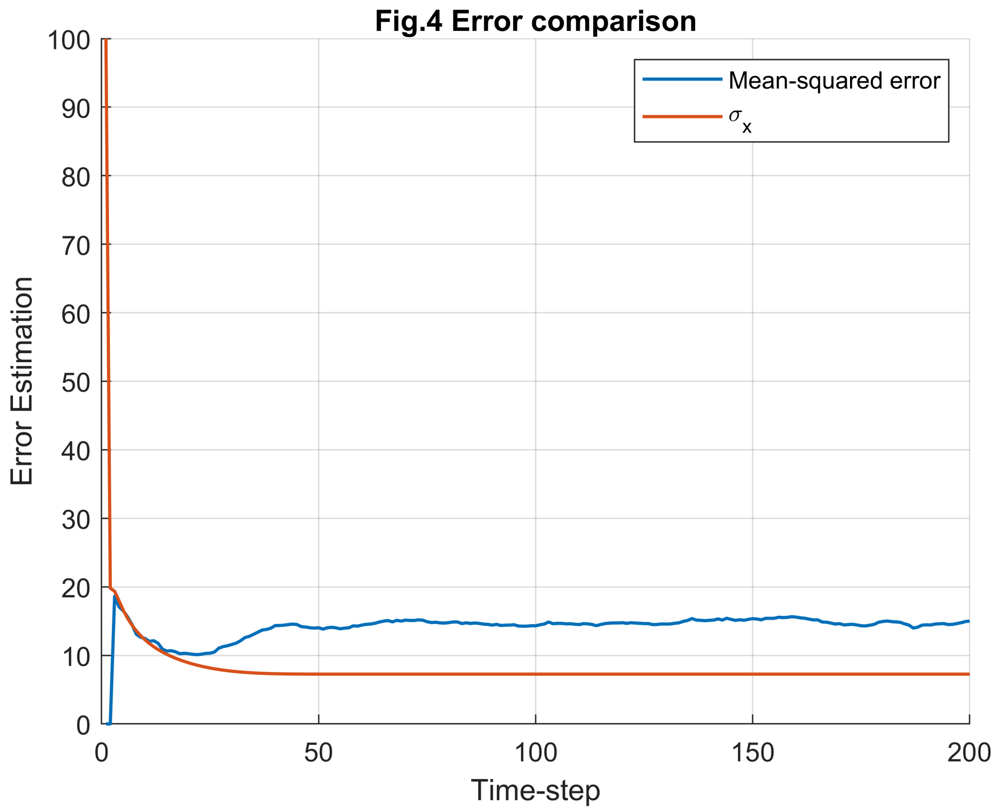
```
figure
plot(x,'b','linewidth',1.2)
hold on
plot(V,'r','linewidth',1.2)
plot(z,'m','linewidth',0.9)
xlabel('Time-step')
ylabel('Estimation of coordinate')
title('Fig.3 Tracking in conditions of correlated state noise and uncorrelated measurement nois
legend('True trajectory','True velocity','Measurements of position')
grid on
```



Task 5: Compare true estimation errors with errors of estimation (non-optimal Kalman filter for correlated state noise and uncorrelated measurement noise)

```
M = 500;

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;      % covariance matrix of measurements noise
Q = G*G'*var_a;   % covariance matrix of state noise

K = zeros(2,n);
Final_Error_filt_2 = zeros(1,n);
error_filt = zeros(1,n);
for run = 1:M
```

```
    x = 5*ones(1,n);
    V = ones(1,n);
    a = sigma_a*randn(1,n);
    eta = sqrt(var_eta)*randn(1,n);
    ksi = sqrt(var_ksi)*randn(1,n);  % random uncorrelated unbiased noise

    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;
        a(i) = exp(-lambda*T)*a(i-1)+sigma_a*ksi(i)*sqrt(1-exp(-2*lambda*T));
    end

    z = eta+x; % measurements of coordinate x

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

    for i = 3:n
        error_filt(i) = (x(1,i)-X(1,i))^2;
        Final_Error_filt_2(i)= Final_Error_filt_2(i)+error_filt(i);
    end
end
Final_Error_filt_2 = sqrt(Final_Error_filt_2/(M-1));

figure
hold on
plot(Final_Error_filt_2,'linewidth',1.2)
plot(sigma_x,'linewidth',1.2)
grid on
title('Fig.4 Error comparison')
legend('Mean-squared error','\sigma_x')
ylabel('Error Estimation')
xlabel('Time-step')
```



Task 6: Compare results for optimal and non-optimal filter

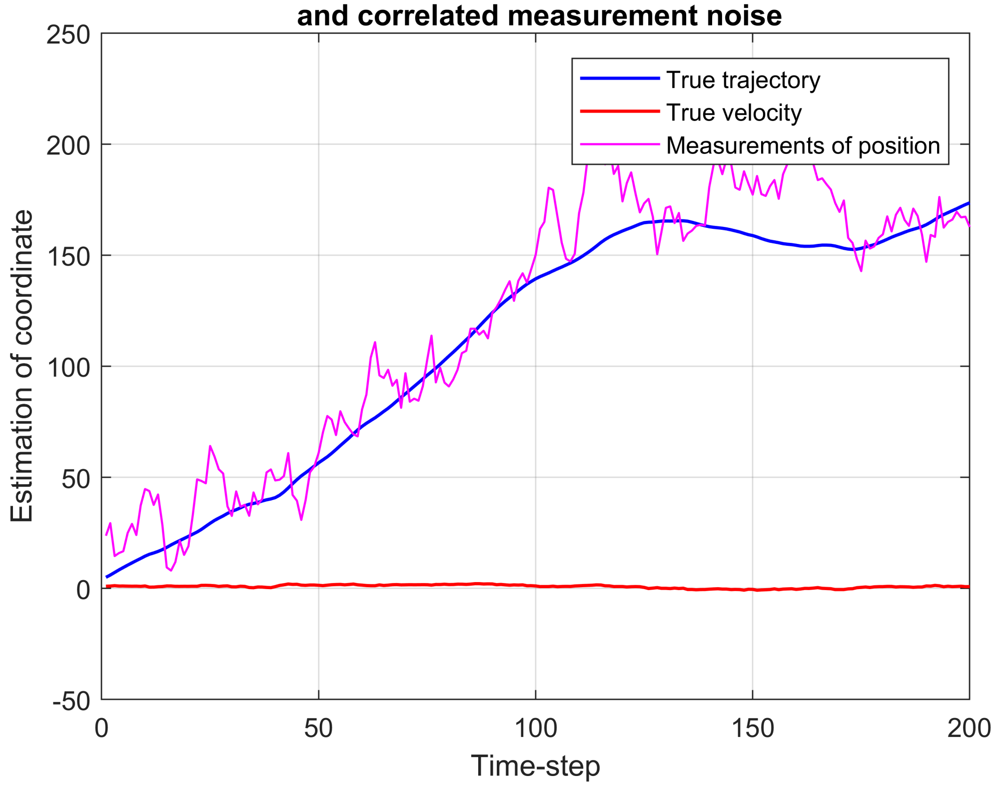
```
figure
subplot(1,2,1)
sgtitle(' Error comparison for:')
hold on
plot(Final_Error_filt_1,'linewidth',1.2)
plot(sigma_x,'linewidth',1.2)
grid on
title('Optimal Kalman filter')
legend('Mean-squared error','\sigma_x')
ylabel('Error Estimation')
xlabel('Time-step')

subplot(1,2,2)
hold on
plot(Final_Error_filt_2,'linewidth',1.2)
plot(sigma_x,'linewidth',1.2)
grid on
title('Fig.5 Kalman filter neglecting correlated state noise')
legend('Mean-squared error','\sigma_x')
ylabel('Error Estimation')
xlabel('Time-step')
```

By comparing the results we can clearly see the differences in the estimation errors for optimal and non-optimal Kalman filters. As the correlation is introduced in the stae noise there is a rise in the estimation errors.

Task 7: Generate trajectory and measurements (uncorrelated state noise and correlated measurement noise)

```
n = 200;

T = 1;              % time interval between measurements
var_ksi = 1;        % variance of ksi
ksi = sqrt(var_ksi)*randn(1,n);  % random uncorrelated unbiased noise
sigma_a = 0.2;      % standard deviation of random acceleration.
var_a = sigma_a^2; % variance of random acceleration.

x = 5*ones(1,n);
V = ones(1,n);

lambda_1 = 1000; % => a is uncorrelated noise
a = sigma_a*ksi;

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

lambda_2 = 0.1; % => eta is correlated noise
var_eta = 20^2;
var_s = var_eta*(1-exp(-2*lambda_2*T));
s = sqrt(var_s)*randn(1,n);
eta = sqrt(var_eta)*randn*ones(1,n);

for i = 2:n
    eta(i) = exp(-lambda_2*T)*eta(i-1) + s(i);
end

z = eta+x; % measurements of coordinate x

figure
plot(x,'b','linewidth',1.2)
hold on
plot(V,'r','linewidth',1.2)
plot(z,'m','linewidth',0.8)
xlabel('Time-step')
ylabel('Estimation of coordinate')
title({'Fig.6 Tracking in conditions of uncorrelated state noise';' and correlated measurement
legend('True trajectory','True velocity','Measurements of position')
grid on
```

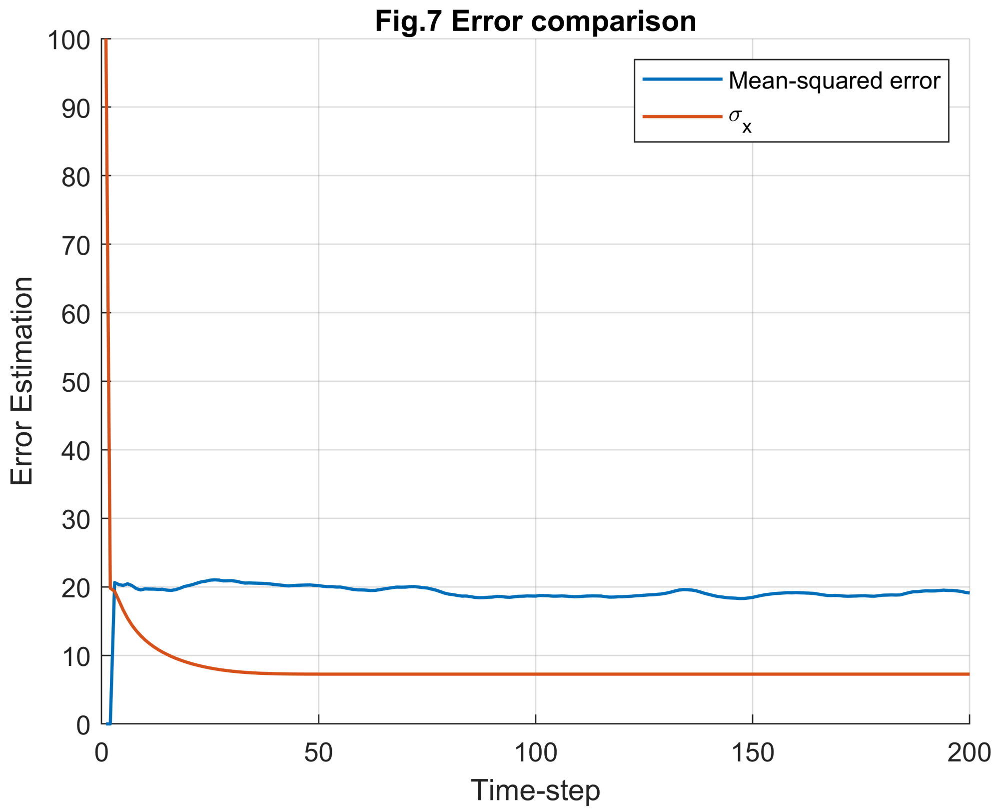
Fig.6 Tracking in conditions of uncorrelated state noise and correlated measurement noise



Task 8: Compare true estimation errors with errors of estimation (non-optimal Kalman filter for uncorrelated state noise and correlated measurement noise)

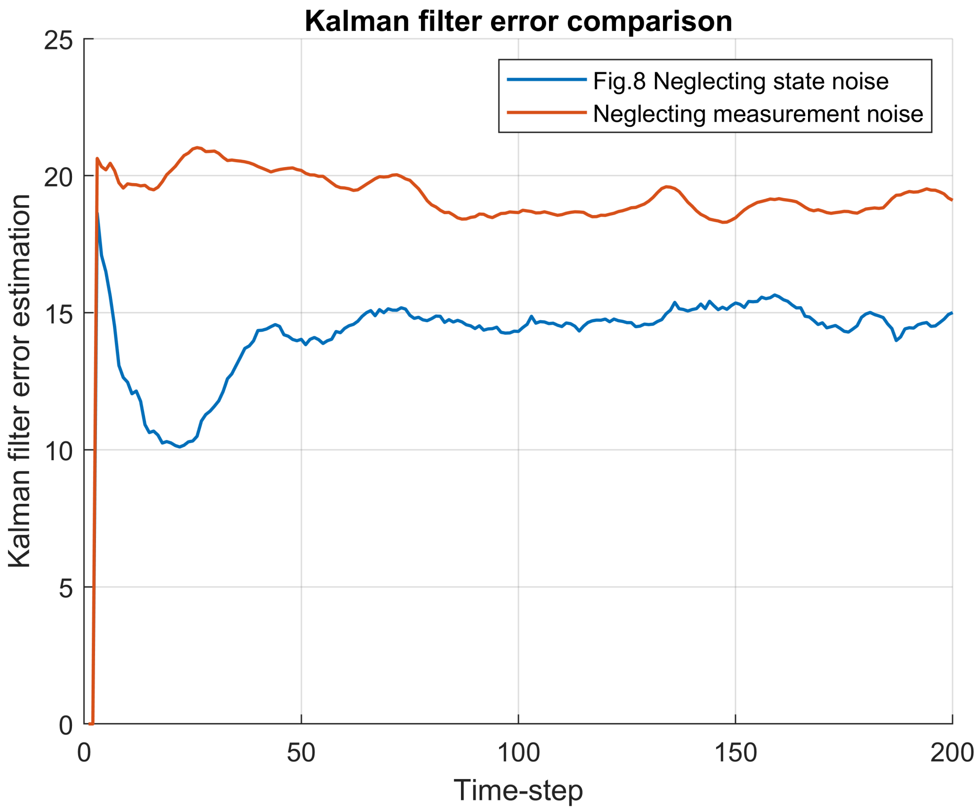
```
M = 500;

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;      % covariance matrix of measurements noise
Q = G*G'*var_a;   % covariance matrix of state noise

K = zeros(2,n);
Final_Error_filt_3 = zeros(1,n);
error_filt = zeros(1,n);
for run = 1:M
    x = 5*ones(1,n);
    V = ones(1,n);
    ksi = sqrt(var_ksi)*randn(1,n);  % random uncorrelated unbiased noise
    a = sigma_a*ksi;

    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

    lambda_2 = 0.1; % => eta is correlated noise
```

```
    var_eta = 20^2;
    var_s = var_eta*(1-exp(-2*lambda_2*T));
    s = sqrt(var_s)*randn(1,n);
    eta = sqrt(var_eta)*randn*ones(1,n);

    for i = 2:n
        eta(i) = exp(-lambda_2*T)*eta(i-1) + s(i);
    end
    z = eta+x; % measurements of coordinate x

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

    for i = 3:n
        error_filt(i) = (x(1,i)-X(1,i))^2;
        Final_Error_filt_3(i)= Final_Error_filt_3(i)+error_filt(i);
    end
end
Final_Error_filt_3 = sqrt(Final_Error_filt_3/(M-1));

figure
hold on
plot(Final_Error_filt_3,'linewidth',1.2)
plot(sigma_x,'linewidth',1.2)
grid on
title('Fig.7 Error comparison')
legend('Mean-squared error','\sigma_x')
ylabel('Error Estimation')
xlabel('Time-step')
```



Task 9: Conclude neglecting of which noise leads to greater accuracy decrease

```
figure
hold on
plot(Final_Error_filt_2,'linewidth',1.2)
plot(Final_Error_filt_3,'linewidth',1.2)
legend('Fig.8 Neglecting state noise','Neglecting measurement noise')
title('Kalman filter error comparison')
ylabel('Kalman filter error estimation')
xlabel('Time-step')
grid on
```

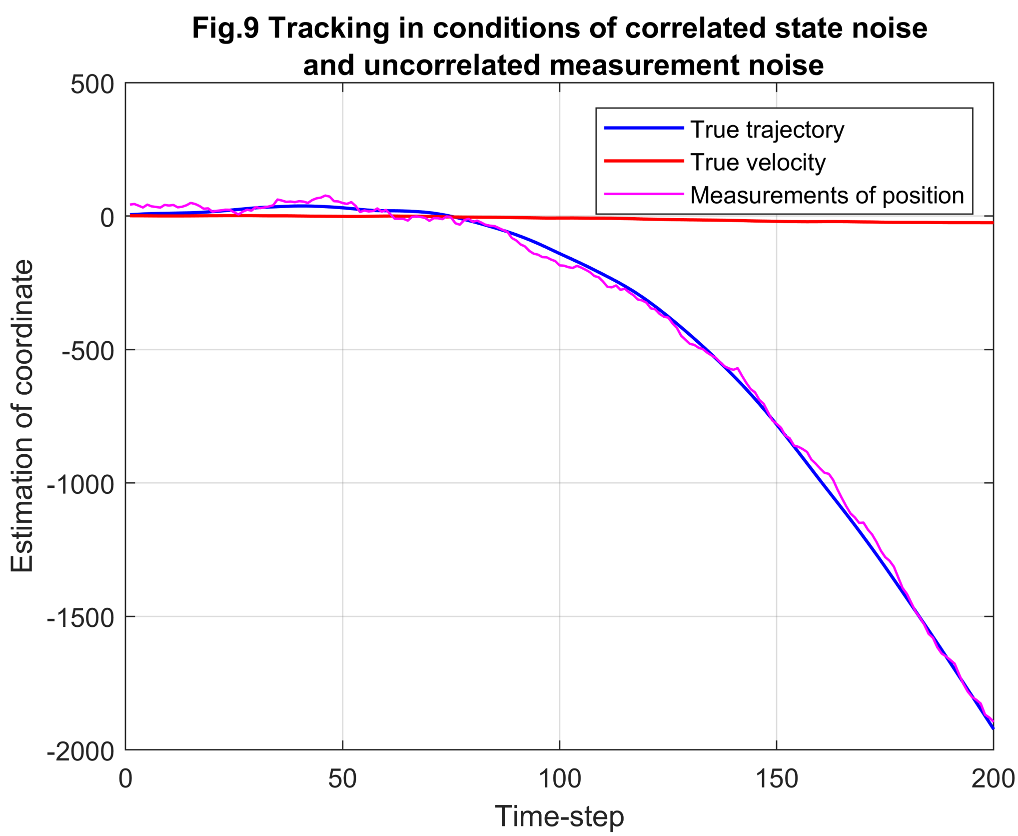


Using previous tasks' results, a graph comparing Kalman estimation errors for two cases was plotted. With it one can see that true error for uncorrelated measurement noise and correlated state noise is lower than error for opposite situation. This is due to bigger value of measurement noise than it is of state noise. So we can make a conclusion that neglecting measurements noise leads to greater accuracy decrease.

Task 10: Generate trajectory and measurements (correlated state and measurement noise)

```
n = 200;

T = 1;              % time interval between measurements
var_ksi = 1;        % variance of ksi
ksi = sqrt(var_ksi)*randn(1,n);  % random uncorrelated unbiased noise
sigma_a = 0.2;      % standard deviation of random acceleration.
var_a = sigma_a^2; % variance of random acceleration.

x = 5*ones(1,n);
V = ones(1,n);
a = sigma_a*randn*ones(1,n);

lambda = 0.1; % => a and eta are correlated noise
var_zeta = sigma_a^2*(1-exp(-2*lambda*T));
zeta = sqrt(var_zeta)*randn(1,n);

var_eta = 20^2;
var_s = var_eta*(1-exp(-2*lambda*T));
```

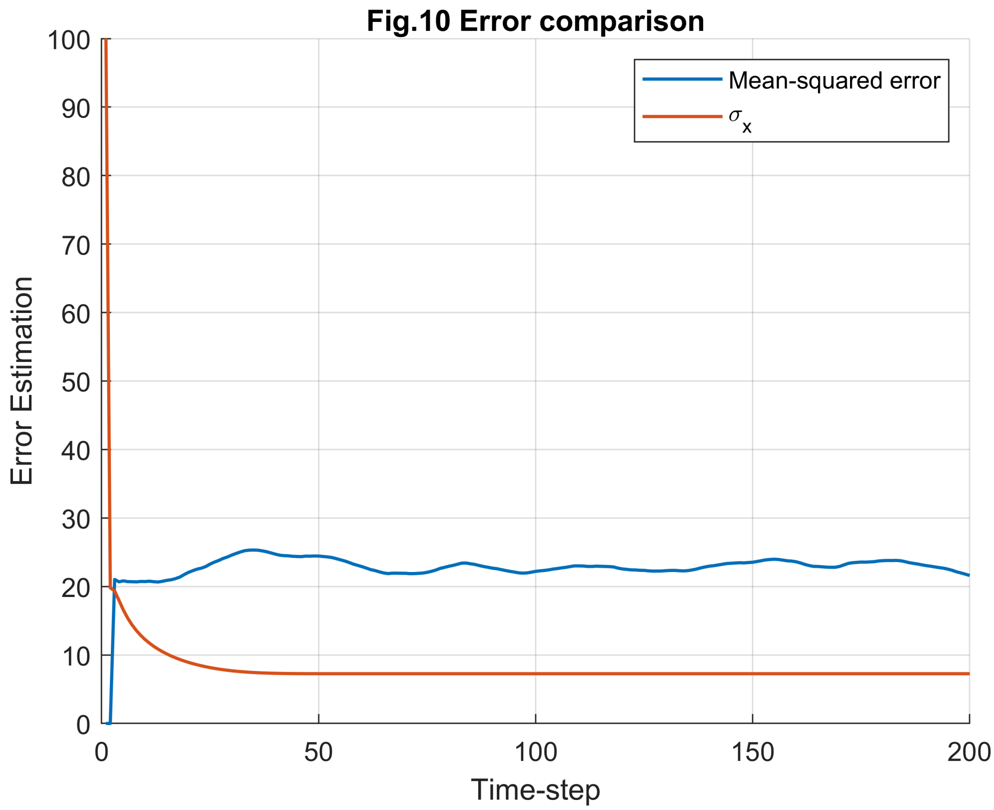
```
s = sqrt(var_s)*randn(1,n);
eta = sqrt(var_eta)*randn*ones(1,n);

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;
    a(i) = exp(-lambda*T)*a(i-1)+sigma_a*ksi(i)*sqrt(1-exp(-2*lambda*T));
    eta(i) = exp(-lambda*T)*eta(i-1) + s(i);
end

z = eta+x; % measurements of coordinate x

figure
plot(x,'b','linewidth',1.2)
hold on
plot(V,'r','linewidth',1.2)
plot(z,'m','linewidth',0.9)
xlabel('Time-step')
ylabel('Estimation of coordinate')
title({'Fig.9 Tracking in conditions of correlated state noise';' and uncorrelated measurement
legend('True trajectory','True velocity','Measurements of position')
grid on
```



Task 11: Compare true estimation errors with errors of estimation (non-optimal Kalman filter for correlated state and measurement noise)

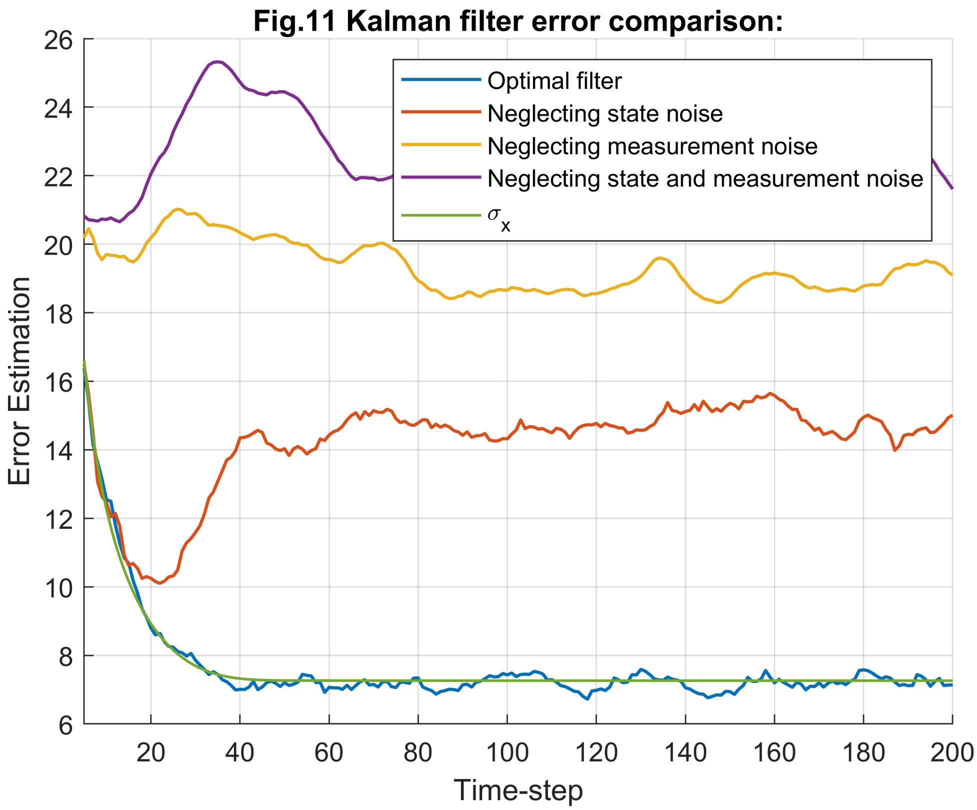
```
M = 500;
```

```
K = zeros(2,n);
Final_Error_filt_4 = zeros(1,n);
error_filt = zeros(1,n);
for run = 1:M
    x = 5*ones(1,n);
    V = ones(1,n);
    a = sigma_a*randn(1,n);
    s = sqrt(var_s)*randn(1,n);
    eta = sqrt(var_eta)*randn*ones(1,n);
    ksi = sqrt(var_ksi)*randn(1,n);  % random uncorrelated unbiased noise

    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;
        a(i) = exp(-lambda*T)*a(i-1)+sigma_a*ksi(i)*sqrt(1-exp(-2*lambda*T));
        eta(i) = exp(-lambda*T)*eta(i-1) + s(i);
    end

    z = eta+x; % measurements of coordinate x

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

    for i = 3:n
        error_filt(i) = (x(1,i)-X(1,i))^2;
        Final_Error_filt_4(i)= Final_Error_filt_4(i)+error_filt(i);
    end
end
Final_Error_filt_4 = sqrt(Final_Error_filt_4/(M-1));

figure
hold on
plot(Final_Error_filt_4,'linewidth',1.2)
plot(sigma_x,'linewidth',1.2)
grid on
title('Fig.10 Error comparison')
legend('Mean-squared error','\sigma_x')
ylabel('Error Estimation')
xlabel('Time-step')
```



Final comparison:

```
figure
title('Fig.11 Kalman filter error comparison:')
hold on
plot(Final_Error_filt_1,'linewidth',1.2)
plot(Final_Error_filt_2,'linewidth',1.2)
plot(Final_Error_filt_3,'linewidth',1.2)
plot(Final_Error_filt_4,'linewidth',1.2)
plot(sigma_x,'linewidth',1)

grid on
legend('Optimal filter','Neglecting state noise','Neglecting measurement noise','Neglecting sta
ylabel('Error Estimation')
xlabel('Time-step')
xlim([5 200])
```



Conclusions:

- **Case 1 (uncorrelated state and measurement noise):** The true trajectory and measurements for the uncorrelated state and measurement noise are generated. For the uncorrelated noises the taken value of lambda (inverse to correlation interval) is 1000. Thus, the uncorrelated noise is calculated. Comparing the true estimation errors with the errors of estimations the filter behaves as a optimal Kalman filter, as the assumptions about the uncorrelated noise are true.
- **Case 2 (correlated state noise and uncorrelated measurement noise):** The true trajectory and measurements for the correlated state noise and uncorrelated measurement noise are generated. For the correlated state noise the taken value of lambda (inverse to correlation interval) is 0.1 and the measurement noise remaines the same. For this lambda, the correlated noise is on interval of 10 steps, i.e. inside every 10 steps correlation is significant. This random acceleration is considered as a first order Gauss-Markov process. On comapring the true estimation errors and errors of estimations the filter is a non-optimal Kalman filter as the estimation error coincides with the standard deviation only for the first 20 time-steps approximately, but then it starts to diverge at a constant error of almost 15. This happens because Standar Kalman filter is neglecting the correlation in time of the state noise.
- **Case 3 (uncorrelated state noise and correlated measurement noise):** The true trajectory and measurements for the uncorrelated state noise and correlated measurement noise are genearated. For the uncorrelated state noise the taken value of lambda is 1000 and the value of lambda for the measurements is 0.1. For the correlated measurement noise first order Gauss-Markov process is used. By comparing the true estimation errors and errors of estimations one can see that the Kalman filter

is again a non-optimal filter. In this case the true estimation error is nearly 20 whereas the standard deviation is ~7. The filter achieves constant estimation error in the early time-steps.

- **Case 4 (correlated state and measurement noise):** The true trajectory and measurements for the correlated state noise and correlated measurement noise is generated. The value of lambda for the correlated state and measurement noise is taken 0.1. For both the noises first order Gauss-Markov process is used. On comparing the true estimation errors and errors of estimations the filter is non-optimal Kalman filter, in which the estimations is at a constant trend of nearly more than 20 from very early time-steps and the standard deviation is at ~7.
- **Final Comparison:** By comparing the results, we can clearly see the differences in the estimation errors for optimal and non-optimal Kalman filters. By the last graph, one can observe that true error for filtered estimates of coordinate x, obtained neglecting correlated measurement noise, is higher than true error obtained neglecting state noise and obviously than error for optimal Kalman estimates. It is due to the standard deviation of measurement noise, which is bigger than the one of state noise. As it could be expected, the highest value for true error is achieved by filtered estimate of coordinate which neglects correlation in time of both measurement and state noise. It can be concluded that when correlation is introduced, whether in state noise or measurement noise or both, the Kalman filter becomes non-optimal and gives high estimation errors. To compensate these correlations the Kalman filter should also estimate the dynamics of the correlated acceleration.