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

Group 6:

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

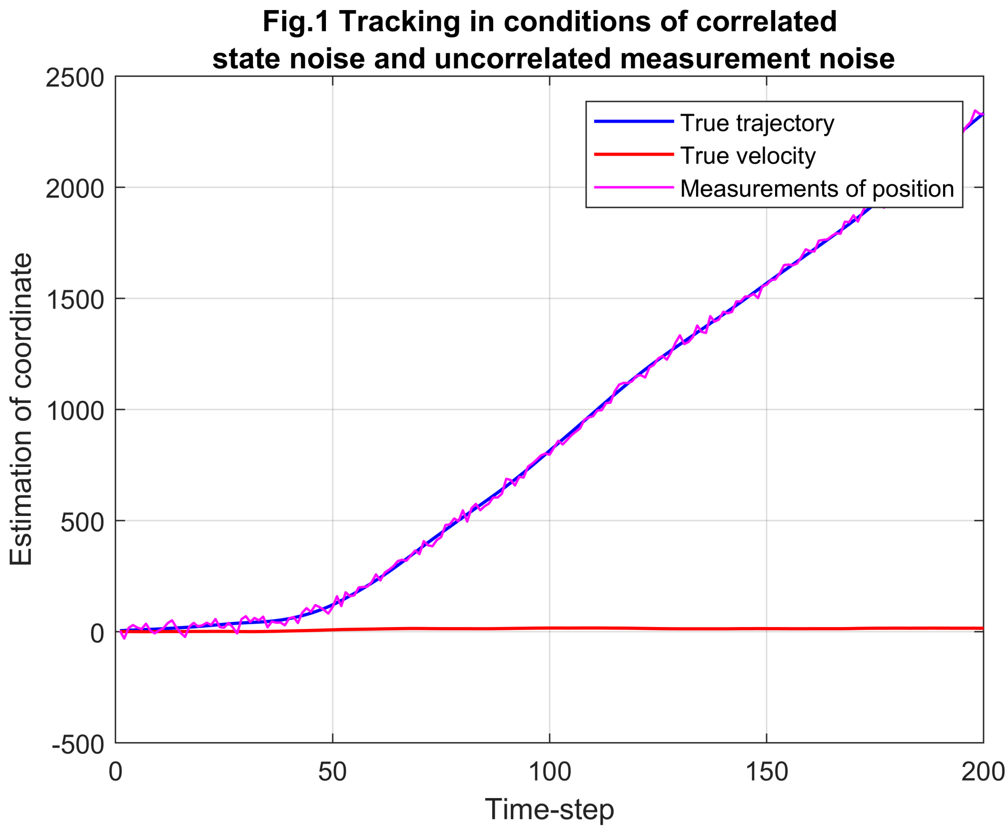
PART II: Development of optimal Kalman filter in conditions of correlated state noise

```
clc
clear
close all
```

Task 1: Generate trajectory and measurements

```
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(1,n);

lambda = 0.1; % => a is correlated noise
var_zeta = var_a*(1-exp(-2*lambda*T));

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); % uncorrelated 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',0.9)
xlabel('Time-step')
ylabel('Estimation of coordinate')
title({'Fig.1 Tracking in conditions of correlated';' state noise and uncorrelated measurement
legend('True trajectory','True velocity','Measurements of position')
grid on
```



Task 2: Present the system at state space

```
phi = [1 T (T^2)/2; 0 1 T; 0 0 exp(-lambda*T)]; % transition matrix that relates X(i) and X(i-1
G = [0 0 1]';     % input matrix, that determines how random acceleration affects state vector;
H = [1 0 0];      % observation matrix
R = var_eta;      % covariance matrix of measurements noise
Q = G*G'*var_zeta;   % covariance matrix of state noise
```

Task 3: Initial conditions

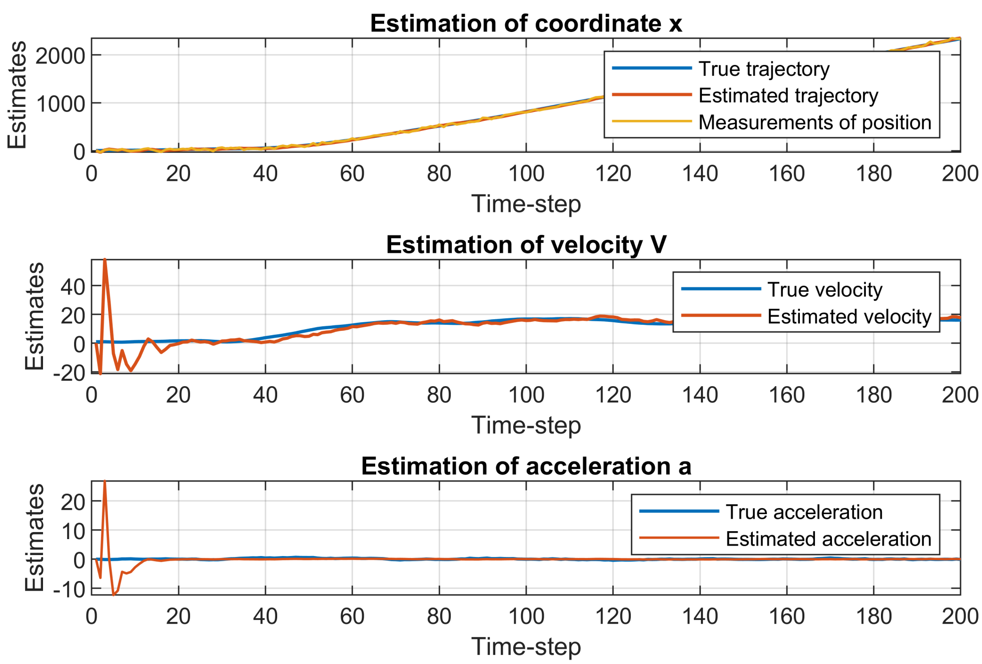
```
X(:,1) = [2 0 0]'; % Initial filtered estimate
P = 1e4*eye(3);     % Initial filtration error covariance matrix
```

Task 4: Comparison of errors of filtered estimates and errors of extrapolated estimates

```
sigma_x = sqrt(P(1,1)*ones(1,n));
sigma_V = sqrt(P(2,2)*ones(1,n));
```

```
sigma_A = sqrt(P(3,3)*ones(1,n));

K = zeros(3,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);
    X(:,i) = X_pred(:,i) + K(:,i)*(z(i)-H*X_pred(:,i));
    P = (eye(3)-K(:,i)*H)*P_pred;
    sigma_x(i) = sqrt(P(1,1));
    sigma_V(i) = sqrt(P(2,2));
    sigma_A(i) = sqrt(P(3,3));
end
K(:,1)=[NaN NaN NaN]';


figure
subplot(3,1,1)
sgtitle([{'Fig.2 Tracking in conditions of correlated state noise' ;
    ' and uncorrelated measurement noise'}])
plot(x,'linewidth',1.2)
hold on
plot(X(1,:),'linewidth',1.2)
plot(z,'linewidth',0.9)
xlabel('Time-step')
ylabel('Estimates')
title('Estimation of coordinate x')
legend('True trajectory','Estimated trajectory','Measurements of position')
grid on

subplot(3,1,2)
plot(V,'linewidth',1.2)
hold on
plot(X(2,:),'linewidth',1.2)
xlabel('Time-step')
ylabel('Estimates')
title('Estimation of velocity V')
legend('True velocity','Estimated velocity')
grid on

subplot(3,1,3)
plot(a,'LineWidth',1.2)
hold on
plot(X(3,:),'LineWidth',0.9)
xlabel('Time-step')
ylabel('Estimates')
title('Estimation of acceleration a')
legend('True acceleration','Estimated acceleration')
grid on
```

Fig.2 Tracking in conditions of correlated state noise and uncorrelated measurement noise

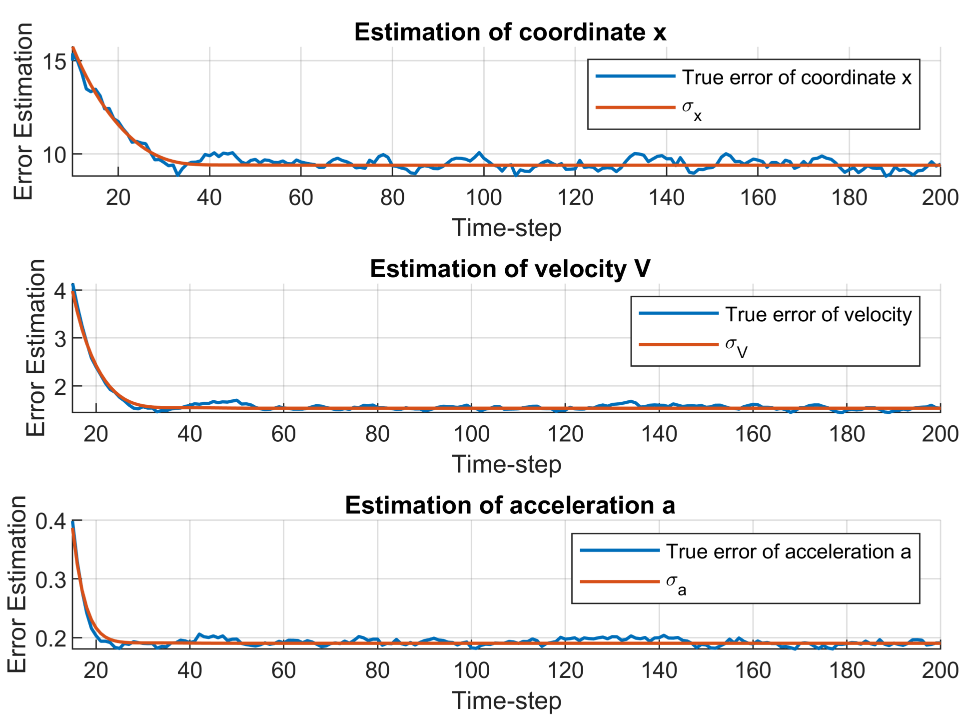



```
M = 500;

K = zeros(3,n);
Final_Error_x = zeros(1,n);
Final_Error_V = zeros(1,n);
Final_Error_a = zeros(1,n);
error_filt_x = zeros(1,n);
error_filt_V = zeros(1,n);
error_filt_a = zeros(1,n);

Final_Error_estr_x = zeros(1,n);
Final_Error_estr_V = zeros(1,n);
Final_Error_estr_a = zeros(1,n);
error_estr_x = zeros(1,n);
error_estr_V = zeros(1,n);
error_estr_a = zeros(1,n);

for run = 1:M
    x = 5*ones(1,n);
    V = ones(1,n);
    a = sigma_a*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
```

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

    X(:,1) = [2 0 0].';
    P = 1e4*eye(3);
    sigma_x = sqrt(P(1,1)*ones(1,n));
    sigma_V = sqrt(P(2,2))*ones(1,n);
    sigma_A = sqrt(P(3,3))*ones(1,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);
        X(:,i) = X_pred(:,i) + K(:,i)*(z(i)-H*X_pred(:,i));
        P = (eye(3)-K(:,i)*H)*P_pred;
        sigma_x(i) = sqrt(P(1,1));
        sigma_V(i) = sqrt(P(2,2));
        sigma_A(i) = sqrt(P(3,3));
    end

    for i = 3:n
        error_filt_x(i) = (x(1,i)-X(1,i))^2;
        error_filt_V(i) = (V(1,i)-X(2,i))^2;
        error_filt_a(i) = (a(1,i)-X(3,i))^2;

        error_estr_x(i) = (x(1,i)-X_pred(1,i))^2;
        error_estr_V(i) = (V(1,i)-X_pred(2,i))^2;
        error_estr_a(i) = (a(1,i)-X_pred(3,i))^2;

        Final_Error_x(i)= Final_Error_x(i)+error_filt_x(i);
        Final_Error_V(i)= Final_Error_V(i)+error_filt_V(i);
        Final_Error_a(i)= Final_Error_a(i)+error_filt_a(i);

        Final_Error_estr_x(i)= Final_Error_estr_x(i)+error_estr_x(i);
        Final_Error_estr_V(i)= Final_Error_estr_V(i)+error_estr_V(i);
        Final_Error_estr_a(i)= Final_Error_estr_a(i)+error_estr_a(i);
    end
end
Final_Error_x = sqrt(Final_Error_x/(M-1));
Final_Error_V = sqrt(Final_Error_V/(M-1));
Final_Error_a = sqrt(Final_Error_a/(M-1));

Final_Error_estr_x = sqrt(Final_Error_estr_x/(M-1));
Final_Error_estr_V = sqrt(Final_Error_estr_V/(M-1));
Final_Error_estr_a = sqrt(Final_Error_estr_a/(M-1));

figure
subplot(3,1,1)
hold on
plot(Final_Error_x,'linewidth',1.2)
plot(sigma_x,'linewidth',1.2)
grid on
```

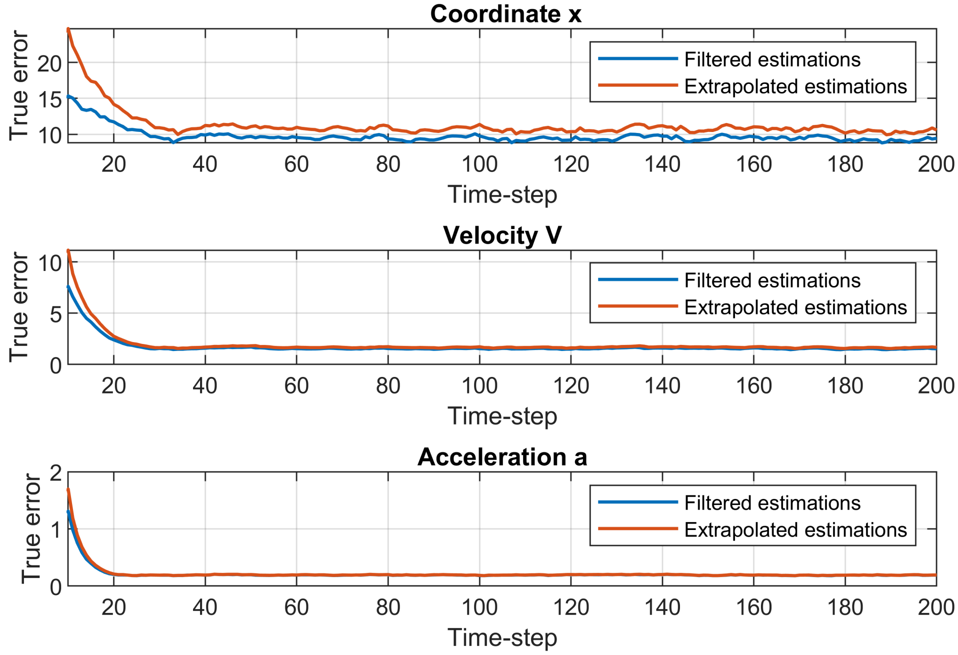
```
title(' Estimation of coordinate x')
legend('True error of coordinate x','\sigma_x')
ylabel('Error Estimation')
xlabel('Time-step')
xlim([10 200])

subplot(3,1,2)
hold on
plot(Final_Error_V,'linewidth',1.2)
plot(sigma_V,'linewidth',1.2)
grid on
title(' Estimation of velocity V')
legend('True error of velocity','\sigma_V')
ylabel('Error Estimation')
xlabel('Time-step')
xlim([15 200])

subplot(3,1,3)
hold on
plot(Final_Error_a,'linewidth',1.2)
plot(sigma_A,'linewidth',1.2)
legend('True error of acceleration a','\sigma_a')
title(' Estimation of acceleration a')
xlim([15 200])
grid on
sgtitle(' Fig.3 Error comparison for all parameters: ')
ylabel('Error Estimation')
xlabel('Time-step')
```

Fig.3 Error comparison for all parameters:



```
figure
subplot(3,1,1)
sgtitle({'Fig.4 Comparison of error of filtered estimations';' with error of extrapolated estim
plot(Final_Error_x,'linewidth',1.2)
hold on
plot(Final_Error_estr_x,'linewidth',1.2)
grid on
title(' Coordinate x')
legend('Filtered estimations','Extrapolated estimations')
ylabel('True error')
xlabel('Time-step')
xlim([10 200])

subplot(3,1,2)
plot(Final_Error_V,'linewidth',1.2)
hold on
plot(Final_Error_estr_V,'linewidth',1.2)
grid on
title('Velocity V')
legend('Filtered estimations','Extrapolated estimations')
ylabel('True error')
xlabel('Time-step')
xlim([10 200])

subplot(3,1,3)
plot(Final_Error_a,'linewidth',1.2)
hold on
plot(Final_Error_estr_a,'linewidth',1.2)
grid on
title('Acceleration a')
legend('Filtered estimations','Extrapolated estimations')
ylabel('True error')
xlabel('Time-step')
xlim([10 200])
```

Fig.4 Error comparison of filtered estimations with extrapolated estimations of:



Conclusions:

- The true trajectory and measurements for the correlated state noise and uncorrelated measurement noise are generated (Fig.1). For the correlated state noise the value of lambda (inverse to correlation interval) taken is 0.1 and the measurement noise is taken for the given variance. 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 first order Gauss-Markov process.
- After determining the transition matrix and input matrix from the state equation and observation matrix from measurement equation, the comparison between filtered estimates and true trajectory of position, velocity and acceleration is shown on Fig.2, in which the estimated trajectory of postion, velocity and accelration follows with the true trajectory of position, velocity and acceleration respectively with minimal fluctuations/ noise.
- To develop the optimal Kalman filter the correlated acceleration should be taken ito consideration, i.e. the new state space vector is extended with the inclusion of correlated random acceleration. Upon taking the initial conditions for the filtered estimates and error covariance matrix the final error estimates for the position, velocity and acceleration is calculated along with standard deviation. Since covariance matrix of state noise is a 3x3 matrix and acceleration is correlated, there will be changes in the covariance matrix with the introduction of the uncorrelated random noise (variance zeta)
- Comparasement of the true estimation errors with the errors of estimation P(i,i) on Fig.3 shows that the value of standard deviation for position is 9.38 and the estimation errors fluctuates over this value. Same for the standard deviation for velocity and acceleration at 1.51 and 0.19 repectively, the estimation error for velocity and accelration also fluctuates near their respective ranges. Hence, a optimal Kalman filter is achieved with consideration of correlated state noise.

- True estimation error and errors of estimation . provided by Kalman filter algorithm are almost equal for coordinate estimation.
- The highest true error is for coordinate estimation with a peak in the initial time-steps. For acceleration it is about 20 times less; the lowest true error is gained for estimation of acceleration.
- Fig.4 shows that error of extrapolated estimation tends to be greater than error of filtered estimation for all three parameters X,V,a.