```
clear; close all; clc
set(0,'DefaultFigureWindowStyle','docked')
%% Linear estimation in one dimension

dt = 1;                 % timestep [sec]
T = 10;                 % duration [sec]
n = T/dt;               % number of steps
t = linspace(dt,T,n);   % time vector

% speed
v = 5;      % [m/s]

% Noise on the motion model
std_x = 1; % [m]
sig_x = std_x^2;

% Noise on the measurement model
std_z = 2; % [m]
sig_z = std_z^2;

x0 = 0 + std_x * randn;

x = zeros(1,n);
x(1) = x0;

z = zeros(1,n);

xhat = zeros(1,n);
xhat_0 = 0 + std_x * randn;
xhat(1) = xhat_0;
xhat_p = xhat;

sig = zeros(1,n);
sig(1) = sig_x;
sig_p = sig;

for k = 2:n
    x(k) = x(k-1) + dt*v + randn * std_x;
    z(k) = x(k) + randn * std_z;

    xhat_p(k) = xhat(k-1) + v*dt;
    sig_p(k) = sig(k-1) + sig_x;

    K = sig_p(k) / (sig_p(k) + sig_z);

    xhat(k) = xhat_p(k) + K * (z(k) - xhat_p(k));
    %sig(k) = sig(k) * sig_z / (sig(k) + sig_z);
    sig(k) = sig_p(k) - K * sig_p(k);
end

% RMS of the estimation error
RMSE = sqrt(sum((x-xhat).^2)/n)
RMSE = sqrt(mean((x-xhat).^2))
RMSE = sqrt((x-xhat)*(x-xhat)'/n)
RMSE = rms(x-xhat)

figure
hold on; box on;

plot(x,'b')
plot(z,'go')
plot(xhat,'r')
xlabel('Time [sec]')
ylabel('Position [m]')
title(['RMSE = ',num2str(RMSE)])
yyaxis right
plot(x-xhat)
plot([dt,T],[RMSE,RMSE])
ylim([-3,3])
leg = {'True state','Measurement','Estimated state','Error','RMSE'};
legend(leg,'Location','NorthWest')
```