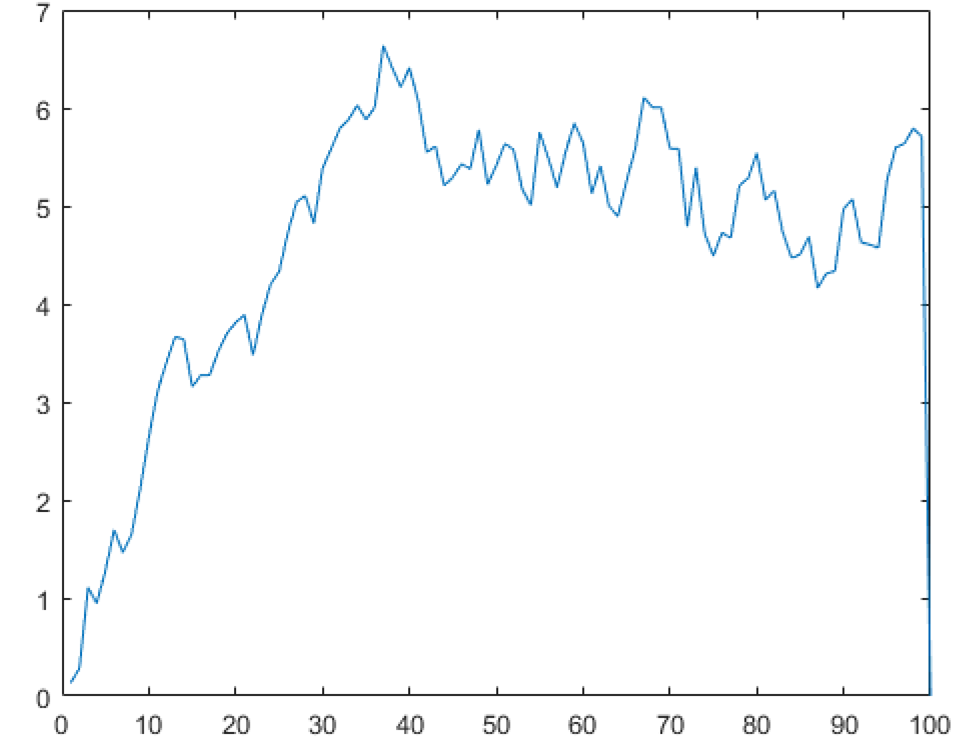
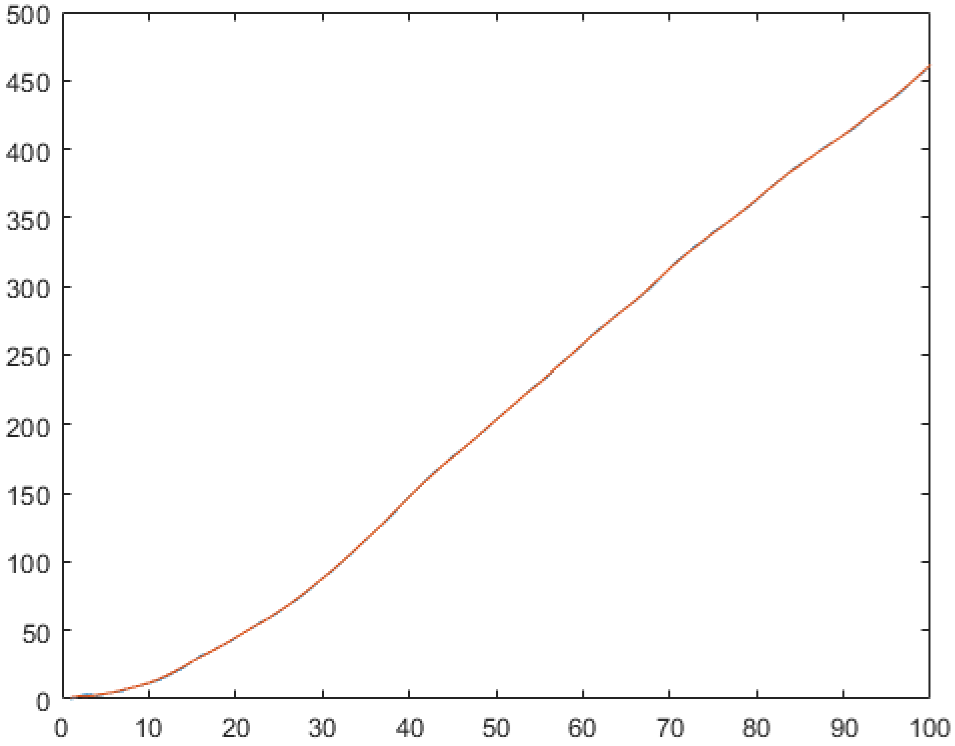
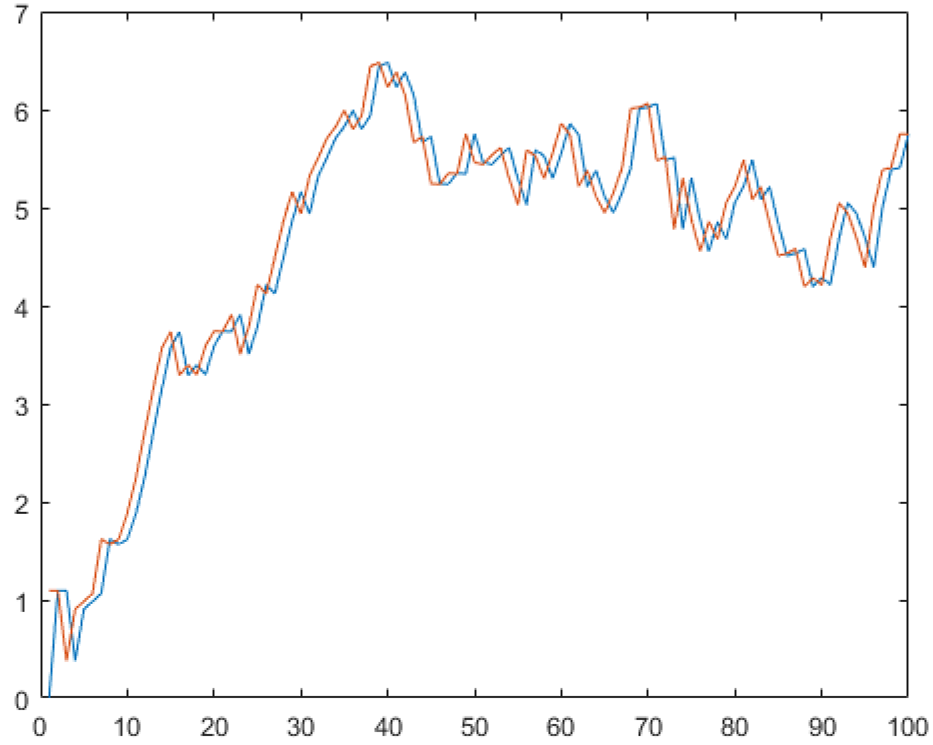
```
% Project 4 exercise 1
clear all; clc
% State space
A = [1  1
     0  1];
B = [0  1]';
C = [1  0];
D = 0;
% White noise
g = [0  1]';
wk = [0     0
      0   0.1];

vk = 0.01;

% Random initialization
x_0 = [1    1]';

p_0 = [1    0
       0    1];
% Number of steps
N = 100;
% INITIALIZATIONS
x_est = zeros(2,N);
y_est = zeros(1,N);
x_bar = zeros(2,N);
x_hat = zeros(2,N);
y_hat = zeros(1,N);
y     = zeros(1,N);
p     = zeros(2,2,N);
p_hat = zeros(2,2,N);
meansqrerr = zeros(1,N);
% Set initial values
w  = normrnd(0,sqrt(0.1));
v  = normrnd(0,sqrt(0.01));
x_est(:,1) = x_0;
y_est(:,1) = C * x_est(:,1) + v;
x_bar(:,1) = [0 0]';
x_hat(:,1) = x_0;
p(:,:,1)   = p_0;
p_hat(:,:,1) = p_0;
% Assign initial observer value
L = [1  1]';

% Estimation
for i = 1 : N-1
    % get noise
    w  = normrnd(0,sqrt(0.1));
    v  = normrnd(0,sqrt(0.01));
    % next step
    x_est(:,i+1) = A * x_est(:,i) + B * w;
    y_est(:,i+1) = C * x_est(:,i) + D + v;
```

```
end

% Prediction
for i = 1 : N-1
    y(i) = y_est(i);
    x_hat(:,i) = x_bar(:,i) + L * (y(i) - C * x_bar(:,i));
    p(:,:,i) = (eye(length(L)) - L * C) * squeeze(p_hat(:,:,i));

    x_bar(:,i+1) = A * x_hat(:,i);
    p_hat(:,:,i+1) = A * squeeze(p(:,:,i)) * A' + wk;
    L = squeeze(p_hat(:,:,i+1)) * C' * inv(C * squeeze(p_hat(:,:,i+1))
 * C' + vk);
end

% Update mean-square error
for i = 1 : N - 1
    diff_1 = x_hat(1,i) - x_est(1,i);
    diff_2 = x_hat(2,i) - x_est(2,i);
    meansqrerr(1,i) = sqrt(diff_1^2 + diff_2^2);
end

% Get last prediction
x_hat(:,end) = x_bar(:,end) + L * (y_est(end) - C * x_bar(:,end));

% PLOTS
figure(1)
plot(1:N,meansqrerr);
figure(2)
plot(1:N,x_bar(1,:),1:N,x_hat(1,:));
figure(3)
plot(1:N,x_bar(2,:),1:N,x_hat(2,:));
```


7
6
5
4
3
2
1
0
0
10
20
30
40
50
60
70
80
90
100


500
450
400
350
300
250
200
150
100
50
0
0
10
20
30
40
50
60
70
80
90
100



Published with MATLAB® R2020b