04|27|2020 **MAE 6292: Homework 5**

Harshvardhan Uppaluru
G42420576

34/34

Problem 1:

(a) Given

$$K = P^- H^T (H P^- H^T + R)^{-1}$$
$$P^+ = (I - KH) P^-$$

Substituting K in P^+

$$P^+ = P^- - K H P^-$$
$$P^+ = P^- - P^- H^T (H P^- H^T + R)^{-1} H P^- \quad \text{— (1)}$$

Matrix Inversion lemma.

$$F = [A + BCD]^{-1} = A^{-1} - A^{-1} B (D A^{-1} B + C^{-1})^{-1} D A^{-1} \quad \text{— (2)}$$

Comparing equation (1) with (2), we get

$$\boxed{P^+ = [(P^-)^{-1} + H^T R^{-1} H]^{-1}}$$

We know that $(P^+)(P^+)^{-1} = 1$

Using this result, we have

$$K = P^- H^T (H P^- H^T + R)^{-1}$$
$$K = (P^+)(P^+)^{-1} P^- H^T (H P^- H^T + R)^{-1}$$
$$K = P^+ [(P^-)^{-1} + H^T R^{-1} H] P^- H^T [H P^- H^T + R]^{-1}$$

Expanding, we have

$$K = (P^+ (P^-)^{-1} + P^+ H^T R^{-1} H) P^- H^T (H P^- H^T + R)^{-1}$$
$$K = (P^+ (P^-)^{-1} P^- H^T + P^+ H^T R^{-1} H P^- H^T)(H P^- H^T + R)^{-1}$$

$$K = P^+ H^T (I + R^{-1} H P^- H^T)(H P^- H^T + R)^{-1}$$

$$\boxed{K = P^+ H^T R^{-1}}$$

(b) Joseph form:

$$P^+ = (I-KH)P^-(I-KH)^T + KRK^T$$

Expanding:

$$P^+ = (P^- - KHP^-)(I-KH)^T + KRK^T$$

$$P^+ = P^- - KHP^- - P^-H^TK^T + KHP^-H^TK^T + KRK^T$$

$$P^+ = P^- - KHP^- - P^-H^TK^T + K[HP^-H^T+R]K^T$$

$$P^+ = P^- - KHP^- - P^-H^TK^T + KK^{-1}P^-H^TK^T$$

$$P^+ = P^- - KHP^-$$

$$\boxed{P^+ = (I-KH)P^-}$$

from this form, we can also derive that

$$\boxed{P^+ = ((P^-)^{-1} + H^TR^{-1}H)^{-1}}$$

Using matrix inversion lemma (part (a))

Problem 2: Information filter.

Part (a):

Using corresponding prediction equations of the Kalman filters,

$$\bar{x}_{K+1} = A_K \bar{x}_K + B_K u_K$$

$$P_{K+1} = A_K P_K A_K^T + Q_K$$

Substituting $\bar{x} = \Omega^{-1}\xi$ $\quad P = \Omega^{-1}$ we get

$$\Omega_{K+1}^{-1}\xi_{K+1} = A_K \Omega_K^{-1}\xi_K + B_K u_K \longrightarrow (1)$$

$$\Omega_{K+1}^{-1} = A_K \Omega_K^{-1} A_K^T + Q_K$$

from this we have

$$\boxed{\Omega_{K+1} = \left(A_K \Omega_K^{-1} A_K^T + Q_K\right)^{-1}}$$

from (1), we have

$$\boxed{\xi_{K+1} = \Omega_{K+1}\left(A_K \Omega_K^{-1}\xi_K + B_K u_K\right)}$$

✓

Part(b): Using the correction step of KF, we have

$$P^+ = ((P^-)^{-1} + H^T R^{-1} H)^{-1}$$

$P = \Omega^{-1}$ → given.

$$(\Omega^+)^{-1} = (((\Omega^-)^{-1})^{-1} + H^T R^{-1} H)^{-1}$$

$$(\Omega^+)^{-1} = (\Omega^- + H^T R^{-1} H)^{-1}$$

$$\boxed{\Omega^+ = H^T R^{-1} H + \Omega^-}$$

Also, we have $\bar{x}^+ = \bar{x}^- + K(z - H\bar{x}^-)$ where $K = P^+ H^T R^{-1}$

Hence

$$\bar{x}^+ = \bar{x}^- + Kz - KH\bar{x}^-$$

$$(\Omega^+)^{-1} \xi^+ = (\Omega^-)^{-1} \xi^- + Kz - KH(\Omega^-)^{-1} \xi^-$$

$$(\Omega^+)^{-1} \xi^+ = (\Omega^-)^{-1} \xi^- + P^+ H^T R^{-1} z - P^+ H^T R^{-1} H (\Omega^-)^{-1} \xi^-$$

$$(\Omega^+)^{-1} \xi^+ = (\Omega^-)^{-1} \xi^- + (\Omega^+)^{-1} H^T R^{-1} z - (\Omega^+)^{-1} H^T R^{-1} H (\Omega^-)^{-1} \xi^-$$

$$\xi^+ = (\Omega^+)(\Omega^-)^{-1} \xi^- + H^T R^{-1} z - H^T R^{-1} H (\Omega^-)^{-1} \xi^-$$

$$\xi^+ = H^T R^{-1} \cancel{H(\Omega^-)^{-1}} \xi^- + (\Omega^-)(\Omega^-)^{-1} \xi^- + H^T R^{-1} z - H^T R^{-1} \cancel{H(\Omega^-)^{-1}} \xi^-$$

$$\boxed{\xi^+ = H^T R^{-1} z + \xi^-}$$

(c)

The prediction steps of Information filter involves the inversion of two matrices of the size $n \times n$, where n is the dimension of the state space.

complexity for Inversion — $O(n^{2.8})$ (Time)

But in the case of Kalman filter, this prediction/update step is additive and requires $O(n^2)$ time

The computational load can be even cheaper

(i) if only a subset of variables is affected by a control

(ii) if variables transition independently of each other

(d) The correction step for information filter is additive i.e. requires at most $O(n^2)$ time. Information filter is efficient if measurements carry only information about a subset of all state variables at a time.

In the case of Kalman filter, the correction step requires matrix inversion which has a worst case complexity $O(n^{2.8})$

(c)

Advantages of Information filter over Kalman filter:

① with respect to robotics problems, representing global uncertainty is easier in information filter ($\Omega = 0$) Such uncertainty representation leads to a covariance of infinite magnitude in Kalman filter.

② Information filter is numerically more stable than Kalman filter.

③ Information filter is a natural fit for multi-robot problems. The canonical parameters of IF represent a probability in logarithmic form. IF integrate information in arbitrary order and in a completely decentralized manner. In the case of KF, the necessary overhead for doing so is very high

④ Complexity:

Kalman filter
- efficient prediction step : $O(n^2)$
- costly correction step : $O(n^{2.8})$ → slow

Information filter
- costly prediction step : $O(n^{2.8})$ → slow
- efficient correction step : $O(n^2)$

```
clc;
clear all;
close all;
```

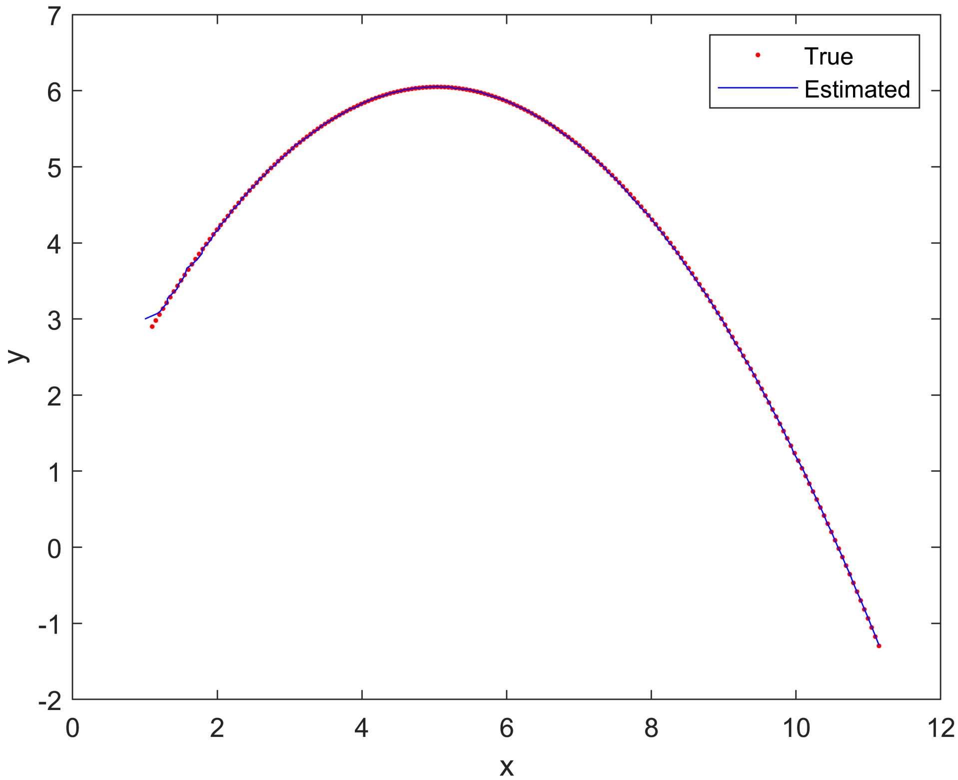
```
% Simulation Parameters
N=201;
t=linspace(0,20,N);
h=t(2)-t(1);

sigma_x=1e-3;
sigma_y=1e-3;
sigma_t1=3*pi/180;
sigma_t2=5*pi/180;

Q=diag([0 sigma_x^2 0 sigma_y^2]);
R=diag([sigma_t1^2 sigma_t2^2]);

X0=[1 0.5 3 0.8]';
P0=diag([0.1^2 0.1^2 0.1^2 0.1^2]);
```

```
% True Trajectory
w_x=normrnd(0,sigma_x,1,N);
w_y=normrnd(0,sigma_y,1,N);
w=[zeros(1,N); w_x; zeros(1,N); w_y];
v_t1=normrnd(0,sigma_t1,1,N);
v_t2=normrnd(0,sigma_t2,1,N);
v=[v_t1;v_t2];

X_true=zeros(4,N);
X_true(:,1)=X0+[0.1 0 -0.1 0]';

Ak=[1 h 0 0;
    0 1 0 0;
    0 0 1 h;
    0 0 0 1];
B=[0;0;h^2/2;h];
u=-0.1;

for k=1:N-1
    X_true(:,k+1) = Ak*X_true(:,k) + B*u + w(:,k);
end


for k = 1:N
    x(:,k) = X_true(1,k);
    y(:,k) = X_true(3,k);
    z(:,k) = [sqrt((x(:,k)-4)^2 + y(:,k)^2);
        sqrt((x(:,k)-8)^2 + y(:,k)^2)] + v(:,k);
end

%figure(2);
%subplot(2,1,1);
```

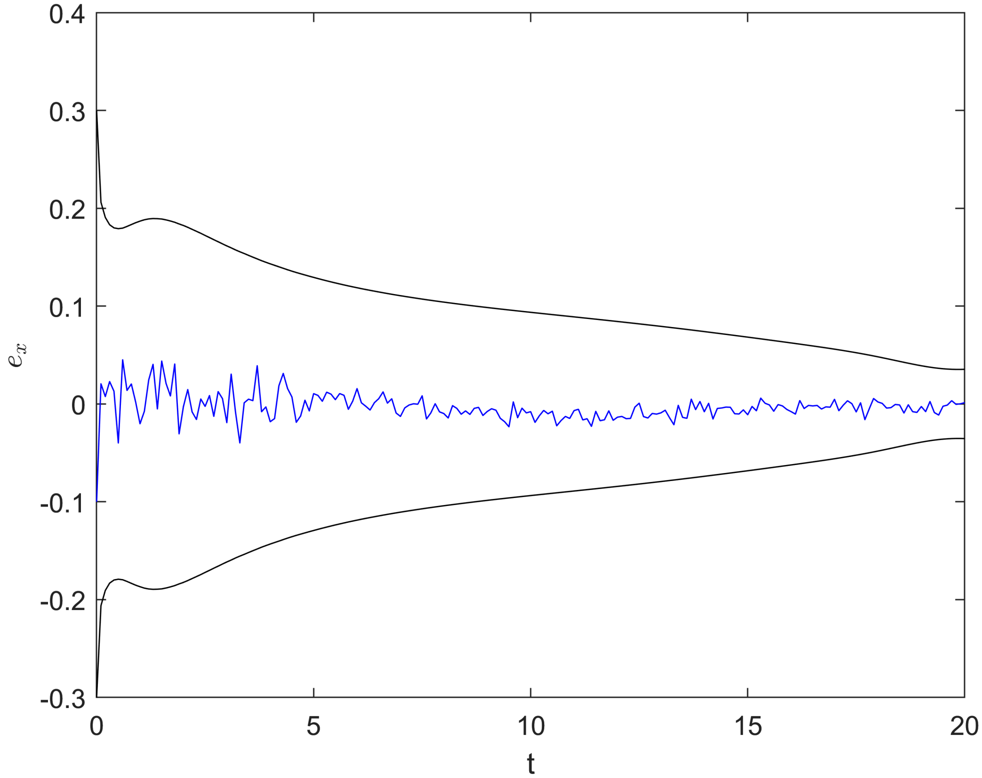
```
%plot(t,x,'r');
%plot(t,z(1,:),'k.');
%ylabel('$r$','interpreter','latex');
%subplot(2,2,2);
%plot(t,y,'r');
%plot(t,z(2,:),'k.');
%ylabel('$r$','interpreter','latex');
```

```
% EKF
X_bar = zeros(4,N);
P = zeros(4,4,N);
K = zeros(2,N);
X_bar(:,1) = X0;
P(:,:,1) = P0;

for k=1:N-1
    % prediction
    X_bar_kp = Ak*X_bar(:,k) + B*u;
    P_kp = Ak*P(:,:,k)*Ak' + Q;

    % Correction
    x = X_bar_kp(1);
    y = X_bar_kp(3);
    z_bar = [sqrt((x-4)^2 + y^2); sqrt((x-8)^2 + y^2)] + v(:,k);
    H = [(2*x - 8)/(2*sqrt((x-4)^2 + y^2)) 0 y/(sqrt((x-4)^2 + y^2)) 0;
        (2*x - 16)/(2*sqrt((x-8)^2 + y^2)) 0 y/(sqrt((x-8)^2 + y^2)) 0];
    K = P_kp*H'*inv(H*P_kp*H' +R);
    X_bar_kp = X_bar_kp + K * (z(:,k+1)-z_bar);
    P_kp = (eye(4) - K*H)*P_kp;
    P(:,:,k+1) = P_kp;
    X_bar(:,k+1) = X_bar_kp;
end
```

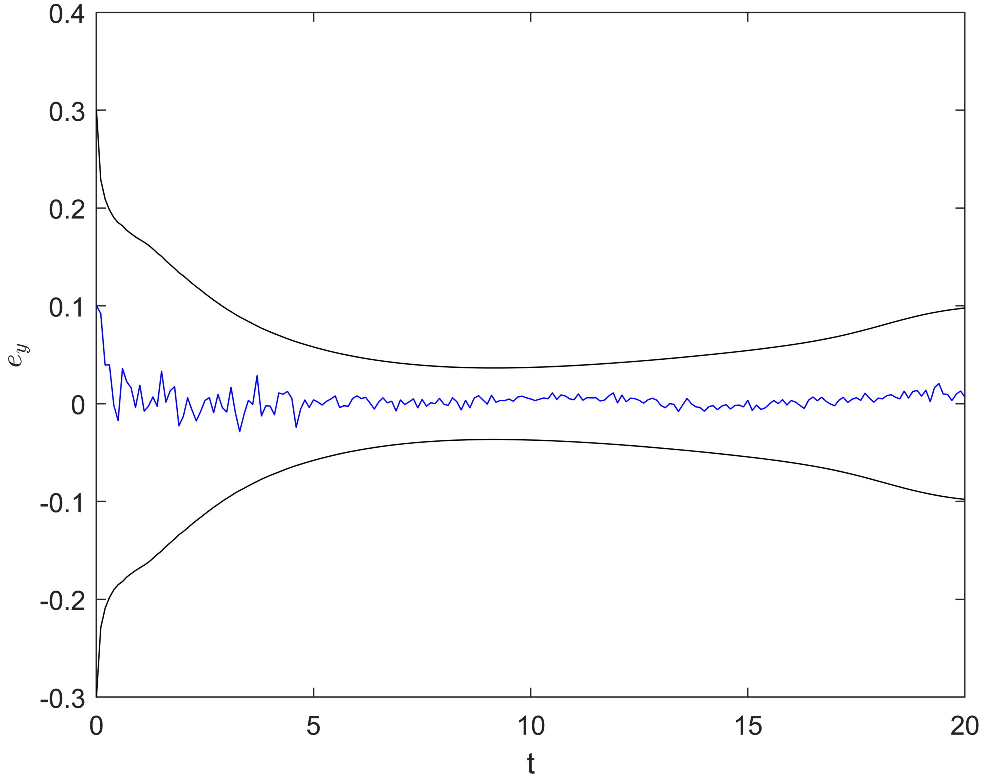
```
% Estimated Trajectory
figure
plot(X_true(1,:),X_true(3,:), 'r.'); hold on;
plot(X_bar(1,:),X_bar(3,:),'b');
xlabel('x');
ylabel('y');

legend('True', 'Estimated');
hold off
```



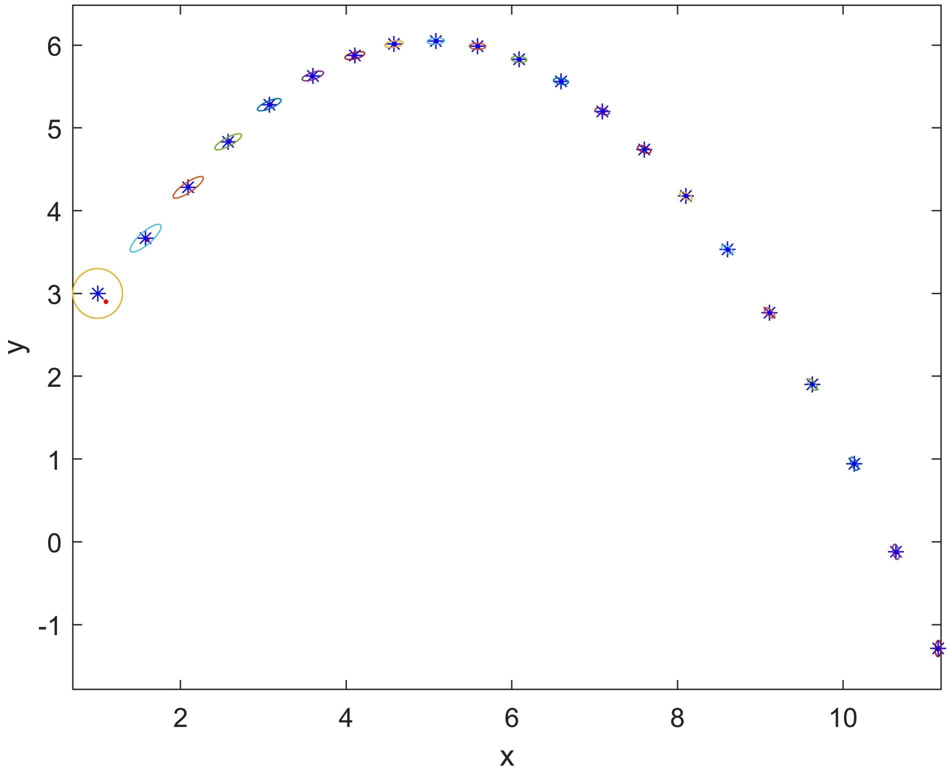

```
% Part(c) - Estimation Error
for k = 1:N
    sigma_x(k) = sqrt(P(1,1,k));
    sigma_y(k) = sqrt(P(3,3,k));
end

% 3sigma bounds with respect to t for x
figure
plot(t,X_bar(1,:)-X_true(1,:),'b',t,3*sigma_x,'k',t,-3*sigma_x,'k');
xlabel('t');
ylabel('$e_x$','interpreter','latex');
```




```
% 3sigma bounds with respect to t for y
figure
plot(t,X_bar(3,:)-X_true(3,:),'b',t,3*sigma_y,'k',t,-3*sigma_y,'k');
xlabel('t');
ylabel('$e_y$','interpreter','latex');
```




```
% Part(d): Plot the 3sigma Gaussian ellipsoid.
figure
for k=1:10:N
    plot(X_true(1,k),X_true(3,k),'r.');hold on;
    plot(X_bar(1,k),X_bar(3,k),'b*');
    plot_gaussian_ellipsoid(X_bar([1,3],k),P([1 3],[1 3],k),3);
end
axis equal;
xlabel('x');
ylabel('y');
```


6
5
4
3
2
1
0
-1
y
2
4
6
8
10
x