Aniket.M.Anbhule

# HW 08

Q.1.

→ $x_{A(k+1)} = F_A x_{A_k} + N\sim(0, Q_A)$

$x_{B(k+1)} = F_B x_{B_k} + N\sim(0, Q_B)$

Using Van loan's method for calculating $Q_A$ & $Q_B$ matrices.
$(F_A, Q_A)$ & $(F_B, Q_B)$ are represented in MATLAB

Q.2.

A)
→ $y = Hx + \text{mvnrnd}(\text{mean}, \text{cov})^T$

B)
→ $\hat{x}_{(0)} = Mu_{(0)}$

Using KF measurement update & Time update for estimation of states $\hat{x}$.

Q.3.

A)
→ $r_A = \begin{bmatrix} \xi_A \\ \eta_A \end{bmatrix}$, $r_B = \begin{bmatrix} \xi_B \\ \eta_B \end{bmatrix}$, $R_d = \begin{bmatrix} 10 & 0.15 \\ 0.15 & 10 \end{bmatrix}$

$y_{(D)} = r_{A(k)} - r_{B(k)} + v_{D(k)}$, $E[v_D(k)] = 0$

$E[v_D(k) v_D^T(j)] = R_D \delta(i,j)$

$$\begin{bmatrix} y_{A(k+1)} \\ y_{D(k+1)} \end{bmatrix} = \begin{bmatrix} H & 0 \\ H & -H \end{bmatrix} \begin{bmatrix} x_{A(k)} \\ x_{B(k)} \end{bmatrix} + V$$

$$x_{(k+1)} = F x_{(k)}$$

$$x_{(k+1)} = F^{k+1} x_{(0)}$$

OR $$x_{(k)} = F^{k} x_{(0)}$$

$$x_{A(k)} = F_A^k \, x_{A(0)}$$
$$x_{B(k)} = F_B^k \, x_{D(0)}$$
$$\Rightarrow \begin{bmatrix} x_{A(k)} \\ x_{B(k)} \end{bmatrix} = \begin{bmatrix} F_A^k & 0 \\ 0 & F_B^k \end{bmatrix} \begin{bmatrix} x_{A(0)} \\ x_{B(0)} \end{bmatrix}$$

$$R = \begin{bmatrix} R_a & 0 \\ 0 & R_d \end{bmatrix}, \quad x_{A(0)} = Mu_{a(0)}$$
$$x_{B(0)} = Mu_{b(0)}$$

$$P = \begin{bmatrix} P_a & 0 \\ 0 & P_b \end{bmatrix}, \quad \varphi = \begin{bmatrix} \varphi_A & 0 \\ 0 & \varphi_B \end{bmatrix}$$

B)

→ For part B everything remains the same except that now we do not consider ground/true measurements, hence, this is the case of pure prediction.

H matrix & R matrix changes as follows,

$$H = [H \quad -H], \quad R = R_d$$

# HW - 08

## Q1

```
dt = 0.5;
xa = 0.045;
A = [0 1 0 0; 0 0 0 -xa; 0 0 0 1; 0 xa 0 0];
Fa = expm(A*dt)
```

```
Fa = 4×4
    1.0000    0.5000         0   -0.0056
         0    0.9997         0   -0.0225
         0    0.0056    1.0000    0.5000
         0    0.0225         0    0.9997
```

```
xb = -0.045;
B = [0 1 0 0; 0 0 0 -xb; 0 0 0 1; 0 xb 0 0];
Fb = expm(B*dt)
```

```
Fb = 4×4
    1.0000    0.5000         0    0.0056
         0    0.9997         0    0.0225
         0   -0.0056    1.0000    0.5000
         0   -0.0225         0    0.9997
```

```
gammaA = [0 0; 1 0; 0 0; 0 1];
gammaB = [0 0; 1 0; 0 0; 0 1];

qw = 10;
W = qw * [2 0.05; 0.05 0.5];
Z_a = dt * [-A gammaA*W*(gammaA)'; zeros(4,4) A'];
eZ_a = expm(Z_a);

Qa = eZ_a(5:8,5:8)' * eZ_a(1:4,5:8)
```

```
Qa = 4×4
    0.8329    2.4983    0.0261    0.0953
    2.4983    9.9931    0.0719    0.3343
    0.0261    0.0719    0.2087    0.6266
    0.0953    0.3343    0.6266    2.5069
```

```
Z_b = dt * [-B gammaB*W*(gammaB)'; zeros(4,4) B'];
eZ_b = expm(Z_b);

Qb = eZ_b(5:8,5:8)' * eZ_b(1:4,5:8)
```

```
Qb = 4×4
    0.8336    2.5011    0.0156    0.0297
    2.5011   10.0044    0.0531    0.1656
    0.0156    0.0531    0.2080    0.6238
    0.0297    0.1656    0.6238    2.4956
```

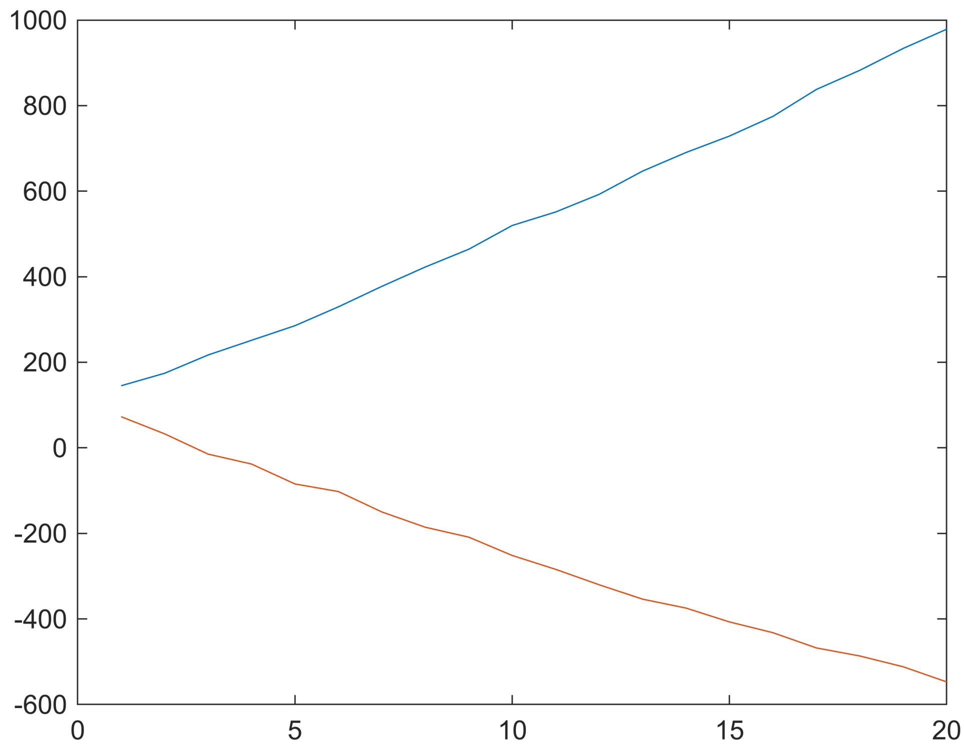
## Q2. A)

```
rng(100)
```

```
H = [1 0 0 0; 0 0 1 0];
Ra = [20 0.05; 0.05 20];

y_all = [];
for n = 1:201
    y = H*xasingle_truth(:,n) + (mvnrnd(zeros(2,1),Ra))';
    y_all = [y_all,y];
end

y_all;
y_100 = y_all(:,2:101)
```

```
y_100 = 2×100
10^3 ×
    0.1452    0.1743    0.2170    0.2514    0.2855    0.3296    0.3775    0.4227 ···
    0.0728    0.0327   -0.0146   -0.0379   -0.0844   -0.1022   -0.1499   -0.1856
```

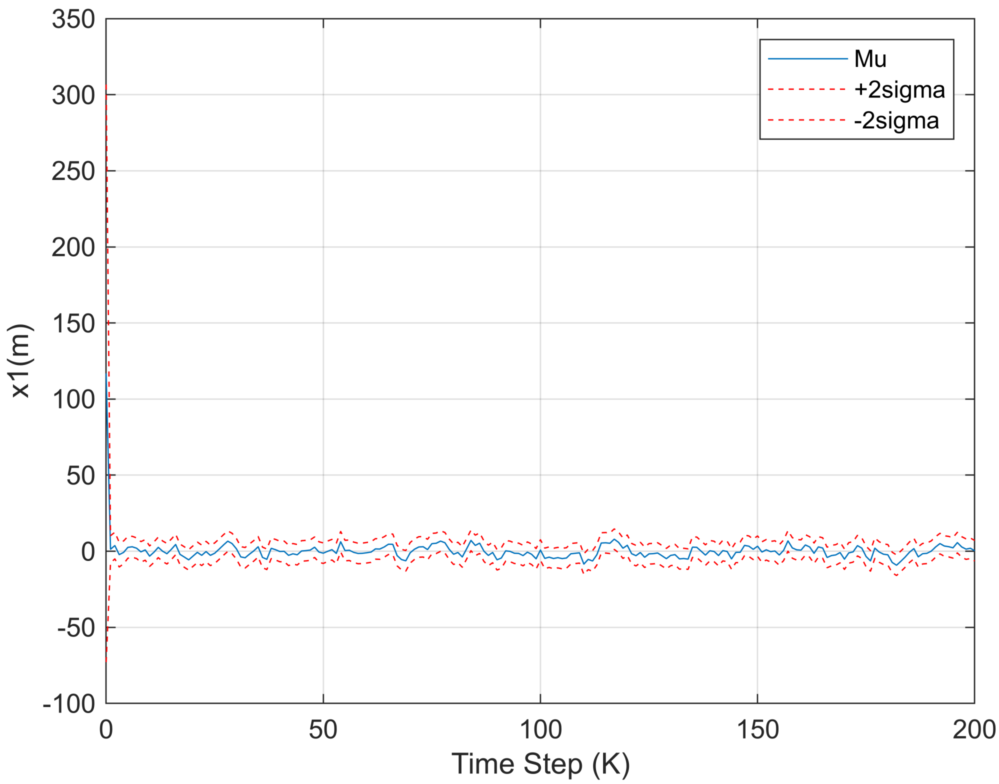
```
y_20 = y_all(:,2:21);
t = 1:20;
plot (t,y_20)
```



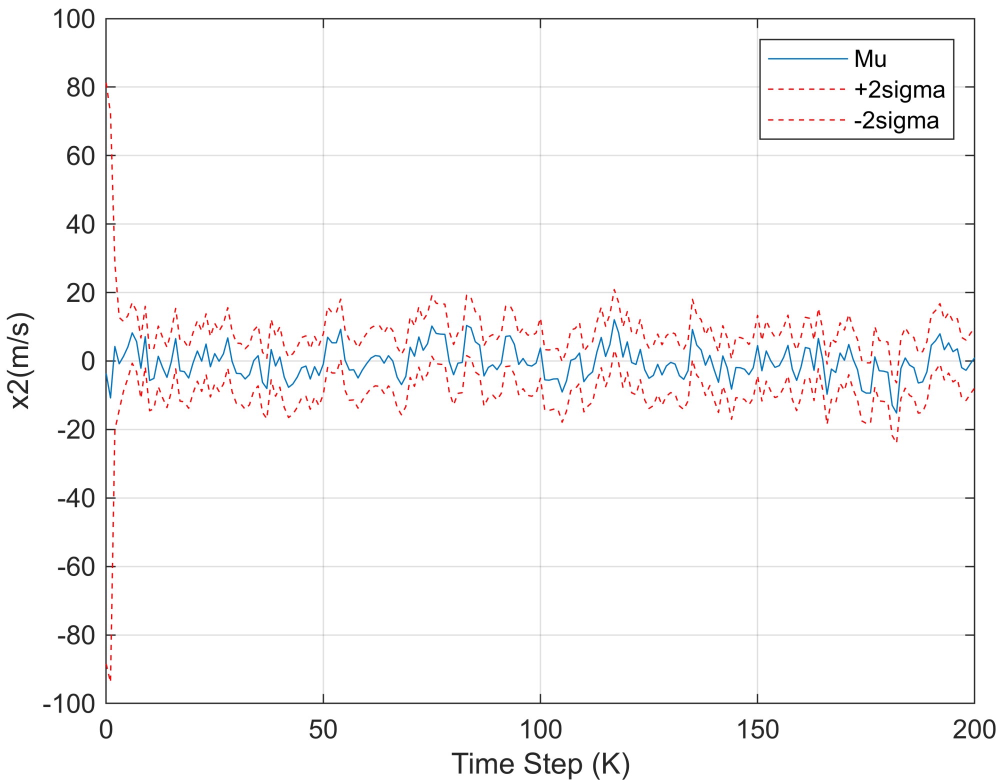

## Q2. B)

```
A = [0 1 0 0; 0 0 0 -xa; 0 0 0 1; 0 xa 0 0];
Fa = expm(A*dt);
H = [1 0 0 0; 0 0 1 0];
Ra = [20 0.05; 0.05 20];
```

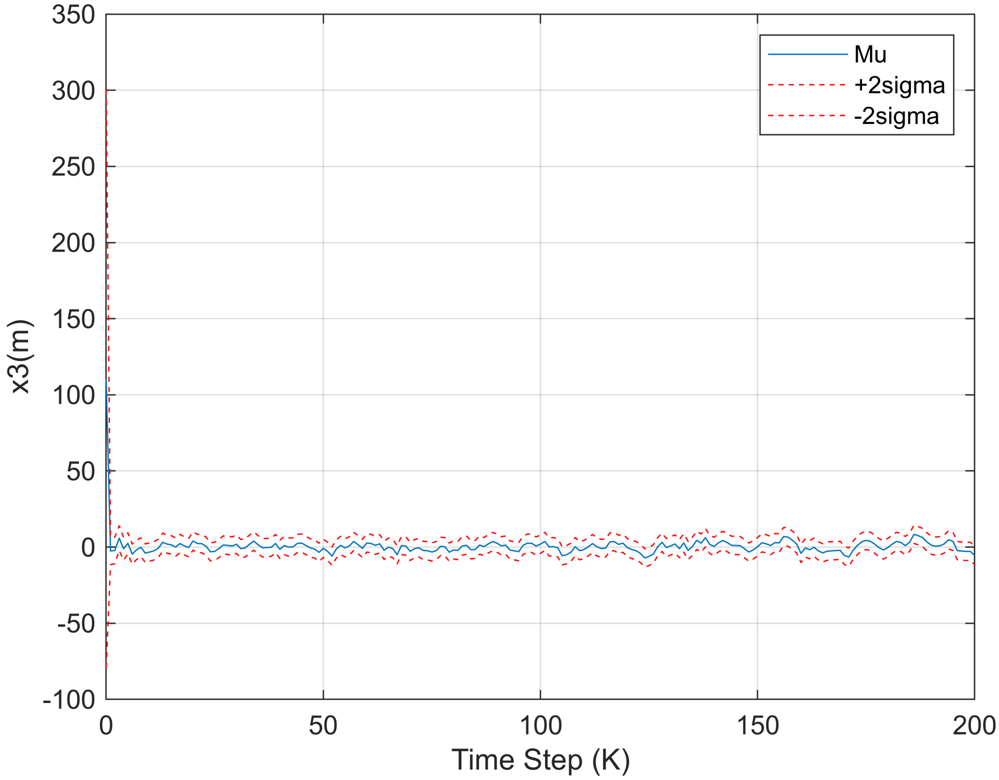
```
Hall = repmat(H,201,1);
Rall = [];

y_all;

for n = 1:201
    Ra = [20 0.05; 0.05 20];
    Rall = blkdiag(Rall,Ra);
end
Rall;

% Kalman filtering
Mua0 = [0; 85*cos(pi/4); 0; -85*sin(pi/4)];
xall = zeros(4,201);
xall(:,1) = Mua0;
v = [10 2 10 2];
Pa0 = 900 * diag(v);
Pall = zeros(4,200);
Pall(:,1) = 900*[10;2;10;2];

for k = 1:200
    % KF Time update
    xall(:,k+1) = Fa*xall(:,k);
    Pa0 = Fa*Pa0*Fa'+ Qa;
    Kk = Pa0*H'*inv(H*Pa0*H' + Ra);
    % KF Measurement update
    xall(:,k+1) = xall(:,k+1) + Kk*(y_all(:,k+1)-H*xall(:,k+1));
    Pa0 = (eye(4,4) - Kk*H)*Pa0;
    Pall(:,k+1) = diag(Pa0);
end

xall;
Pall;
x_err = xasingle_truth - xall;
sigma = sqrt(Pall);
sigma2p = x_err + 2*sigma;
sigma2n = x_err - 2*sigma;

time = 0:200;
plot(time,x_err(1,:)),grid on,ylabel('x1(m)'),xlabel("Time Step (K)")
hold on
plot(time,sigma2p(1,:),'r--'),grid on
plot(time,sigma2n(1,:),'r--'),grid on
legend("Mu","+2sigma","-2sigma")
hold off
```



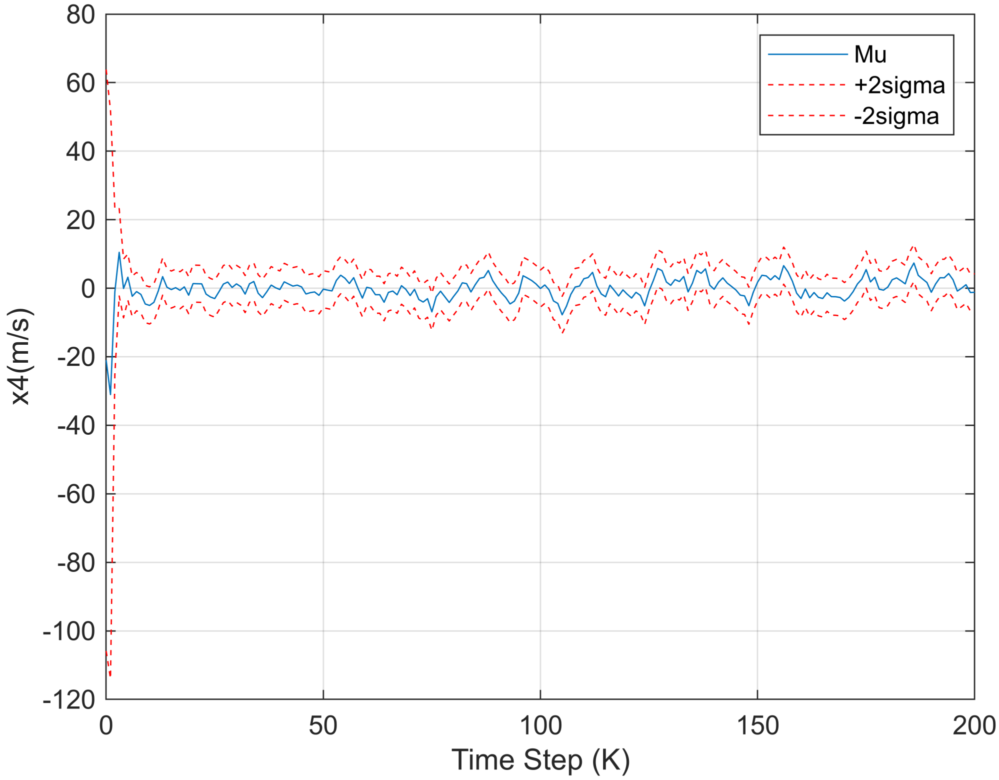

```
plot(time,x_err(2,:)),grid on,ylabel('x2(m/s)'),xlabel("Time Step (K)")
hold on
plot(time,sigma2p(2,:),'r--'),grid on
plot(time,sigma2n(2,:),'r--'),grid on
legend("Mu","+2sigma","-2sigma")
hold off
```




```
plot(time,x_err(3,:)),grid on,ylabel('x3(m)'),xlabel("Time Step (K)")
hold on
plot(time,sigma2p(3,:),'r--'),grid on
plot(time,sigma2n(3,:),'r--'),grid on
legend("Mu","+2sigma","-2sigma")
hold off
```




```
plot(time,x_err(4,:)),grid on,ylabel('x4(m/s)'),xlabel("Time Step (K)")
hold on
plot(time,sigma2p(4,:),'r--'),grid on
plot(time,sigma2n(4,:),'r--'),grid on
legend("Mu","+2sigma","-2sigma")
hold off
```

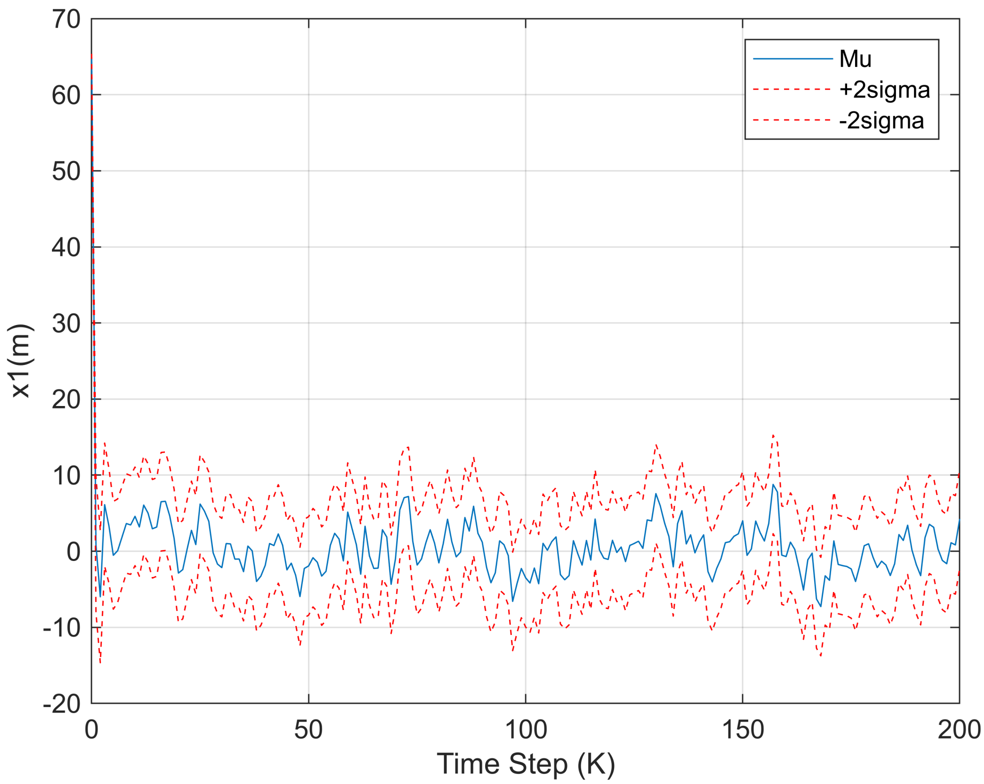



Q2. B) The KF output for states X1 & X3 have more certanity than states X2 & X4 because it depends on the measurements y since we have measurements only for position, velocity propagates with just prediction, hence more uncertainity.

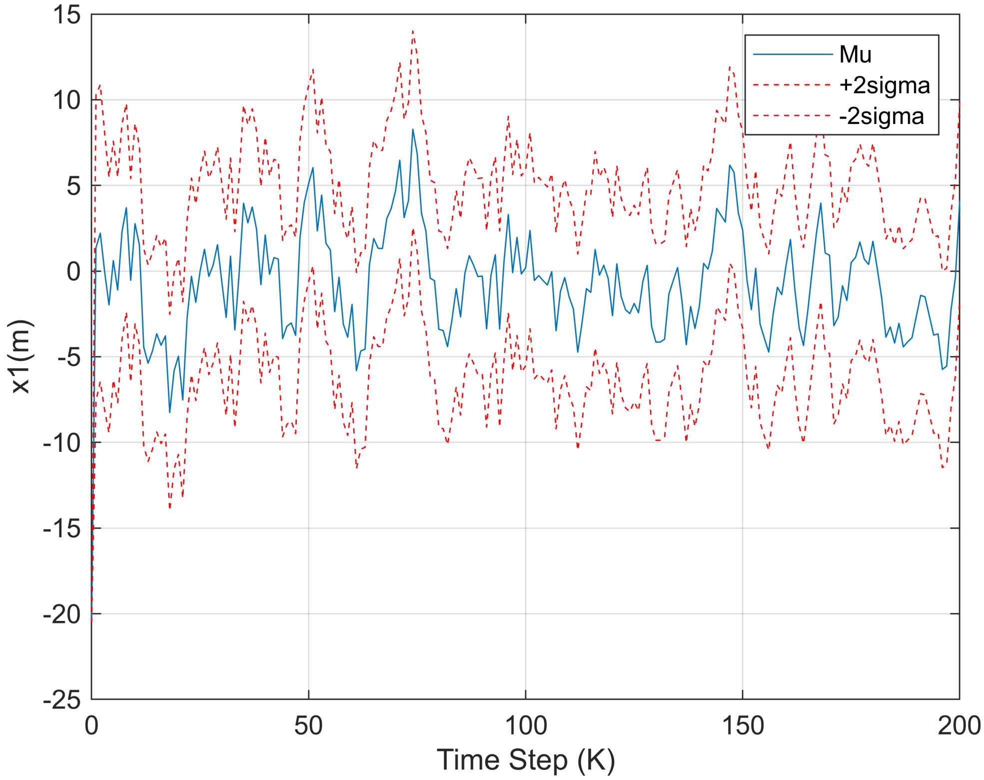
## Q3.A)

```
Ra = [20 0.05; 0.05 20];
Rd = [10 0.15; 0.15 10];

Hall = [H zeros(2,4); H -H];
R = [Ra zeros(2,2); zeros(2,2) Rd];
yd_k = [];
ya_new_k = [];

for n = 1:201
    yd = H*xadouble_truth(:,n) - H*xbdouble_truth(:,n) + (mvnrnd(zeros(2,1),Rd))';
    yd_k = [yd_k,yd];
    ya_new = H*xadouble_truth(:,n) + (mvnrnd(zeros(2,1),Ra))';
    ya_new_k = [ya_new_k,ya_new];
end

ys = vertcat(ya_new_k,yd_k);

% Kalman filtering
Mua0 = [0; 85*cos(pi/4); 0; -85*sin(pi/4)];
```

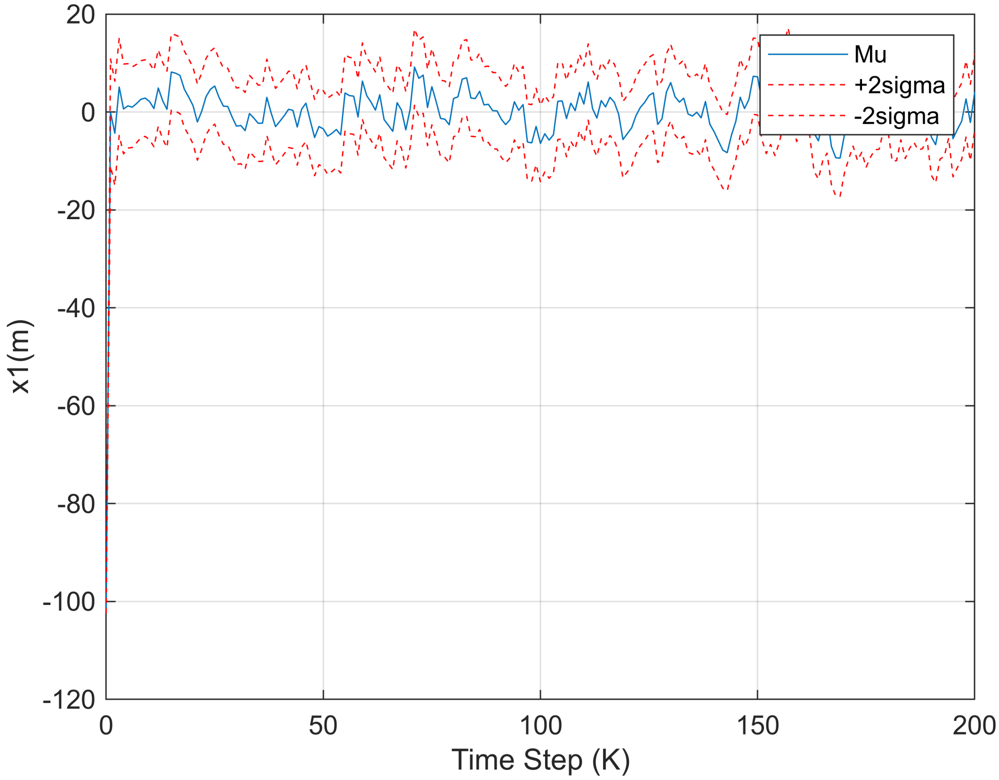
```
Pa0 = 900*diag([10,2,10,2]);
Mub0 = [3200; 85*cos(pi/4); 3200; -85*sin(pi/4)];
Pb0 = 900*diag([11,4,11,4]);
xa_all = zeros(4,201);
xa_all(:,1) = Mua0;
xb_all = zeros(4,201);
xb_all(:,1) = Mub0;
xs_all = vertcat(Mua0,Mub0);
Q_new = blkdiag(Qa,Qb);
P_new = blkdiag(Pa0,Pb0);
F_new = blkdiag(Fa,Fb);
H_new = [H zeros(2,4); H -H];

for k = 1:200
    % KF Time update
    xs_all(:,k+1) = F_new*xs_all(:,k);
    P_new = F_new*P_new*F_new'+ Q_new;
    Kk = P_new*H_new'*inv(H_new*P_new*H_new' + R);
    % KF Measurement update
    xs_all(:,k+1) = xs_all(:,k+1) + Kk*(ys(:,k+1)-H_new*xs_all(:,k+1));
    P_new = (eye(8,8) - Kk*H_new)*P_new;
    P_new_all(:,k+1) = diag(P_new);
end

P_new;
xs_all;
P_new_all;
xs_err = vertcat(xadouble_truth,xbdouble_truth) - xs_all;
sigma_s = sqrt(P_new_all);
sigma2p_s = xs_err + 2*sigma_s;
sigma2n_s = xs_err - 2*sigma_s;

time = 0:200;
plot(time,xs_err(1,:)),grid on,ylabel('x1(m)'),xlabel("Time Step (K)")
hold on
plot(time,sigma2p_s(1,:),'r--'),grid on
plot(time,sigma2n_s(1,:),'r--'),grid on
legend("Mu","+2sigma","-2sigma")
hold off
```



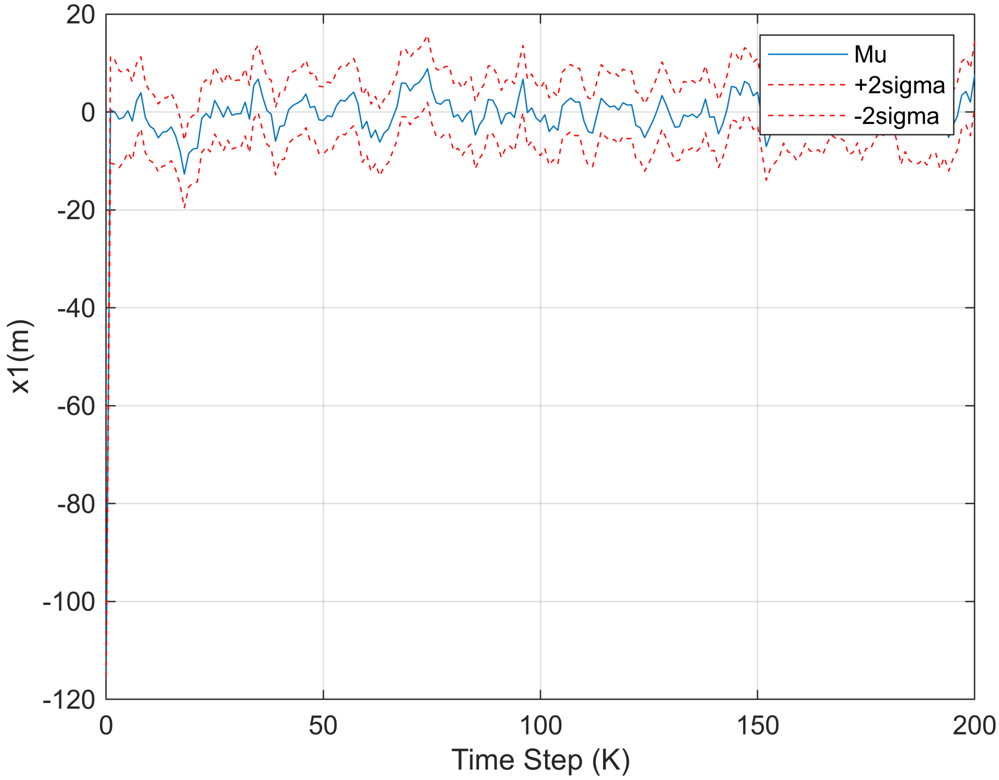

```
plot(time,xs_err(3,:)),grid on,ylabel('x1(m)'),xlabel("Time Step (K)")
hold on
plot(time,sigma2p_s(3,:),'r--'),grid on
plot(time,sigma2n_s(3,:),'r--'),grid on
legend("Mu","+2sigma","-2sigma")
hold off
```




```
plot(time,xs_err(5,:)),grid on,ylabel('x1(m)'),xlabel("Time Step (K)")
hold on
plot(time,sigma2p_s(5,:),'r--'),grid on
plot(time,sigma2n_s(5,:),'r--'),grid on
legend("Mu","+2sigma","-2sigma")
hold off
```



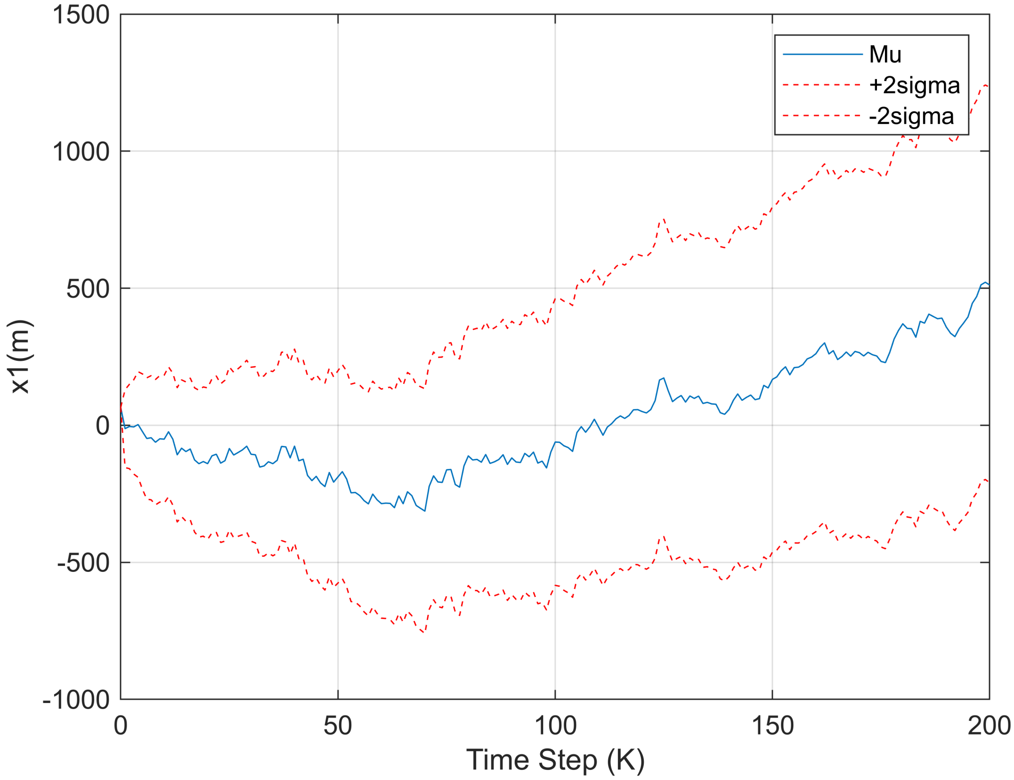

```
plot(time,xs_err(7,:)),grid on,ylabel('x1(m)'),xlabel("Time Step (K)")
hold on
plot(time,sigma2p_s(7,:),'r--'),grid on
plot(time,sigma2n_s(7,:),'r--'),grid on
legend("Mu","+2sigma","-2sigma")
hold off
```



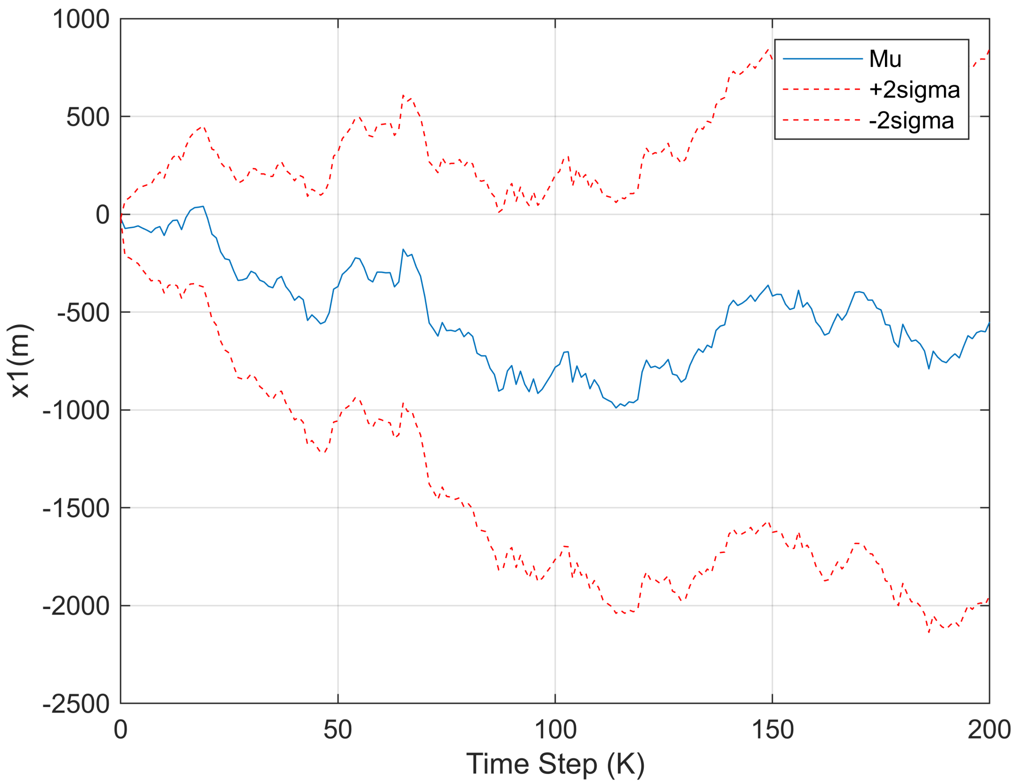

## Q3. B)

```
Rd = [10 0.15; 0.15 10];

Hd_all = [H -H];
yd_k = [];
ya_new_k = [];

for n = 1:201
    yd = H*xadouble_truth(:,n) - H*xbdouble_truth(:,n) + (mvnrnd(zeros(2,1),Rd))';
    yd_k = [yd_k,yd];
end

% Kalman filtering
Mua0 = [0; 85*cos(pi/4); 0; -85*sin(pi/4)];
Pa0 = 900*diag([10,2,10,2]);
Mub0 = [3200; 85*cos(pi/4); 3200; -85*sin(pi/4)];
Pb0 = 900*diag([11,4,11,4]);
xa_all = zeros(4,201);
xa_all(:,1) = Mua0;
xb_all = zeros(4,201);
xb_all(:,1) = Mub0;
xs_all = vertcat(Mua0,Mub0);
Q_new = blkdiag(Qa,Qb);
P_d = blkdiag(Pa0,Pb0);
```

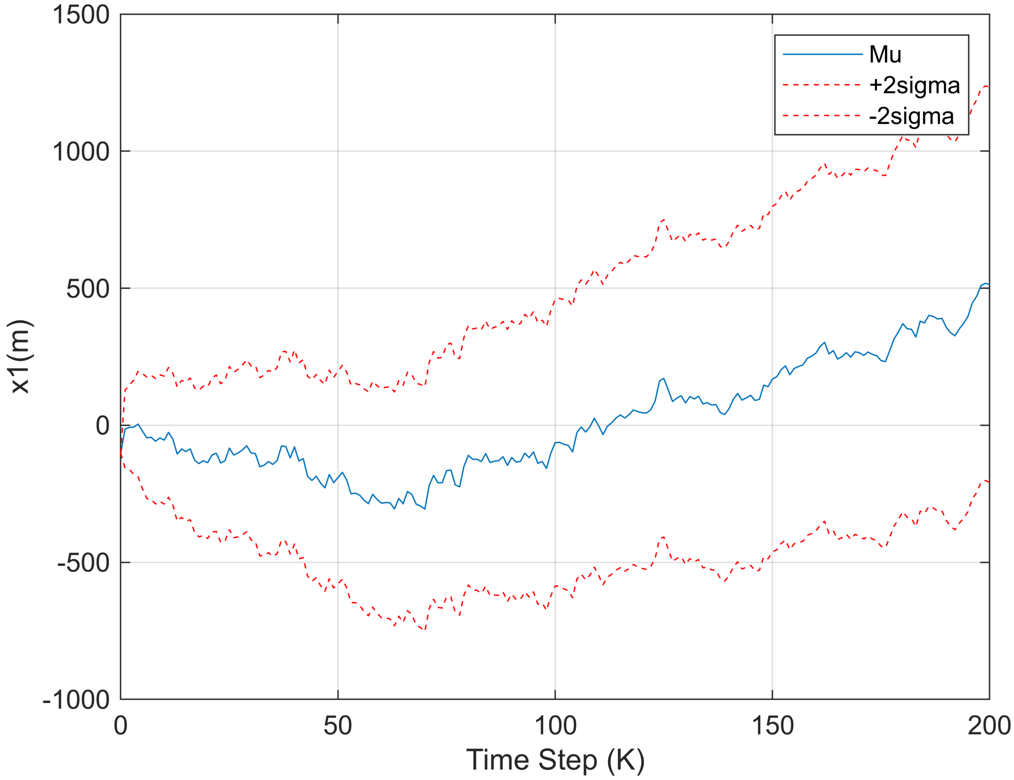
```
F_new = blkdiag(Fa,Fb);

for k = 1:200
    % KF Time update
    xs_all(:,k+1) = F_new*xs_all(:,k);
    P_d = F_new*P_d*F_new'+ Q_new;
    Kk = P_d*Hd_all'*inv(Hd_all*P_d*Hd_all' + Rd);
    % KF Measurement update
    xs_all(:,k+1) = xs_all(:,k+1) + Kk*(yd_k(:,k+1)-Hd_all*xs_all(:,k+1));
    P_d = (eye(8,8) - Kk*Hd_all)*P_d;
    Pd_new_all(:,k+1) = diag(P_d);
end

P_d;
xs_all;
Pd_new_all;
xs_err = vertcat(xadouble_truth,xbdouble_truth) - xs_all;
sigma_s = sqrt(Pd_new_all);
sigma2p_s = xs_err + 2*sigma_s;
sigma2n_s = xs_err - 2*sigma_s;

time = 0:200;
plot(time,xs_err(1,:)),grid on,ylabel('x1(m)'),xlabel("Time Step (K)")
hold on
plot(time,sigma2p_s(1,:),'r--'),grid on
plot(time,sigma2n_s(1,:),'r--'),grid on
legend("Mu","+2sigma","-2sigma")
hold off
```



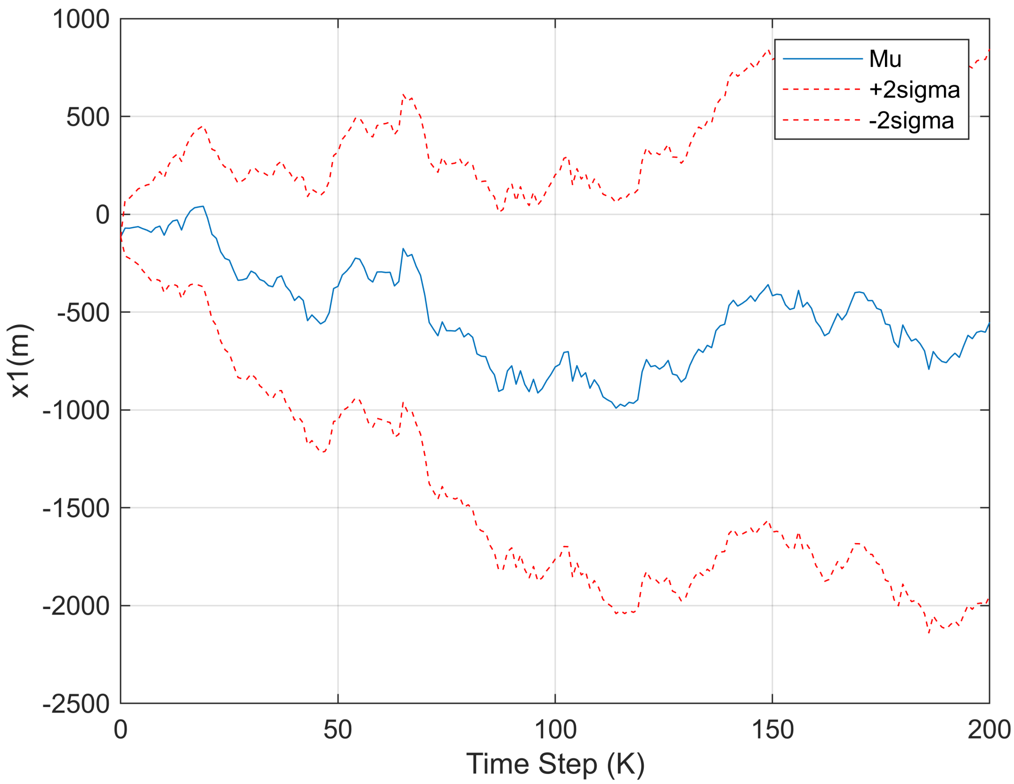

```
plot(time,xs_err(3,:)),grid on,ylabel('x1(m)'),xlabel("Time Step (K)")
hold on
plot(time,sigma2p_s(3,:),'r--'),grid on
plot(time,sigma2n_s(3,:),'r--'),grid on
legend("Mu","+2sigma","-2sigma")
hold off
```




```
plot(time,xs_err(5,:)),grid on,ylabel('x1(m)'),xlabel("Time Step (K)")
hold on
plot(time,sigma2p_s(5,:),'r--'),grid on
plot(time,sigma2n_s(5,:),'r--'),grid on
legend("Mu","+2sigma","-2sigma")
hold off
```




```
plot(time,xs_err(7,:)),grid on,ylabel('x1(m)'),xlabel("Time Step (K)")
hold on
plot(time,sigma2p_s(7,:),'r--'),grid on
plot(time,sigma2n_s(7,:),'r--'),grid on
legend("Mu","+2sigma","-2sigma")
hold off
```




Q3. B) In part A, we are using measurements Y(a) hence the error and sigma bounds are converging but in part B we have completely eliminated the true marusements the entire kalman filtering is done on pure prediction because of which the covariance matrix matrix keeps on increasing which in turn results in increasing error. Therefore the error ends up around 500 units and the bounds keep on divering.

## Q3. C)

```
for k = 1:200
    xs_all(:,k+1) = F_new*xs_all(:,k);
    P_new = F_new*P_new*F_new'+ Q_new;
    Ps_new (:,k+1) = diag(P_new);
end


xc_all = zeros(8,n);
xc_all(:,1) = [Mua0;Mub0];
Pc = blkdiag(Pa,Pb);
pcb2 = zeros(8,n);
pcb2(:,1) = diag(blkdiag(Pa,Pb));

for i= 1:n-1
    %prediction step
    xc_all(:,i+1) = F_new*xc_all(:,i);
    Pc = F_new*Pc*transpose(F_new) + Q_new;
    pcb2(:,i+1) = diag(Pc);
```

```
end
```

The structure of covariance matrix in part A differs from covariance matrix in part B because in part A we have correlation between flight A & flight B whereas in part B we do not have any correlation, hence, the covariance matrix sturcture is different.