1. (a) $\begin{cases} x_{k+1} = A x_k \\ y_k = g(x_k) + D W_k \end{cases}$ system $\quad A = I_3$

$$\hat{x}_{k+1|k} = A \hat{x}_{k|k}$$

$$\hat{x}_{k+1|k+1} = \hat{x}_{k+1|k} + K_k (y_{k+1} - g(\hat{x}_{k+1|k}))$$

$$K_k = P_{k+1|k} C_{k+1}^T (C_{k+1} P_{k+1|k} C_{k+1}^T + R_{k+1})^{-1}$$

$$P_{k+1|k} = A_k P_{k|k} A_k^T + Q_k$$

$$P_{k+1|k+1} = P_{k+1|k} - K_k C_{k+1} P_{k+1|k}$$

$$C_{k+1} = \frac{\partial g}{\partial x}\Big|_{\hat{x}_{k+1|k}} = \begin{bmatrix} \frac{\hat{x}_k - x_1}{\sqrt{(\hat{x}_k - x_1)^2 + (\hat{y}_k - y_1)^2}} , & \frac{\hat{y}_k - y_1}{\sqrt{(\hat{x}_k - x_1)^2 + (\hat{y}_k - y_1)^2}} \\ \frac{\hat{x}_k - x_2}{\sqrt{(\hat{x}_k - x_2)^2 + (\hat{y}_k - y_2)^2}} , & \frac{\hat{y}_k - y_2}{\sqrt{(\hat{x}_k - x_2)^2 + (\hat{y}_k - y_2)^2}} \\ \frac{\hat{x}_k - x_3}{\sqrt{(\hat{x}_k - x_3)^2 + (\hat{y}_k - y_3)^2}} , & \frac{\hat{y}_k - y_3}{\sqrt{(\hat{x}_k - x_3)^2 + (\hat{y}_k - y_3)^2}} \end{bmatrix}$$

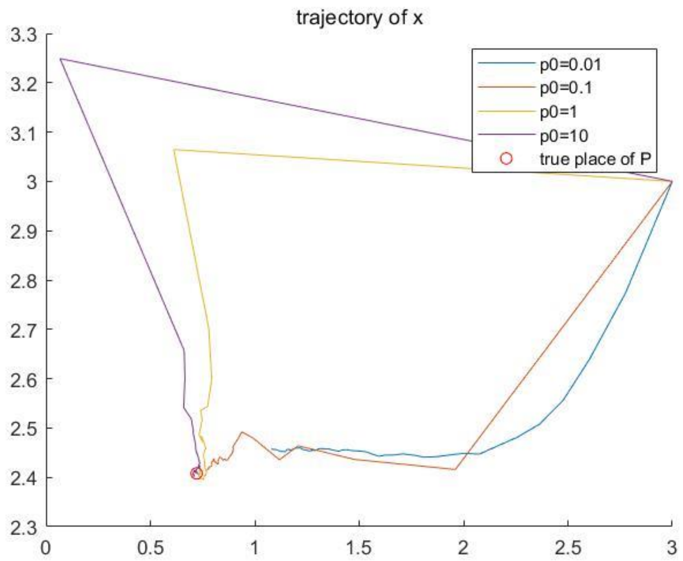
For 1(b) we use following codes.

```
clc;
clear all;

%%
%init parameters
p0 = 0.001;
X_all_all = {};
P_all_all = {};
for scale = 1:1:4
    p0 = p0*10;
    Q = 0;
    R = 0.1*eye(3);
    X_0 =[3;3];
    D = diag([0.1,0.1,0.1]);
    X_all = zeros(2,51);
    X_all(:,1) = X_0;
    A = eye(2);
    L1 = [0;0];
    L2 = [5;5];
    L3 = [2.5;0];
    P_true = [0.7212; 2.4080];
    P_0_0 = p0*eye(2);
    %%
    %iternation parameters
    P_all = zeros(2,2,51);
    P_all(:,:,1) = P_0_0;
    y = zeros(3,51);
    l_y = zeros(3,51);
    %%
    %kalman filter
    for k = 1:1:50
        w_k = normrnd(0,1,[3,1]);
        x_k_k = X_all(:,k);
        x_k1_k = A*x_k_k;
        y_k1 = [2.5;
                5;
                 3]+D*w_k;
        y(:,k) = y_k1;
        P_k_k = P_all(:,:,k);
        P_k1_k = A*P_k_k*A'+Q;
        C_k1 = linerity(x_k1_k,L1,L2,L3);
        Kk = P_k1_k*C_k1'/(C_k1*P_k1_k*(C_k1')+R);
```

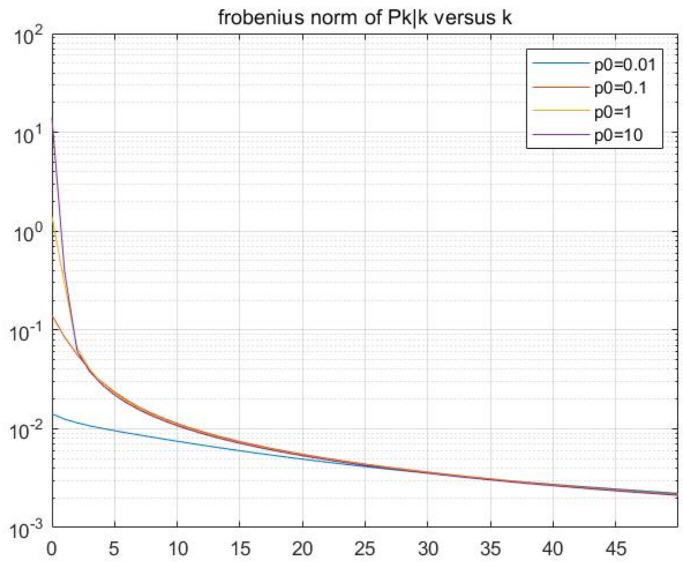
```
        P_k1_k1 = P_k1_k-Kk*C_k1*P_k1_k;
        l_y(:,k) = measure(x_k1_k,L1,L2,L3);
        x_k1_k1 = x_k1_k+Kk*(y_k1-l_y(:,k));
        P_all(:,:,k+1) = P_k1_k1;
        X_all(:,k+1) = x_k1_k1;
    end
    X_all_all{end+1} = X_all;
    P_all_all{end+1} = P_all;
end
figure(1)
hold on
for i = 1:1:length(X_all_all)
    X_all = X_all_all{i};
    plot(X_all(1,:),X_all(2,:));
end
plot(P_true(1),P_true(2),'ro');
legend("p0=0.01","p0=0.1","p0=1","p0=10","true place of P");
title("trajectory of x");
hold off

figure(2)
p_norm = zeros(4,51);
tt = 0:1:50;
for j = 1:1:length(P_all_all)
    P_all = P_all_all{j};
    for i=1:51
        p_norm(j,i) = norm(P_all(:,:,i),'fro');
    end
end
semilogy(tt,p_norm(1,:));
hold all
semilogy(tt,p_norm(2,:));
semilogy(tt,p_norm(3,:));
semilogy(tt,p_norm(4,:));
legend("p0=0.01","p0=0.1","p0=1","p0=10");
title("frobenius norm of Pk|k versus k");
grid on

%%
%linerity function for   EKF
function C_k1 = linerity(xk,L1,L2,L3)
    C_k1 = [(xk-L1)'/norm(xk-L1);
             (xk-L2)'/norm(xk-L2);
             (xk-L3)'/norm(xk-L3)];
```

```
end
function g_x = measure(xk,L1,L2,L3)
    g_x = [norm(xk-L1);
              norm(xk-L2);
              norm(xk-L3)];
end
```

And the pictures look like below:



P1: trajectory of x



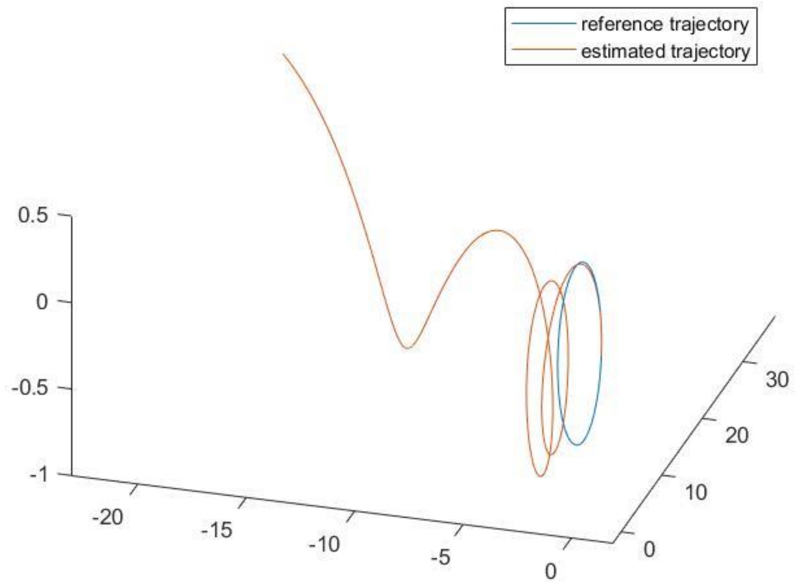
P2: forbenius norm of Pk|k versus k

For 2(a) we use following codes.

```
clc;
clear all;
load("rcwA.mat");
%plot3(rcwA_Ts_0_01(1,:),rcwA_Ts_0_01(2,:),rcwA_Ts_0_01(3,:));
T = 0.01;
A = [eye(3), T*eye(3);
        zeros(3,3), eye(3)];
B = [T^2/2*eye(3);
        T*eye(3)];
g = 9.80665;
D1 = diag([0.1, 0.1 0.1]);
D2 = diag([0.1, 0.1 0.1]);
phi = pi/6;
O_ba_0 = [1,0,0;
    0,   cos(phi),     sin(phi);
    0,   -sin(phi),   cos(phi)];
r_0 = [1,0,0]';
v_0 = [0,     cos(phi),     sin(phi)]';
O_k_all = zeros(3,3,2001);
O_k_all(:,:,1) = O_ba_0;
x_k_all = zeros(6,2001);
x_k_all(:,1) = [r_0;v_0];

for k = 1:1:2000
    w_k = [0,0,1]'+D1*normrnd(0,1,[3,1]);
    a_k = [-1-g*sin(phi)*sin(k*T);
                -g*sin(phi)*cos(k*T);
                -g*cos(phi)]+D2*normrnd(0,1,[3,1]);
    O_k1 = expm(-T*vec_to_mat(w_k)) *O_k_all(:,:,k);
    O_k_all(:,:,k+1)=O_k1;
    x_k = x_k_all(:,k);
    x_k1 = A*x_k+B*(O_k1'*a_k-[0,0,-g]');
    x_k_all(:,k+1)=x_k1;
end
figure(1)
hold on
plot3(rcwA_Ts_0_01(1,:),rcwA_Ts_0_01(2,:),rcwA_Ts_0_01(3,:));
plot3(x_k_all(1,:),x_k_all(2,:),x_k_all(3,:));
hold off
legend("reference trajectory","estimated trajectory");
function matrix = vec_to_mat(w)
wx = w(1);
```

```
wy = w(2);
wz = w(3);
matrix = [0,-wz,wy;wz,0,-wx;-wy,wx,0];
end
```

And the picture look like below:



P3: trajectory with only gyro and accelerometer measurements

For 2(b) we use following codes.

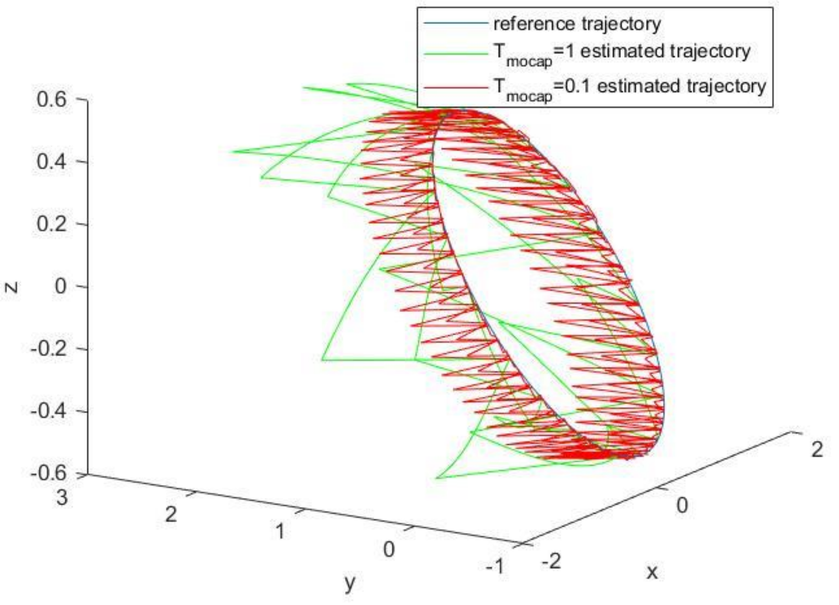
```
clc;
clear all;
load("rcwA.mat");

T = 0.01;
A = [eye(3), T*eye(3);
            zeros(3,3), eye(3)];
B = [T^2/2*eye(3);
            T*eye(3)];
g = 9.80665;
D1 = diag([0.1, 0.1 0.1]);
D2 = diag([0.1, 0.1 0.1]);
D3 = diag([0.005, 0.005, 0.005]);
phi = pi/6;
O_ba_0 = [1,0,0;
```

```
        0,   cos(phi),     sin(phi);
        0,   -sin(phi),   cos(phi)];
r_0 = [1,0,0]';
v_0 = [0,     cos(phi),     sin(phi)]';
O_k_all = zeros(3,3,2001);
O_k_all(:,:,1) = O_ba_0;
R = 0.001*eye(3);
Q = 10*eye(6);
P_all = zeros(6,6,2001);
P_all(:,:,1) = 10*eye(6);
x_k_all = zeros(6,2001);
x_k_all(:,1) = [r_0;v_0];
C = [eye(3), zeros(3,3)];
Tmocap = 1;

y_k1 = rcwA_Ts_0_01(:,1);
for k = 1:1:2000
    if(mod((k+1)*T,Tmocap)==0)
        Ck1=C;
        y_k1 = rcwA_Ts_0_01(:,k+1)+D3*normrnd(0,1,[3,1]);
    else
        Ck1 = zeros(3,6);
    end
    w_k = [0,0,1]'+D1*normrnd(0,1,[3,1]);
    a_k = [-1-g*sin(phi)*sin(k*T);
              -g*sin(phi)*cos(k*T);
              -g*cos(phi)]+D2*normrnd(0,1,[3,1]);
    O_k1 = expm(-T*vec_to_mat(w_k)) *O_k_all(:,:,k);
    O_k_all(:,:,k+1)=O_k1;
    x_k_k = x_k_all(:,k);
    x_k1_k = A*x_k_k+B*(O_k1'*a_k-[0,0,-g]');
    Pkk = P_all(:,:,k);
    P_k1_k = A*Pkk*A'+Q;
    Kk = P_k1_k*Ck1'*inv(Ck1*P_k1_k*Ck1'+R);
    P_k1_k1 = P_k1_k-Kk*Ck1*P_k1_k;
    x_k1_k1 = x_k1_k+Kk*(y_k1-Ck1*x_k1_k);
    x_k_all(:,k+1)=x_k1_k1;
end
x_t_1 = x_k_all;

Tmocap = 0.1;
y_k1 = rcwA_Ts_0_01(:,1);
for k = 1:1:2000
    if(mod((k+1)*T,Tmocap)==0)
```

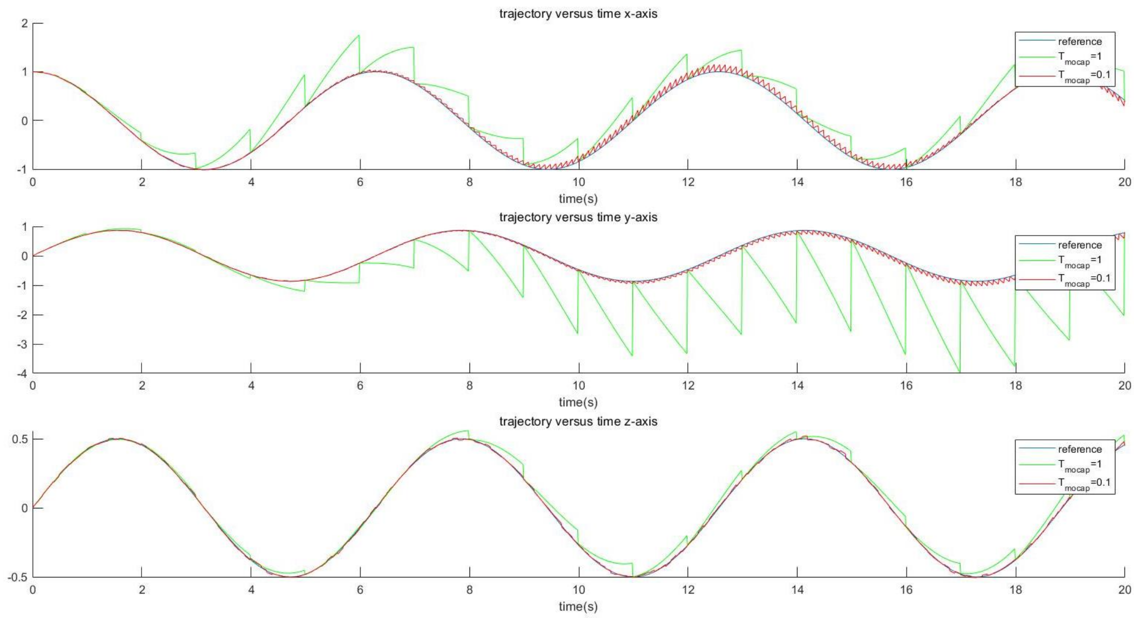
```
            Ck1=C;
            y_k1 = rcwA_Ts_0_01(:,k+1)+D3*normrnd(0,1,[3,1]);
        else
            Ck1 = zeros(3,6);
        end
        w_k = [0,0,1]'+D1*normrnd(0,1,[3,1]);
        a_k = [-1-g*sin(phi)*sin(k*T);
                    -g*sin(phi)*cos(k*T);
                    -g*cos(phi)]+D2*normrnd(0,1,[3,1]);
        O_k1 = expm(-T*vec_to_mat(w_k)) *O_k_all(:,:,k);
        O_k_all(:,:,k+1)=O_k1;
        x_k_k = x_k_all(:,k);
        x_k1_k = A*x_k_k+B*(O_k1'*a_k-[0,0,-g]');
        Pkk = P_all(:,:,k);
        P_k1_k = A*Pkk*A'+Q;
        Kk = P_k1_k*Ck1'*inv(Ck1*P_k1_k*Ck1'+R);
        P_k1_k1 = P_k1_k-Kk*Ck1*P_k1_k;
        x_k1_k1 = x_k1_k+Kk*(y_k1-Ck1*x_k1_k);
        x_k_all(:,k+1)=x_k1_k1;
    end

figure(1)
hold on
plot3(rcwA_Ts_0_01(1,:),rcwA_Ts_0_01(2,:),rcwA_Ts_0_01(3,:));
plot3(x_t_1(1,:),x_t_1(2,:),x_t_1(3,:),'g');
plot3(x_k_all(1,:),x_k_all(2,:),x_k_all(3,:),'r');
hold off
axis equal;
xlabel("x");
ylabel("y");
zlabel("z");
legend("reference trajectory","T_{mocap}=1 estimated trajectory","T_{mocap}=0.1 estimated
trajectory");


figure(2)
for i=1:1:3
    subplot(3,1,i)
    hold on
    plot(0:0.01:20,rcwA_Ts_0_01(i,:));
    plot(0:0.01:20,x_t_1(i,:),'g');
    plot(0:0.01:20,x_k_all(i,:),'r');
    hold off
    legend("reference","T_{mocap}=1","T_{mocap}=0.1");
```

```
    if(i==1)
        st = "trajectory versus time x-axis";
    elseif(i==2)
        st = "trajectory versus time y-axis";
    else
        st = "trajectory versus time z-axis";
    end
    title(st);
    xlabel("time(s)");
end

function matrix = vec_to_mat(w)
wx = w(1);
wy = w(2);
wz = w(3);
matrix = [0,-wz,wy;wz,0,-wx;-wy,wx,0];
end
```

And the pictures look like below:



P4: trajectories with measurements at different frequencies



P4: trajectories on different axis