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)

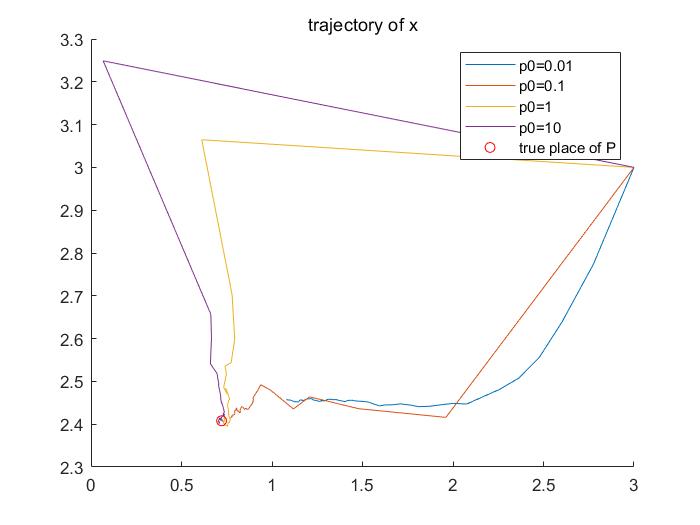
g\_x = [norm(xk-L1);

norm(xk-L2);

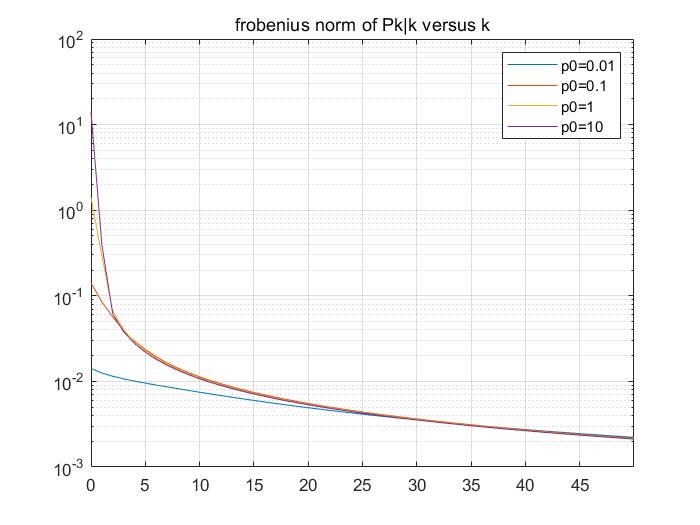
norm(xk-L3)];

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);

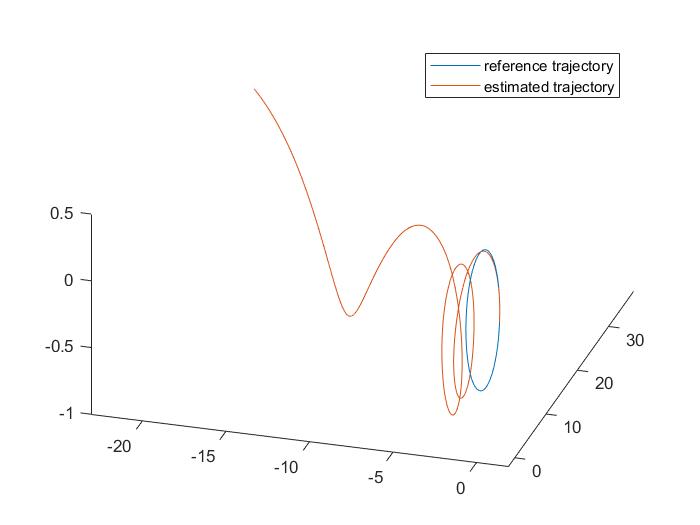
wy = w(2);

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);

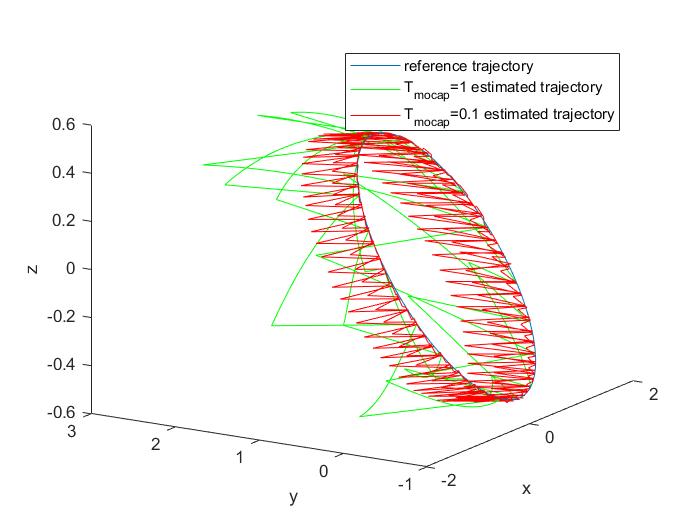
wy = w(2);

wz = w(3);

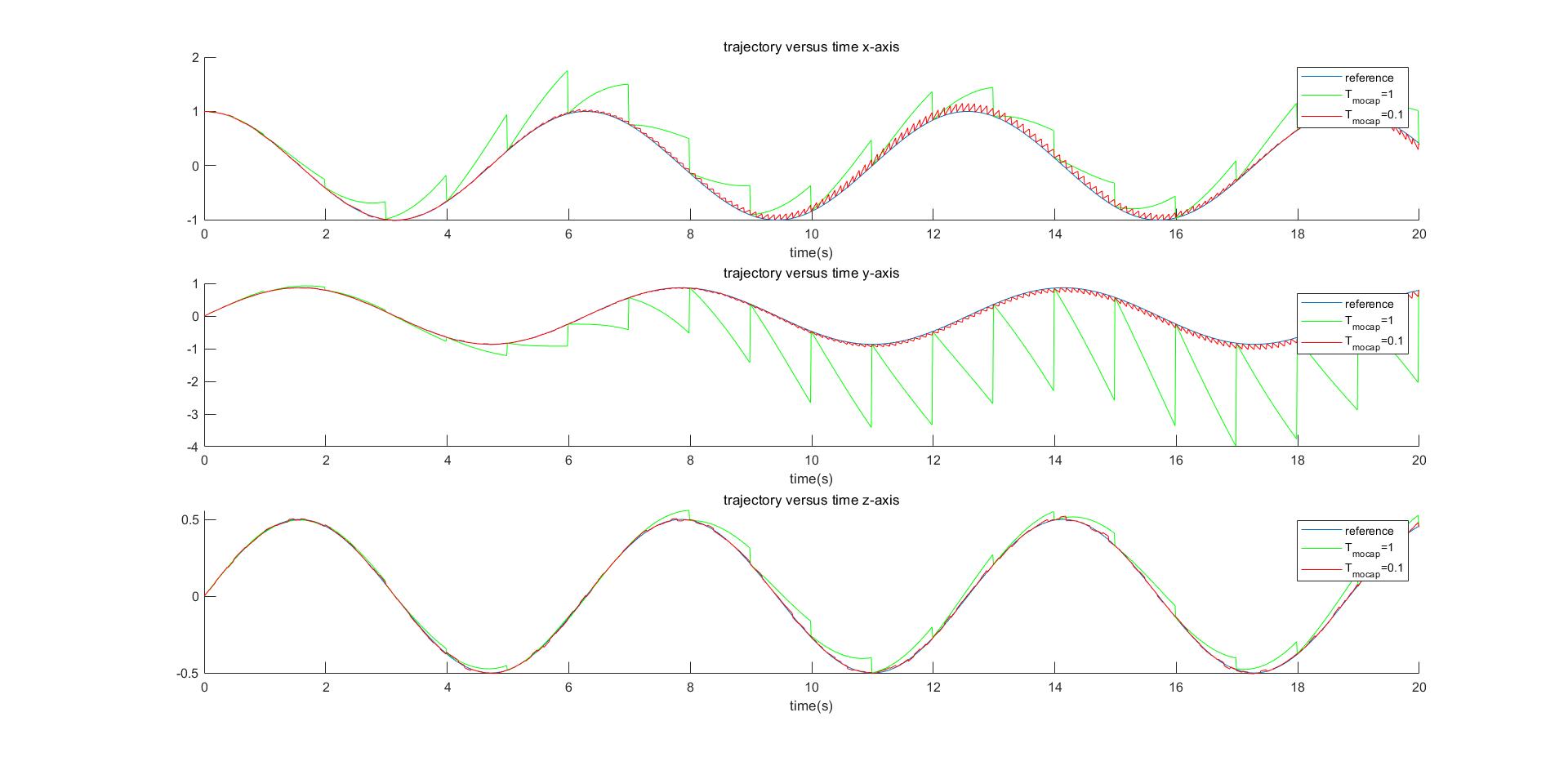
matrix = [0,-wz,wy;wz,0,-wx;-wy,wx,0];

end

And the pictures look like below:



P4: trajectories with measurements at different frequencies



P4: trajectories on different axis