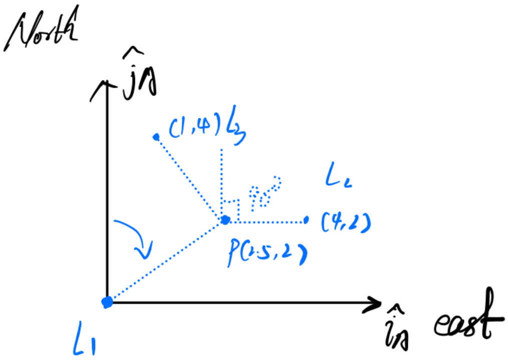



$$\theta_1 = \arctan\frac{2.5}{2} - \pi = -2.2455$$
$$\theta_2 = \frac{\pi}{2}$$
$$\theta_3 = -\arctan\frac{1.5}{2} = -0.6345$$

## For question1 we use following codes.

```
clc;

clear all;

theta1 = atan(2.5/2)-pi;

theta2 = pi/2;

theta3 =-atan(1.5/2);

theta_all = [theta1;theta2;theta3];

L1 = [0,0];

L2 = [4,2];

L3 = [1,4];

pos1 = get_pos(theta1,theta2,L1,L2);

pos2 = get_pos(theta2,theta3,L2,L3);

pos3 = get_pos(theta1,theta3,L1,L3);

Px = (pos1(1)+pos2(1)+pos3(1))/3;

Py = (pos1(2)+pos2(2)+pos3(2))/3;

p = [Px,Py];
```

```
len_each = 100000;
percentage = zeros(1,10);
dis = zeros(10,len_each);
for j = 1:1:10
    sigma = j/10;
    inside_times = 0;
    for i=1:1:len_each
        noised_theta = theta_all+normrnd(0,sigma,[3,1]);
        %get 3 potential pos
        pos12 = get_pos(noised_theta(1),noised_theta(2),L1,L2);
        pos23 = get_pos(noised_theta(2),noised_theta(3),L2,L3);
        pos13 = get_pos(noised_theta(1),noised_theta(3),L1,L3);
        c_x = (pos12(1)+pos23(1)+pos13(1))/3;
        c_y =   (pos12(2)+pos23(2)+pos13(2))/3;
        c_hat = [c_x,c_y];
        S_all = cal_tri_area(pos12,pos23,pos13);
        S1  = cal_tri_area(p,pos23,pos13);
        S2  = cal_tri_area(pos12,p,pos13);
        S3  = cal_tri_area(pos12,pos23,p);
        if(abs(S_all-(S1+S2+S3))<10e-6)
            inside_times = inside_times+1;
        end
```

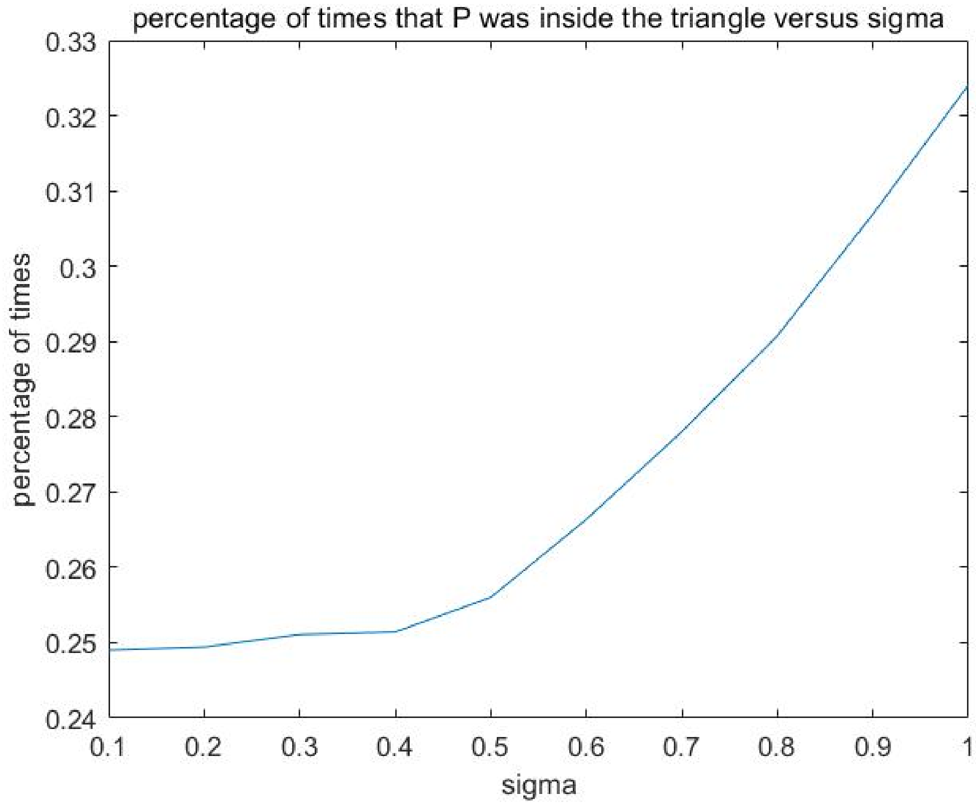
```
            dis(j,i) = norm(c_hat-p);
        end
        percentage(j) = inside_times/len_each;
end
mean_dis = mean(dis,2);
figure(1)
plot(0.1:0.1:1,percentage);
xlabel("sigma");
ylabel("percentage of times");
title("percentage of times that P was inside the triangle versus sigma");

figure(2)
plot(0.1:0.1:1,mean_dis);
xlabel("sigma");
ylabel("average dis from p to chat");
title("average distance of P to chat versus sigma");
```

## The code includes 2 functions "get_pos", "cal_tri_area"

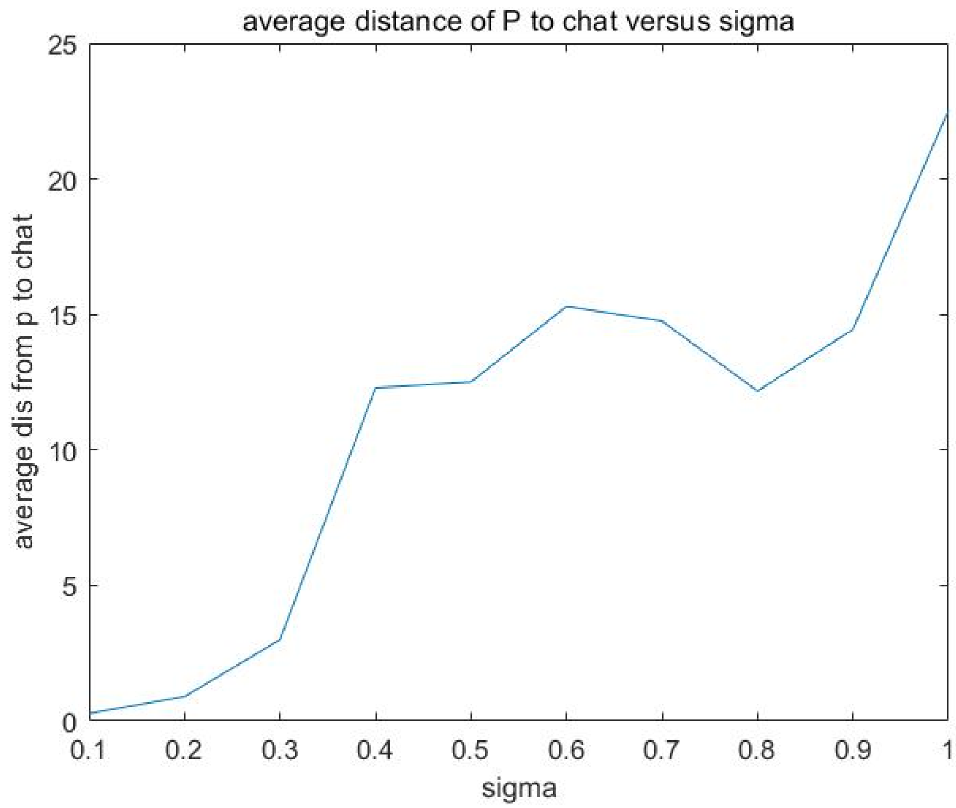
```
function pos = get_pos(theta1,theta2,L1,L2)
%from L1 and L2 to get the pos of P
x1 = L1(1);
y1 = L1(2);
```

```
x2 = L2(1);
y2 = L2(2);
T1 = tan(pi/2-theta1);
T2 = tan(pi/2-theta2);
M1 = [T1,-1;
          T2,-1];
V2 = [T1*x1-y1;
        T2*x2-y2];
pos = inv(M1)*V2;
end
function area = cal_tri_area(A1,A2,A3)
x1 = A1(1);
x2 = A2(1);
x3 = A3(1);

y1 = A1(2);
y2 = A2(2);
y3 = A3(2);
area = abs( (x1*(y2-y3)+x2*(y3-y1)+x3*(y1-y2))/2   );
end
```

## And the picture look like below:




P1: percentage that P was inside the triangle versus sigma



P2: average distance of P to c^ versus sigma

From the picture, we can observe that in general, the average distance from P to c^ is increasing and the percentage of times that P was inside the triangle is increasing with the increase of sigma.

Explanation: sigma represents the deviation from the true angle value. A large deviation means $\hat{P}_{\{1,2\}}, \hat{P}_{\{1,3\}}, \hat{P}_{\{2,3\}}$ are far from where they should be. Then the triangle formed by these 3 points will become bigger and leads to the increasing possibility that P is inside the triangle. At the same time, as the triangle becomes bigger, the centroid of the triangle will move away from the true position P gradually, and leads to the increasing average distance.

For question2 we use the following code:

```
clc;
clear all;
load('AE584_Midterm_P2.mat');
L1 = [1.52,0,0]';
L2 = [0,0,0]';
star1 = [0,1,0]';
star2 = [0,0,1]';
```

```
P0 = [0.52 0 -1]';

P_all = zeros(3,length(bearingL2St1));

P_all(:,1) = P0;

time = 0:1:length(bearingL2St1);

for i=1:length(bearingL2St1)

    theta = subAngL1L2(i);

    phi1 = bearingL2St1(i);

    phi2 = bearingL2St2(i);

    P0 = [0.52 0 -1]';

    fun = @(x) two_star_one_angle(L1,L2,x,theta,phi1,star1,phi2,star2);

    x0 = P_all(:,i);

    options = optimoptions('fminunc','OptimalityTolerance',10e-16);

    [x_ans,fval] = fminunc(fun,x0,options);

    P_all(:,i+1)=x_ans;

end

figure(1)

subplot(3,1,1)

plot(time,P_all(1,:));

title("position of spacecraft versus time on x-axis");

xlabel("time");

ylabel("position(AU)");
```

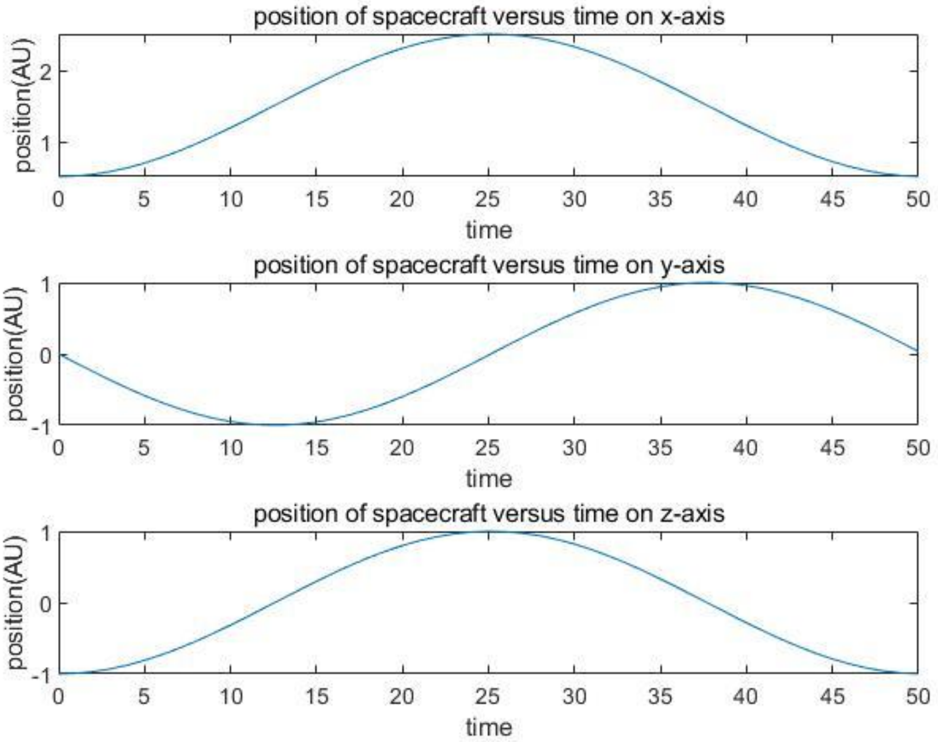
```
subplot(3,1,2)

plot(time,P_all(2,:));

title("position of spacecraft versus time on y-axis");

xlabel("time");

ylabel("position(AU)");


subplot(3,1,3)

plot(time,P_all(3,:));

title("position of spacecraft versus time on z-axis");

xlabel("time");

ylabel("position(AU)");


figure(2)

hold on

scatter3(P_all(1,:),P_all(2,:),P_all(3,:),'filled');

scatter3(L1(1),L1(2),L1(3),'filled','r');

scatter3(L2(1),L2(2),L2(3),'filled','y');

hold off

title("3D plot of stars and spacecraft trajectory");

legend("spacecraft trajectory","Mar","Sun");
```

## The code includes a function “two_star_one_angle”:

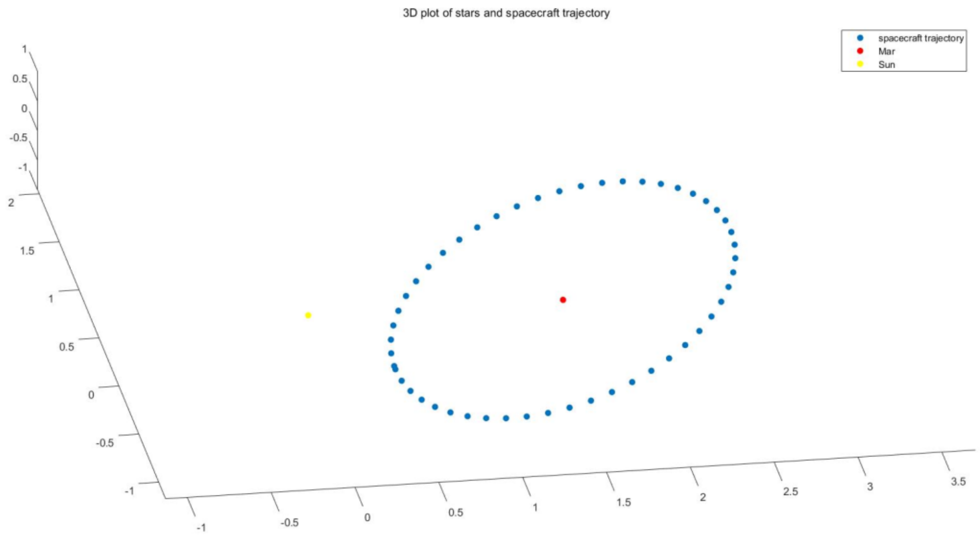
```
function fun = two_star_one_angle(L1,L2,x,theta,phi1,star1,phi2,star2)

eq1 = dot(x-L1,L2-L1)+norm(x-L1)...

    *norm(x-L2)*cos(theta)-norm(x-L1)^2;

eq2 = dot(L2-x,star1)-cos(phi1)*norm(L2-x);

eq3 = dot(L2-x,star2)-cos(phi2)*norm(L2-x);

fun = eq1^2+eq2^2+eq3^2;

if(norm(x-L1)<0.1 || norm(x-L2)<0.1)

    fun = fun+10001/(norm(x-L1)*norm(x-L2)+0.01);

end

end
```

For question2 we use the following code:

And the pictures are shown below:



P3: x y z position of the spacecraft versus step K




P4: 3D trajectory of the spacecraft in solar system

## For question3 we use the following code:

```
clc;

clear all;

time = 0:0.01:10;

phi_E = 0;

theta_E = pi/6;

psi_E = 0;

%[~,phi_e] = ode45(@(t,phi) sin(0.05*t), time, phi_E);

%[~,theta_e] = ode45(@(t,theta) 0.3*cos(0.01*t), time, theta_E);

%[~,psi_e] = ode45(@(t,psi) 0.5*sin(0.01*t), time, psi_E);

phi_e = 20 * (1-cos(0.05*time));

theta_e=   30*sin(0.01*time) + pi/6;
```

```
psi_e = 50*(1-cos(0.01*time));


O_EI = zeros(3,3,length(0:0.01:10));

for i = 1:1:length(0:0.01:10)

    O_EI(:,:,i) = zxz_angle2mat(phi_e(i),theta_e(i),psi_e(i));

end

%A = vec_to_mat(w0);   % Some arbitrary matrix we will use

F0 = eye(3);   % matrix initial value

odefun = @(t,y) deriv(t,y);   % Anonymous derivative function with A

tspan = 0:0.01:10;

f0 = reshape(F0,[1,9])';

[T,F] = ode45(odefun,tspan,f0);   % Pass in column vector initial value

F = reshape(F.',3,3,[]);   % Reshape the output as a sequence of 3x3 matrices

O_BI = F;

O_BE = zeros(3,3,length(0:0.01:10));

for i = 1:1:length(0:0.01:10)

   O_BE(:,:,i) = O_BI(:,:,i)*O_EI(:,:,i)';

end

figure(1)

hold on

for i=1:3

    for j=1:3
```

```
        subplot(3,3,3*(i-1)+j);
        plot(T,squeeze(O_BE(i,j,:)),'color','#D95319','LineWidth',2);
        %plot(T,o_indexs_2(j,:),'b');
        txt = [int2str(3*(i-1)+j),'th value of O_{BE}'];
        title(txt);
        xlabel('time(s)')
    end
end
hold off
O1_solutions = zeros(6,length(O_BE(1,1,:)));
for i=1:length(O_BE)
    [O1_solutions(1:3,i),O1_solutions(4:6,i)] = cal_Eular(O_BE(:,:,i));
end
T = 0:0.01:10;
figure(2)
subplot(3,1,1);
hold on
plot(T,O1_solutions(1,:),'color','#D95319','LineWidth',2);
title('phi versus time');
xlabel('time(s)');
hold off
subplot(3,1,2);
```

```
hold on
plot(T,O1_solutions(2,:),'color','#D95319','LineWidth',2);
title('theta versus time');
xlabel('time(s)')
hold off
subplot(3,1,3);
hold on
plot(T,O1_solutions(3,:),'color','#D95319','LineWidth',2);
title('psi versus time');
xlabel('time(s)')
hold off

function dy = deriv(t,y)
A = vec_to_mat([cos(2*t),cos(2*t),0.025*t]);
F = reshape(y,size(A));   % Reshape input y into matrix
FA = -A*F;   % Do the matrix multiply
dy = reshape(FA,[1,9])';   % Reshape output as a column vector
end
```

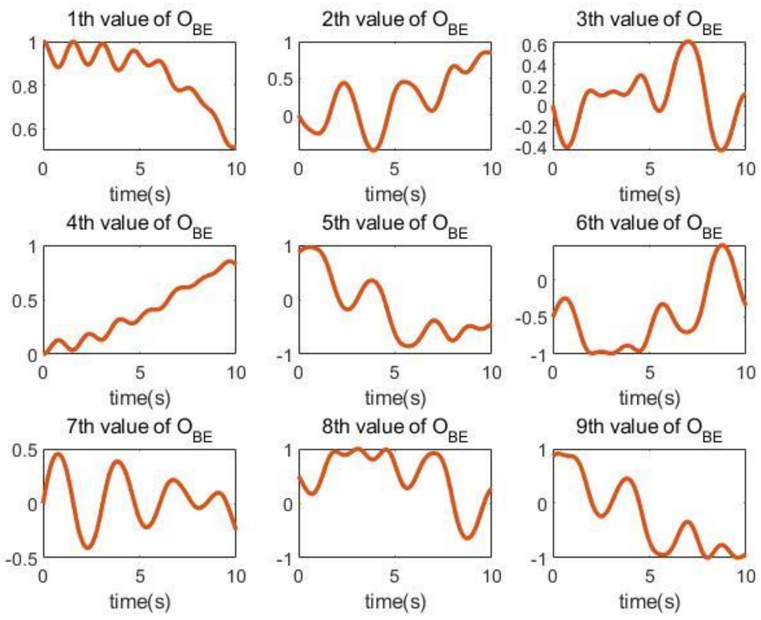
## The code includes 3 functions “zxz_angle2mat”, “cal_Eular”, “vec_to_mat”

```
function O_matrix = zxz_angle2mat(phi,theta,psi)
```

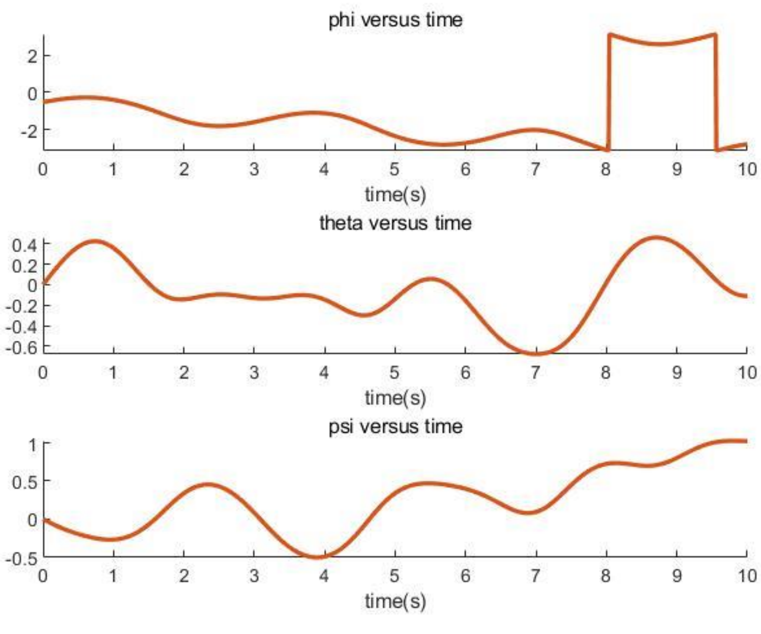
```
r1 = [cos(phi), sin(phi), 0;
      -sin(phi),  cos(phi),0;
      0,           0,          1];
r2 = [1,     0,   0;
      0, cos(theta),      sin(theta);
      0, -sin(theta),       cos(theta)];
r3 =   [cos(psi), sin(psi), 0;
      -sin(psi),  cos(psi),0;
      0,           0,          1];
O_matrix = r3*r2*r1;
end


function [solution1, solution2] = cal_Eular(o_matrix)
%get orientation_matrix in, Eular angles out
    theta_1 = -asin(o_matrix(1,3));
    theta_2 = pi-theta_1;
    if(theta_1<0)
        theta_2 = -pi-theta_1;
    end


    Psi_1 = atan2(o_matrix(1,2)/cos(theta_1),...
        o_matrix(1,1)/cos(theta_1));
```

```
    Psi_2 = atan2(o_matrix(1,2)/cos(theta_2),...
        o_matrix(1,1)/cos(theta_2));

    Phi_1 = atan2(o_matrix(2,3)/cos(theta_1),...
        o_matrix(3,3)/cos(theta_1));
    Phi_2 = atan2(o_matrix(2,3)/cos(theta_2),...
        o_matrix(3,3)/cos(theta_2));

    solution1 = [Phi_1,theta_1,Psi_1];
    solution2 = [Phi_2,theta_2,Psi_2];

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 are shown below:




P5: all components of $O_{\{B|E\}}$ versus time




P6: $\phi, \Theta, \Psi$ versus time

For question4 we use the following code:

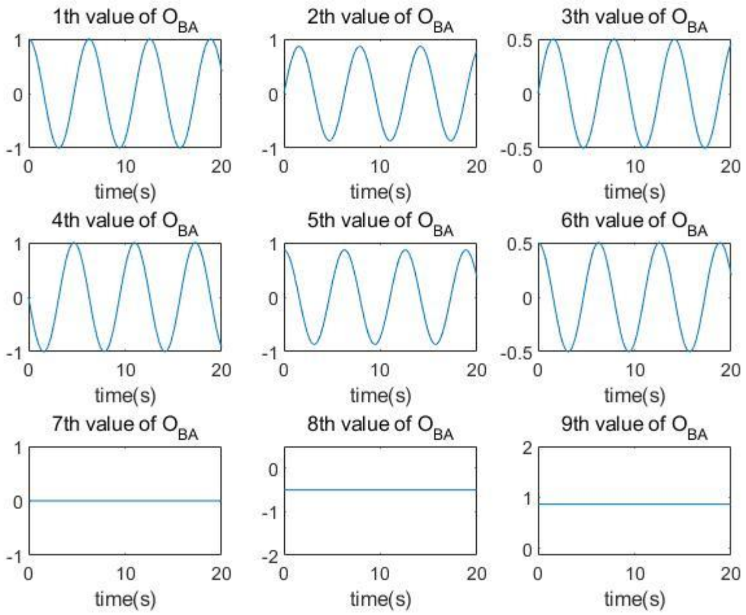
```
clc;

clear all;

time = 0:0.01:20;

phi = pi/6;

g = 9.80665;


O_BA0 = [1,                0,              0;

                    0,    cos(phi),     sin(phi);

                    0,    -sin(phi),    cos(phi)];    % matrix initial value

v_0 = [0;cos(phi);sin(phi)];

r_0 = [1;0;0];

w_BA = [0;0;1];

opts = odeset('RelTol',1e-5,'AbsTol',1e-16);

odefun = @(t,y) deriv(t,y);    % Anonymous derivative function with A

tspan = time;

f0 = reshape(O_BA0,[1,9])';

f0 = [f0;r_0;v_0];

[T,F] = ode45(odefun,tspan,f0,opts);    % Pass in column vector initial value

%T = F';

O_BA = F(:,1:9);

O_BA = reshape(O_BA.',3,3,[]);    % Reshape the output as a sequence of 3x3

matrices
```

```
r_cw = F(:,10:12);


figure(1)

hold on

for i=1:3

    for j=1:3

        subplot(3,3,3*(i-1)+j);

        plot(T,squeeze(O_BA(i,j,:)));

        %plot(T,o_indexs_2(j,:),'b');

        txt = [int2str(3*(i-1)+j),'th value of O_{BA}'];

        title(txt);

        xlabel('time(s)')

    end

end

hold off


figure(2)

hold on

for i=1:3

    subplot(3,1,i);

    plot(T,r_cw(:,i));

    txt = [int2str(i),'th component of r'];
```

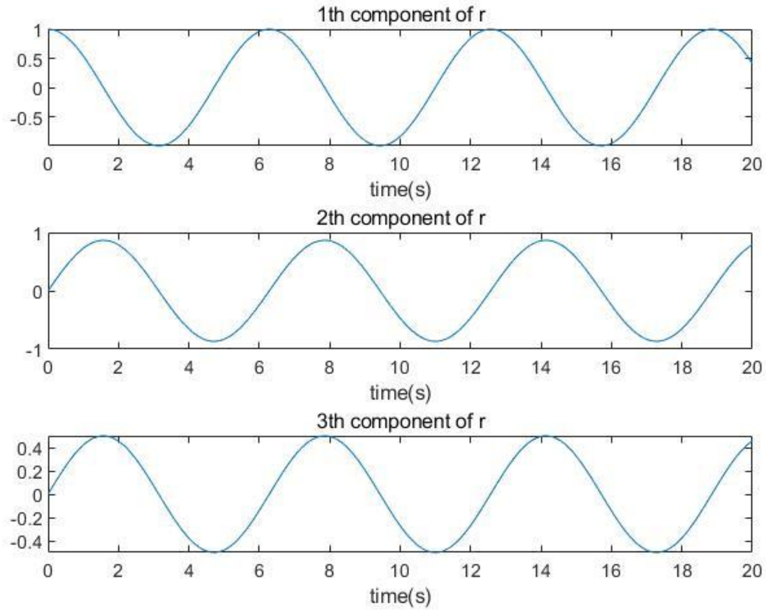
```
    title(txt);

    xlabel('time(s)')

end

hold off


figure(3)

plot3(r_cw(:,1),r_cw(:,2),r_cw(:,3));

title("3D trajectory of the center");

grid on;

axis equal;

xlabel("x")

zlabel("z")

ylabel("y")


function dy = deriv(t,y)

w_x = vec_to_mat([0;0;1]);

O_BA = reshape(y(1:9),size(w_x));   % Reshape input y into matrix

O_BA_diff = -w_x*O_BA;   % Do the matrix multiply

dy1 = reshape(O_BA_diff,[1,9])';   % Reshape output as a column vector

dr = y(13:15);% same as velocity


g = 9.80665;
```

```
phi = pi/6;

a_mean = [-1-g*sin(phi)*sin(t);

                    -g*sin(phi)*cos(t);

                    -g*cos(phi)];

dv = O_BA'*a_mean-[0;0;-g]; %acceleration

dy = [dy1;dr;dv];

end
```

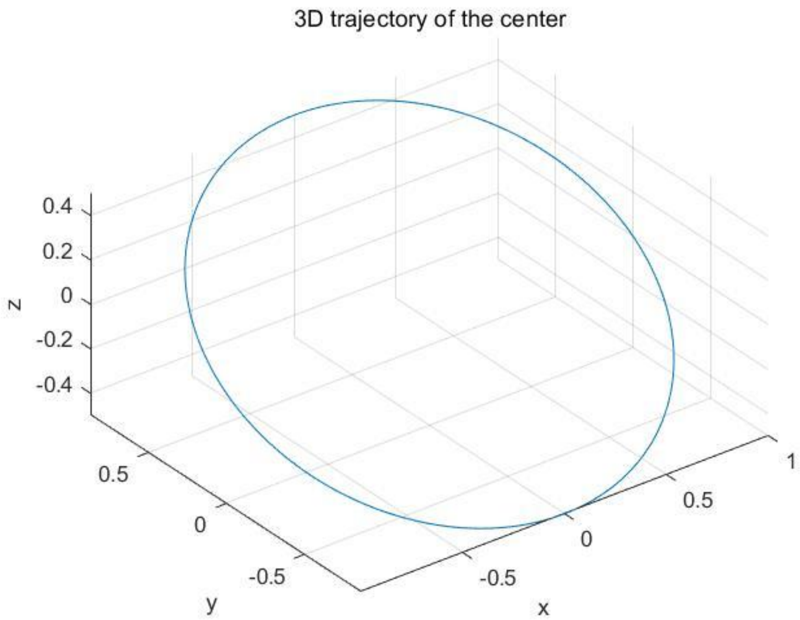
And the pictures are shown below:




P7: all components of $O_{B/A}$ versus time




P8: 3 components of $r_{c/w}|_A$ versus time




P9: 3D trajectory of the center of mass of the quadcopter