For 7(a) we use following codes. And the code includes a fuction “vdp1”

clc;

clear all;

time = 0:0.01:10;

[t,y] = ode45(@vdp1,time,[0;0;0]);

%y(1) = phi, y(2) = theta, y(3) = psi

w = [cos(2\*t),cos(2\*t),0.025\*t];

figure(1)

title('w versus time')

hold on

plot(t,w(:,1),'b');

plot(t,w(:,2),'g');

plot(t,w(:,3),'r');

hold off

legend('wx','wy','wz');

xlabel('time')

ylabel('rotation velocity')

figure(2)

title('angles versus time')

hold on

plot(t,y(:,1),'b');

plot(t,y(:,2),'g');

plot(t,y(:,3),'r');

hold off

legend('phi','theta','psi');

xlabel('time')

ylabel('angles in radian')

O\_matrix\_1 = zeros(3,3,1001);

o\_indexs = zeros(9,length(t));

for i=1:length(t)

ang = y(i,:);

o\_matrix\_t = angle(ang(1),ang(2),ang(3))';

o\_matrix = o\_matrix\_t';

O\_matrix\_1(:,:,i) = o\_matrix;

for j=1:9

o\_indexs(j,i) = o\_matrix\_t(j);

end

end

figure(3)

hold on

for j=1:9

subplot(3,3,j);

plot(t,o\_indexs(j,:));

txt = [int2str(j),'th value of O-matrix'];

title(txt);

xlabel('time(s)')

end

hold off

save("O\_matrix.mat","o\_indexs","O\_matrix\_1");

Following is function vdp1:

function dydt = vdp1(t,y)

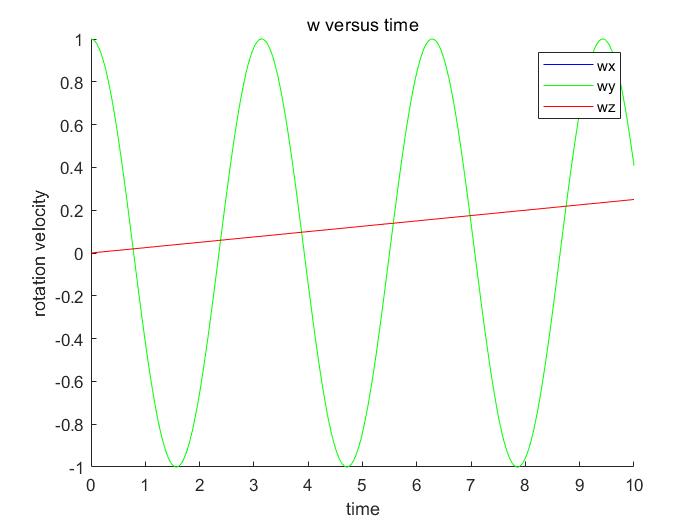
dydt = [cos(2\*t)\*(1+sin(y(1))\*tan(y(2)))+cos(y(1))\*tan(y(2))\*0.025\*t;...

cos(2\*t)\*cos(y(1))-sin(y(1))\*0.025\*t;...

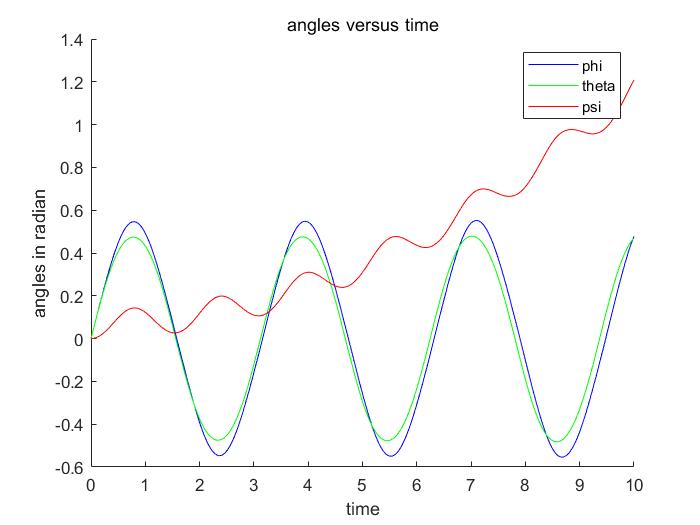
cos(2\*t)\*sin(y(1))\*sec(y(2))+0.025\*t\*cos(y(1))\*sec(y(2))];

end

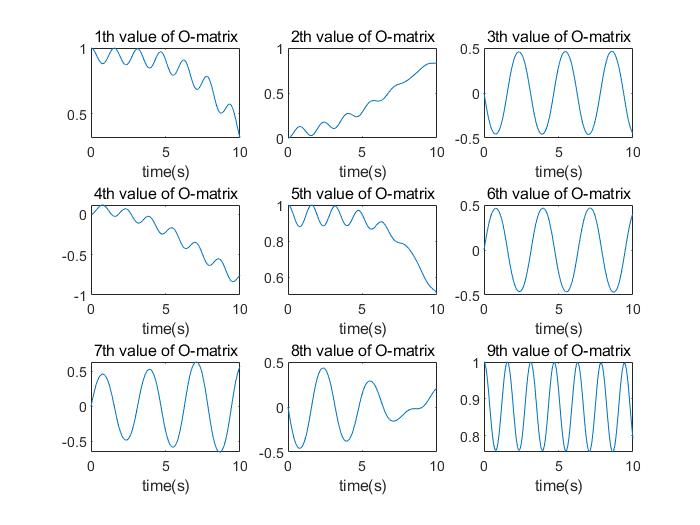
and the pictures look like below:



P1: w versus time



P2: angles versus time



P3: values in orientation matrix versus time

For 7(b) we use the following codes. And the code includes a fuction “vec\_to\_mat”, “angle”

clc;

clear all;

w0 = [1,1,0]';

%A = vec\_to\_mat(w0); % Some arbitrary matrix we will use

F0 = angle(0,0,0); % matrix initial value

F1=[1,2,3;4,5,6;7,8,9];

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

tspan = 0:0.01:10;

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

f1 = reshape(F1,[1,9])';

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

%T = F';

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

o\_indexs\_2 = zeros(9,length(T));

for i=1:length(T)

o\_matrix\_t\_2 = F(:,:,i)';

for j=1:9

o\_indexs\_2(j,i) = o\_matrix\_t\_2(j);

end

end

last\_method = load("O\_matrix.mat");

o\_indexs\_1 = last\_method.o\_indexs;

figure(1)

hold on

for j=1:9

subplot(3,3,j);

hold on

plot(T,o\_indexs\_2(j,:),'b','LineWidth',2);

plot(T,o\_indexs\_1(j,:),'color','#D95319','LineStyle','--','LineWidth',2);

hold off

legend('in b','in a');

%plot(T,o\_indexs\_2(j,:),'b');

txt = [int2str(j),'th value of O-matrix'];

title(txt);

xlabel('time(s)')

end

hold off

save("O\_matrix\_2.mat","F");

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

Following is function angle:

function o\_matrix = angle(a,b,c)

o\_matrix = [cos(b)\*cos(c), cos(b)\*sin(c), -sin(b);...

cos(c)\*sin(a)\*sin(b)-cos(a)\*sin(c), cos(a)\*cos(c)+sin(a)\*sin(b)\*sin(c), cos(b)\*sin(a);...

sin(a)\*sin(c)+cos(a)\*cos(c)\*sin(b), cos(a)\*sin(b)\*sin(c)-cos(c)\*sin(a), cos(a)\*cos(b)];

end

Following is function vec\_to\_mat：

function matrix = vec\_to\_mat(w)

wx = w(1);

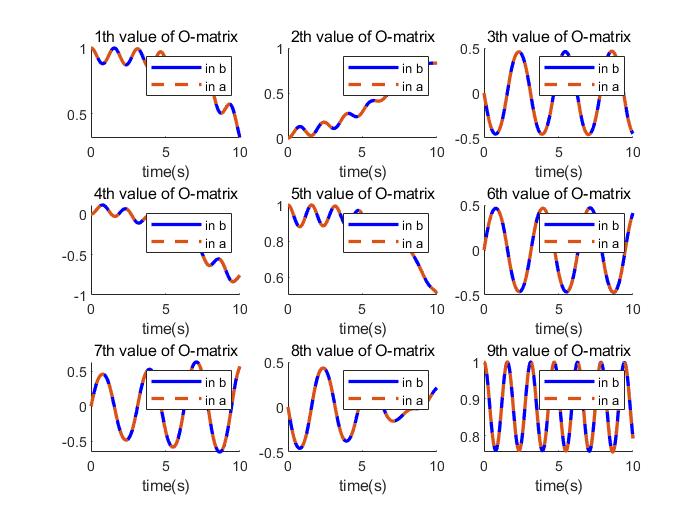
wy = w(2);

wz = w(3);

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

end

and the pictures look like below:



P4：values in 2 orientation matrix

The values concide

For 7(c) we use the following codes. And the code includes a fuction “cal\_Eular”

clc;

clear all;

%from a and b import Orientation matrix

O\_1 = load("O\_matrix.mat");

O\_2 = load("O\_matrix\_2.mat");

w\_1 = O\_1.y;

O\_matrix\_1 = O\_1.O\_matrix\_1;

O\_matrix\_2 = O\_2.F;

%then use cal\_Eular fuction to calculate the angles

O1\_solutions = zeros(6,length(O\_matrix\_1(1,1,:)));

O2\_solutions = zeros(6,length(O\_matrix\_2(1,1,:)));

for i=1:length(O\_matrix\_1)

[O1\_solutions(1:3,i),O1\_solutions(4:6,i)] = cal\_Eular(O\_matrix\_1(:,:,i));

[O2\_solutions(1:3,i),O2\_solutions(4:6,i)] = cal\_Eular(O\_matrix\_2(:,:,i));

end

T = 0:0.01:10;

figure(1)

subplot(1,3,1);

hold on

plot(T,w\_1(:,1),'LineWidth',3);

plot(T,O1\_solutions(1,:),'color','#D95319','LineStyle','--','LineWidth',2);

plot(T,O2\_solutions(1,:),'LineWidth',2,'color','y','LineStyle','-.');

title('phi versus time');

xlabel('time(s)');

legend('angle','angle from O matrix in a','angle from O matrix in b');

hold off

subplot(1,3,2);

hold on

plot(T,w\_1(:,2),'LineWidth',3);

plot(T,O1\_solutions(2,:),'color','#D95319','LineStyle','--','LineWidth',2);

plot(T,O2\_solutions(2,:),'LineWidth',2,'color','y','LineStyle','-.');

title('theta versus time');

xlabel('time(s)')

legend('angle','angle from O matrix in a','angle from O matrix in b');

hold off

subplot(1,3,3);

hold on

plot(T,w\_1(:,3),'LineWidth',3);

plot(T,O1\_solutions(3,:),'color','#D95319','LineStyle','--','LineWidth',2);

plot(T,O2\_solutions(3,:),'LineWidth',2,'color','y','LineStyle','-.');

title('psi versus time');

xlabel('time(s)')

legend('angle','angle from O matrix in a','angle from O matrix in b');

hold off

%----------------------------------------

figure(2)

subplot(1,3,1);

hold on

plot(T,w\_1(:,1),'LineWidth',2);

plot(T,O1\_solutions(4,:),'color','#D95319','LineStyle','--','LineWidth',2);

plot(T,O2\_solutions(4,:),'LineWidth',2,'color','y','LineStyle','-.');

title('phi versus time');

xlabel('time(s)');

legend('angle','angle from O matrix in a','angle from O matrix in b');

hold off

subplot(1,3,2);

hold on

plot(T,w\_1(:,2),'LineWidth',2);

plot(T,O1\_solutions(5,:),'color','#D95319','LineStyle','--','LineWidth',2);

plot(T,O2\_solutions(5,:),'LineWidth',2,'color','y','LineStyle','-.');

title('theta versus time');

xlabel('time(s)')

legend('angle','angle from O matrix in a','angle from O matrix in b');

hold off

subplot(1,3,3);

hold on

plot(T,w\_1(:,3),'LineWidth',2);

plot(T,O1\_solutions(6,:),'color','#D95319','LineStyle','--','LineWidth',2);

plot(T,O2\_solutions(6,:),'LineWidth',2,'color','y','LineStyle','-.');

title('psi versus time');

xlabel('time(s)')

legend('angle','angle from O matrix in a','angle from O matrix in b');

hold off

%then plot

Following is function cal\_Eular：

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),...

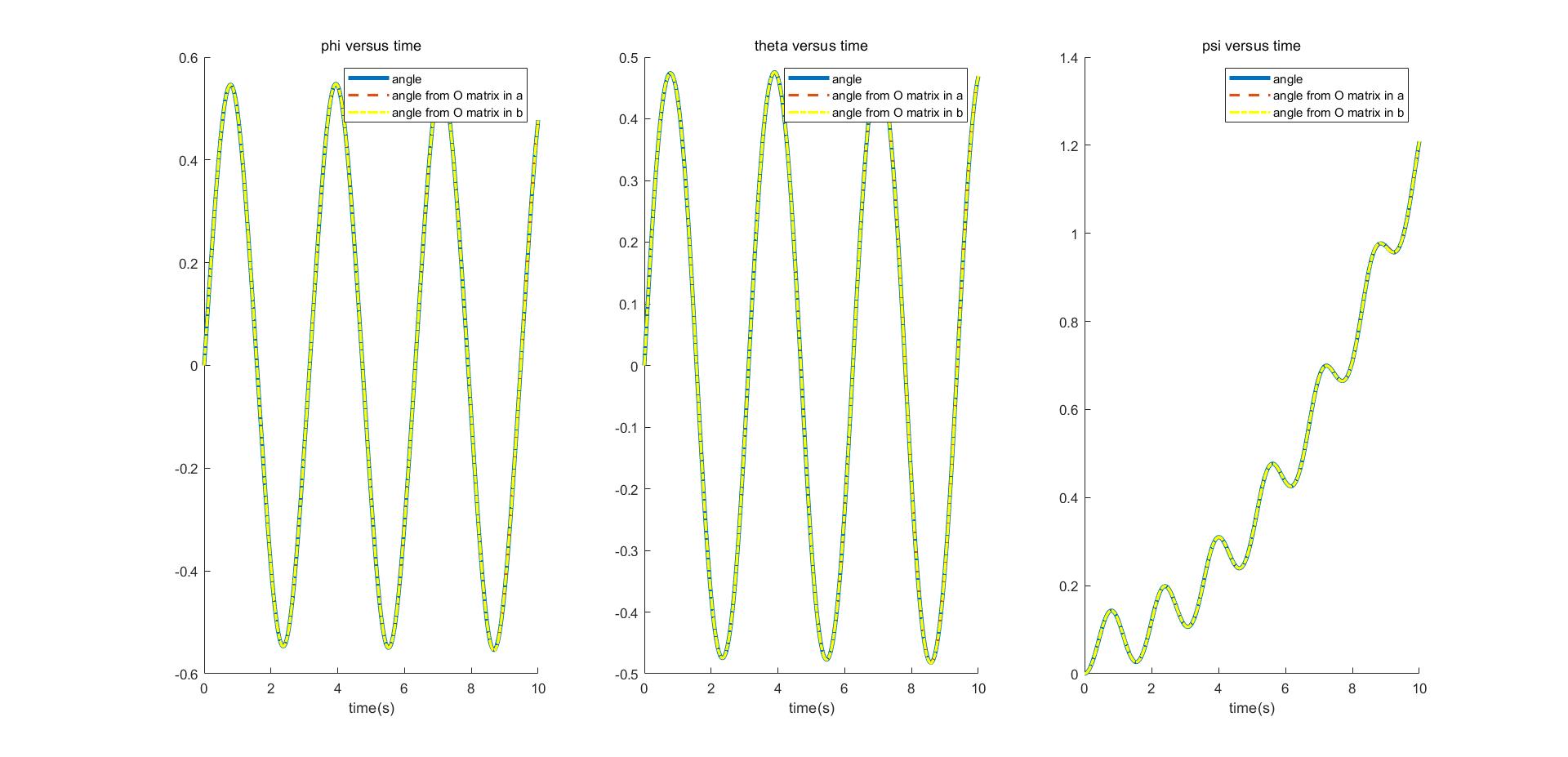
o\_matrix(3,3)/cos(theta\_2));

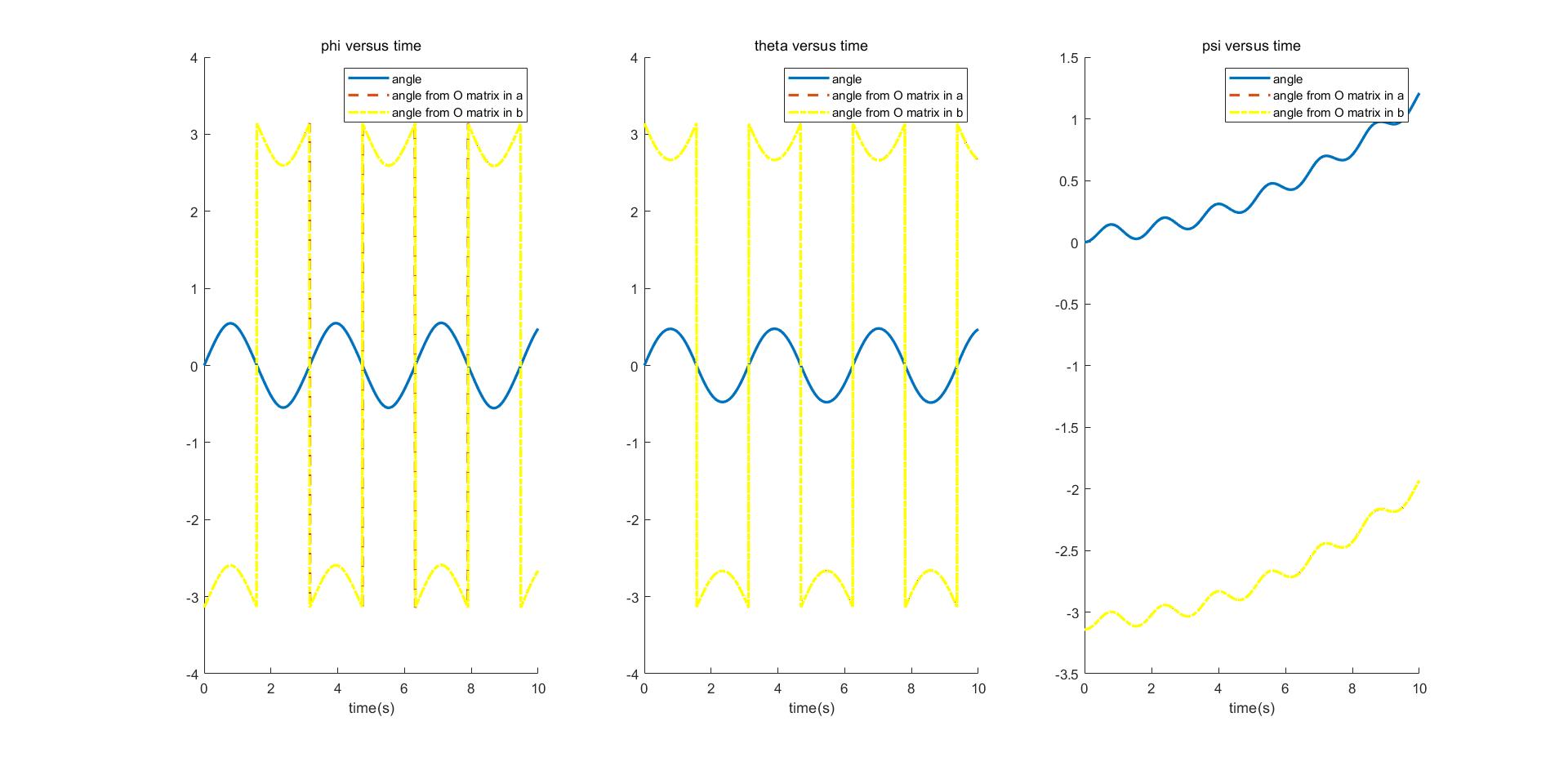
solution1 = [Phi\_1,theta\_1,Psi\_1];

solution2 = [Phi\_2,theta\_2,Psi\_2];

end

and the pictures look like below:





P5: Eular Angles gotten by different methods

Eular Angles have two different solutions, therefore, there are two pictures. However, these two solutions coincide in different methods and one of the solutions coincides with the Eular Angle we integraled in 7(a)