Main function for the simulation of (b) in Problem 6

```
clear;

% IC
format long;
THETA_0 = [0 15*pi/180]';

Csb_0 = eye(3,3);    % Initial DCM
w_ba_s_0  = [-0.002588190451025  0  0.009659258262891]';
w_ba_b_0 = Csb_0'*w_ba_s_0;

X_0 = [THETA_0;w_ba_s_0];

% time span
t = [0 2];

% ODE45
options = odeset('RelTol',1e-6);
[T,Y] = ode45(@HW5_2g_fun, t, X_0,options);

% -----------------------------------plot---------------------------------
figure(1);
subplot(3,2,1)
plot(T,rad2deg(Y(:,1)));
xlabel('Time [s]');ylabel('\theta [{\circ}]','Interpreter','Tex');
title('\theta versus time','Interpreter','Tex')
subplot(3,2,2)
plot(T,rad2deg(Y(:,2)));
xlabel('Time [s]');ylabel('\phi [{\circ}]','Interpreter','Tex');
title('\phi versus time','Interpreter','Tex')
subplot(3,2,3)
plot(T,Y(:,3));
xlabel('Time [s]');ylabel('\omega^{ba}_{b1} [{rad/s}]','Interpreter','Tex');
title('\omega^{ba}_{b1} versus time','Interpreter','Tex')
subplot(3,2,4)
plot(T,Y(:,4));
xlabel('Time [s]');ylabel('\omega^{ba}_{b2} [{rad/s}]','Interpreter','Tex');
title('\omega^{ba}_{b2} versus time','Interpreter','Tex')
subplot(3,2,5)
plot(T,Y(:,5));
xlabel('Time [s]');ylabel('\omega^{ba}_{b3} [{rad/s}]','Interpreter','Tex');
title('\omega^{ba}_{b3} versus time','Interpreter','Tex')

% --------------------Compute the total  energy---------------------------

SIZE = size(Y);
E = zeros(SIZE(1),1);
dE = zeros(SIZE(1),1);

for i=1:(SIZE(1))
[dx E_tmp] = HW5_2g_fun(T(i),Y(i,:));
%caculate dE versus time
E(i) = E_tmp;
```

```
if i ==1
dE(i) = 0;
else
dE(i) = (E(i)-E(i-1));
end
end

% plot energy stuffs
figure(2);
plot(T,E)
xlabel('Time [s]');ylabel('E [J]');title('total energy versus time');
ylim([.45833 .45834])
figure(3)
plot(T,dE);
xlabel('Time [s]');ylabel('\Delta E_{Bw/a} [J]','Interpreter','Tex');
title('Change of energy versus time');

save AE_540_HW5.mat;
```

Function for ODE 45 and computing total energy:

```
function [dx E] = HW5_2g2 (t,x)
% Function for the ODE45 and also used for computing total energy.
% Wei Ding
% 11/8/2014

% get in constants
constants;
eta = deta*t + eta_0;

theta = x(1);
phi   = x(2);

THETA_ba = [x(1) x(2)]';

% --------------------------Kinematics---------------------------------
%       1.DCM
Cqa = [cos(theta)  sin(theta)    0;
-sin(theta) cos(theta)    0
0           0    1];

Cbq = [cos(phi)     0   -sin(phi)
0     1           0
sin(phi)     0    cos(phi)];

Csb = [cos(eta)     sin(eta)    0;
-sin(eta)     cos(eta)    0
0     0           1];

Csa = Csb*Cbq*Cqa;
%       2.Angular Velocity
w_ba_b = [x(3) x(4) x(5)]';
w_ba_s = Csb*w_ba_b;
w_sb = [0 0 deta]';
```

```
w_sa = w_sb + w_ba_s;

%       S_ba
S_ba = Csb*[Cbq*[0 0 1]' [0 1 0]'];

%       3.Positon
r_cw_s = [0 0 l]';
r_cw_b = r_cw_s;

%       4.Velocity
%       5.Acceleration

% --------------------------------FBD---------------------------------
%       1.Forces
%       2.Moments

m_Bw_b = [0 sin(phi)*l*m*g 0]';
m_Bw_s = Csb*m_Bw_b;

% ---------------------------------E2L-------------------------------
%       1. Momentum
%       2. Angular Momentum
%       3. E2L
% ------------------->Get the DEs

% ----------------------------------DE-------------------------------
%          Inertia
J_MbW_b   = -m*CROSS(r_cw_b)*CROSS(r_cw_b);
J_MbW_s   = Csb*J_MbW_b*Csb';
J_BC_s    = [1/12*m*(3*(D/2)^2+h^2)                        0                   0;
0   1/12*m*(3*(D/2)^2+h^2)                  0;
0                         0      1/2*m*(D/2)^2];


%           DE of w_ba
dw_ba_s = inv(J_BC_s+J_MbW_s)*
(m_Bw_s - CROSS(w_sa)*J_BC_s*w_sa - CROSS(w_ba_s)*J_MbW_s*w_ba_s - J_MbW_s*CROSS(w_sb)*w_ba_s);
dw_ba_b = Csb'*(dw_ba_s+CROSS(w_sb)*w_ba_s); % From s frame to b frame
%           DE of theta_ba
dTHETA_ba = S_ba'*w_ba_s;
%           spits out time derivatives
dx = [dTHETA_ba;dw_ba_b];

%---------------------------Total Energy------------------------------
% Kinetic
T_Bw = 1/2*w_sa'*J_BC_s*w_sa+1/2*w_ba_b'*J_MbW_b*w_ba_b;
% Potential
g_a  = [0 0 -g]';
U_Bw = -r_cw_s'*Csa*g_a*m;
% Total
E = T_Bw+U_Bw;

end
```

```
function X_cross = CROSS( X )
% Cross of a Matrix

X_cross = zeros(3,3);
X_cross = [  0    -X(3)   X(2);
X(3)      0    -X(1);
-X(2)   X(1)    0
];

end
```

constants.m:

```
    g = 9.81;
    l = 5/100;
    sigma = 8050;
    D = 60/1000;
    h = 5/1000;
    m = pi*(D/2)^2*h*sigma;
    deta = 1200*(2*pi/60);
    eta_0 = 0;
```