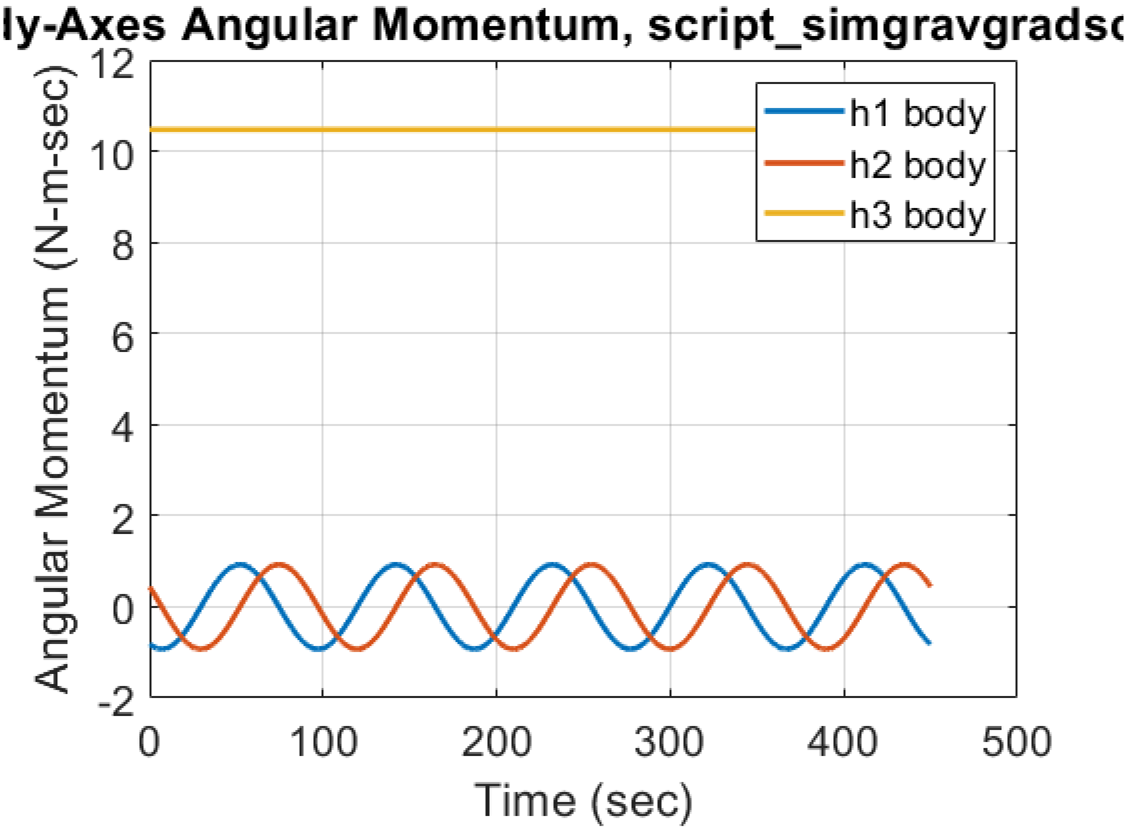
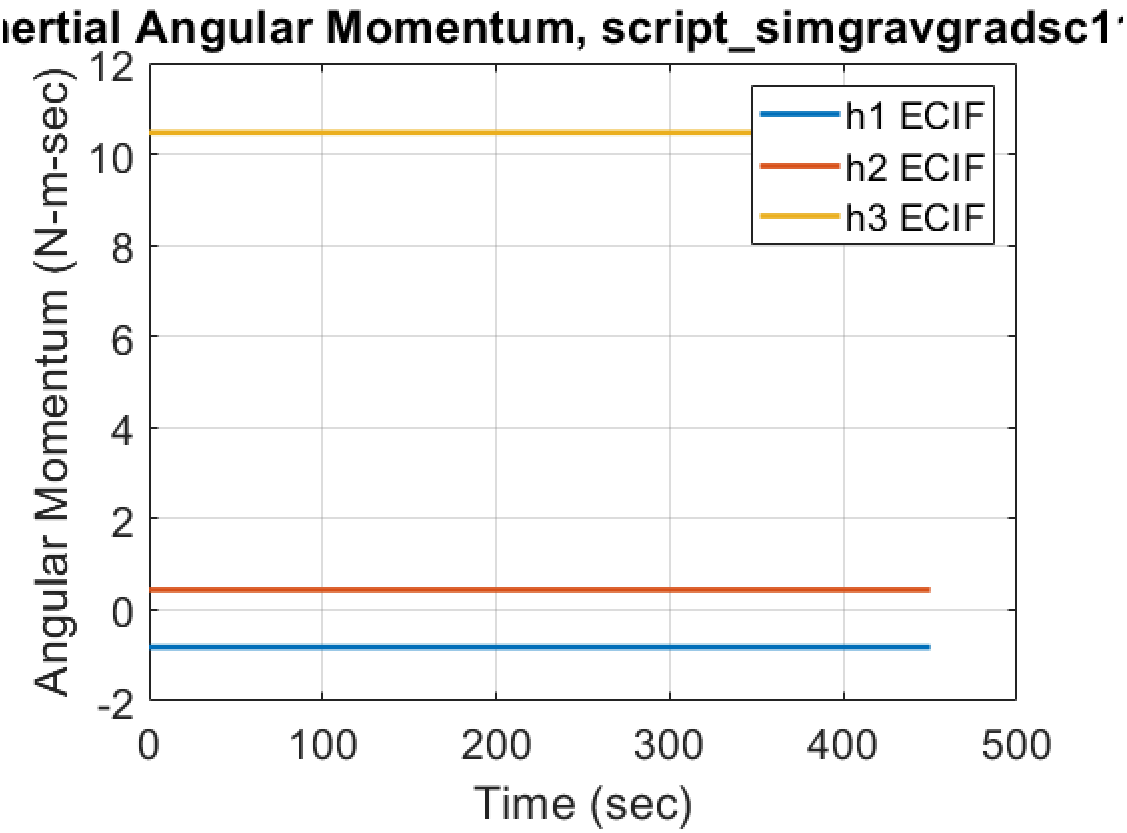
Problem 2

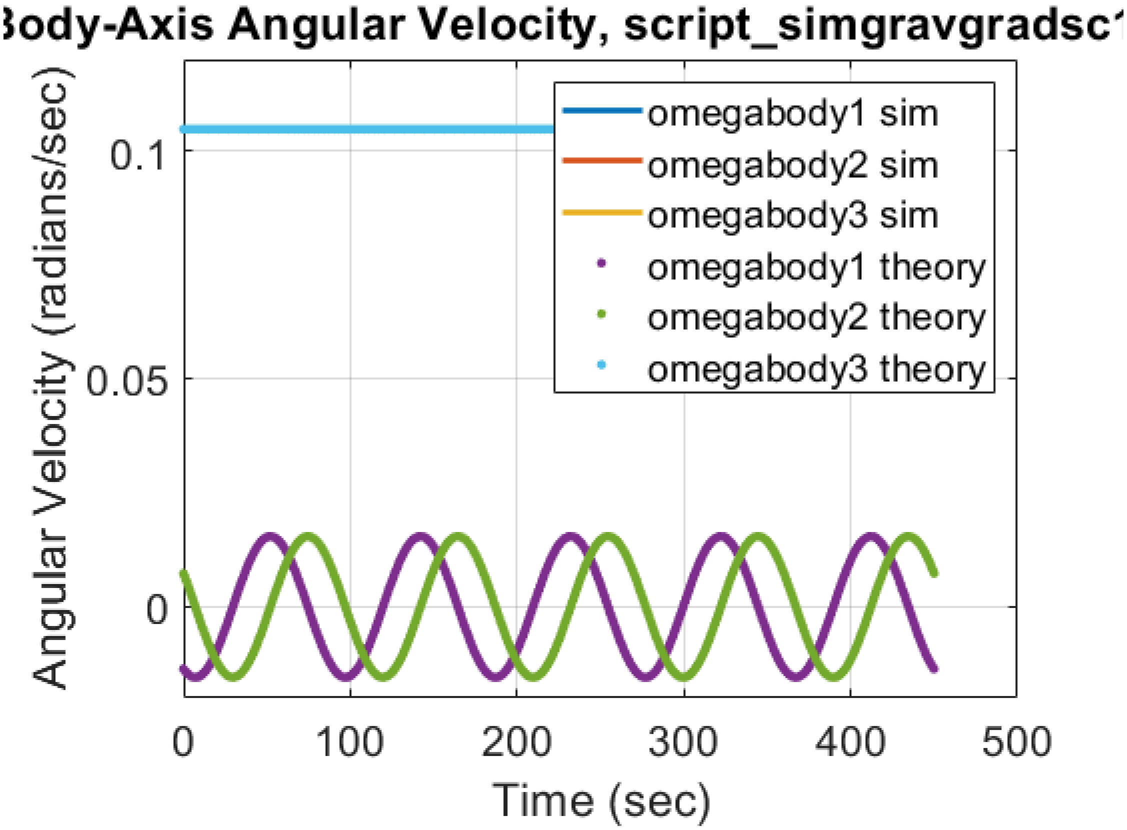
```
%script_simgravgradsc11.m
%
%  Copyright (c) 2019 Mark L. Psiaki.  All rights reserved.
%
%  This Matlab script simulates the torque-free motion of
%  an axi-symmetric spinning satellite.
%
%  This script makes a plot of the
%  angular momentum time history in body-fixed
%  coordinates and in inertial
%  coordinates.  It also makes plots
%  of the time histories of the 3 body-axis spin-
%  rate vector elements.
%
%  Clear the Matlab workspace.
%
   clear;clc;close all;
%
%  Set up the simulation parameters.
%
   Itr = 60;
   Ispin = 100;
   IMoIbody = diag([Itr;Itr;Ispin]);
   omegabody0 = [(-0.13*(2*pi/60));(0.07*(2*pi/60));(2*pi/60)];
   norbit = 0; % eliminate gravity-gradient torque and
               % rotation of the reference frame relative to which
               % x(1:4,1) defines the attitude quaternion so that
               % it becomes an inertial reference frame rather
               % than a non-inertial orbit-following local-level
               % reference frame.
   q0 = [0;0;0;1];
   x0 = [q0;omegabody0];
%
%  Define the aircraft dynamics function handle
%  in a form that is suitable for input to ode45.m.
%
   ffunctode45 = @(tdum,xdum) ...
             ffunctgravgradsc02(tdum,xdum,IMoIbody,norbit);
%
%  Define the time span of the simulation, computing outputs
%  every 0.5.  This time span should be large enough
%  to see several spins periods and several nutation periods.
%
   tspan = ((0:900)')*0.5;
%
%  Set up numerical integration options for ode45.m
%  in a way that uses a tighter relative tolerance than
%  is normally used.
%
   optionsode45 = odeset('RelTol',1.e-10);
```

```
%
%  Call ode45.m in order to perform numerical integration.
%
   tic
   [thist,xhist] = ode45(ffunctode45,tspan,x0,optionsode45);
   timetosim = toc;
%
%  Compute the angular momentum vector time history in
%  inertial coordinates.
%
   tic
   N = size(thist,1);
   hvecbodyhist = zeros(N,3);
   hvecinertialhist = zeros(N,3);
   for k = 1:N
      xk = xhist(k,:)';
      hvecbodyk = IMoIbody*xk(5:7,1);
      hvecbodyhist(k,:) = hvecbodyk';
      qk = xk(1:4,1);
      qknorm = qk*(1/sqrt(sum(qk.^2)));
      Rk = rotmatquaternion(qknorm);
      hvecinertialk = (Rk')*hvecbodyk;
      hvecinertialhist(k,:) = hvecinertialk';
   end
   timetohvecinertial = toc
   clear k xk hvecbodyk qk qknorm Rk hvecinertialk
%
%  Compute the nutation frequency.
%
   omeganut = abs(Ispin-Itr)*omegabody0(3)/Itr;

%
%  Compute the theoretical body-axis spin vector component
%  time histories that are valid for this axially-symmetric
%  spacecraft.
%
   signItrminusIspin = sign(Itr - Ispin);
   omegabody1hist = omegabody0(1)*cos(omeganut*thist)
 +omegabody0(2)*signItrminusIspin*sin(omeganut*thist);
   omegabody2hist = omegabody0(2)*cos(omeganut*thist) -
omegabody0(1)*signItrminusIspin*sin(omeganut*thist);
   omegabody3hist = ones(N,1)*omegabody0(3);
%
%  Plot the body-axes angular momentum time history.
%
   figure(1)
   hold off
   plot(thist,hvecbodyhist,'LineWidth',2)
   set(get(gcf,'CurrentAxes'),'FontSize',16)
   grid
   xlabel('Time (sec)')
   ylabel('Angular Momentum (N-m-sec)')
   title(['Body-Axes Angular Momentum,',...
          ' script\_simgravgradsc11.m'])
   legend('h1 body','h2 body','h3 body')
%
%  Plot the inertial angular momentum time history.
%
   figure(2)
   hold off
   plot(thist,hvecinertialhist,'LineWidth',2)
   set(get(gcf,'CurrentAxes'),'FontSize',16)
   grid
   xlabel('Time (sec)')
   ylabel('Angular Momentum (N-m-sec)')
   title(['Inertial Angular Momentum,',...
          ' script\_simgravgradsc11.m'])
   legend('h1 ECIF','h2 ECIF','h3 ECIF')
%
%  Plot the body-axis spin vector component time histories,
%  both from the numerical integration and the theoretical
%  values.
%
   figure(3)
   hold off
   plot(thist,xhist(:,5:7),'-','LineWidth',2)
   set(get(gcf,'CurrentAxes'),'FontSize',16)
   hold on
   plot(thist,[omegabody1hist,omegabody2hist,omegabody3hist],'.',...
        'MarkerSize',10)
   grid
   xlabel('Time (sec)')
   ylabel('Angular Velocity (radians/sec)')
   title(['Body-Axis Angular Velocity,',...
          ' script\_simgravgradsc11.m'])
   legend('omegabody1 sim','omegabody2 sim','omegabody3 sim',...
          'omegabody1 theory','omegabody2 theory','omegabody3 theory')
%
%  Save the results.
%
   textcommands = ['These data have been generated by the',...
                   ' commands in script_simgravgradsc11.m'];
   save simgravgradsc11

   format long
   xfinal = xhist(end,:)'
   qfinalmag = norm(xfinal(1:4,1))


timetohvecinertial =

   0.010625500000000


xfinal =

   0.076778204795434
  -0.041341864928188
  -0.984339512015986
   0.153186692608215
  -0.013613444068028
   0.007330204591270
   0.104719755119660


qfinalmag =

   0.999997240113923
```





ıertial Angular Momentum, script_simgravgradsc1
Angular Momentum (N-m-sec)
12
10
8
6
4
2
0
-2
0
100
200
300
400
500
Time (sec)
h1 ECIF
h2 ECIF
h3 ECIF



3ody-Axis Angular Velocity, script_simgravgradsc1
Angular Velocity (radians/sec)
0.1
0.05
0
0
100
200
300
400
500
Time (sec)
omegabody1 sim
omegabody2 sim
omegabody3 sim
omegabody1 theory
omegabody2 theory
omegabody3 theory

Published with MATLAB® R2019a