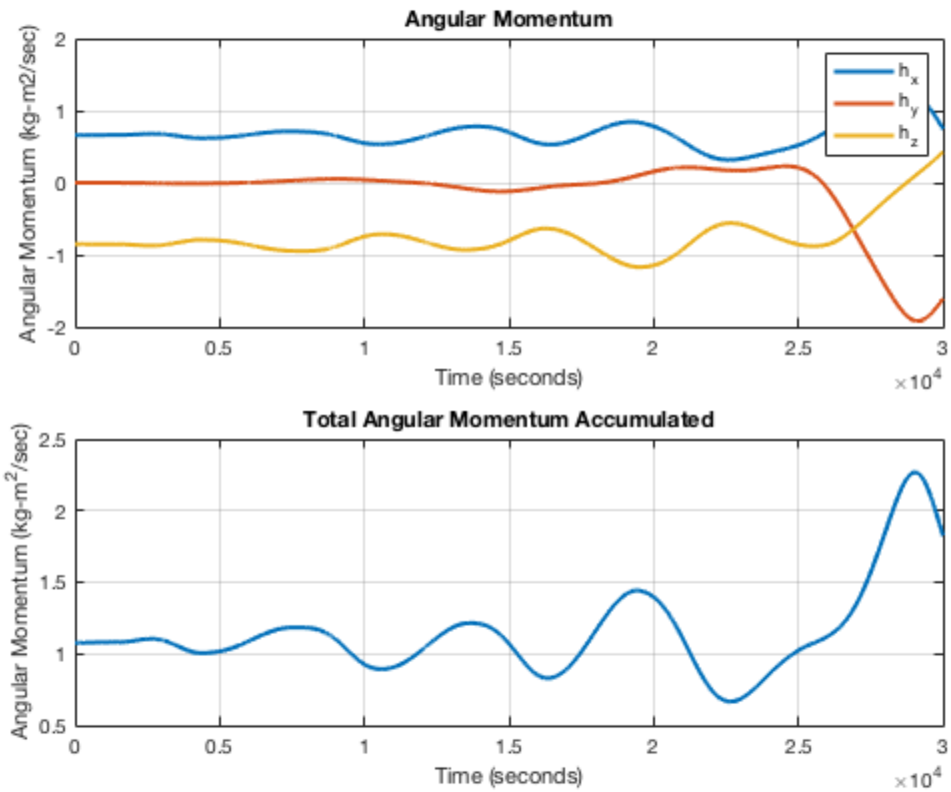
Table of Contents

orbital parameters

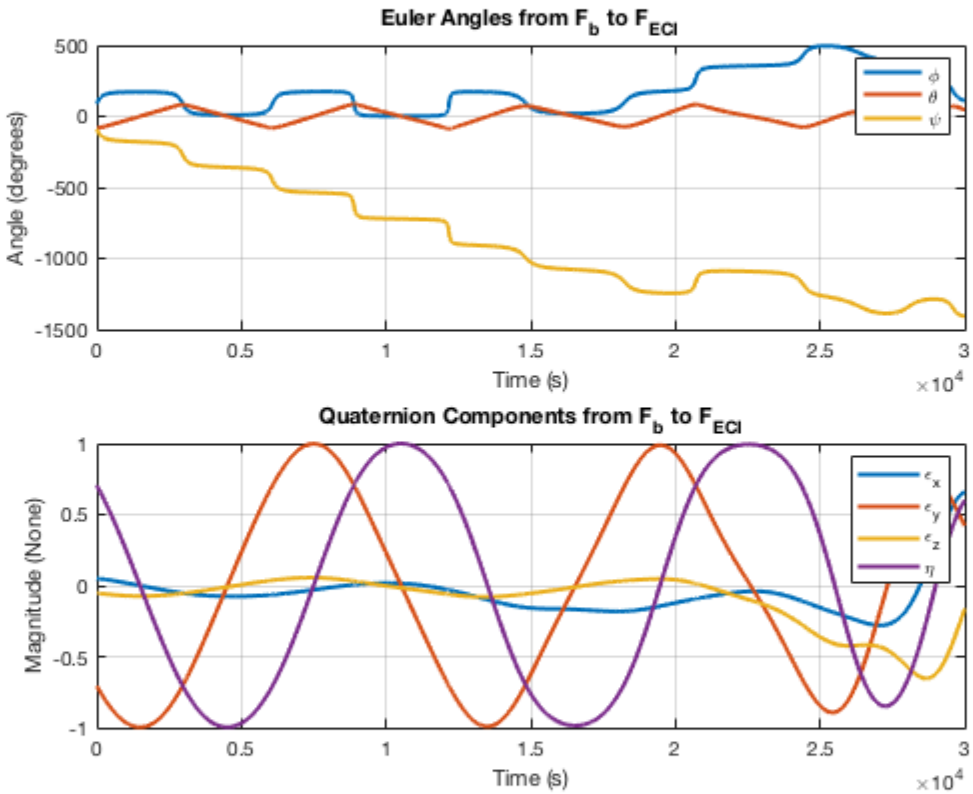
rigid body prop

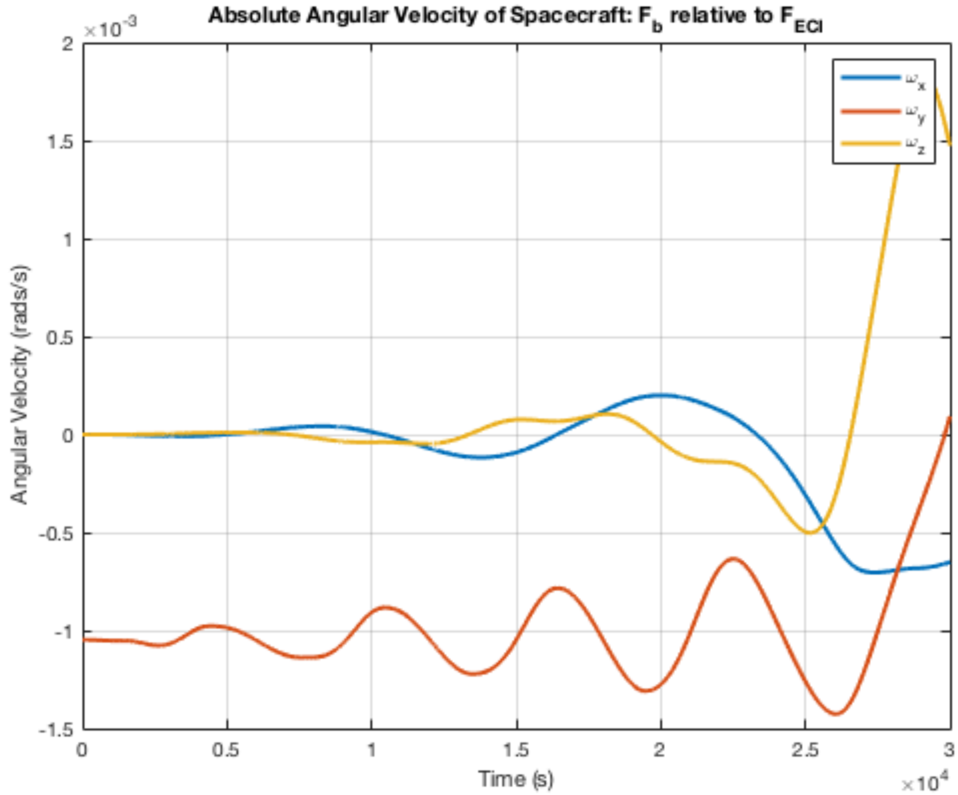
```
The inertial tensor is:

  813.0396         0         0
         0  546.3729         0
         0         0  628.7083

kg-m2

The mass of the spacecraft is 640 kg.

ODE Finished
```

Angular Momentum Accumulated



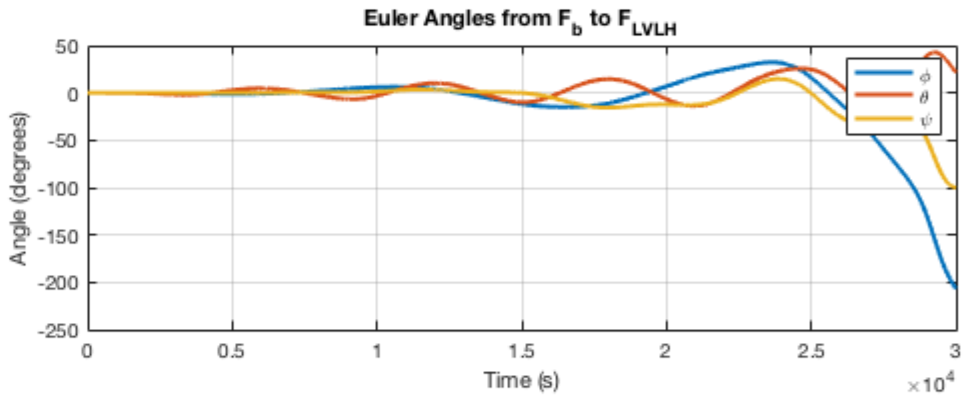
Body Relativeto ECI Plots

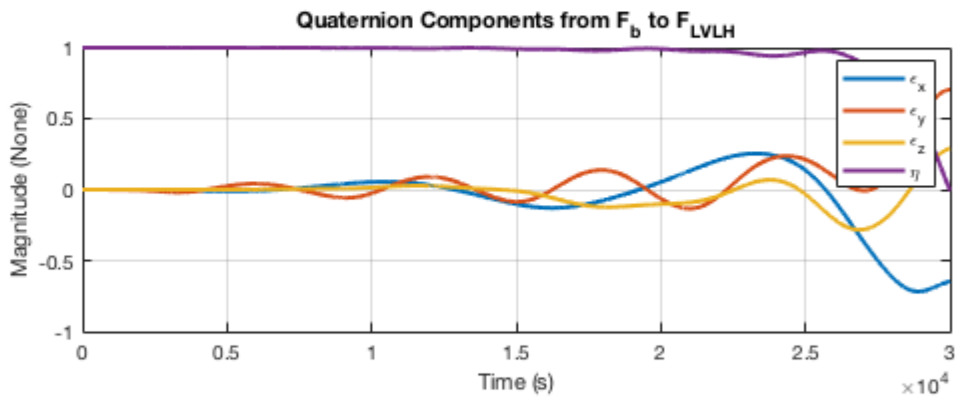


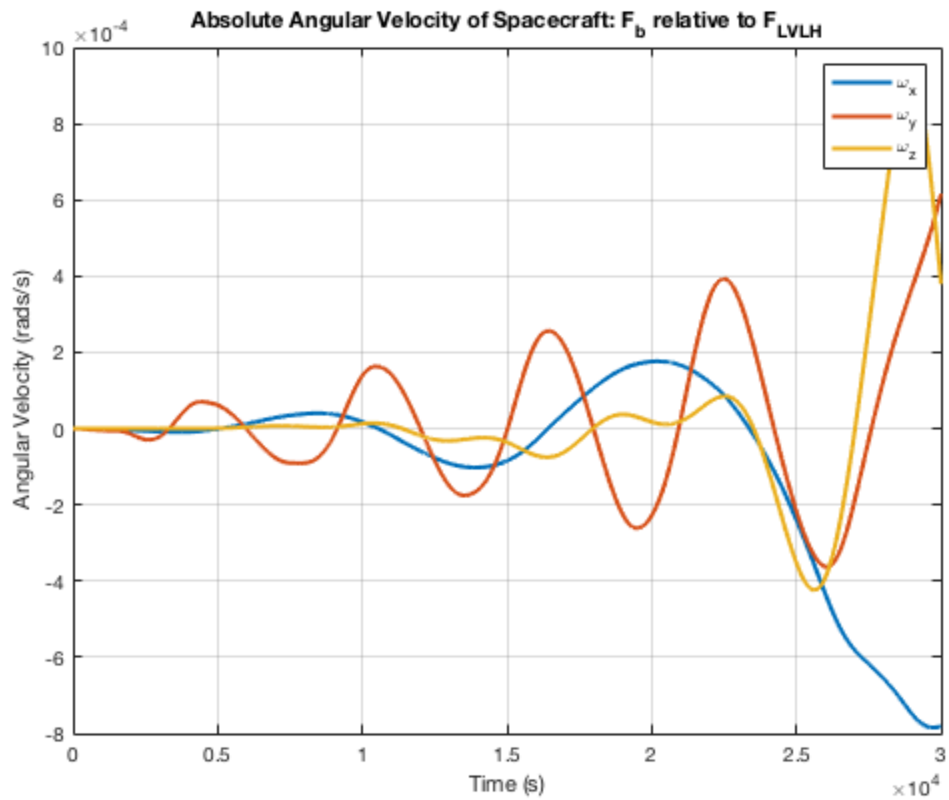


Absolute Angular Velocity of Spacecraft: F_b relative to F_{ECI}
$\times 10^{-3}$
2
1.5
1
0.5
0
-0.5
-1
-1.5
Angular Velocity (rads/s)
0
0.5
1
1.5
2
2.5
3
Time (s)
$\times 10^{4}$
ω_x
ω_y
ω_z

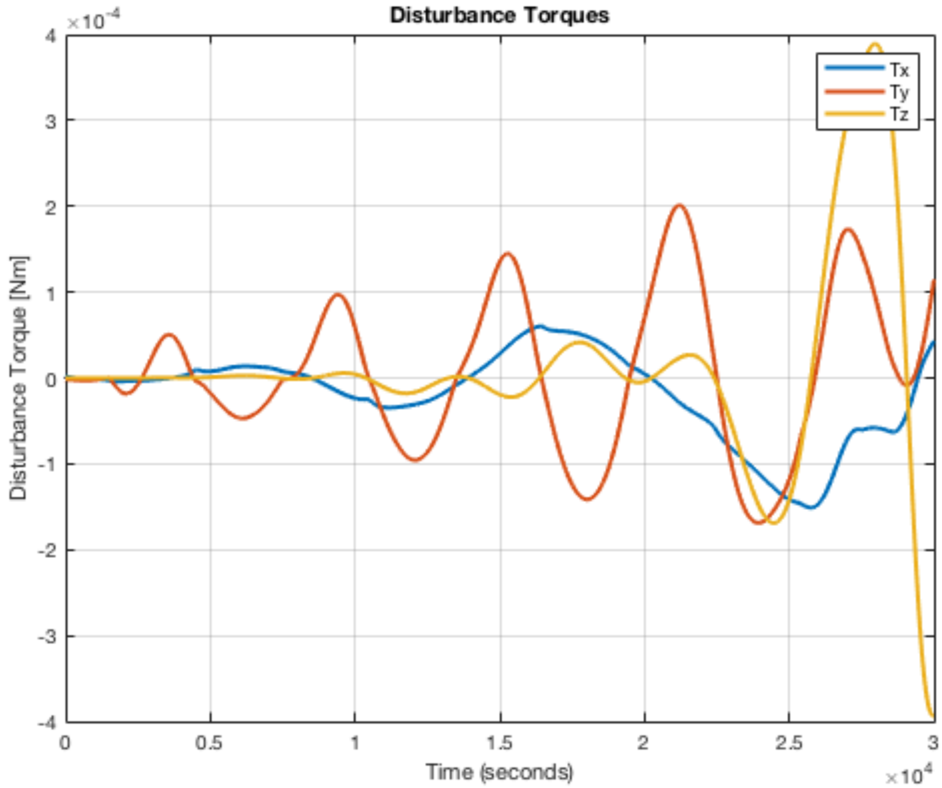
Body Relative to LVLH Plots



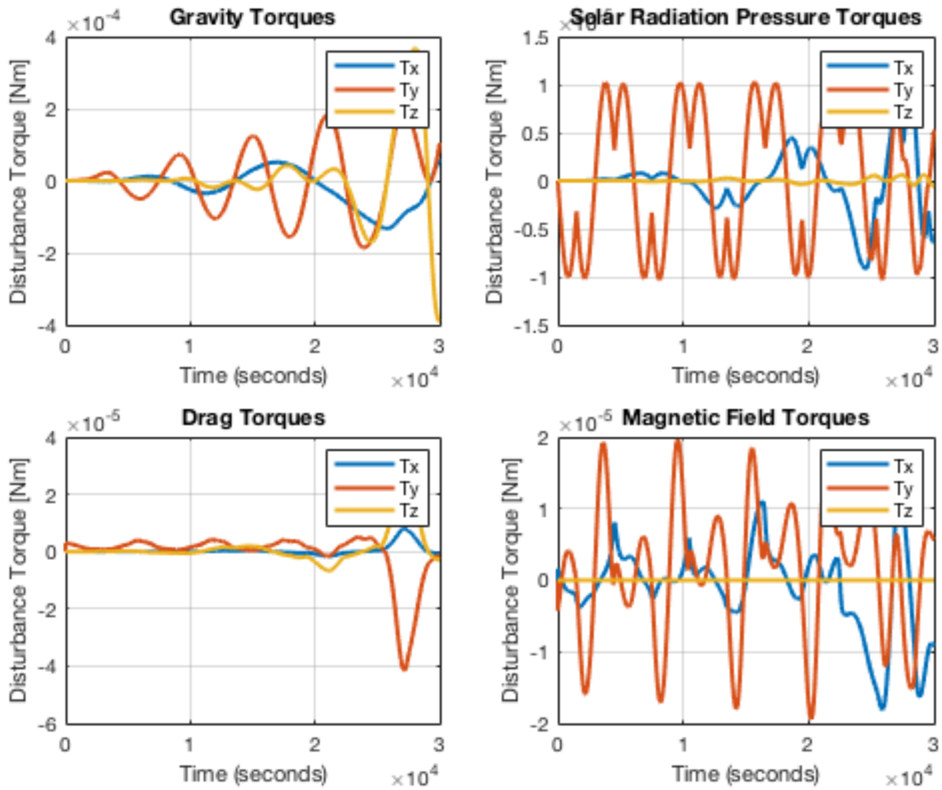



Absolute Angular Velocity of Spacecraft: F_b relative to F_{LVLH}
$\times 10^{-4}$
10
8
6
4
2
0
-2
-4
-6
-8
Angular Velocity (rads/s)
0
0.5
1
1.5
2
2.5
3
Time (s)
$\times 10^{4}$
ω_x
ω_y
ω_z

Total Torque



Individual Torques



Published with MATLAB® R2018a

Aero 421 Homework 6 Discussion

Henry Macanas
Michael Johnston
Eric Ekstrom

Discuss your results. Do they make sense physically? Why does the spacecraft do what it does?

Based upon the plot of angular momentum accumulated over the course of the five orbits, it is clear that the spacecraft initially has a relatively small, fixed angular momentum. Almost immediately the torques due to solar pressure, gravity gradient, magnetic field, and drag all contribute to the increase in angular momentum about the x and z axes. Angular momentum about y remains relatively constant until about the third orbit where tumbling about the y causes the greatest change angular momentum.

Physically this makes sense as the initial spin about the y axis which matches the orbital period initially keeps the spacecraft probe pointed nadier. However because of the torques applied, the spacecraft begins to tumble in a way which corresponds to the periodic nature of a spacecraft's orbit. Solar pressure and drag contribute most to rotation about the x-axis due to the large surface area of the solar panels. There is also a secondary effect about the y-axis due to the panels. The magnetic field torque also produces torques about the y and x axes due to the residual dipole moment in the spacecraft being in the z-axis, and the force attempting to align it with the Earth's magnetic field.

In regard to euler angles of the spacecraft relative to eci, for approximately the first three orbits, the roll and pitch angles oscillate between repeated positions as a result minimal disturbance torques shown by the total disturbance torque plot. Angular position is accumulated about the z axis of the eci frame at an almost constant rate since the orbit is near circular. This explains the steady increase in the yaw angle relative to eci plot. After three orbits the effects of the torques experience by the spacecraft cause the spacecraft to begin tumbling and as a result the role, pitch, and yaw angles become sporadic. This coincides with the increases in disturbance torques seen on the total torque plot.

A similar trend can be seen in the euler angle relative to the lvlh frame plots. For the first three orbits the body and lvlh frame are nearly aligned. After three orbits the frames become more and more unaligned as a result of the disturbance torques increasing. This same trend can also be seen examining the quaternion relative to lvlh.

Table of Contents


```
% Henry Macanas
% Eric Ekstrom
% Michael Johnston
% 421 Assignment 6
clc, close all, clear all

addpath('overhead_functions/')
```

orbital parameters

```
h = 53335.2;
ecc = 0;
RAAN = 0;
inc = deg2rad(98.43);
W = 0;
nu = 0;

% initial r and v vectors
[r0,v0 ] = coes2rv(ecc,inc,RAAN,W,h,nu );

% retrieves period
[ ~,~,~,~,~,~,P,~ ] = coes( r0,v0 );

% angular velcoity of lvlh frame
w_lvlh_eci_eci = cross(r0,v0)/norm(r0)^2; % this is wrong, should be a
 3x1 not 1x1
```

rigid body prop

```
% lvlh to eci tranformation
C_lvlh_eci= eci2lvlh(r0,v0);
C_eci_lvlh = C_lvlh_eci';
C_body_lvlh = eye(3);
C_body_eci = C_body_lvlh*C_lvlh_eci;

% initial states inn reference to eci
w0_body_eci = [0;-2*pi/P;0];
r0_eci_eci = r0;
```

```
v0_eci_eci = v0;
euler_angles0_eci = euler_angs(C_body_eci);
q0_eci_eci = quaternion(C_body_eci);

% states relative to lvlh
euler_angs0_lvlh = [0;0;0];
w_body_eci = w0_body_eci;
w_body_lvlh0 = w_body_eci - C_body_eci*w_lvlh_eci_eci;
q0_body_lvlh = [0;0;0;1];

state =
 [euler_angles0_eci;w0_body_eci;q0_eci_eci;r0_eci_eci;v0_eci_eci;...
    euler_angs0_lvlh;w_body_lvlh0;q0_body_lvlh];

% Distances to individual centers of mass
Rbb = zeros(3,1); % Distance from bus COM to bus COM
Rbsp = [0; 2.5; 0]; % Distance from bus COM to solar panel COM
Rbsens = [0; 0; 1.5]; % Distance from bus COM to sensor COM
masses = [500,20,20,100]; % Component masses
dims = [2 2 2 0.25; 2 3 3 0.25; 2 0.05 0.05 1]; % Component dimensions
[consts.COM, consts.I] = getCOM(masses,...
 [Rbb Rbsp -Rbsp Rbsens],...
 dims); % Spacecraft center of mass

% inertial parameters
consts.n   = [1 0 -1  0 0  0    1 1 -1 -1 0 0  0  0    1 -1 0  0;
              0 1  0 -1 0  0    0 0  0  0 0 0  0  0    0  0 1 -1;
              0 0  0  0 1 -1    0 0  0  0 1 1 -1 -1    0  0 0  0]; %
 Bus     Solar Panels    Sensor
consts.rho = [1 0 -1  0 0  0      1    1   -1   -1    0    0     0
 0    .125 -.125    0      0;
              0 1  0 -1 0  0    2.5 -2.5  2.5 -2.5  2.5 -2.5   2.5
 -2.5       0      0 .125 -.125;
              0 0  0  0 1 -1      0    0    0    0 .025 .025 -.025
 -.025     1.5    1.5  1.5    1.5] - consts.COM; % Bus    Solar Panels
  Sensor
consts.A   = [4 4 4 4 4 4    .15 .15 .15 .15 6 6 6 6
  .25 .25 .25 .25]; % Bus    Solar Panels    Sensor

% Activity 1 & 2 outputs
 fprintf('The inertial tensor is:  \n')
 fprintf('\n')
 disp(consts.I)
 fprintf('kg-m2 \n\n')
 fprintf('The mass of the spacecraft is %d kg.\n\n',640)

%--ode call

Torque = 'yes';
tspan = [0 5*P];
options =
 odeset('RelTol',1e-8,'AbsTol',1e-8, 'OutputFcn',@(t,y,flag,varargin)
 odeOutFunc(t,y,flag));
[tnew, statenew] = ode45(@day_func,tspan,state,options,Torque,consts);
```

```
% Save and load solutions for speed
% save('soln','tnew','statenew', 'Torques')
% load('soln')

h = zeros(length(statenew),3);
mag_h = zeros(length(statenew),1);
for i = 1:length(statenew)
 h(i,:) = cross(diag(consts.I,0), statenew(i,4:6));
    mag_h(i,:) = norm(h(i,:));
end
```

Angular Momentum Accumulated

```
figure
subplot(2,1,1)
plot(tnew,h(:,:), 'lineWidth', 2)
grid on

title('Angular Momentum')
xlabel('Time (seconds)'), ylabel('Angular Momentum (kg-m2/sec)')
legend('h_x','h_y','h_z')

subplot(2,1,2)
plot(tnew,mag_h(:,:), 'lineWidth', 2)
grid on
title('Total Angular Momentum')
xlabel('Time (seconds)'), ylabel('Angular Momentum (kg-m2/sec)')

title('Total Angular Momentum Accumulated')
xlabel('Time (seconds)'), ylabel('Angular Momentum (kg-m^2/sec)')
```

Body Relativeto ECI Plots

```
figure
subplot(2,1,1)
plot(tnew,rad2deg(statenew(:,1:3)),'LineWidth',2)
title('Euler Angles from F_b to F_{ECI}')
xlabel('Time (s)')
ylabel('Angle (degrees)')
legend('\phi','\theta','\psi')

subplot(2,1,2)
plot(tnew,statenew(:,7:10),'LineWidth',2)
title('Quaternion Components from F_b to F_{ECI}')
xlabel('Time (s)')
ylabel('Magnitude (None)')
legend('\epsilon_x','\epsilon_y','\epsilon_z','\eta')

figure
set(groot,'DefaultAxesXGrid','on', 'DefaultAxesYGrid','on')
plot(tnew,statenew(:,4:6),'LineWidth',2)
title('Absolute Angular Velocity of Spacecraft: F_b relative to
 F_{ECI}')
```

```
xlabel('Time (s)')
ylabel('Angular Velocity (rads/s)')
legend('\omega_x','\omega_y','\omega_z')
```

Body Relative to LVLH Plots

```
figure
subplot(2,1,1)
plot(tnew,rad2deg(statenew(:,17:19)),'LineWidth',2)
title('Euler Angles from F_b to F_{LVLH}')
xlabel('Time (s)')
ylabel('Angle (degrees)')
legend('\phi','\theta','\psi')

subplot(2,1,2)
plot(tnew,statenew(:,23:26),'LineWidth',2)
title('Quaternion Components from F_b to F_{LVLH}')
xlabel('Time (s)')
ylabel('Magnitude (None)')
legend('\epsilon_x','\epsilon_y','\epsilon_z','\eta')

figure
plot(tnew,statenew(:,20:22),'LineWidth',2)
title('Absolute Angular Velocity of Spacecraft: F_b relative to
 F_{LVLH}')
xlabel('Time (s)')
ylabel('Angular Velocity (rads/s)')
legend('\omega_x','\omega_y','\omega_z')

% orbit plot
% figure
% hold on
% plot3(statenew(:,11),statenew(:,12),statenew(:,13))
% plot3(statenew(1,11),statenew(1,12),statenew(1,13),'*')
```

Total Torque

```
figure
plot(Torques.tot(:,1),Torques.tot(:,2:4), 'lineWidth', 2)
grid on
title('Disturbance Torques')
xlabel('Time (seconds)'), ylabel('Disturbance Torque [Nm]')
legend('Tx', 'Ty', 'Tz')
```

Individual Torques

```
figure
subplot(2,2,1)
plot(Torques.grav(:,1),Torques.grav(:,2:4), 'lineWidth', 2)
grid on
title('Gravity Torques')
xlabel('Time (seconds)'), ylabel('Disturbance Torque [Nm]')
```

```
legend('Tx', 'Ty', 'Tz')

subplot(2,2,2)
plot(Torques.srp(:,1),Torques.srp(:,2:4), 'lineWidth', 2)
grid on
title('Solar Radiation Pressure Torques')
xlabel('Time (seconds)'), ylabel('Disturbance Torque [Nm]')
legend('Tx', 'Ty', 'Tz')

subplot(2,2,3)
plot(Torques.drag(:,1),Torques.drag(:,2:4), 'lineWidth', 2)
grid on
title('Drag Torques')
xlabel('Time (seconds)'), ylabel('Disturbance Torque [Nm]')
legend('Tx', 'Ty', 'Tz')

subplot(2,2,4)
plot(Torques.mag(:,1),Torques.mag(:,2:4), 'lineWidth', 2)
grid on
title('Magnetic Field Torques')
xlabel('Time (seconds)'), ylabel('Disturbance Torque [Nm]')
legend('Tx', 'Ty', 'Tz')
```

Published with MATLAB® R2018a