Programming Project 1

Stephen Baird II*
Embry-Riddle Aeronautical University, Daytona-Beach, Fl, 32114

This project addresses Spacecraft Attitude Dynamics through the calculation of the center of mass and moments of inertia of a satellite, followed by principal axes analysis and calculation of angular velocity. Using constants defined as a = 21 and b = 41, derived from the author's name and birth date, respectively, the satellite is modeled with six components placed in a body-fixed reference frame. Part 1 involves determining the center of mass of the satellite (0.0091, 0.0457, 0.1189) and calculating both the inertia matrix about the origin and about the center of mass. In Part 2, the principal moments and axes of inertia are computed from the inertia matrix at the center of mass. Finally, in Part 3, Euler's equations are solved using MATLAB's ODE45 solver to simulate the angular velocity vector over 100 seconds, assuming zero external torque. The rotational kinetic energy is computed throughout the simulation, and its constancy is validated. All results are presented through code-generated outputs and plots.

I. Nomenclature

X_G	=	X component of the center of mass
Y_G	=	Y component of the center of mass
Z_G	=	Z component of the center of mass
I_o	=	moments of inertia matrix of the satellite about the origin
x_I	=	X component with respect to centroid
y_I	=	Y component with respect to centroid
z_I	=	Z component with respect to centroid
I_G	=	moments of inertia matrix of the satellite about the center of mass
D	=	principle moments of inertia
V	=	principle axes of inertia
M_{net}	=	net torque
ω_{to}	=	initial angular velocity
t_{span}	=	time span
T	=	rotational kinetic energy
T_{error}	=	error T

II. Introduction

The attitude dynamics of spacecraft play a crucial role in ensuring stability, orientation control, and mission success for satellites operating in space. Understanding how a satellite's mass distribution affects its rotational behavior is fundamental to designing effective control systems. This project explores the rotational characteristics of a small satellite composed of six distinct components, each with a fixed mass and location relative to a body-fixed coordinate system.

The primary objective of this project is to analyze the satellite's inertial properties and use them to simulate its rotational motion in the absence of external torques. The project is divided into three main parts: determining the satellite's center of mass and inertia matrices, calculating the principal moments and axes of inertia, and simulating the time evolution of angular velocity using numerical methods.

By combining theoretical principles with MATLAB-based simulation, this project aims to build a comprehensive understanding of spacecraft attitude dynamics, develop coding proficiency in solving rigid body dynamics problems, and visualize key physical behaviors such as conservation of rotational kinetic energy.

*Undergraduate Aerospace Engineer Student, Embry-Riddle Aeronautical University, Daytona-Beach, Fl, 32114

Table 1 Major Components for selected Satellite

Point, i	Mass m_i (kg)	x_i (m)	y_i (m)	z_i (m)
1	$(a/10) \times 10$	$(a/b) \times 1$	$(a/b) \times 1$	$(a/b) \times 1$
2	$(a/10) \times 7$	$-(a/b) \times 1$	$-(a/b) \times 1$	$-(a/b) \times 1$
3	$(a/10) \times 5$	$(a/b) \times 4$	$-(a/b) \times 4$	$(a/b) \times 4$
4	$(a/10) \times 8$	$-(a/b) \times 2$	$(a/b) \times 2$	$-(a/b) \times 2$
5	$(a/10) \times 12$	$(a/b) \times 3$	$-(a/b) \times 3$	$-(a/b) \times 3$
6	$(a/10) \times 14$	$-(a/b) \times 3$	$(a/b) \times 3$	$(a/b) \times 3$

III. Theoretical Background

A. Part 1

For part 1 the following table was given for a small satellite consisting of 6 major components with constant masses about the origin.

Using this table the center of mass can be found using the following equations:

$$X_G = \Sigma\{\frac{(x_i \times m_i)}{m_{total}}\} \tag{1}$$

$$Y_G = \Sigma\{\frac{(y_i \times m_i)}{m_{total}}\} \tag{2}$$

$$Z_G = \Sigma\{\frac{(z_i \times m_i)}{m_{total}}\} \tag{3}$$

After plugging in values from Table 1:

$$X_G = 0.0091 \tag{4}$$

$$Y_G = 0.0457 \tag{5}$$

$$Z_G = 0.1189 \tag{6}$$

Both moments of inertia can be found using this matrix:

$$I = \begin{bmatrix} m(y^2 + z^2) & -mxy & -mxz \\ -mxy & m(x^2 + z^2) & -myz \\ -mxz & -myz & m(x^2 + y^2) \end{bmatrix}$$

Determine the moments of inertia matrix of the satellite about the origin by plugging in all 6 components points from the table. Sum all of the matrices to determine the inertial matrix.

$$x = x_i \tag{7}$$

$$y = y_i \tag{8}$$

$$z = z_i \tag{9}$$

$$I_o = \Sigma I_{oi} \tag{10}$$

$$I_o = \begin{bmatrix} 399.964 & 181.25534 & 57.8468 \\ 181.2534 & 399.9694 & -76.5782 \\ 57.8468 & -76.5782 & 399.9694 \end{bmatrix}$$

The moments of inertia matrix about the center of mass can be found similarly to the previous solution. In order to find the points with respect to the center of mass, each point must be subtraacted by the centroid.

$$x = x_I = x_i - X_G \tag{11}$$

$$y = y_I = y_i - Y_G \tag{12}$$

$$z = z_I = z_i - Z_G \tag{13}$$

$$I_G = \Sigma I_{Gi} \tag{14}$$

$$I_G = \begin{bmatrix} 398.0609 & 181.3026 & 57.9747 \\ 181.3026 & 398.2970 & -75.9387 \\ 57.9747 & -75.9387 & 399.7136 \end{bmatrix}$$

B. Part 2

The principle axes and moments of inertia can be found using the EIG function in MATLAB. The principal axes will be the eigen vectors. The principal moments will be the eigen values. The principle axes of inertia (V) and principle axes of inertia (D).

$$V = \begin{bmatrix} -0.6421 & 0.3352 & -0.6894 \\ 0.6592 & -0.2176 & -0.7198 \\ 0.3913 & 0.9167 & 0.0813 \end{bmatrix}$$

$$D = \begin{bmatrix} 176.6166 & 0 & 0 \\ 0 & 438.9350 & 0 \\ 0 & 0 & 580.5198 \end{bmatrix}$$

C. Part 3

ω was found using the ODE45 in MATLAB for the first 100 s, the net moment is M_{net}= 0 N-m and $\omega(t_0) = [-0.91; 0.0986; 1.1945]$ rad/s.

T was calculated using the following formula for each ω:

$$T = 1/2 * \omega^T * I_G * \omega \tag{15}$$

To determine error in each T the following equation was used:

$$T_{error} = T - T_0 \tag{16}$$

IV. Numerical Results

A. plot showing the change of $\omega_x, \omega_y, \omega_z$ over time

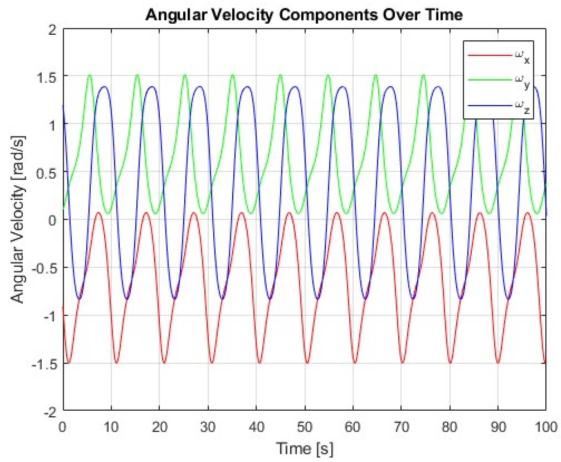

Fig. 1 Angular Velocity Components Over Time.
NOTE: This plot displays angular velocity vector components $\omega_x, \omega_y, \omega_z$ over a 100-second interval, derived by solving Euler's equations of motion using MATLAB's ODE45 solver. Since the satellite is assumed to experience no external torque (free rotation), the variations reflect the natural dynamics of a rigid body rotating about its center of mass. The oscillating behavior, particularly the phase differences and amplitude variation among the components, is due to the nonlinear rotational motion about the satellite's principal axes.

B. plot showing that Rotational T remains constant over the simulation period

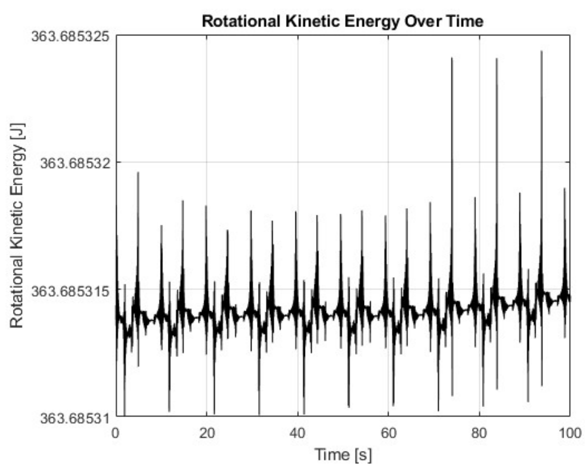

Fig. 2 Rotational Kinetic Energy over Time.
NOTE: This plot tracks the rotational kinetic energy of the satellite throughout the time span. Assuming external torque is zero, the kinetic energy should ideally remain constant. The near-constant value with noticeable variation confirms that the integration of Euler's equations conserves energy, consistent with theoretical expectations for torque-free rigid body rotation.

C. Error Plot

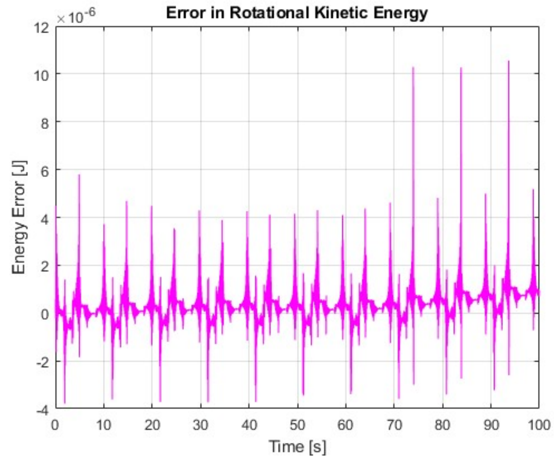

Fig. 3 Error in Rotational Kinetic Energy

NOTE: This plot displays the deviation of the rotational kinetic energy from its expected constant value over time. By highlighting small fluctuations in energy. The low magnitude of the energy error, generally within the range of $4 * 10^-6$ to $10 * 10^-6$, supports that the ODE45 solver preserves energy well over long simulation times.

V. Conclusion

In this project, we successfully analyzed the rotational dynamics of a satellite using its mass properties and numerical integration techniques. First, the satellite's center of mass and inertia matrices were calculated with respect to both the body-fixed origin and the center of mass. We then performed eigenvalue decomposition on the inertia matrix about the center of mass,I_G , to obtain the principal moments of inertia and corresponding principal axes, which simplify the analysis of rotational motion.

Using the principal inertia matrix, Euler's equations were solved numerically over a 100-second interval assuming zero external torque. The results showed that the angular velocity components $\omega_x, \omega_y, \omega_z$varied over time due to the internal dynamics of the rotating satellite, consistent with torque-free motion behavior.

The rotational kinetic energy was computed at each time step and remained approximately constant throughout the simulation, verifying the conservation of energy in the absence of external torques. This consistency also confirmed the accuracy of the numerical integration.

Overall, the project demonstrated key principles of rigid body dynamics, including the importance of the inertia matrix, principal axes, and energy conservation in rotational motion. These methods and results are fundamental in satellite attitude dynamics and control.

Appendix

A. Matlab Code

```
clear
clc
close all


%Stephen Delaney Baird II
%08/29/04

a=21;
b=8+29+4;
```

```
a_10=a/10;
a_b=a/b;


mi=[ a_10*10 a_10*7 a_10*5 a_10*8 a_10*12 a_10*14 ];
xi=[ a_b -a_b a_b*4 -a_b*2 a_b*3 -a_b*3 ];
yi=[ a_b -a_b -a_b*4 a_b*2 -a_b*3 a_b*3];
zi=[ a_b -a_b a_b*4 -a_b*2 -a_b*3 a_b*3];

%Center of Mass
xm=xi.*mi;
ym=yi.*mi;
zm=zi.*mi;

x_G= sum(xm)/sum(mi);
y_G= sum(ym)/sum(mi);
z_G= sum(zm)/sum(mi);



%Moment of Inertia of Origin

I_o= zeros(3,3);
for i=1:6
    m=mi(i);
    x=xi(i);
    y=yi(i);
    z=zi(i);

I = [ m*(y^2 + z^2), -m*x*y,        -m*x*z;
     -m*x*y,         m*(x^2 + z^2), -m*y*z;
     -m*x*z,        -m*y*z,         m*(x^2 + y^2) ];
I_o= I_o+I;
end



%Moment of Inertia of center of gravity


I_G= zeros(3,3);
for i=1:6
    m=mi(i);
    x=xi(i)-x_G;
    y=yi(i)-y_er_G;
    z=zi(i)-z_G;

Ig = [ m*(y^2 + z^2), -m*x*y,        -m*x*z;
     -m*x*y,         m*(x^2 + z^2), -m*y*z;
     -m*x*z,        -m*y*z,         m*(x^2 + y^2) ];

I_G= I_G+Ig;
end
```

```
% ) Find the principal axes of inertia of the moments of inertia matrix, IG
[V, D] = eig(I_G);  %d is the principle moment and v is the principle axes

Mnet=0;
w_t0= [-0.91 0.0986 1.1945];
tspan= [0 100];

% ODE function for d(omega)/dt
eulerEq = @(t, omega) I_G \ (-cross(omega, I_G * omega));

% Solve using ODE45
opts = odeset('RelTol',1e-8,'AbsTol',1e-10); % better accuracy
[t, w_t0] = ode45(eulerEq, tspan, w_t0, opts);

figure;
plot(t, w_t0(:,1), 'r', 'DisplayName','\omega_x'); hold on;
plot(t, w_t0(:,2), 'g', 'DisplayName','\omega_y');
plot(t, w_t0(:,3), 'b', 'DisplayName','\omega_z');
xlabel('Time [s]');
ylabel('Angular Velocity [rad/s]');
title('Angular Velocity Components Over Time');
legend;
grid on;

KE = zeros(length(t), 1);
for i = 1:length(t)
    w = w_t0(i, :)';
    KE(i) = 0.5 * w' * I_G * w;
end

%Compute error from initial energy
KE_error = KE - KE(1);

figure;
plot(t, KE, 'k');
xlabel('Time [s]');
ylabel('Rotational Kinetic Energy [J]');
title('Rotational Kinetic Energy Over Time');
grid on;

% Error plot
figure;
plot(t, KE_error, 'm');
xlabel('Time [s]');
ylabel('Energy Error [J]');
title('Error in Rotational Kinetic Energy');
grid on;
```