Table of Contents


```
%Roshan Jaiswal-Ferri
%Section - 01
%Aero 320 HW 5 - 10/29/24
```

Workspace Prep

```
format long;  % Allows for more accurate decimals
close all;    % Clears all
clear all;    % Clears Workspace
clc;          % Clears Command Window
addpath('C:\Users\rosha\Documents\MATLAB\RBSim FIles');
```

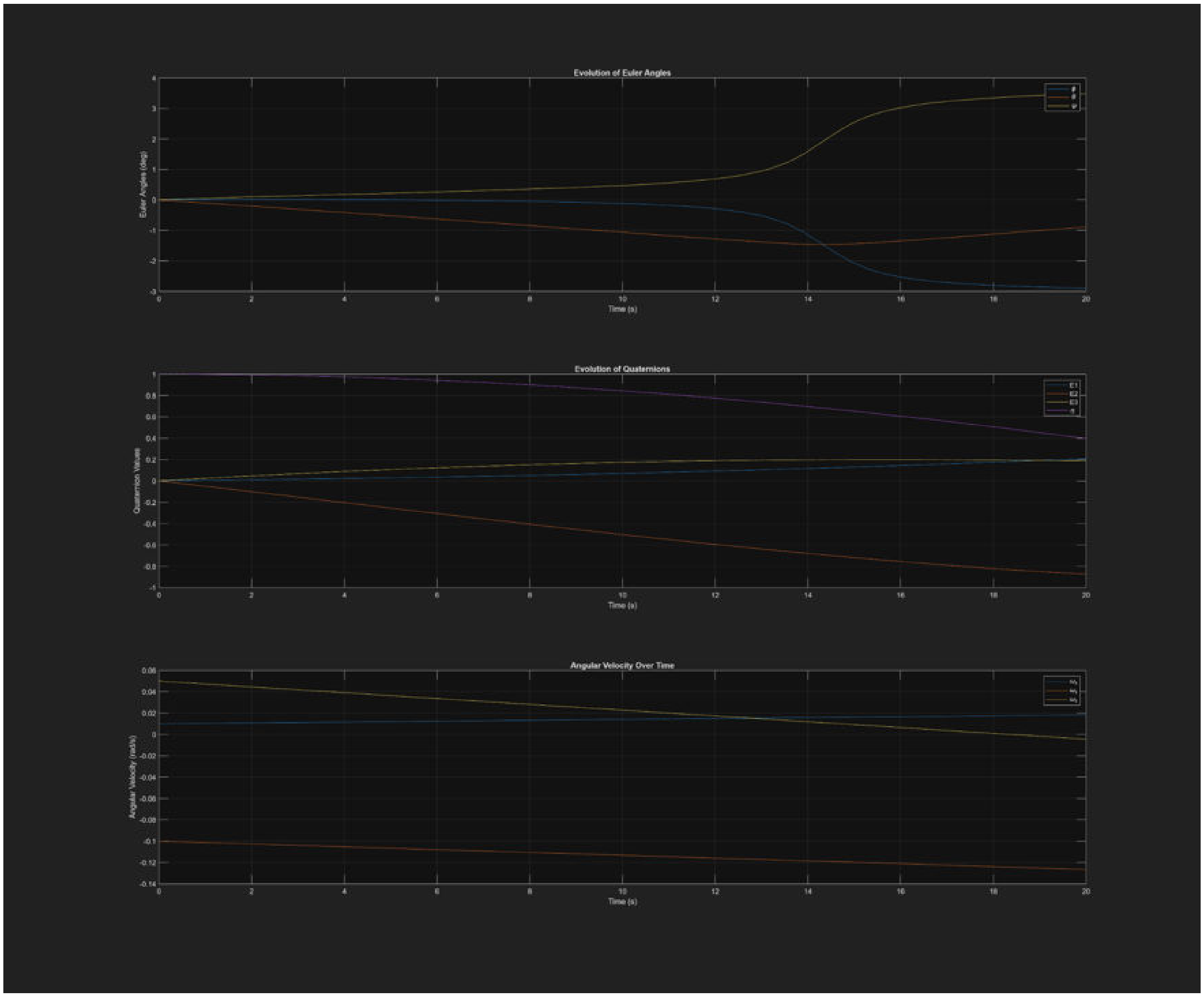
PART 1: Problem 1 Part A

```
wb0 = [0.01; -0.1; 0.05]; %angular vel rad/s
q0 = [0; 0; 0; 1]; %initial quaternion
eulA = [0; 0; 0]; %initial euler angles of 0
Torque = [0; 0; 0]; %initial torque in Nm
time = [0, 20]; %sec

J = [17, -3, 2; -3, 20, -4; 2, -4, 15];

[eigenvectors, eigenvalues_matrix] = eig(J);
principal_moments = diag(eigenvalues_matrix);

disp('Principle Moments are:');
disp(['J1: ', num2str(principal_moments(1))]);
disp(['J2: ', num2str(principal_moments(2))]);
disp(['J3: ', num2str(principal_moments(3))]);
disp(' ');
```

```
Principle Moments are:
J1: 12.7528
J2: 15.1967
J3: 24.0505
```

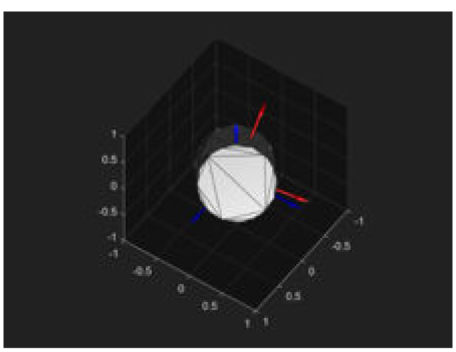
PART 2: Problem 1 Part B (No Initial Torque)

```
[t3, w] = ode45(@(t,states) angularV(t, wb0, J, Torque), time, wb0, []);
w_interp = @(t) interp1(t3, w, t);

[t1, q] = ode45(@(t, y) quaternion(t, y, w_interp(t).'), time, q0);
[t2, e] = ode45(@(t, y) eula(t, y, w_interp(t).'), time, eulA);


% Plot results for No Initial Torque
figure('Name', 'No Initial Torque');
% Increase figure pixel resolution and font sizes (for publishing)
Gx = gcf;
Gx.Position(3:4) = Gx.Position(3:4)*5;
Ax = gca;
Ax.FontSize = Ax.FontSize *1.5;
subplot(3, 1, 1);  % 3 rows, 1 column, 1st subplot
plot(t2, e(:,1), t2, e(:,2), t2, e(:,3));
grid on;
xlabel('Time (s)');
ylabel('Euler Angles (deg)');
legend('\phi', '\theta', '\psi');
title('Evolution of Euler Angles');

subplot(3, 1, 2);  % 3 rows, 1 column, 2nd subplot
plot(t1, q(:,1), t1, q(:,2), t1, q(:,3), t1, q(:,4));
grid on;
xlabel('Time (s)');
ylabel('Quaternion Values');
legend('E1', 'E2', 'E3', '\eta');
title('Evolution of Quaternions');

subplot(3, 1, 3);  % 3 rows, 1 column, 3rd subplot
plot(t3, w(:,1), t3, w(:,2), t3, w(:,3));
grid on;
xlabel('Time (s)');
ylabel('Angular Velocity (rad/s)');
legend('\omega_x', '\omega_y', '\omega_z');
title('Angular Velocity Over Time');
```



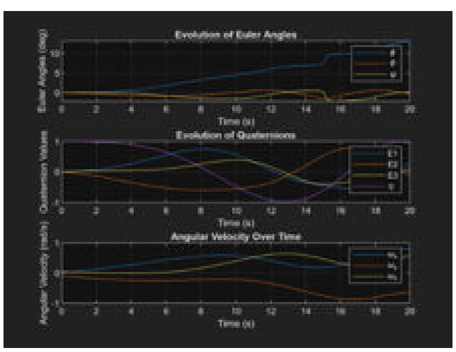
Simulation 1 (No Torque)

```
figure('Name','Simulation Without Torque')
psi = e(:,3);
theta = e(:,2);
phi = e(:,1);
M = RBMotionSim(psi, theta, phi);
```



PART 3: Problem 1 Part C (Including Initial Torque)

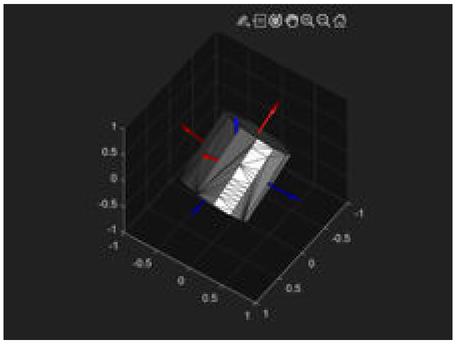
```
Torque = [1; -1; 0];  % Applied torque in Nm

[t3, w] = ode45(@(t, states) angularV(t, states, J, Torque), time, wb0);
w_interp = @(t) interp1(t3, w, t);

[t1, q] = ode45(@(t, y) quaternion(t, y, w_interp(t).'), time, q0);
[t2, e] = ode45(@(t, y) eula(t, y, w_interp(t).'), time, eulA);

figure('Name', 'Including Initial Torque');
subplot(3, 1, 1);  % 3 rows, 1 column, 1st subplot
plot(t2, e(:,1), t2, e(:,2), t2, e(:,3));
grid on;
xlabel('Time (s)');
ylabel('Euler Angles (deg)');
legend('\phi', '\theta', '\psi');
title('Evolution of Euler Angles');

subplot(3, 1, 2);  % 3 rows, 1 column, 2nd subplot
plot(t1, q(:,1), t1, q(:,2), t1, q(:,3), t1, q(:,4));
grid on;
xlabel('Time (s)');
ylabel('Quaternion Values');
legend('E1', 'E2', 'E3', '\eta');
title('Evolution of Quaternions');

subplot(3, 1, 3);  % 3 rows, 1 column, 3rd subplot
plot(t3, w(:,1), t3, w(:,2), t3, w(:,3));
grid on;
xlabel('Time (s)');
ylabel('Angular Velocity (rad/s)');
legend('\omega_x', '\omega_y', '\omega_z');
title('Angular Velocity Over Time');
```



Simulation 2 (With Torque)

```
figure('Name','Simulation With Torque')
psi = e(:,3);
```

```
theta = e(:,2);
phi = e(:,1);
M = RBMotionSim(psi, theta, phi);
```



PART 4: Problem 2

```
m = 17474; %kg
w = [0.5; -0.1; 0.1]; %ang vel
v = [20; 105; -10]; %vel
J = [2.44, 0, -1.2; 0, 27, 0; -1.2, 0, 30]; %mom of inert
J = J.*1e+6;

KE = norm(0.5*m*v.^2+0.5*w'*J*w);
disp(['Kinetic Energy of the Airplane: ', num2str(KE/1000), ' KJ']);
```

```
Kinetic Energy of the Airplane: 96949.1761 KJ
```

Functions:

```
function dw = angularV(t,w,J,T)

    %Euler's equation of motion (torques)
    dw(1:3,1) = inv(J)*(T-vcross(w)*J*w);

end

function [dq] = quaternion(t,x,o)
    ep = x(1:3,1);
    n = x(4);

    dq(1:3,1) = .5*(n*eye(3)+vcross(ep))*o;
    dq(4,1) = -.5*ep'*o;
end

function [de] = eula(t,x,o)
    ph = x(1);
    th = x(2);

    de(1:3,1) = 1/cos(th)*[cos(th), sin(ph)*sin(th), cos(ph)*sin(th);...
        0, cos(ph)*cos(th), -sin(ph)*cos(th);...
        0, sin(ph), cos(ph)]*o;
```

```
end

function [out] = vcross(v)
    out = [0, -v(3), v(2); v(3), 0, -v(1); -v(2), v(1), 0];
end
```

Published with MATLAB® R2023b