Table of Contents


```
%ASEN 3128 Lab 2: Problem 1
%9/7/2021
%By Ben Arnold, Brian Trybus, Chieri Kamada
```

Discription

```
%Simulation of quadrotor with attitude, and kinematics using Euler
 angle
%attitude.
```

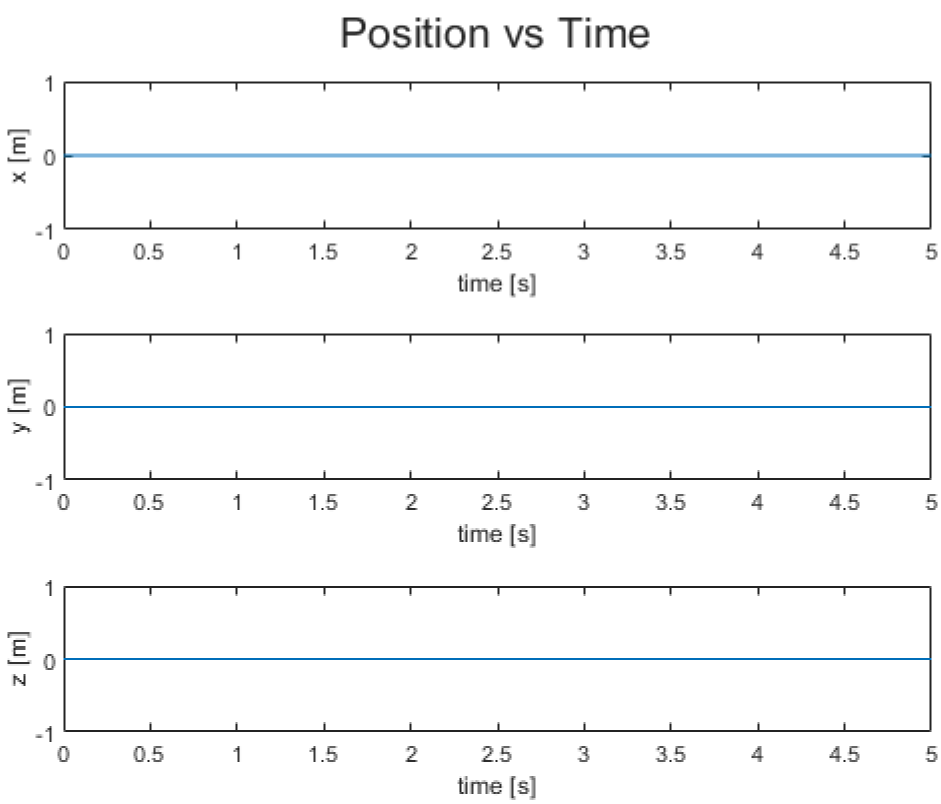
Main

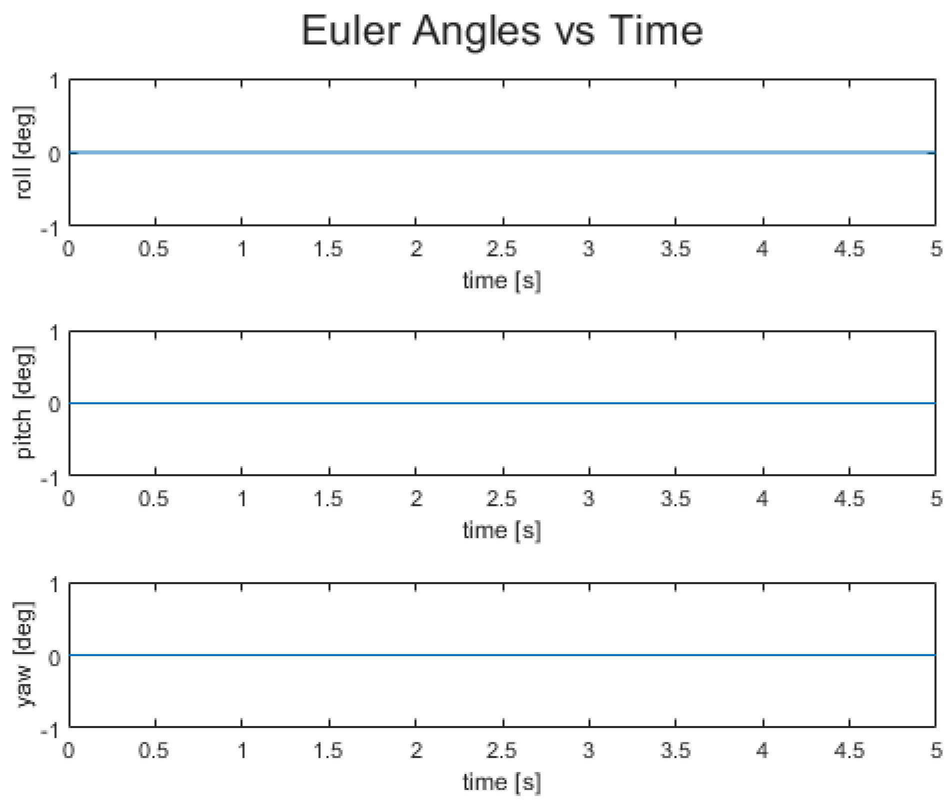
```
%Given Variblaes:

m = 0.068; %(kg) mass of craft
r = 0.06; %(m) distance from rotor to cg
k_m = 0.0024; % (N*m/N) Control moment coeff
I_x = 6.8*(10^-5); %(kg*(m^2))
I_y = 9.2*(10^-5); %(kg*(m^2))
I_z = 1.35*(10^-4); %(kg*(m^2))
nu = 1*(10^-3); %(N/(m/s)^2)
mu = 2*(10^-6); % (N*m/(rad/s)^2)
g = 9.81; % m/s^2
v_air = 5; % m/s, airspeed/quadrotor's speed

phi = atan((v_air^2)*nu/(m*g));
%Zc = m*g*cos(phi)+(v_air^2)*nu*sin(phi);

Zc = (m*g)/cos(phi);



%Equations
%use 12 varible state vector in ode45

%state = [x;y;z;phi;theta;psi;u;v;w;p;q;r]
state_0 = [0;0;0;phi;0;0;0;v_air*cos(phi);-
v_air*sin(phi);0;0;0];%Inital condisions of the aircraft
```

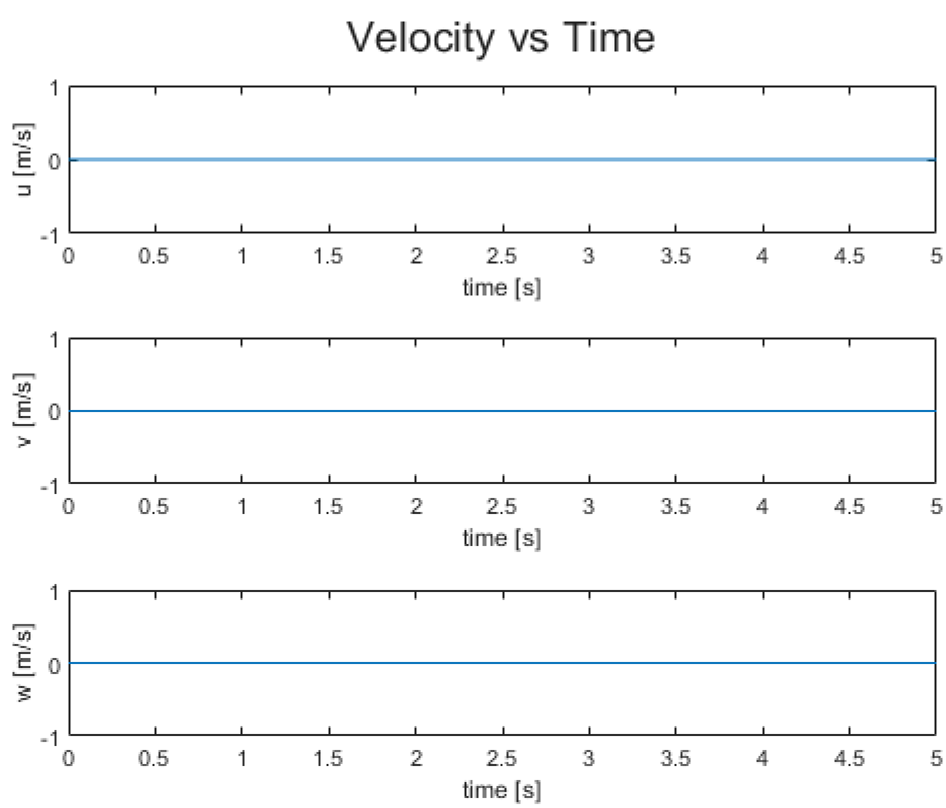
```
%f1-f4 is rotor force;

f1 = Zc/4;
f2 = f1;
f3 = f1;
f4 = f1;
constants = [m;r;k_m;I_x;I_y;I_z;nu;mu;f1;f2;f3;f4;g];

% Call ODE45
```

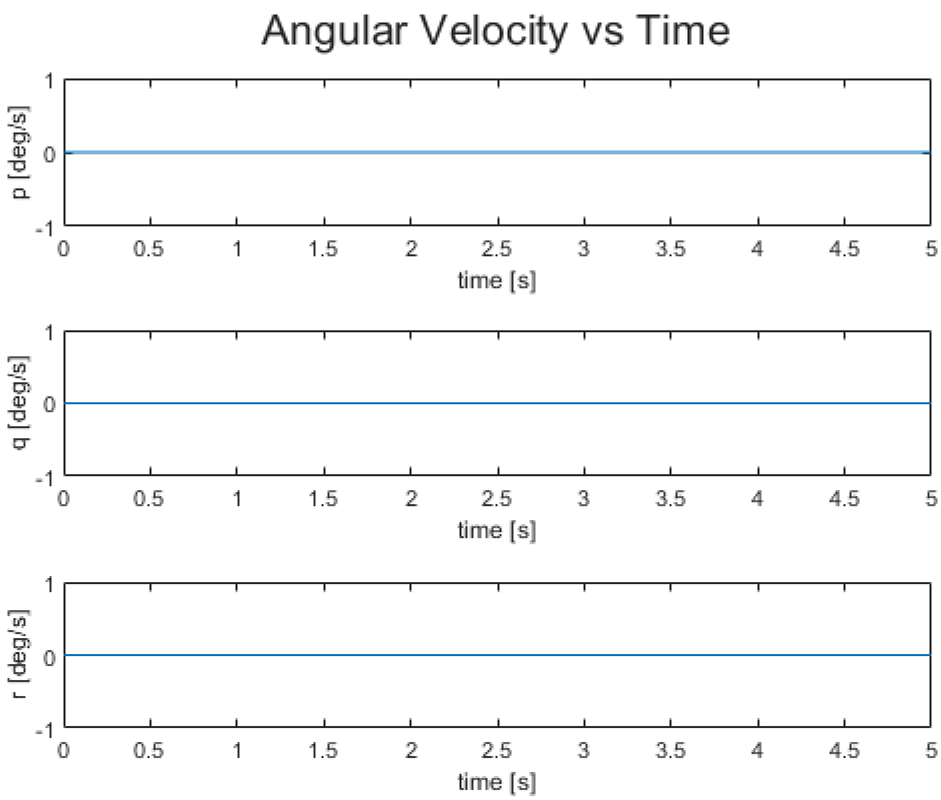
Problem 1

```
state_0 = [0;0;0;0;0;0;0;0;0;0;0;0];%Inital condisions of the aircraft
%Rotor Force
f1 = (m*g)/4;
f2 = f1;
f3 = f1;
f4 = f1;
%Constants
constants = [m;r;k_m;I_x;I_y;I_z;0;0;f1;f2;f3;f4;g];
%ODE45 Call
tSpan = [0,5];
[t,state] = ode45(@(t,state)EOMQuad(t,state,constants),tSpan,state_0);
%Plots the state.
PlotAircraftSim(t,state,"");
```



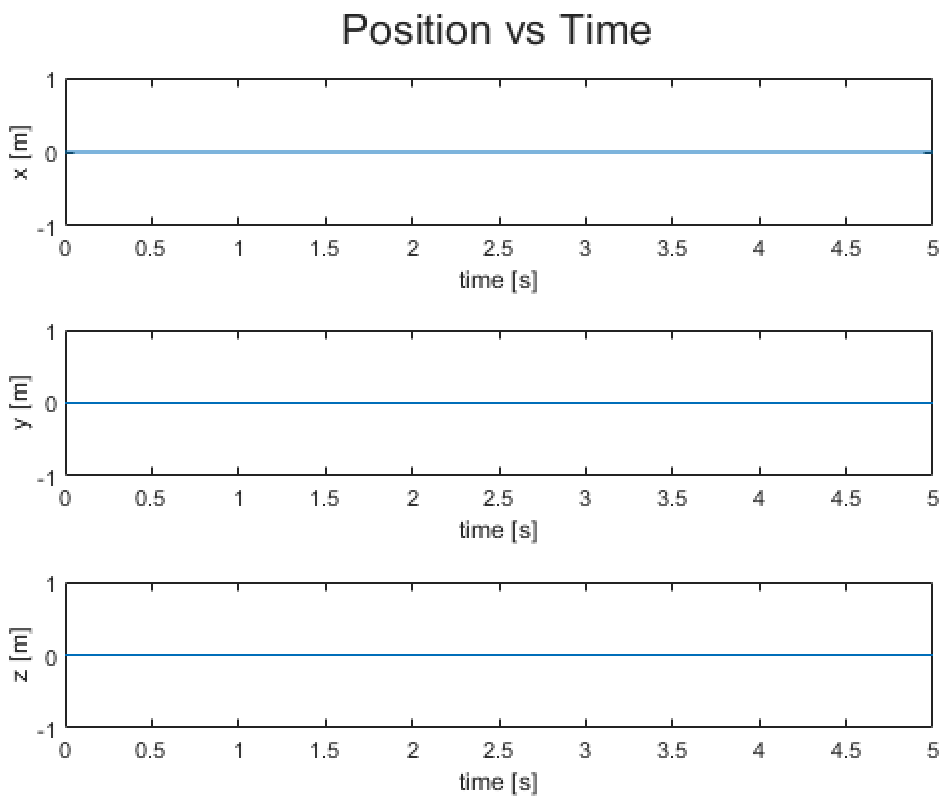

Euler Angles vs Time
roll [deg]
pitch [deg]
yaw [deg]
time [s]

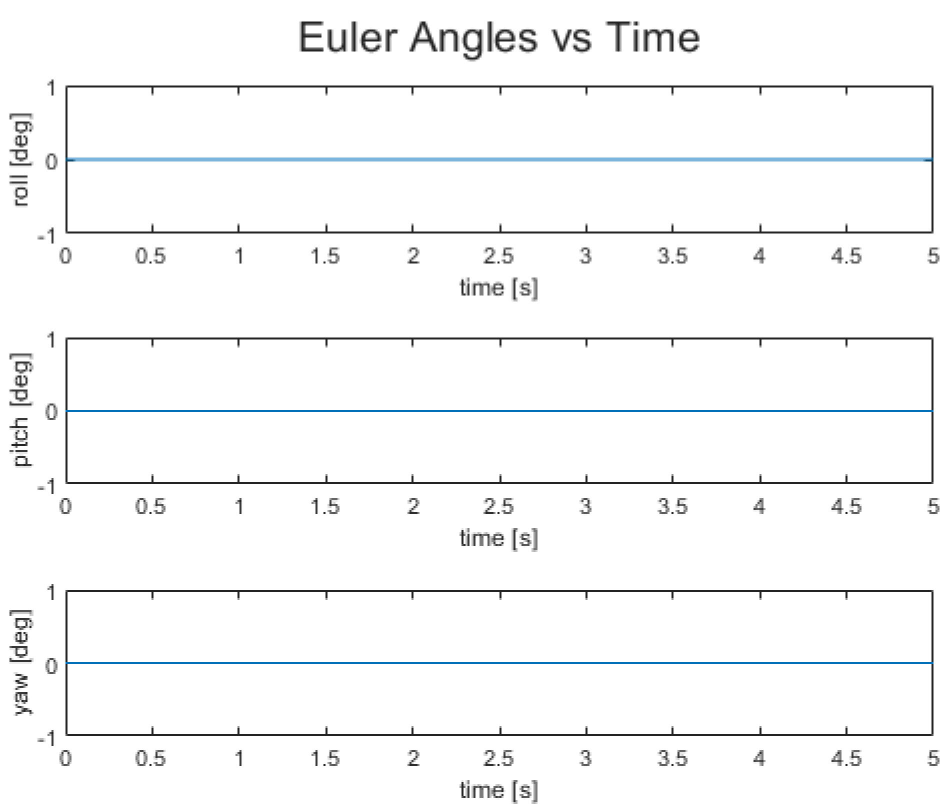

Velocity vs Time
u [m/s]
v [m/s]
w [m/s]
time [s]

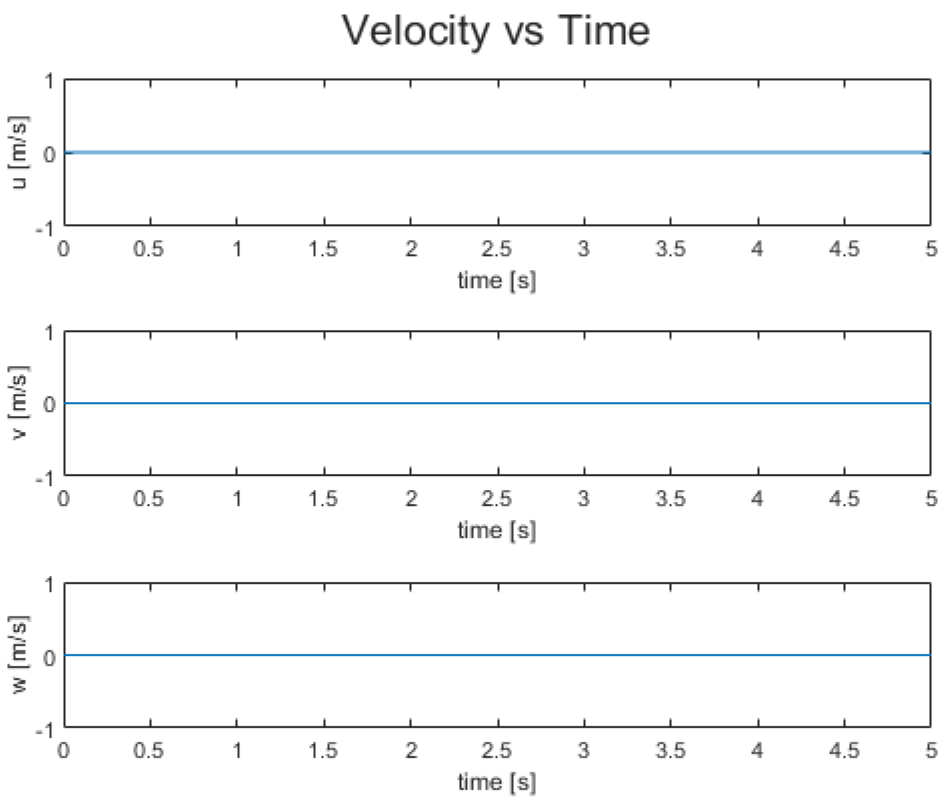


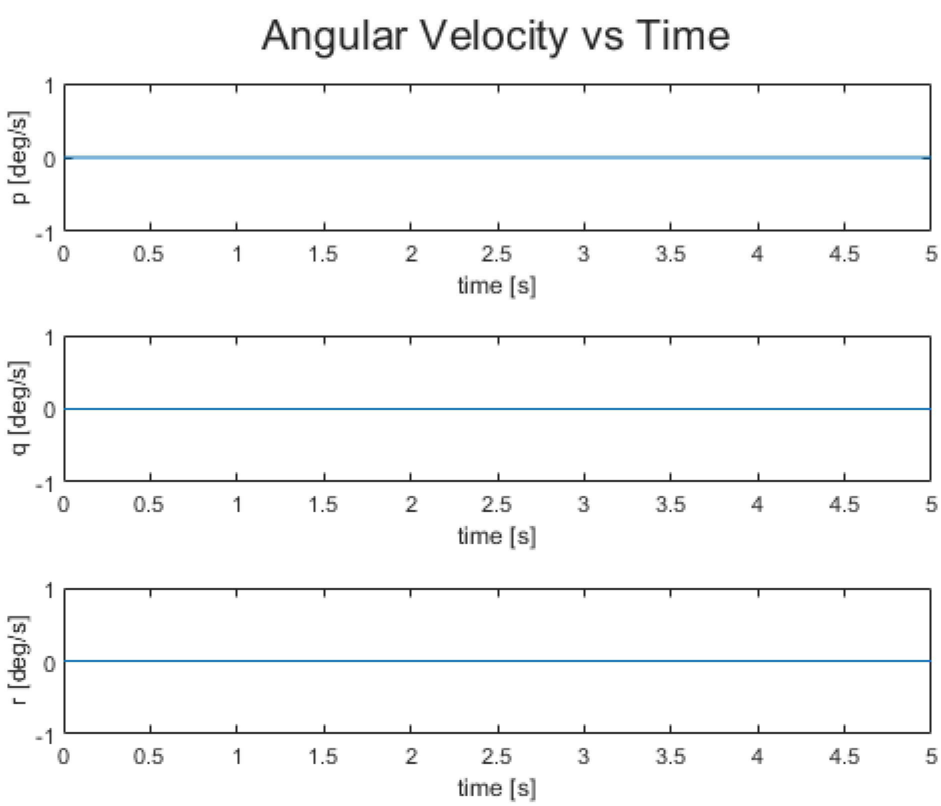
Problem 2a

```
state_0 = [0;0;0;0;0;0;0;0;0;0;0;0];%Inital condisions of the aircraft
%Rotor Force
f1 = (m*g)/4;
f2 = f1;
f3 = f1;
f4 = f1;
%Constants
constants = [m;r;k_m;I_x;I_y;I_z;nu;mu;f1;f2;f3;f4;g];
%ODE45 Call
tSpan = [0,5];
[t,state] = ode45(@(t,state)EOMQuad(t,state,constants),tSpan,state_0);
%Plots the state.
PlotAircraftSim(t,state,"");
```


Position vs Time
x [m]
y [m]
z [m]
time [s]
1
0
-1
0 0.5 1 1.5 2 2.5 3 3.5 4 4.5 5

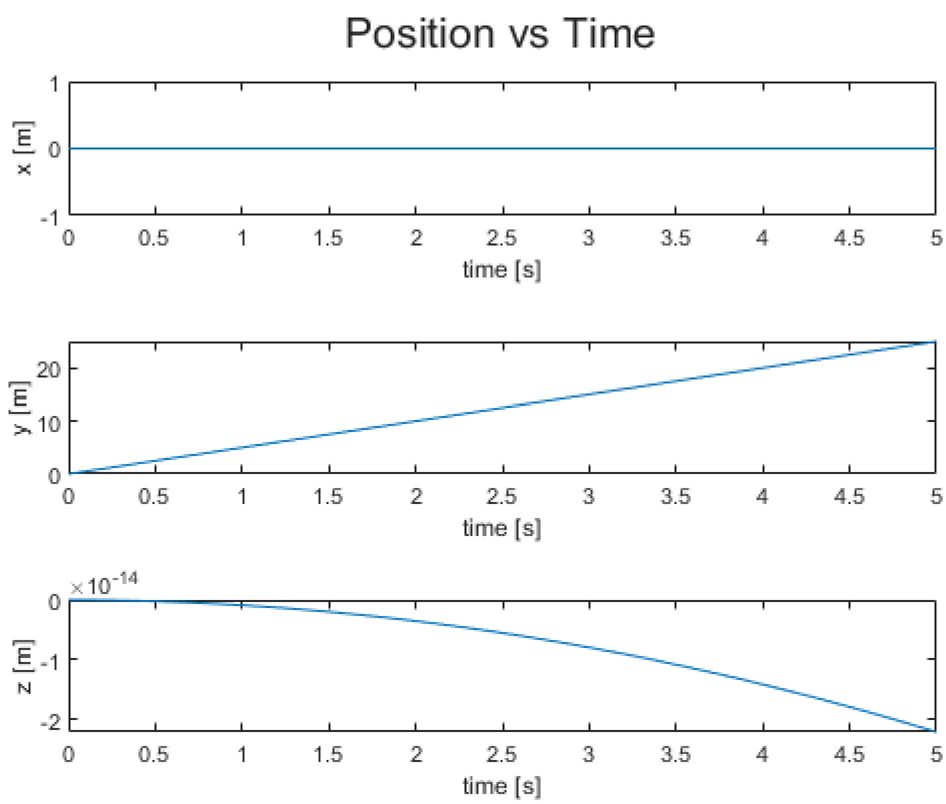

Euler Angles vs Time
roll [deg]
pitch [deg]
yaw [deg]
time [s]
1
0
-1
0 0.5 1 1.5 2 2.5 3 3.5 4 4.5 5

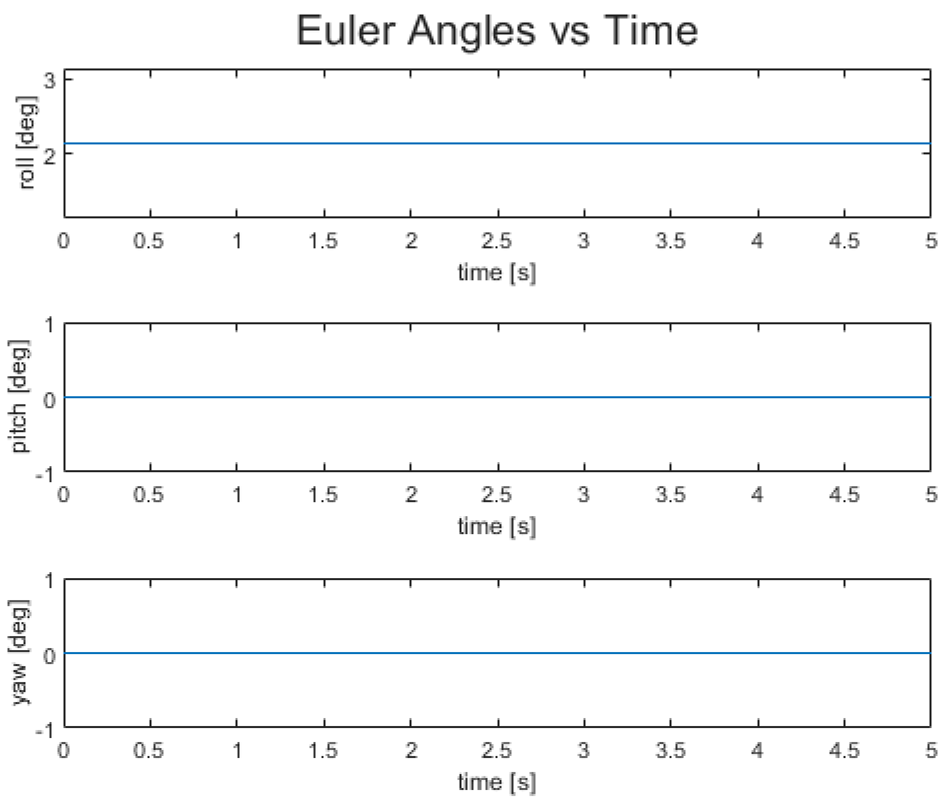

Velocity vs Time
u [m/s]
v [m/s]
w [m/s]
1
0
-1
0 0.5 1 1.5 2 2.5 3 3.5 4 4.5 5
time [s]

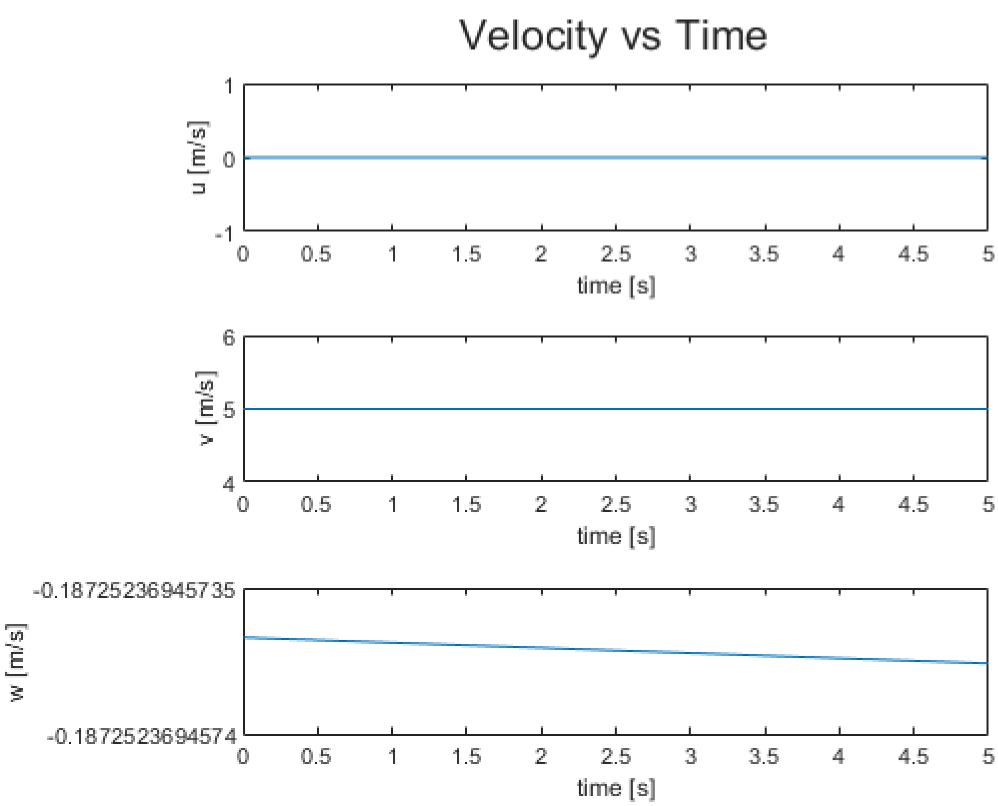

Angular Velocity vs Time
p [deg/s]
q [deg/s]
r [deg/s]
1
0
-1
0 0.5 1 1.5 2 2.5 3 3.5 4 4.5 5
time [s]

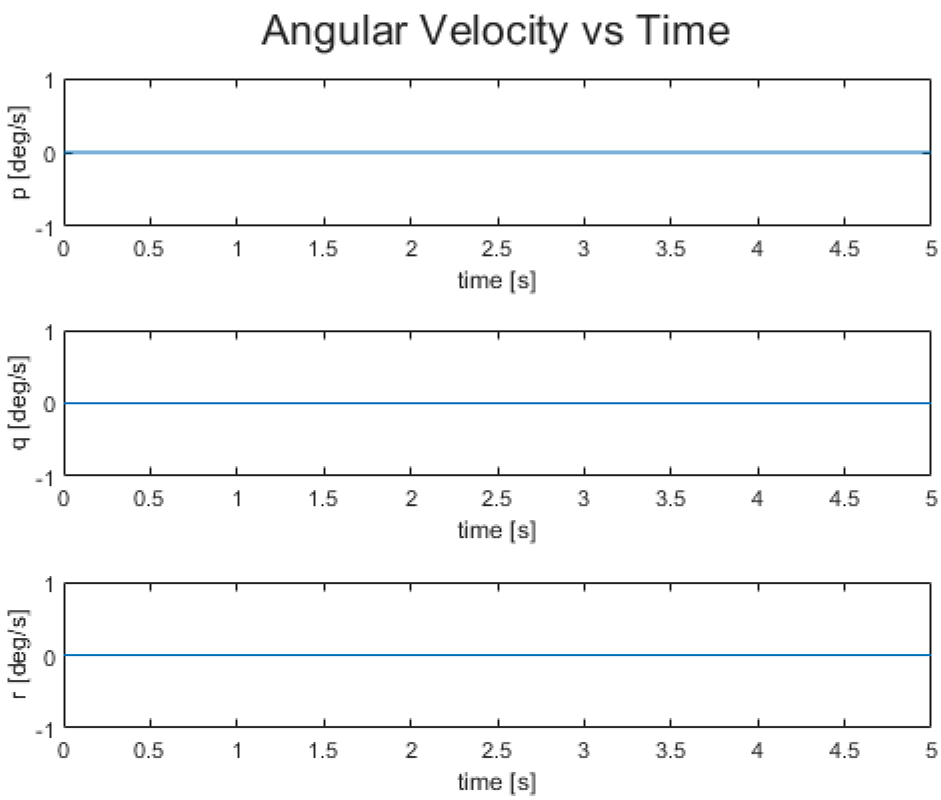
Problem 2b

```
state_0 = [0;0;0;phi;0;0;0;v_air*cos(phi);-
v_air*sin(phi);0;0;0];%Inital condisions of the aircraft
%Caculate angle and power for steady speed.
phi = atan((v_air^2)*nu/(m*g));
Zc = (m*g)/cos(phi);
%f1-f4 is rotor force;
f1 = Zc/4;
f2 = f1;
f3 = f1;
f4 = f1;

constants = [m;r;k_m;I_x;I_y;I_z;nu;mu;f1;f2;f3;f4;g];
%ODE45 Call
tSpan = [0,5];
[t,state] = ode45(@(t,state)EOMQuad(t,state,constants),tSpan,state_0);
%Plots the state.
PlotAircraftSim(t,state,"");
```




Euler Angles vs Time
roll [deg]
pitch [deg]
yaw [deg]
time [s]


Velocity vs Time
u [m/s]
v [m/s]
w [m/s]
-0.18725236945735
-0.18725236945740
time [s]



Problem 2c

```
state_0 = [0;0;0;phi;0;pi/2;0;v_air*cos(phi);-
v_air*sin(phi);0;0;0];%Inital condisions of the aircraft
%Caculate angle and power for steady speed.
phi = atan((v_air^2)*nu/(m*g));
Zc = (m*g)/cos(phi);
%f1-f4 is rotor force;
f1 = Zc/4;
f2 = f1;
f3 = f1;
f4 = f1;

constants = [m;r;k_m;I_x;I_y;I_z;nu;mu;f1;f2;f3;f4;g];
%ODE45 Call
tSpan = [0,5];
[t,state] = ode45(@(t,state)EOMQuad(t,state,constants),tSpan,state_0);
%Plots the state.
PlotAircraftSim(t,state,"");
```

Functions

```
function dsdt = EOMQuad(t,state,constants)
    %Takes in the state of the aircraft and constants about the craft
 to
```

```
    %simulate motion.
    dsdt = zeros(12,1);
    dsdt(1:3,1)=
 BodytoInertial(state(4),state(5),state(6))*state(7:9);

    tempmatrix = [1,(sin(state(4))*tan(state(5))),
(cos(state(4))*tan(state(5)));0,(cos(state(4))),(-sin(state(4)));0,
(sin(state(4))*sec(state(5))),(cos(state(4))*sec(state(5)))];

    dsdt(4:6,1)= tempmatrix*state(10:12);

    vec1 = [((state(12)*state(8))-(state(11)*state(9)));
((state(10)*state(9))-(state(12)*state(7)));((state(11)*state(7))-
(state(10)*state(8)))];
    vec2 = constants(13)*[(-sin(state(5)));
(cos(state(5))*sin(state(4)));(cos(state(5))*cos(state(4)))];
    f_aero = -
constants(7)*(norm(([state(7);state(8);state(9)])))*([state(7);state(8);state(9)])
    vec3 = (1/constants(1))*f_aero;
    vec4 = (1/constants(1))*([0;0;-
(constants(11)+constants(10)+constants(9)+constants(12))]);

    dsdt(7:9,1) = vec1+vec2+vec3+vec4;

    %constants = [m;r;k_m;4 I_x;I_y;I_z;nu;mu;f1;f2;f3;f4;g];
    v1 = [(((constants(5)-constants(6))/
constants(4))*(state(11)*state(12)));(((constants(6)-constants(4))/
constants(5))*(state(10)*state(12)));(((constants(4)-constants(5))/
constants(6))*(state(10)*state(11)))];
    m_aero = -constants(8)*(norm(state(10:12)))*(state(10:12));
    v2 = m_aero./constants(4:6);
    m_cntl = [(constants(2)/sqrt(2))*(-constants(9)-
constants(10)+constants(11)+constants(12));(constants(2)/
sqrt(2))*(constants(9)-constants(10)-constants(11)+constants(12));
(constants(3))*(constants(9)-constants(10)+constants(11)-
constants(12))];
    v3 = m_cntl./constants(4:6);
    dsdt(10:12,1) = v1+v2+v3;


end

function PlotAircraftSim(time, aircraft_state_array, col)


% Plots State Vecotor
figure(1);

tplot1 = tiledlayout(3,1);
title(tplot1,'\fontsize{16}Position vs Time');

nexttile();
plot(time,aircraft_state_array(:,1),col);
xlabel('time [s]');
```

```
ylabel('x [m]');

nexttile();
plot(time,aircraft_state_array(:,2),col);
xlabel('time [s]');
ylabel('y [m]');

nexttile();
plot(time,aircraft_state_array(:,3),col);
xlabel('time [s]');
ylabel('z [m]');

figure(2);

tplot2 = tiledlayout(3,1);
title(tplot2,'\fontsize{16}Euler Angles vs Time');

nexttile();
plot(time,aircraft_state_array(:,4)*(180)/pi,col);
xlabel('time [s]');
ylabel('roll [deg]');

nexttile();
plot(time,aircraft_state_array(:,5)*(180)/pi,col);
xlabel('time [s]');
ylabel('pitch [deg]');

nexttile();
plot(time,aircraft_state_array(:,6)*(180)/pi,col);
xlabel('time [s]');
ylabel('yaw [deg]');

figure(3);

tplot3 = tiledlayout(3,1);
title(tplot3,'\fontsize{16}Velocity vs Time');

nexttile();
plot(time,aircraft_state_array(:,7),col);
xlabel('time [s]');
ylabel('u [m/s]');

nexttile();
plot(time,aircraft_state_array(:,8),col);
xlabel('time [s]');
ylabel('v [m/s]');

nexttile();
plot(time,aircraft_state_array(:,9),col);
xlabel('time [s]');
ylabel('w [m/s]');

figure(4);
```

```
tplot4 = tiledlayout(3,1);
title(tplot4,'\fontsize{16}Angular Velocity vs Time');

nexttile();
plot(time,aircraft_state_array(:,10)*180/pi,col);
xlabel('time [s]');
ylabel('p [deg/s]');

nexttile();
plot(time,aircraft_state_array(:,11)*180/pi,col);
xlabel('time [s]');
ylabel('q [deg/s]');

nexttile();
plot(time,aircraft_state_array(:,12)*180/pi,col);
xlabel('time [s]');
ylabel('r [deg/s]');

end

function matrix = BodytoInertial(phi,theta,psi)

    m1 = [1,0,0;0,cos(phi),sin(phi);0,-sin(phi),cos(phi)];
    m2 = [cos(theta),0,-sin(theta);0,1,0;sin(theta),0,cos(theta)];
    m3 = [cos(psi),sin(psi),0;-sin(psi),cos(psi),0;0,0,1];

    matrix = (m1*m2*m3)';
end
```

Published with MATLAB® R2019b