Table of Contents


```
% ASEN 3128 Lab 3
% Code by Zak Reichenbach
% With help from: Elena Bauer, Chris Nylund, Alex Pichler
```

```
Error using dbstatus
Error: File: C:\Users\runeb\OneDrive\MATLAB\ASEN 3128 Aircraft\Lab
 3\ASEN_3128_Lab3_main.m Line: 785 Column: 15
Function 'controlZc' has already been declared within this scope.
```

House Keeping

```
clc
clear all
close all
```

Constants

```
g = 9.81; %[m/s^2]
m = 0.068; %[kg]
R = 0.06; %[m]
```

```
km = 0.0024; %[N*m/N]
Ix = 6.8*10^(-5); %[kg*m^2]
Iy = 9.2*10^(-5); %[kg*m^2]
Iz = 1.35*10^(-4); %[kg*m^2]
vCoef = 1*10^(-3); %[N/(m/s)^2]
mu = 2*10^(-6); %[N*m/(rad/s)^2]

C = {g m R km Ix Iy Iz vCoef mu};
```

motor forces

```
F = ((m*g)/4);
f1 = F;
f2 = F;
f3 = F;
f4 = F;
```

Problem 1

```
col = 'b-';
% PlotAircraftSim(t,s,control,col)
```

Problem 2

```
%keep for now, only need the actual function later
Z_c = -f1-f2-f3-f4;
L_c = (R/sqrt(2))*(-f1-f2+f3+f4);
M_c = (R/sqrt(2))*(f1-f2-f3+f4);
N_c = km * (f1-f2+f3-f4);

C = [C L_c M_c N_c];

%For plot Overlay
fig_a = [1,2,3,4,5,6]; % figure a number vector
fig_b = [1,2,3,4,5,6] + 6; % figure b number vector
fig_c = [1,2,3,4,5,6] + 2*6; % figure c number vector
fig_d = [1,2,3,4,5,6] + 3*6; % figure d number vector
fig_e = [1,2,3,4,5,6] + 4*6; % figure e number vector
fig_f = [1,2,3,4,5,6] + 5*6; % figure f number vector
fig_5 = [1,2,3,4,5,6] + 6*6;


% call on motor forces function
motor_forces = ComputeMotorForces(Z_c,L_c,M_c,N_c, R, km);

tspan = [0 5];
```

Problem 3

```
A = deg2rad(5);

% %Do 3.a-3.f indiviually
```

```
% 3.a phi = 5deg
% 3.b theta = 5deg
% 3.c psi = 5deg
% 3.d phi_dot = 0.1 rad/s
% 3.e theta_dot = 0.1 rad/s
% 3.f psi_dot = 0.1 rad/s
```

3.a

```
pos_0 = [0 0 -5]; %[m]
ang_0 = [A 0 0];
vel_0 = [0 0 0];
rat_0 = [0 0 0]; %[rad/s]


s_0 = [pos_0 ang_0 vel_0 rat_0];



%Rate of change of ODE at t=0;
odeSim(0,s_0,C);
% Initial func
% C = 12x12 Identity
% D = 0
%

%[t2c,s2c] = ode45(@(t,x) odefun(t,s_0_2c),t_span,s_0_2c);
[t1,s1] = ode45(@(t,x) odeSim(t,x,C),tspan,s_0);

control = '3.a 5deg roll';

PlotAircraftSim(t1,s1,control, col,fig_a,ones(length(t1),4)*Z_c/4)
```

3.b

```
pos_0 = [0 0 -5]; %[m]
ang_0 = [0 A 0];
vel_0 = [0 0 0];
rat_0 = [0 0 0]; %[rad/s]


s_0 = [pos_0 ang_0 vel_0 rat_0];

[t2,s2] = ode45(@(t,x) odeSim(t,x,C),tspan,s_0);

control = '3.b 5deg pitch';

PlotAircraftSim(t2,s2,control, col,fig_b,ones(length(t2),4)*Z_c/4)
```

3.c

```
pos_0 = [0 0 -5]; %[m]
```

```
ang_0 = [0 0 A];
vel_0 = [0 0 0];
rat_0 = [0 0 0]; %[rad/s]


s_0 = [pos_0 ang_0 vel_0 rat_0];

[t3,s3] = ode45(@(t,x) odeSim(t,x,C),tspan,s_0);

control = '3.c 5deg yaw';

PlotAircraftSim(t3,s3,control, col, fig_c,ones(length(t3),4)*Z_c/4)
```

3.d

```
pos_0 = [0 0 -5]; %[m]
ang_0 = [0 0 0];
vel_0 = [0 0 0];
rat_0 = [0.1 0 0]; %[rad/s]


s_0 = [pos_0 ang_0 vel_0 rat_0];

[t4,s4] = ode45(@(t,x) odeSim(t,x,C),tspan,s_0);

control = '3.d 0.1rad/s roll';

PlotAircraftSim(t4,s4,control, col, fig_d,ones(length(t4),4)*Z_c/4)
```

3.e

```
pos_0 = [0 0 -5]; %[m]
ang_0 = [0 0 0];
vel_0 = [0 0 0];
rat_0 = [0 0.1 0]; %[rad/s]


s_0 = [pos_0 ang_0 vel_0 rat_0];

[t5,s5] = ode45(@(t,x) odeSim(t,x,C),tspan,s_0);

control = '3.e 0.1rad/s pitch';

PlotAircraftSim(t5,s5,control, col, fig_e,ones(length(t5),4)*Z_c/4)
```

3.f

```
pos_0 = [0 0 -5]; %[m]
ang_0 = [0 0 0];
vel_0 = [0 0 0];
rat_0 = [0 0 0.1]; %[rad/s]
```

```
s_0 = [pos_0 ang_0 vel_0 rat_0];

[t6,s6] = ode45(@(t,x) odeSim(t,x,C),tspan,s_0);

control = '3.f 0.1rad/s yaw';

PlotAircraftSim(t6,s6,control, col, fig_f,ones(length(t6),4)*Z_c/4)
```

Problem 4

```
tFinal = 5;

col = 'b--';
```

4a

```
pos_0 = [0 0 -5]; %[m]
ang_0 = [A 0 0];
vel_0 = [0 0 0];
rat_0 = [0 0 0]; %[rad/s]

state_vec_0 = [pos_0 ang_0 vel_0 rat_0]';
[state_vec_a,t_a] = odeLinSim(state_vec_0,tFinal);

control = '3.a & 4.a 5deg roll';

PlotAircraftSim(t_a,state_vec_a,control, col,
 fig_a,ones(length(t_a),4)*Z_c/4)
```

4b

```
pos_0 = [0 0 -5]; %[m]
ang_0 = [0 A 0];
vel_0 = [0 0 0];
rat_0 = [0 0 0]; %[rad/s]

state_vec_0 = [pos_0 ang_0 vel_0 rat_0]';
[state_vec_b,t_b] = odeLinSim(state_vec_0,tFinal);

control = '3.b & 4.b 5deg pitch';

PlotAircraftSim(t_b,state_vec_b,control, col,
 fig_b,ones(length(t_b),4)*Z_c/4)
```

4c

```
pos_0 = [0 0 -5]; %[m]
ang_0 = [0 0 A];
vel_0 = [0 0 0];
rat_0 = [0 0 0]; %[rad/s]
```

```
state_vec_0 = [pos_0 ang_0 vel_0 rat_0]';
[state_vec_c,t_c] = odeLinSim(state_vec_0,tFinal);

control = '3.c & 4.c 5deg yaw';

PlotAircraftSim(t_c,state_vec_c,control, col,
 fig_c,ones(length(t_c),4)*Z_c/4)
```

4d

```
%input angles, and change in angles
[p,q,r] = angularRate2Velocity(0,0,0,0.1,0,0);

pos_0 = [0 0 -5]; %[m]
ang_0 = [0 0 0];
vel_0 = [0 0 0];
rat_0 = [p q r]; %[rad/s]

state_vec_0 = [pos_0 ang_0 vel_0 rat_0]';
[state_vec_c,t_d] = odeLinSim(state_vec_0,tFinal);

control = '3.d & 4.d 0.1rad/s roll';

PlotAircraftSim(t_d,state_vec_c,control, col,
 fig_d,ones(length(t_d),4)*Z_c/4)
```

4e

```
%input angles, and change in angles
[p,q,r] = angularRate2Velocity(0,0,0,0,0.1,0);

pos_0 = [0 0 -5]; %[m]
ang_0 = [0 0 0];
vel_0 = [0 0 0];
rat_0 = [p q r]; %[rad/s]

state_vec_0 = [pos_0 ang_0 vel_0 rat_0]';
[state_vec_c,t_e] = odeLinSim(state_vec_0,tFinal);

control = '3.e & 4.e 0.1rad/s pitch';

PlotAircraftSim(t_e,state_vec_c,control, col,
 fig_e,ones(length(t_e),4)*Z_c/4)
```

4f

```
%input angles, and change in angles
[p,q,r] = angularRate2Velocity(0,0,0,0,0,0.1);

pos_0 = [0 0 -5]; %[m]
ang_0 = [0 0 0];
vel_0 = [0 0 0];
```

```
rat_0 = [p q r]; %[rad/s]

state_vec_0 = [pos_0 ang_0 vel_0 rat_0]';
[state_vec_c,t_f] = odeLinSim(state_vec_0,tFinal);

control = '3.f & 4.f 0.1rad/s yaw';

PlotAircraftSim(t_f,state_vec_c,control, col,
 fig_f,ones(length(t_f),4)*Z_c/4)
```

Plot Labeling

```
for N = 1:6
    figure(6*N-5)
    legend('Non-Linear Path','Linearized Path','location','best');
    legend('boxoff');

    figure(6*N-4)
    legend('Non-Linear Path','Linearized Path','location','best');
    legend('boxoff');

    figure(6*N-3)
    legend('Non-Linear Path','Linearized Path','location','best');
    legend('boxoff');

    figure(6*N-2)
    legend('Non-Linear Path','Linearized Path','location','best');
    legend('boxoff');

    figure(6*N-1)
    legend('Non-Linear Path','Linearized Path','location','best');
    legend('boxoff');

    figure(6*N)
    legend('Non-Linear Path','Start','End','Linearized Path');
end
```

5d

```
[p,q,r] = angularRate2Velocity(0,0,0,0.1,0,0);


pos_0 = [0 0 -5]; %[m]
ang_0 = [0 0 0];
vel_0 = [0 0 0];
rat_0 = [p q r]; %[rad/s]
stuff = [m*g -0.004*p -0.004*q -0.004*r];

state_vec_0 = [pos_0 ang_0 vel_0 rat_0 stuff]';

[t5d,s5d] = ode45(@(t,x) odeSimControlled(t,x,C),tspan,state_vec_0);

control = '5.d 0.1rad/s roll';
```

```
PlotAircraftSimOrg(t5d,s5d,control, col)


%function motor_forces = ComputeMotorForces(Z_c,L_c,M_c,N_c, R, km)
% compute forces
```

5e

```
[p,q,r] = angularRate2Velocity(0,0,0,0,0.1,0);


pos_0 = [0 0 -5]; %[m]
ang_0 = [0 0 0];
vel_0 = [0 0 0];
rat_0 = [p q r]; %[rad/s]
stuff = [m*g -0.004*p -0.004*q -0.004*r];

state_vec_0 = [pos_0 ang_0 vel_0 rat_0 stuff]';

[t5e,s5e] = ode45(@(t,x) odeSimControlled(t,x,C),tspan,state_vec_0);

control = '5.e 0.1rad/s pitch';

PlotAircraftSimOrg(t5e,s5e,control, col)


%function motor_forces = ComputeMotorForces(Z_c,L_c,M_c,N_c, R, km)
% compute forces
```

5f

```
[p,q,r] = angularRate2Velocity(0,0,0,0,0,0.1);


pos_0 = [0 0 -5]; %[m]
ang_0 = [0 0 0];
vel_0 = [0 0 0];
rat_0 = [p q r]; %[rad/s]
stuff = [m*g -0.004*p -0.004*q -0.004*r];

state_vec_0 = [pos_0 ang_0 vel_0 rat_0 stuff]';

[t5f,s5f] = ode45(@(t,x) odeSimControlled(t,x,C),tspan,state_vec_0);

control = '5.f 0.1rad/s yaw';

PlotAircraftSimOrg(t5f,s5f,control, col)


%function motor_forces = ComputeMotorForces(Z_c,L_c,M_c,N_c, R, km)
% compute forces
```

```
hold off


for i = 1:length(s5d)
    F_5d(i,:) =
 ComputeMotorForces(s5d(i,13),s5d(i,14),s5d(i,15),s5d(i,16),R,km);

end

figure()
subplot(3,1,1)
plot(t5d,F_5d(:,1),'o','linewidth',1); hold on;
plot(t5d,F_5d(:,2),'-','linewidth',1); hold on;
plot(t5d,F_5d(:,3),'o','linewidth',1); hold on;
plot(t5d,F_5d(:,4),'-','linewidth',1); hold on;
title(['p = ' num2str(p)]);
legend('motor 1','motor 2','motor 3','motor 4');
hold on;

for i = 1:length(s5e)
    F_5e(i,:) =
 ComputeMotorForces(s5e(i,13),s5e(i,14),s5e(i,15),s5e(i,16),R,km);

end

subplot(3,1,2)
plot(t5e,F_5e(:,1),'o','linewidth',1); hold on;
plot(t5e,F_5e(:,2),'-','linewidth',1); hold on;
plot(t5e,F_5e(:,3),'o','linewidth',1); hold on;
plot(t5e,F_5e(:,4),'-','linewidth',1); hold on;
title(['q = ' num2str(q)]);
legend('motor 1','motor 2','motor 3','motor 4');
hold on;

for i = 1:length(s5f)
    F_5f(i,:) =
 ComputeMotorForces(s5f(i,13),s5f(i,14),s5f(i,15),s5f(i,16),R,km);

end

subplot(3,1,3)
plot(t5f,F_5f(:,1),'o','linewidth',1); hold on;
plot(t5f,F_5f(:,2),'-','linewidth',1); hold on;
plot(t5f,F_5f(:,3),'o','linewidth',1); hold on;
plot(t5f,F_5f(:,4),'-','linewidth',1); hold on;
title(['r = ' num2str(r)]);
legend('motor 1','motor 2','motor 3','motor 4');
hold on;
sgtitle(sprintf('Motor Control Forces Cases D-F'))

hold off;

% Functions
```

```
function PlotAircraftSim(t,s,control, col,fig,Zc)

    figure( fig(1))    %Position
    subplot(3,1,1), plot(t,s(:,1),col); hold on;
    title('X Position (North +)');
    xlabel('Time [seconds]');
    ylabel('X Position [m]');
    subplot(3,1,2), plot(t,s(:,2),col); hold on;
    title('Y Position (East +)');
    xlabel('Time [seconds]');
    ylabel('Y Position [m]');
    subplot(3,1,3), plot(t,s(:,3),col); hold on;
    title('Z Position (Down +)');
    xlabel('Time [seconds]');
    ylabel('Z Position [m]');
    sgtitle(sprintf('Position of %s',control))

    figure(fig(2))    %Angular Position
    subplot(3,1,1), plot(t,s(:,4),col); hold on;
    title('Roll(\phi)');
    xlabel('Time [seconds]');
    ylabel('Roll(\phi) [rad]');
    subplot(3,1,2), plot(t,s(:,5),col); hold on;
    title('Pitch(\theta)');
    xlabel('Time [seconds]');
    ylabel('Pitch(\theta) [rad]');
    subplot(3,1,3), plot(t,s(:,6),col); hold on;
    title('Yaw(\psi)');
    xlabel('Time [seconds]');
    ylabel('Yaw(\psi)Position [rad]');
    sgtitle(sprintf('Angular Position of %s',control))

    figure(fig(3))    %Linear Velocity
    subplot(3,1,1), plot(t,s(:,7),col); hold on;
    title('U Inertial Velocity (North +)');
    xlabel('Time [seconds]');
    ylabel('U Inertial Velocity [m/s]');
    subplot(3,1,2), plot(t,s(:,8),col); hold on;
    title('V Velocity (East +)');
    xlabel('Time [seconds]');
    ylabel('V Inertial Velocity [m/s]');
    subplot(3,1,3), plot(t,s(:,9),col); hold on;
    title('W Velocity (Down +)');
    xlabel('Time [seconds]');
    ylabel('W Inertial Velocity [m/s]');
    sgtitle(sprintf('Linear Velocity of %s',control))

    figure(fig(4))    %Angular Velocity
    subplot(3,1,1), plot(t,s(:,10),col); hold on;
    title('p Angular Velocity');
    xlabel('Time [seconds]');
    ylabel('q Angular Velocity [rad/s]');
    subplot(3,1,2), plot(t,s(:,11),col); hold on;
```

```
    title('p Angular Velocity');
    xlabel('Time [seconds]');
    ylabel('p Angular Velocity [rad/s]');
    subplot(3,1,3), plot(t,s(:,12),col); hold on;
    title('r Angular Velocity');
    xlabel('Time [seconds]');
    ylabel('r Angular Velocity [rad/s]');
 sgtitle(sprintf('Angular Velocity of %s',control))


 % Plot control forces and moments vs time
    figure(fig(5))
    subplot(2,2,1)
    plot(t,Zc(:,1),col); hold on;
    grid on
    title([control ' Control Force vs. time']);
    ylabel('Control Force');
    xlim([t(1) t(end)]);
    subplot(2,2,2)
    plot(t,Zc(:,2),col,'linewidth',2); hold on;
    grid on
    title([control ' x Control Moment vs. time']);
    ylabel('x Control Moment');
    xlim([t(1) t(end)]);
    subplot(2,2,3)
    plot(t,Zc(:,3),col,'linewidth',2); hold on;
    grid on
    title([control ' y Control Moment vs. time']);
    ylabel('y Control Moment');
    xlabel('time [s]');
    xlim([t(1) t(end)]);
    subplot(2,2,4)
    plot(t,Zc(:,4),col,'linewidth',2); hold on;
    grid on
    title([control ' z Control Moment vs. time']);
    xlabel('time [s]');
    ylabel('z Control Moment');
    xlim([t(1) t(end)]);

    % Plot flight path of the simulated quadrotor
    figure(fig(6))
    plot3(s(:,1),s(:,2),s(:,3),col,'linewidth',2); hold on;
    plot3(s(1,1),s(1,2),s(1,3),'g.','markersize',20); hold on;
    plot3(s(end,1),s(end,2),s(end,3),'r.','markersize',20); hold on;
    set(gca, 'YDir','reverse')
    set(gca, 'ZDir','reverse')
    grid on
    title([control ' Flight Path']);
    xlabel('x');
    ylabel('y');
    zlabel('z');


end
```

```
function PlotAircraftSimOrg(t,s,control, col)

    figure( )    %Position
    subplot(3,1,1), plot(t,s(:,1),col); hold on;
    title('X Position (North +)');
    xlabel('Time [seconds]');
    ylabel('X Position [m]');
    subplot(3,1,2), plot(t,s(:,2),col); hold on;
    title('Y Position (East +)');
    xlabel('Time [seconds]');
    ylabel('Y Position [m]');
    subplot(3,1,3), plot(t,s(:,3),col); hold on;
    title('Z Position (Down +)');
    xlabel('Time [seconds]');
    ylabel('Z Position [m]');
    sgtitle(sprintf('Position of %s',control))

    figure()    %Angular Position
    subplot(3,1,1), plot(t,s(:,4),col); hold on;
    title('Roll(\phi)');
    xlabel('Time [seconds]');
    ylabel('Roll(\phi) [rad]');
    subplot(3,1,2), plot(t,s(:,5),col); hold on;
    title('Pitch(\theta)');
    xlabel('Time [seconds]');
    ylabel('Pitch(\theta) [rad]');
    subplot(3,1,3), plot(t,s(:,6),col); hold on;
    title('Yaw(\psi)');
    xlabel('Time [seconds]');
    ylabel('Yaw(\psi)Position [rad]');
    sgtitle(sprintf('Angular Position of %s',control))

    figure()    %Linear Velocity
    subplot(3,1,1), plot(t,s(:,7),col); hold on;
    title('U Inertial Velocity (North +)');
    xlabel('Time [seconds]');
    ylabel('U Inertial Velocity [m/s]');
    subplot(3,1,2), plot(t,s(:,8),col); hold on;
    title('V Velocity (East +)');
    xlabel('Time [seconds]');
    ylabel('V Inertial Velocity [m/s]');
    subplot(3,1,3), plot(t,s(:,9),col); hold on;
    title('W Velocity (Down +)');
    xlabel('Time [seconds]');
    ylabel('W Inertial Velocity [m/s]');
    sgtitle(sprintf('Linear Velocity of %s',control))

    figure()    %Angular Velocity
    subplot(3,1,1), plot(t,s(:,10),col); hold on;
    title('p Angular Velocity');
    xlabel('Time [seconds]');
    ylabel('q Angular Velocity [rad/s]');
```

```
    subplot(3,1,2), plot(t,s(:,11),col); hold on;
    title('p Angular Velocity');
    xlabel('Time [seconds]');
    ylabel('p Angular Velocity [rad/s]');
    subplot(3,1,3), plot(t,s(:,12),col); hold on;
    title('r Angular Velocity');
    xlabel('Time [seconds]');
    ylabel('r Angular Velocity [rad/s]');
 sgtitle(sprintf('Angular Velocity of %s',control))

     % Plot flight path of the simulated quadrotor
    figure()
    plot3(s(:,1),s(:,2),s(:,3),col,'linewidth',2); hold on;
    plot3(s(1,1),s(1,2),s(1,3),'g.','markersize',20); hold on;
    plot3(s(end,1),s(end,2),s(end,3),'r.','markersize',20); hold on;
    set(gca, 'YDir','reverse')
    set(gca, 'ZDir','reverse')
    grid on
    title([control ' Flight Path']);
    xlabel('x');
    ylabel('y');
    zlabel('z');



end



% ASEN 3128 Lab 3
%
% function to calculate motor forces using control moments
%

function motor_forces = ComputeMotorForces(Z_c,L_c,M_c,N_c, R, km)

%motor_forces = [f1,f2,f3,f4]
motor_forces = [-1,-1,-1,-1;...
                -R/sqrt(2),-R/sqrt(2),R/sqrt(2),R/sqrt(2);...
                R/sqrt(2),-R/sqrt(2),-R/sqrt(2),R/sqrt(2);...
                km,-km,km,-km]\[Z_c;L_c;M_c;N_c];

end
```

Non-Linear ODE Func

```
function x = odeSim(t,s,C)
g = C{1}; m = C{2}; R = C{3}; km = C{4}; Ix = C{5}; Iy = C{6}; Iz =
 C{7}; vCoef = C{8}; mu = C{9}; Lc = C{10}; Mc = C{11}; Nc = C{12};



   %S Vector x y z phi theta psi u v w p q r
```

```
    [X,Y,Z] = aeroDragForce(vCoef,s(7),s(8),s(9));
    [L,M,N] = aeroMoments(s(10),s(11),s(12),mu);

    Zc = controlZc(m,g);

    eq_1 = firstEq(s(4),s(5),s(6),s(7),s(8),s(9));
    eq_2 = secondEq(s(4),s(5),s(6),s(10),s(11),s(12));
    eq_3 =
  thirdEq(s(4),s(5),s(6),s(10),s(11),s(12),s(7),s(8),s(9),X,Y,Z,Zc,g,m);
    eq_4 = fourthEq(Ix,Iy,Iz,s(10),s(11),s(12),L,M,N,Lc,Mc,Nc);



    x = [eq_1;eq_2;eq_3;eq_4];
end

function Zc = controlZc(m,g)
    Zc = m*g;
end

function [X,Y,Z] = aeroDragForce(vCoef,u,v,w)
    X = -vCoef*sqrt(u^2+v^2+w^2)*u;
    Y = -vCoef*sqrt(u^2+v^2+w^2)*v;
    Z = -vCoef*sqrt(u^2+v^2+w^2)*w;
    %out = vCoef*(u^2+v^2+w^2)*[u;v;w];
end

function [L,M,N] = aeroMoments(p,q,r,mu)
    L = -mu*(sqrt(p^2+q^2+r^2))*p;
    M = -mu*(sqrt(p^2+q^2+r^2))*q;
    N = -mu*(sqrt(p^2+q^2+r^2))*r;
    %out = -mu*(sqrt(p^2+q^2+r^2))*[p;q;r];
end

function out = firstEq(phi,theta,psi,uE,vE,wE)
%Linear Velocities

    velocities = [uE;vE;wE];

    A = [cos(theta)*cos(psi) sin(phi)*sin(theta)*cos(psi)-
cos(phi)*sin(psi) cos(phi)*sin(theta)*cos(psi)+sin(phi)*sin(psi);...
        cos(theta)*sin(psi)
 sin(phi)*sin(theta)*sin(psi)+cos(phi)*cos(psi)
 cos(phi)*sin(theta)*sin(psi)-sin(phi)*cos(psi);...
        -sin(theta) sin(phi)*cos(theta) cos(phi)*cos(theta);];

    out = A*[uE;vE;wE];
end

function out = secondEq(phi,theta,psi,p,q,r)
%Angular Rates
    A = [1 sin(phi)*tan(theta) cos(phi)*tan(theta);...
        0 cos(phi) -sin(phi);...
```

```
        0 sin(phi)*sec(theta) cos(phi)*sec(theta)];
    out = A*[p;q;r];
end

function out = thirdEq (phi,theta,psi,p,q,r,uE,vE,wE,X,Y,Z,Zc,g,m)
%Acceleration
    A = [r*vE - q*wE; p*wE - r*uE; q*uE - p*vE]; %Angular Part
    B = g*[-sin(theta); cos(theta)*sin(phi); -
cos(theta)*cos(phi)]; %Gravity Part
    C = [X; Y; Z]/m;                     %Aero Force Part
    D = [0; 0; Zc]/m;   %Control Force Part
    out = A + B + C + D;
%     out(3,1) = 0;
end

function out = fourthEq(Ix,Iy,Iz,p,q,r,L,M,N,Lc,Mc,Nc)
%Angular Accel. w/ moments
    A = [((Iy-Iz)/Ix)*q*r; ((Iz-Ix)/Iy)*p*r; ((Ix-Iy)/Iz)*p*q];
    B = [L/Ix; M/Iy; N/Iz];
    C = [Lc/Ix; Mc/Iy; Nc/Iz];
    out = A + B + C;
end
```

Linear Func

```
function [x,t] = odeLinSim(x0,tf)
 g = 9.81; % [m/s]

    % Define state space matrix
    A = zeros(12,12);
    A(1,7) = 1;
    A(2,8) = 1;
    A(3,9) = 1;
    A(4,10) = 1;
    A(5,11) = 1;
    A(6,12) = 1;
    A(7,5) = -g;
    A(8,4) = g;

    % Define state space system
    sys = ss(A,zeros(12,1),eye(12),0);

    % Simulate state space system
    [x,t] = initial(sys,x0,tf);

end

function [p,q,r] =
 angularRate2Velocity(phi,theta,psi,phi_dot,theta_dot,psi_dot)
```

EOM 2 for angular motion

```
    A = [1 sin(phi)*tan(theta) cos(phi)*tan(theta);...
```

```
        0 cos(phi) -sin(phi);...
        0 sin(phi)*sec(theta) cos(phi)*sec(theta)];
    out = A*[phi_dot;theta_dot;psi_dot];

    p = out(1);
    q = out(2);
    r = out(3);

end
```

Controlled Non-Linear Func

```
function x = odeSimControlled(t,s,C)
g = C{1}; m = C{2}; R = C{3}; km = C{4}; Ix = C{5}; Iy = C{6}; Iz =
 C{7}; vCoef = C{8}; mu = C{9}; Lc = C{10}; Mc = C{11}; Nc = C{12};



   %S Vector x y z phi theta psi u v w p q r

   [X,Y,Z] = aeroDragForce(vCoef,s(7),s(8),s(9));
   [L,M,N] = aeroMoments(s(10),s(11),s(12),mu);
   Mc = -0.004*[s(10);s(11);s(12)]; % control moment

   Zc = controlZc(m,g);

   eq_1 = firstEq(s(4),s(5),s(6),s(7),s(8),s(9));
   eq_2 = secondEq(s(4),s(5),s(6),s(10),s(11),s(12));

   eq_3 =
 thirdEq(s(4),s(5),s(6),s(10),s(11),s(12),s(7),s(8),s(9),X,Y,Z,Zc,g,m);
   eq_4 =
 fourthEq(Ix,Iy,Iz,s(10),s(11),s(12),L,M,N,Mc(1),Mc(2),Mc(3));


   deltaF = [0 ; -0.004 * eq_4];





   x = [eq_1;eq_2;eq_3;eq_4;deltaF];
end

function Zc = controlZc(m,g)
    Zc = m*g;
end

function [X,Y,Z] = aeroDragForce(vCoef,u,v,w)
    X = -vCoef*sqrt(u^2+v^2+w^2)*u;
    Y = -vCoef*sqrt(u^2+v^2+w^2)*v;
    Z = -vCoef*sqrt(u^2+v^2+w^2)*w;
    %out = vCoef*(u^2+v^2+w^2)*[u;v;w];
```

```
end

function [L,M,N] = aeroMoments(p,q,r,mu)
    L = -mu*(sqrt(p^2+q^2+r^2))*p;
    M = -mu*(sqrt(p^2+q^2+r^2))*q;
    N = -mu*(sqrt(p^2+q^2+r^2))*r;
    %out = -mu*(sqrt(p^2+q^2+r^2))*[p;q;r];
end

function out = firstEq(phi,theta,psi,uE,vE,wE)
%Linear Velocities

    velocities = [uE;vE;wE];

    A = [cos(theta)*cos(psi) sin(phi)*sin(theta)*cos(psi)-
cos(phi)*sin(psi) cos(phi)*sin(theta)*cos(psi)+sin(phi)*sin(psi);...
        cos(theta)*sin(psi)
 sin(phi)*sin(theta)*sin(psi)+cos(phi)*cos(psi)
 cos(phi)*sin(theta)*sin(psi)-sin(phi)*cos(psi);...
        -sin(theta) sin(phi)*cos(theta) cos(phi)*cos(theta);];

    out = A*[uE;vE;wE];
end

function out = secondEq(phi,theta,psi,p,q,r)
%Angular Rates
    A = [1 sin(phi)*tan(theta) cos(phi)*tan(theta);...
        0 cos(phi) -sin(phi);...
        0 sin(phi)*sec(theta) cos(phi)*sec(theta)];
    out = A*[p;q;r];
end

function out = thirdEq (phi,theta,psi,p,q,r,uE,vE,wE,X,Y,Z,Zc,g,m)
%Acceleration
    A = [r*vE - q*wE; p*wE - r*uE; q*uE - p*vE]; %Angular Part
    B = g*[-sin(theta); cos(theta)*sin(phi); -
cos(theta)*cos(phi)]; %Gravity Part
    C = [X; Y; Z]/m;                     %Aero Force Part
    D = [0; 0; Zc]/m;   %Control Force Part
    out = A + B + C + D;
%     out(3,1) = 0;
end

function out = fourthEq(Ix,Iy,Iz,p,q,r,L,M,N,Lc,Mc,Nc)
%Angular Accel. w/ moments
    A = [((Iy-Iz)/Ix)*q*r; ((Iz-Ix)/Iy)*p*r; ((Ix-Iy)/Iz)*p*q];
    B = [L/Ix; M/Iy; N/Iz];
    C = [Lc/Ix; Mc/Iy; Nc/Iz];
    out = A + B + C;

end
```

Published with MATLAB® R2021a