Table of Contents

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%
% Author: Samuel Razumovskiy
% Date written: 11/14/19
% Date modified: 11/21/19
%
% Purpose: Finding the eigenvalues of a 737 and observing its lateral
% flight characteristics
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%

clear,clc,close all
```

Question 1

Defining all the given parameters

```
exz = -6.8*pi/180;
Ix = 2.46767e7;
Iy = 4.488e7;
Iz = 6.7384e7;
Izx = 1.315e6;
m = 6.366e5*4.448/9.81;
u0 = 157.9;
S = 5500*0.3048^2;
b = 195.68*0.3048;
c = 27.31*0.3048;
g = 9.81;
rho = 0.66011;

Ixs = Ix*cos(exz)^2+Iz*sin(exz)^2+Izx*sin(2*exz);
Izs = Ix*sin(exz)^2+Iz*cos(exz)^2-Izx*sin(2*exz);
Izxs = -.5*(Ix-Iz)*sin(2*exz)-Izx*sin(sin(exz)^2-cos(exz)^2);

% Making a matrix of the coefficients
Coeff = [-0.8771, -0.2797,   0.1946;...
               0, -0.3295, -0.04073;...
               0,   0.304,  -0.2737];

% Making a matrix of the conversion factors
```

```
Conv = [1/2*rho*u0*S,1/2*rho*u0*b*S,1/2*rho*u0*b*S;...
    1/4*rho*u0*b*S,1/4*rho*u0*b^2*S,1/4*rho*u0*b^2*S;...
    1/4*rho*u0*b*S,1/4*rho*u0*b^2*S,1/4*rho*u0*b^2*S];

Dim= Conv.*Coeff;

% Dimensionalized stability derivatives
Yv = Dim(1,1); Lv = Dim(1,2); Nv = Dim(1,3);
Yp = Dim(2,1); Lp = Dim(2,2); Np = Dim(2,3);
Yr = Dim(3,1); Lr = Dim(3,2); Nr = Dim(3,3);

% Converting the Stability derivatives to Stability frame
Yvp = Yv;
Ypp = Yp*cos(exz)-Yr*sin(exz);
Yrp = Yr*cos(exz)+Yp*sin(exz);
Lvp = Lv*cos(exz)-Nv*sin(exz);
Lpp = Lp*cos(exz)^2-(Lr+Np)*sin(exz)*cos(exz)+Nr*sin(exz)^2;
Lrp = Lr*cos(exz)^2-(Nr-Lp)*sin(exz)*cos(exz)-Np*sin(exz)^2;
Nvp = Nv*cos(exz)+Lv*sin(exz);
Npp = Np*cos(exz)^2-(Nr-Lp)*sin(exz)*cos(exz)-Lr*sin(exz)^2;
Nrp = Nr*cos(exz)^2+(Lr+Np)*sin(exz)*cos(exz)+Lp*sin(exz)^2;
```

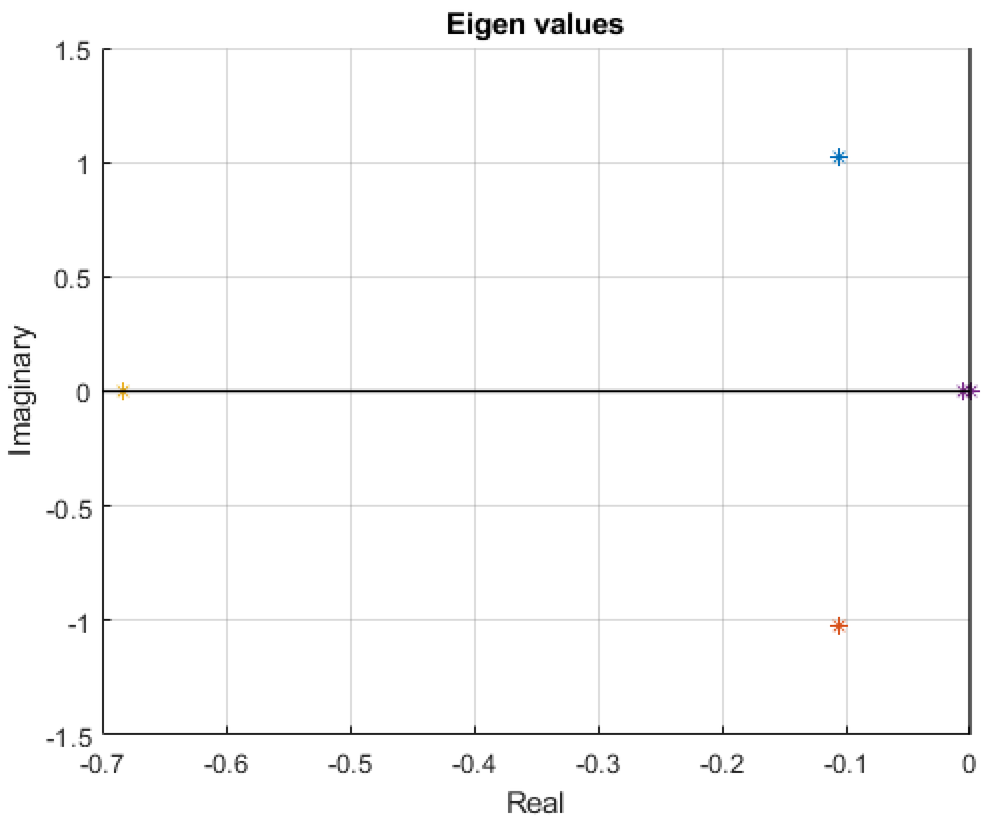
Question 2

```
Ixp = (Ixs*Izs-Izxs^2)/Izs;
Izp = (Ixs*Izs-Izxs^2)/Ixs;
Izxp = Izxs/(Ixs*Izs-Izxs^2);

% A matrix
A = [        Yvp/m,           Ypp/m,        Yr/m-u0, g;...
    Lvp/Ixp+Izxp*Nvp, Lpp/Ixp+Izxp*Npp, Lrp/Ixp+Izxp*Nrp, 0;...
    Izxp*Lvp+Nvp/Izp, Izxp*Lpp+Npp/Izp, Izxp*Lrp+Nrp/Izp, 0;...
                0,               1,               0, 0];
```

Question 3

```
[V,D] = eig(A);

Re = real(D);
Im = imag(D);
figure
hold on
grid on
plot(Re,Im,'*')
title('Eigen values')
xlabel('Real')
ylabel('Imaginary')
line(xlim(), [0,0], 'LineWidth', 1, 'Color', 'k');
line([0,0], ylim(), 'LineWidth', 1, 'Color', 'k');

tau = -1./Re;

% Dutch roll Damping and Natural frequency
```

```
z = sqrt(1/(1+(Im(1)/Re(1))^2));
wn= -Re(1)/z;
% All Stable
```



Question 4

Approximated eigenvalues for Dutch Roll

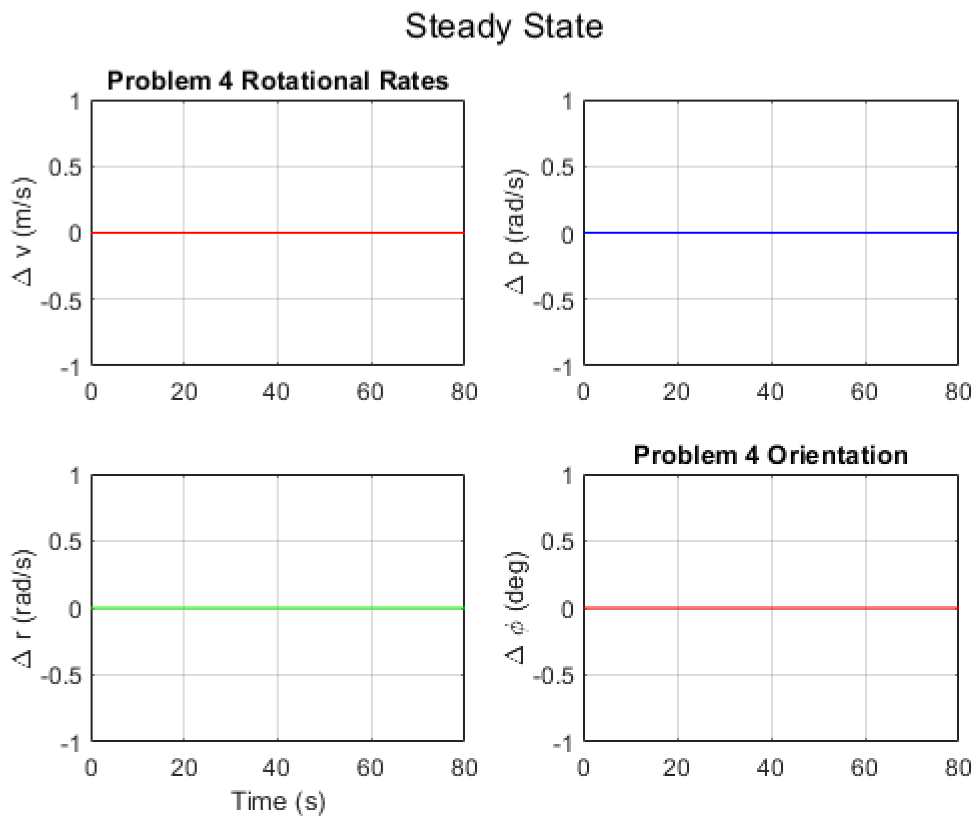
```
approx = roots([1,-(A(1)+A(3,3)), A(1)*A(3,3)+u0*A(3,1)]);
diffRe = abs((Re(1)-real(approx(1)))/Re(1))*100;
diffIm = abs((Im(1)-imag(approx(1)))/Im(1))*100;
fprintf('Relative error of the real parts = %.3f%%\n',diffRe)
fprintf('Relative error of the Imaginary parts = %.3f%%\n\n',diffIm)
```

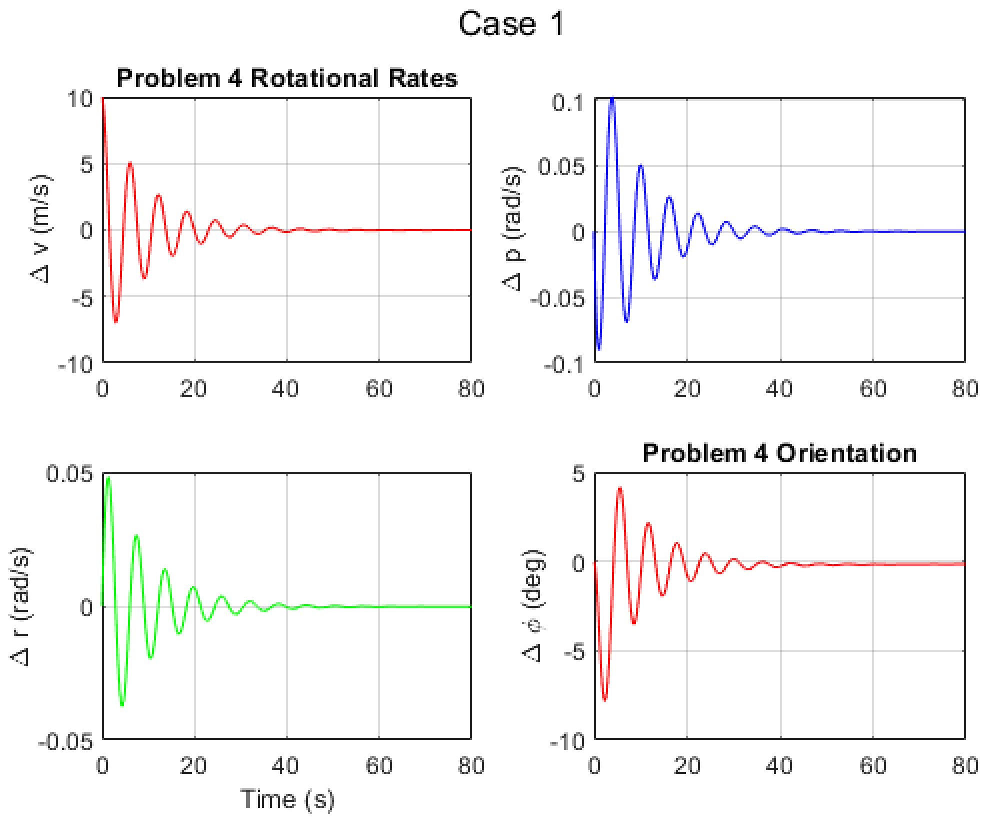
```
Relative error of the real parts = 58.035%
Relative error of the Imaginary parts = 2.533%
```

Question 5

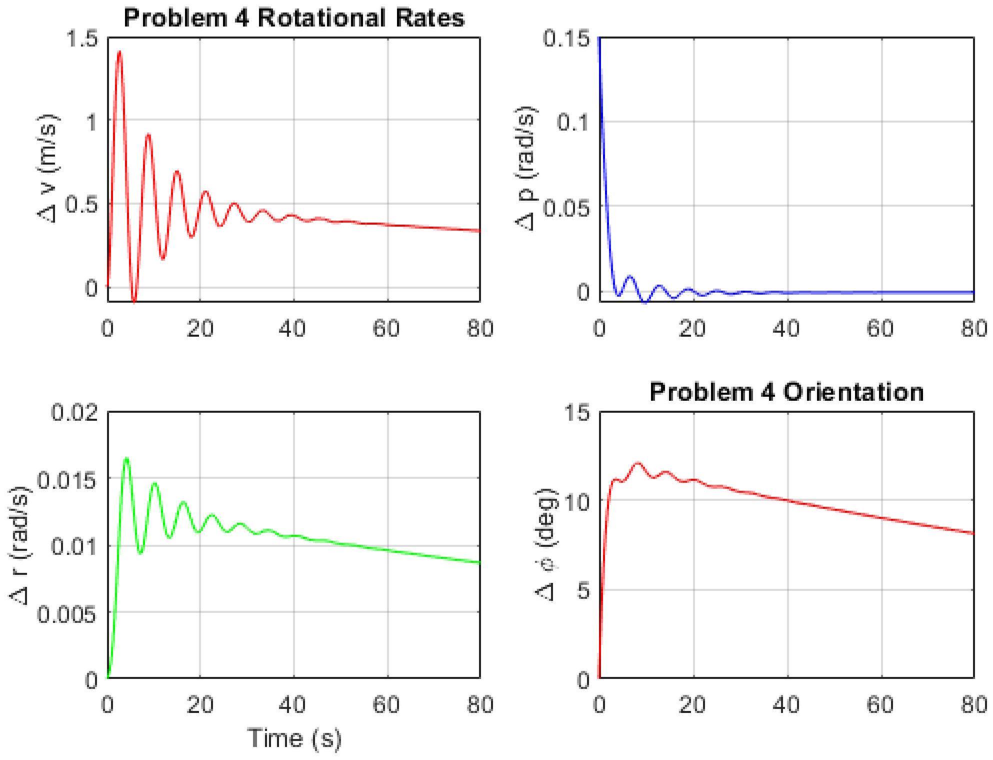
```
close all

dv =  [0,10,    0, -1.8563,  2.9477];
dp =  [0, 0, 0.15, -0.4185, -0.0063];
dr =  [0, 0,    0,  0.0311,  0.0758];
dphi= [0, 0,    0,  0.6148,  1.2431];
```

```matlab
changes = [dv',dp',dr',dphi'];
names = ["Steady State","Case 1","Case 2","Case 3","Case 4"];


for i = 1:5
    CH = changes(i,:);
    tspan = [0 80];

    pos_N = 0; pos_E = 0; pos_D = 0; % m E frame
    u = 0; v = CH(1); w = 0; % m/s B frame
    y_trans = [pos_N; pos_E; pos_D; u; v; w];

    psi = 0; theta = 0; phi = CH(4); % rad
    p = CH(2); q = 0; r = CH(3); % rad/s B frame
    y_rot = [psi; theta; phi; p; q; r];

    y = [y_trans;y_rot];
    opt = odeset('maxstep',1);

    [t,pos] = ode45(@(t,y)linearized_Aircraft_ODE(t,y,A),tspan,y,opt);

    figure
    sgtitle(names(i))
    subplot(2,2,1)
    plot(t,pos(:,5),'r')
    title("Problem 4 Rotational Rates")
    ylabel('\Delta v (m/s)')
    grid on

    subplot(2,2,2)
    plot(t,pos(:,10),'b')
    ylabel('\Delta p (rad/s)')
    grid on

    subplot(2,2,3)
    plot(t,pos(:,12),'g')
    ylabel('\Delta r (rad/s)')
    xlabel('Time (s)')
    grid on

    subplot(2,2,4)
    plot(t,pos(:,9).*180./pi,'r')
    title("Problem 4 Orientation")
    ylabel("\Delta \phi (deg)")
    grid on

end
```

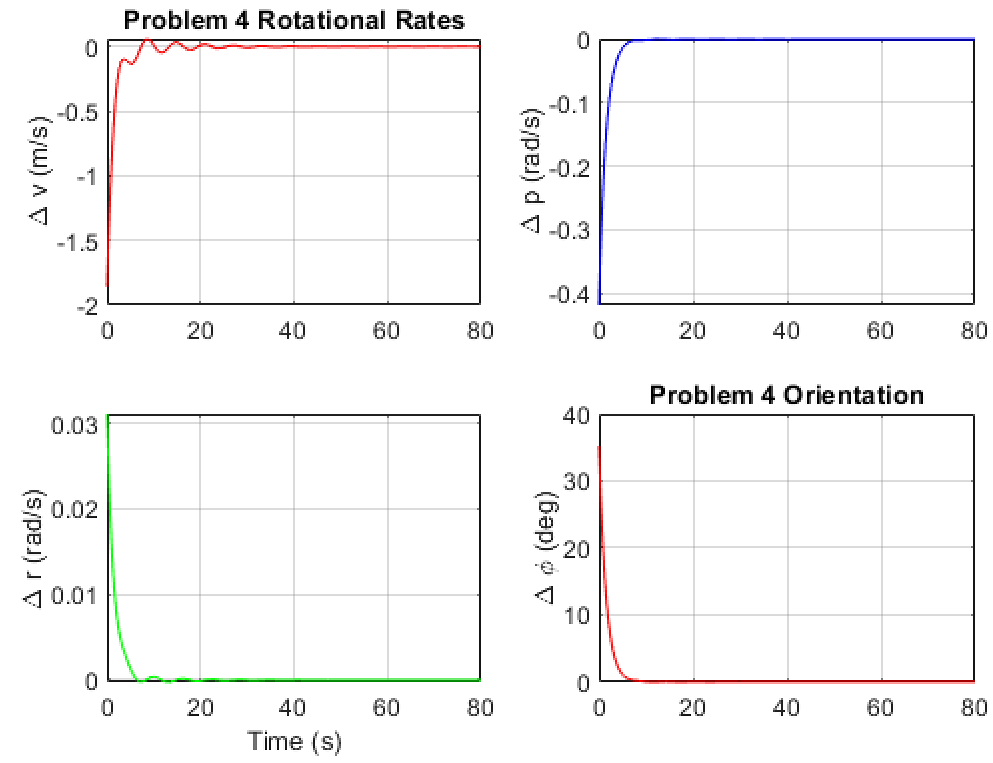

Steady State
Problem 4 Rotational Rates
Δ v (m/s)
Δ p (rad/s)
Δ r (rad/s)
Problem 4 Orientation
Δ ϕ (deg)
Time (s)

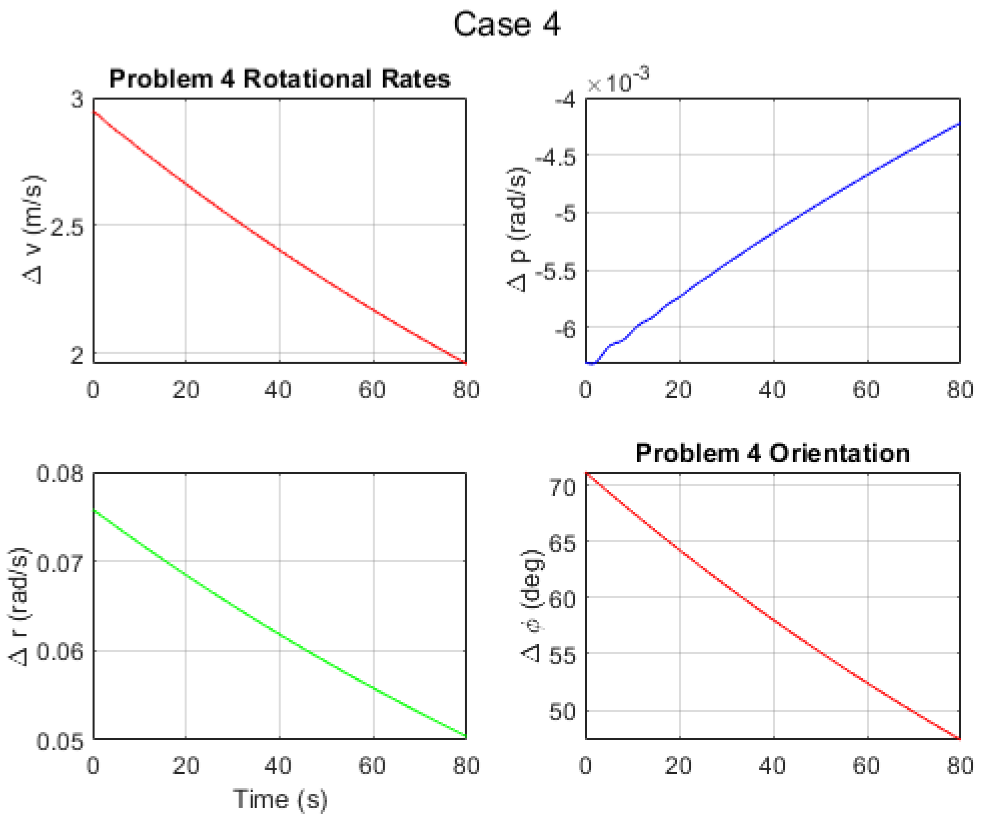

Case 1
Problem 4 Rotational Rates
Δ v (m/s)
Δ p (rad/s)
Δ r (rad/s)
Problem 4 Orientation
Δ ϕ (deg)
Time (s)

Case 2


Problem 4 Rotational Rates
Δ v (m/s)
Δ p (rad/s)
Δ r (rad/s)
Problem 4 Orientation
Δ ϕ (deg)
Time (s)

Case 3


Problem 4 Rotational Rates
Δ v (m/s)
Δ p (rad/s)
Δ r (rad/s)
Problem 4 Orientation
Δ ϕ (deg)
Time (s)



Functions Called

The following functions were built and called as part of this assignment.

```
function [dydt] = linearized_Aircraft_ODE(t,y,A)
% linearized_Aircraft_ODE This function is used to simulate trimmed
 flight
% of an aircraftwith simplified kinematics and dynamics.

% Position in E frame
del_pos_N = y(1); del_pos_E = y(2); del_pos_D = y(3); % m
% Velocity in B frame
del_u = y(4); del_v= y(5); del_w= y(6); % m/s
del_psi = y(7); del_theta = y(8); del_phi = y(9); % rad
del_p = y(10); del_q = y(11); del_r = y(12); % rad/s
del_VE_B = [del_u;del_v;del_w];

% Transfer matrix from B to E frame coordinates
L_EB =
 [cos(del_theta)*cos(del_psi),sin(del_phi)*sin(del_theta)*cos(del_psi)-
cos(del_phi)*sin(del_psi),...

 cos(del_phi)*sin(del_theta)*cos(del_psi)+sin(del_phi)*sin(del_psi);cos(del_theta)

 sin(del_phi)*sin(del_theta)*sin(del_psi)+cos(del_phi)*cos(del_psi),...
```

```
    cos(del_phi)*sin(del_theta)*sin(del_psi)-
sin(del_phi)*cos(del_psi);...
    -
sin(del_theta),sin(del_phi)*cos(del_theta),cos(del_phi)*cos(del_theta)];

% Forces from rotors and aerodynamic drag
del_Xc = 0;
del_Yc = 0;
del_Zc = 0;
del_Ac = [del_Xc;del_Yc;del_Zc];

% Acceleration in B frame coordinates
del_y = [del_v;del_p;del_r;del_phi];
del_y_d = A*del_y;

del_u_d = 0;
del_v_d = del_y_d(1);
del_w_d = 0;

% Velocity in E frame coordinates
VE_E = L_EB*del_VE_B;
vel_NE = 0;
vel_EE = VE_E(2);
vel_DE = 0;

% Finding the aerodynamic moments from rotors and drag
del_Lc = 0;
del_Mc = 0;
del_Nc = 0;
del_G_c = [del_Lc,del_Mc,del_Nc];

% Finding change in p,q,r
del_p_d = del_y_d(2);
del_q_d = 0;
del_r_d = del_y_d(3);

% Finding changes in psi,theta,phi from p,q,r
del_psi_d = 0;
del_theta_d = 0;
del_phi_d = del_y_d(4);

yd_trans = [vel_NE; vel_EE; vel_DE; del_u_d; del_v_d; del_w_d];
yd_rot  = [del_psi_d; del_theta_d; del_phi_d; del_p_d; del_q_d;
 del_r_d];

dydt = [yd_trans;yd_rot];
end
```

Published with MATLAB® R2019b