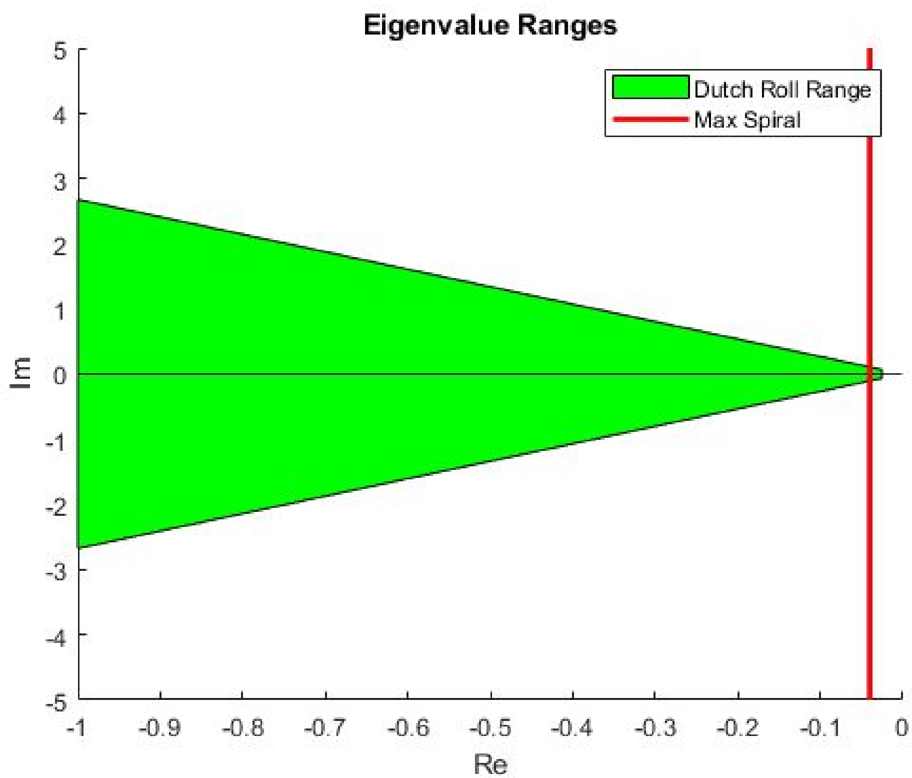
Samuel Rauzmovksiy
ASEN 3128 Assignment 12
12/12/2019

K =

0	0	0	0	0	0
0	0	1.55	0	0	0

1. a)



$\lambda = n \pm \sigma i$

$n = -1/\tau$,

$\omega = (n^2+\sigma^2)^{½}$

$\omega = -n/\zeta$

Substituting ω

$(n^2+\sigma^2)^{½} = -n/\zeta$

Solving for imaginary poriton

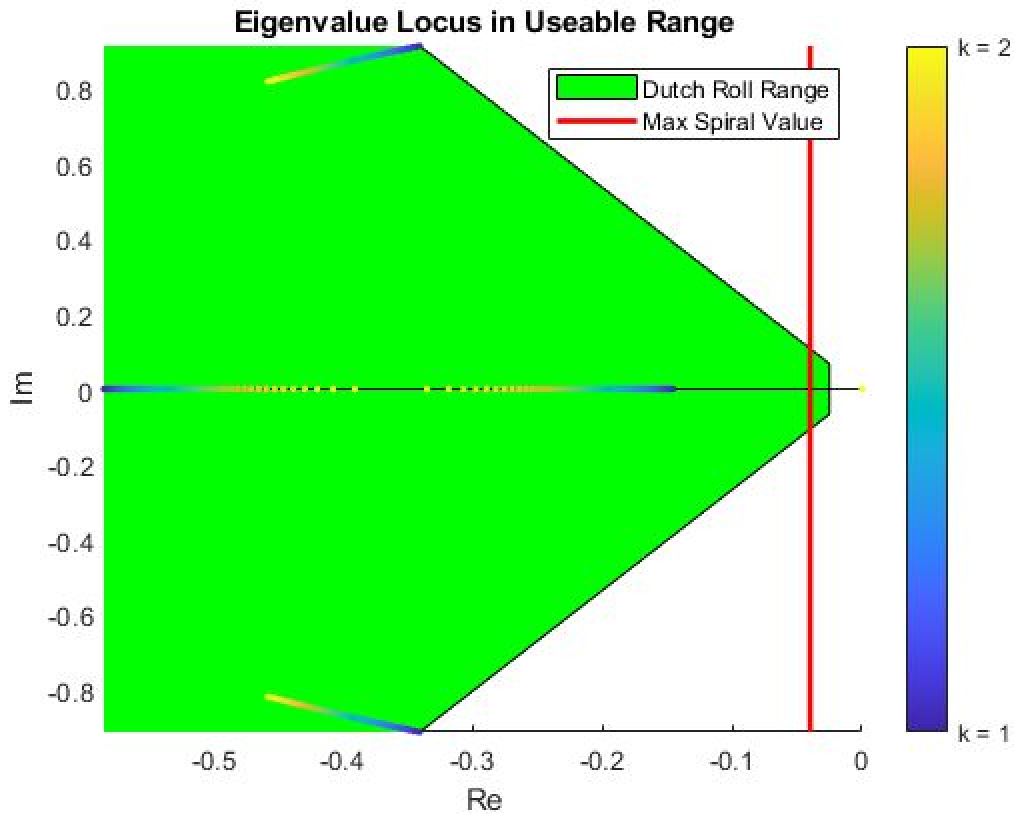
$\sigma = (n^2/\zeta^2-n^2)^{½}$

Dutch Roll:

$n < -1/40$

$n < -0.025$
$\sigma < (n^2/0.35^2 - n^2)^{1/2}$

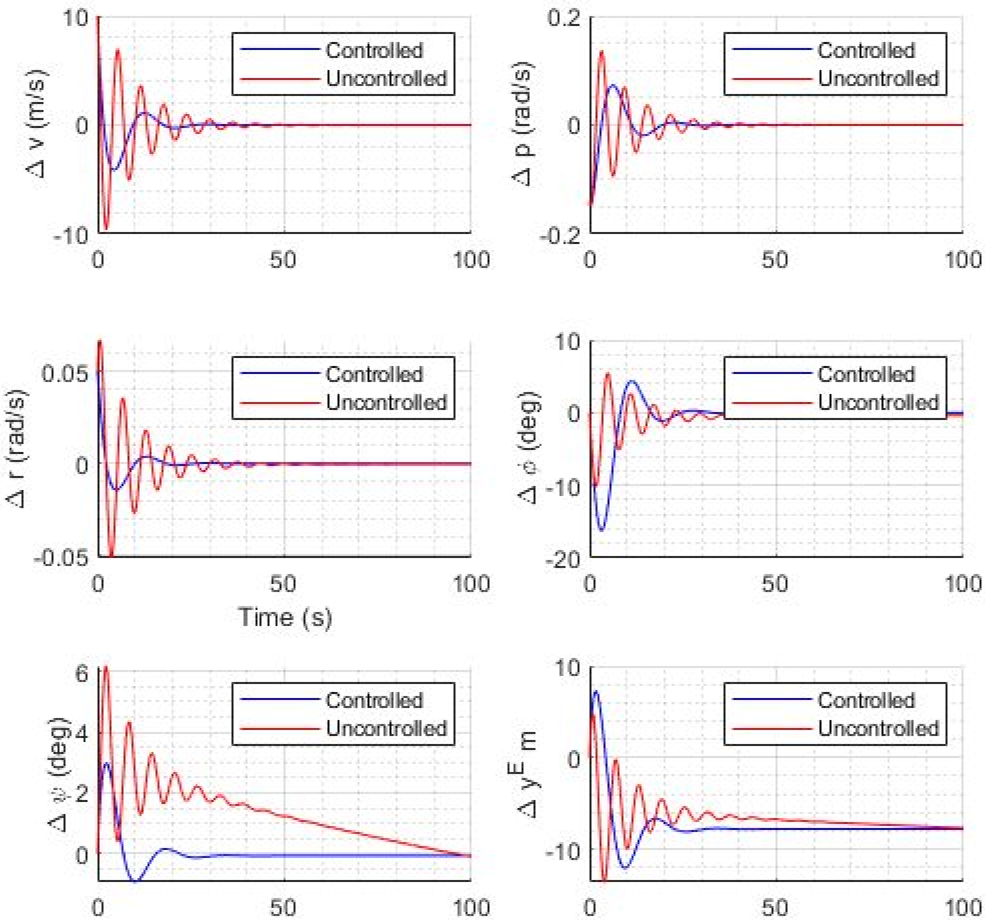
Spiral:
$n < -1/25$



b) The controller I designed only has a K_{rr} value and it seems to do enough to properly stabilize the aircraft with the requirements given. The stability derivatives that are mostly affected are $\mathcal{Y}_r$, $\mathcal{L}_r$, and $\mathcal{N}_r$. The value I chose brings the spiral and roll modes extremely close together while also dampening the dutch roll mode. This may not be optimal since there can be deviations and a new mode can suddenly appear with a slight change in the aircraft's configuration. I came to this choice from the plot provided above.

c) The characteristics of the aircraft to deviations are shown in the plots below. It can be noted that the maximum overshoot in Δv is less than 6 m/s, and the maximum overshoot in $\Delta\psi$ is no where near 5 deg, there is also a maximum peak deflection in the rudder response of 4.44 degrees. Also, there is no aileron deflection since there is no aileron control.

Problem 1.C



Controlled
Uncontrolled
Δ v (m/s)
Δ p (rad/s)
Δ r (rad/s)
Δ φ (deg)
Δ ψ (deg)
Δ y^E m
Time (s)

Table of Contents

Header 1
Question 1 1
Functions Called 5

Header

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%
% Author: Samuel Razumovskiy
% Date written: 12/06/19
% Date modified: 12/12/19
%
% 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;

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];

[V,D] = eig(A);

% Specifying the fancy letter values
Yv = A(1,1); Yp = A(1,2); Yr = A(1,3);
Lv = A(2,1); Lp = A(2,2); Lr = A(2,3);
Nv = A(3,1); Np = A(3,2); Nr = A(3,3);

% Coefficients from the book are multiplied against a conversion
 matrix
Coeff = [    0, -1.368e-2, -1.973e-4;...
    0.1146,  6.976e-3,   -0.1257];
conv = [1/2*rho*u0^2*S, 1/2*rho*u0^2*S*b, 1/2*rho*u0^2*S*b;...
    1/2*rho*u0^2*S, 1/2*rho*u0^2*S*b, 1/2*rho*u0^2*S*b];
dim = Coeff.*conv;

Yda = dim(1,1);
Ydr = dim(2,1);
Lda = dim(1,2);
Ldr = dim(2,2);
Nda = dim(1,3);
Ndr = dim(2,3);

% B matrix
```

```
B = [          Yda/m,              Ydr/m;
    Lda/Ixp+Izxp*Nda, Ldr/Ixp+Izxp*Ndr;
    Izxp*Lda+Nda/Izp, Izxp*Ldr+Ndr/Izp;
    0,               0];

% Augmenting the A and B matricies
Aaug =
 [A(1,:),0,0;A(2,:),0,0;A(3,:),0,0;A(4,:),0,0;0,0,1,0,0,0;1,0,0,0,u0,0];

Baug = [B(1,:);B(2,:);B(3,:);B(4,:);0,0;0,0];
Ddaphi = 0:0.01:10;
Ddap = -(0:0.01:10);
Ddar = -(0:0.01:10);
Ddapsi = 0:0.1:20;
Ddrv = -(0:0.001:0.1);
Ddrp = -(0:0.1:2);
Ddrr = 0:0.01:5;
Ddrphi = -(0:0.1:5);
Ddrpsi = 0:0.1:5;
name = {'Case a';'Case b';'Case c';'Case d';'Case e';'Case f';'Case
 g';...
    'Case h';'Case i'};

tauD = 40;
tauS = 25;

nDmax = -1/tauD;
nD = [-1,nDmax];
iD = sqrt(nD.^2./0.35^2-nD.^2);
nSmax = -1/tauS;

figure
hold on
ar1 = area(nD,iD,0);
ar2 = area(nD,-iD,0);
ar1.FaceColor = 'g';
ar2.FaceColor = 'g';
p = plot([nSmax,nSmax],[-5,5],'r','LineWidth',2);
title('Eigenvalue Ranges')
xlabel('Re')
ylabel('Im')
legend([ar1,p],'Dutch Roll Range','Max Spiral')

Kuse = CheckEig(Aaug,Baug,u0);

k = max(Kuse);

tspan = [0 100];

pos_N = 0; pos_E = 0; pos_D = 0; % m E frame
u = 0; v = 10; w = 0; % m/s B frame
y_trans = [pos_N; pos_E; pos_D; u; v; w];

psi = 0; theta = 0.1; phi = 0; % rad
```

```
p = -0.14; q = 0; r = 0.05; % rad/s B frame
y_rot = [psi; theta; phi; p; q; r];

yin = [y_trans;y_rot];
opt = odeset('maxstep',0.01);

K = [0,0,0,0,0,0;0,0,k,0,0,0];
Acl = Aaug+Baug*K;

[~,pos1] =
 ode45(@(t,y)linearized_Aircraft_ODE(t,y,u0,Acl,Baug,K),tspan,yin,opt);

K = zeros(2,6);
Acl = Aaug+Baug*K;
[t,pos2] =
 ode45(@(t,y)linearized_Aircraft_ODE(t,y,u0,Acl,Baug,K),tspan,yin,opt);

figure
sgtitle('Problem 1.C')
subplot(3,2,1)
hold on
grid minor
grid on
plot(t,pos1(:,5),'b')
plot(t,pos2(:,5),'r')
ylabel('\Delta v (m/s)')
legend('Controlled','Uncontrolled')
ylim([-10,10])

subplot(3,2,2)
hold on
grid minor
grid on
plot(t,pos1(:,10),'b')
plot(t,pos2(:,10),'r')
ylabel('\Delta p (rad/s)')
legend('Controlled','Uncontrolled')

subplot(3,2,3)
hold on
grid minor
grid on
plot(t,pos1(:,12),'b')
plot(t,pos2(:,12),'r')
ylabel('\Delta r (rad/s)')
xlabel('Time (s)')
legend('Controlled','Uncontrolled')

subplot(3,2,4)
hold on
grid minor
grid on
plot(t,pos1(:,9).*180./pi,'b')
plot(t,pos2(:,9).*180./pi,'r')
```

```
ylabel("\Delta \phi (deg)")
legend('Controlled','Uncontrolled')

subplot(3,2,5)
hold on
grid minor
grid on
plot(t,pos1(:,7).*180./pi,'b')
plot(t,pos2(:,7).*180./pi,'r')
ylabel("\Delta \psi (deg)")
legend('Controlled','Uncontrolled')

subplot(3,2,6)
hold on
grid minor
grid on
plot(t,pos1(:,2),'b')
plot(t,pos2(:,2),'r')
ylabel("\Delta y^E m")
legend('Controlled','Uncontrolled')

fprintf('Krr value choosen - %1.2f',k*max(pos(:,12))*190/pi)
```

```
Unrecognized function or variable 'pos'.

Error in Assignment12 (line 217)
fprintf('Krr value choosen - %1.2f',k*max(pos(:,12))*190/pi)
```

Functions Called

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

```
function Kuse = CheckEig(A,B,u0)
% PlotEig is used to generate the eigen values of a bunch of K values

tspan = [0 100];

pos_N = 0; pos_E = 0; pos_D = 0; % m E frame
u = 0; v = 10; w = 0; % m/s B frame
y_trans = [pos_N; pos_E; pos_D; u; v; w];

psi = 0; theta = 0.1; phi = 0; % rad
p = -0.14; q = 0; r = 0.05; % rad/s B frame
y_rot = [psi; theta; phi; p; q; r];


yin = [y_trans;y_rot];
opt = odeset('maxstep',0.01);

k =0:0.01:5;
K = zeros(2,6);
j = 1;
for i = 1:numel(k)
```

```
    K(2,3) = k(i);
    Acl = A+B*K;
    [~,D] = eig(Acl);
    eigen = [D(1,1),D(2,2),D(3,3),D(4,4),D(5,5),D(6,6)];

    omegaN = sqrt(real(eigen).^2+imag(eigen).^2);
    zeta = -real(eigen)./omegaN;
    omegaN(omegaN == 0) = [];
    imaginary = imag(eigen);
    imaginary(imaginary == 0) = [];
    zeta = rmmissing(zeta);
    tau = 1./(zeta.*omegaN);

    if (25 >= tau) & (0.35 <= zeta) & numel(imaginary) == 2
        [~,pos] =
ode45(@(t,y)linearized_Aircraft_ODE(t,y,u0,Acl,B,K),tspan,yin,opt);
        Vover = min(pos(:,5));
        delRud = k(i)*max(pos(:,12));
        maxpsi = max(pos(:,7));
        if Vover > -6 && delRud <10 && maxpsi < 5
            Kuse(j) = k(i);
            lam1(j) = D(1,1);
            lam2(j) = D(2,2);
            lam3(j) = D(3,3);
            lam4(j) = D(4,4);
            lam5(j) = D(5,5);
            lam6(j) = D(6,6);
            j = j+1;
        end
    end

end

tauD = 40;
tauS = 25;

nDmax = -1/tauD;
nD = [-1,nDmax];
iD = sqrt(nD.^2./0.35^2-nD.^2);
nSmax = -1/tauS;

figure
hold on
ar1 = area(nD,iD,0);
ar2 = area(nD,-iD,0);
ar1.FaceColor = 'g';
ar2.FaceColor = 'g';
p = plot([nSmax,nSmax],[-5,5],'r','LineWidth',2);

maxX = max(real([lam1,lam2,lam3,lam4,lam5,lam6]));
minX = min(real([lam1,lam2,lam3,lam4,lam5,lam6]));
maxY = max(imag([lam1,lam2,lam3,lam4,lam5,lam6]));
minY = min(imag([lam1,lam2,lam3,lam4,lam5,lam6]));
c = linspace(1,10,length(Kuse));
```

```
s(1) = scatter(real(lam1),imag(lam1),7,c,'filled');
s(2) = scatter(real(lam2),imag(lam2),7,c,'filled');
s(3) = scatter(real(lam3),imag(lam3),7,c,'filled');
s(4) = scatter(real(lam4),imag(lam4),7,c,'filled');
s(5) = scatter(real(lam5),imag(lam5),7,c,'filled');
s(6) = scatter(real(lam6),imag(lam6),7,c,'filled');
colorbar('Ticks',[1,10],'TickLabels',{sprintf('k =
 %.0f',min(Kuse)),sprintf('k = %.0f',max(Kuse))});
title('Eigenvalue Locus in Useable Range')
xlabel('Re')
ylabel('Im')
xlim([minX,maxX])
ylim([minY,maxY])
legend([ar1,p],'Dutch Roll Range','Max Spiral Value')
end


function [dydt] = linearized_Aircraft_ODE(t,y,u0,A,B,K)
% linearized_Aircraft_ODE This function is used to simulate trimmed
 flight
% of an aircraft with 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_psi;del_pos_E];
del_y_d = (A+B*K)*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 = del_y_d(5);
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