Table of Contents

Header 1
Question 1 1
Question 2 a 2
Question 2 b 3
Question 3 3
Functions Called 9

Header

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%
% Author: Samuel Razumovskiy
% Date written: 11/22/19
% Date modified: 12/5/19
%
% 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);

Re = real(D);
Im = imag(D);

% Roll approximation
RollAp = A(2,2);
e = g*(Lv*Nr-Lr*Nv);
d = -g*Lv+Yv*(Lr*Np-Lv*Np)+Yr*(Lp*Nv-Lv*Np);
SpiralAp = -e/d;
```

Question 2 a

```
% 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];
```

Question 2 b

```
% 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];
```

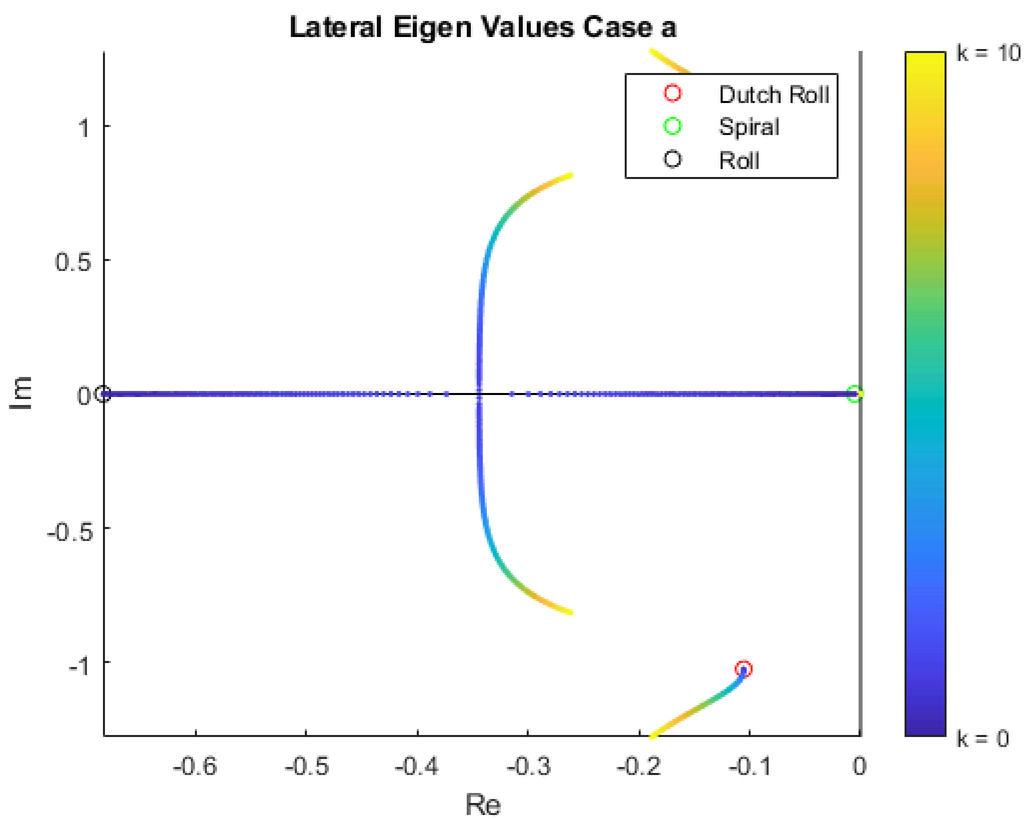
Question 3

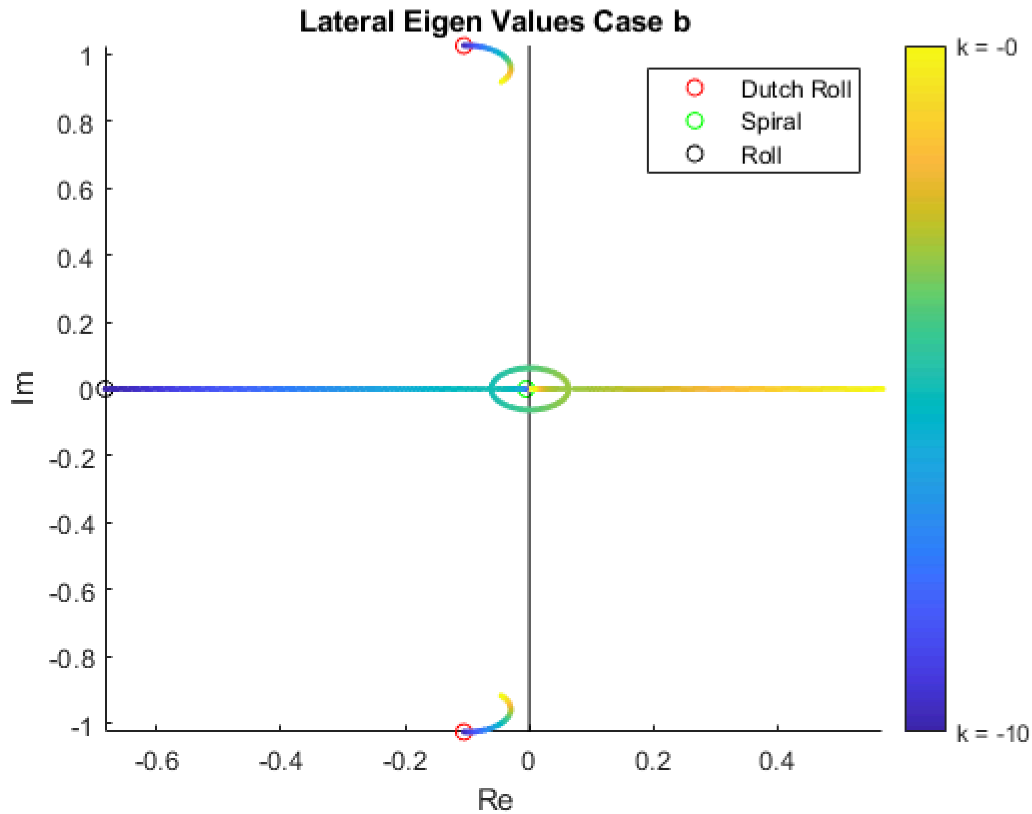
```
% Setting all the variations
Ddaphi = 0:0.01:10;
Ddap = -(0:0.01:10);
Ddar = -(0:0.01:10);
Ddapsi = 0:0.01:20;
Ddrv = -(0:0.0001:0.1);
Ddrp = -(0:0.01:2);
Ddrr = 0:0.01:5;
Ddrphi = -(0:0.01:5);
Ddrpsi = 0:0.01:5;
name = {'Case a';'Case b';'Case c';'Case d';'Case e';'Case f';'Case
 g';...
    'Case h';'Case i'};

% Creating all the K matrices for each case
K = {{   0,     0,     0, Ddaphi,      0,0;...
         0,     0,     0,      0,      0,0},...
     {   0, Ddap,      0,      0,      0,0;...
         0,     0,     0,      0,      0,0},...
     {   0,     0, Ddar,       0,      0,0;...
         0,     0,     0,      0,      0,0},...
     {   0,     0,     0,      0, Ddapsi,0;...
         0,     0,     0,      0,      0,0},...
```

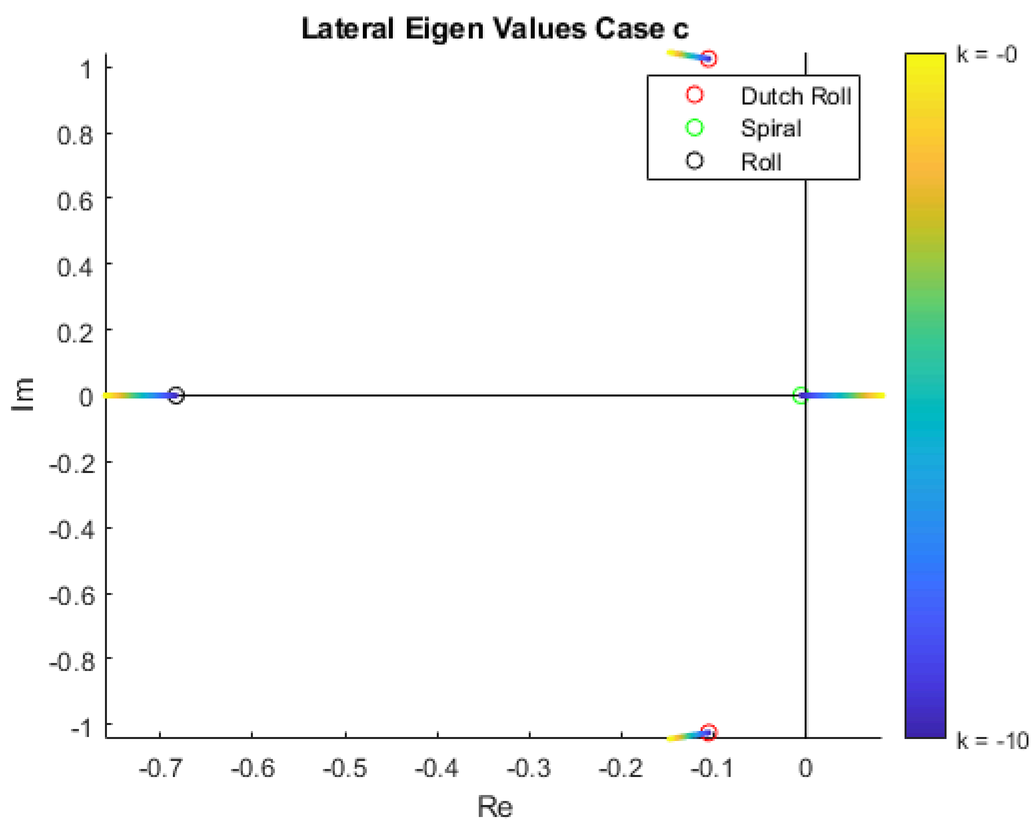
```
    {  0,     0,     0,       0,       0,0;...
    Ddrv,     0,     0,       0,       0,0},...
    {  0,     0,     0,       0,       0,0;...
       0,  Ddrp,     0,       0,       0,0},...
    {  0,     0,     0,       0,       0,0;...
       0,     0,  Ddrr,       0,       0,0},...
    {  0,     0,     0,       0,       0,0;...
       0,     0,     0,  Ddrphi,       0,0},...
    {  0,     0,     0,       0,       0,0;...
       0,     0,     0,       0,  Ddrpsi,0}};

% Loops through all the cases and plots the eigenvalues
for i = 1: numel(K)
    PlotEig(Aaug,Baug,K{i},name{i})
end
```

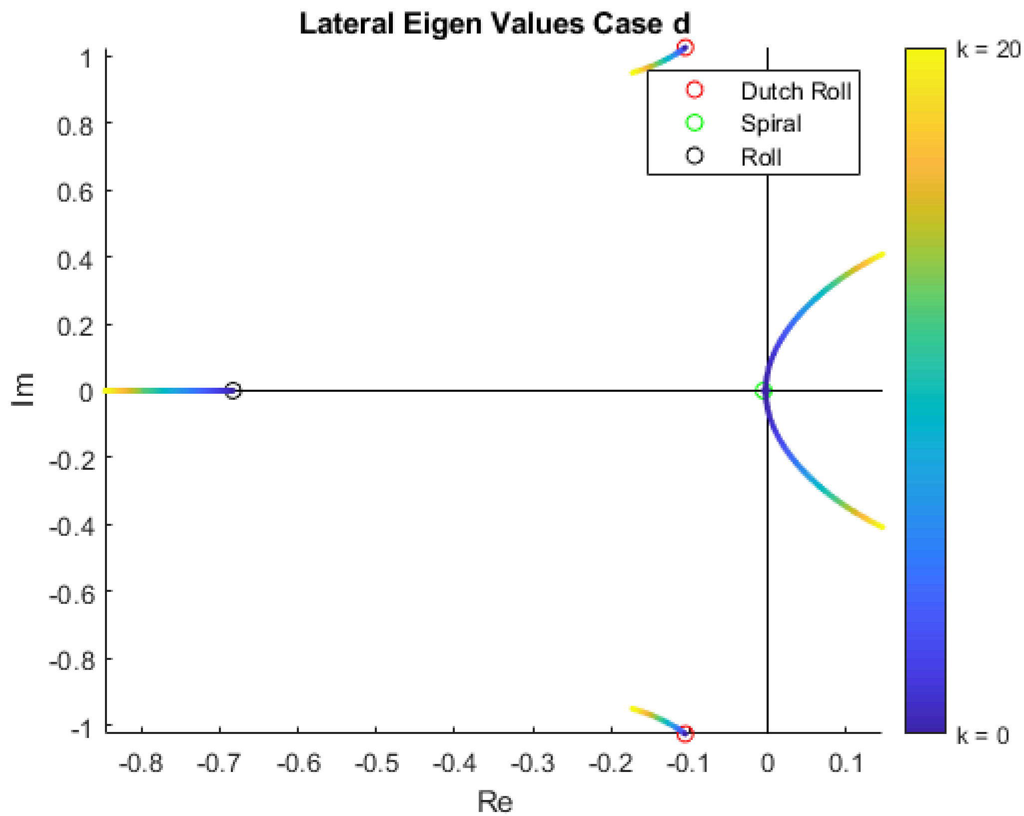


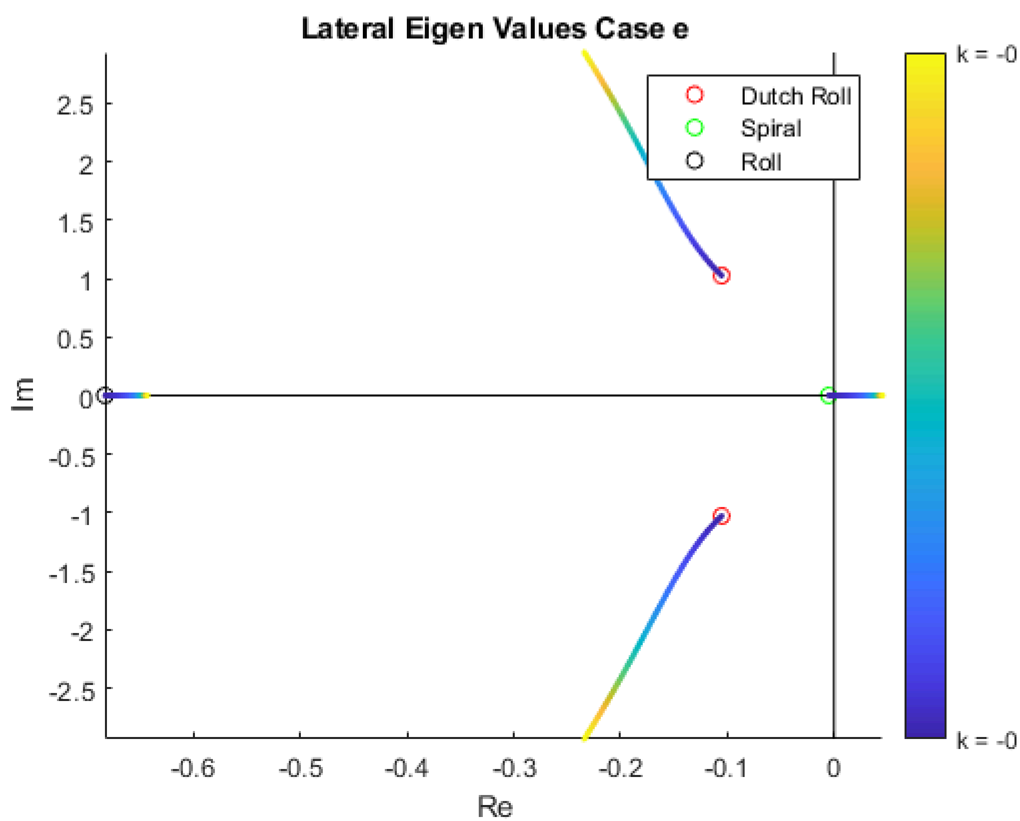


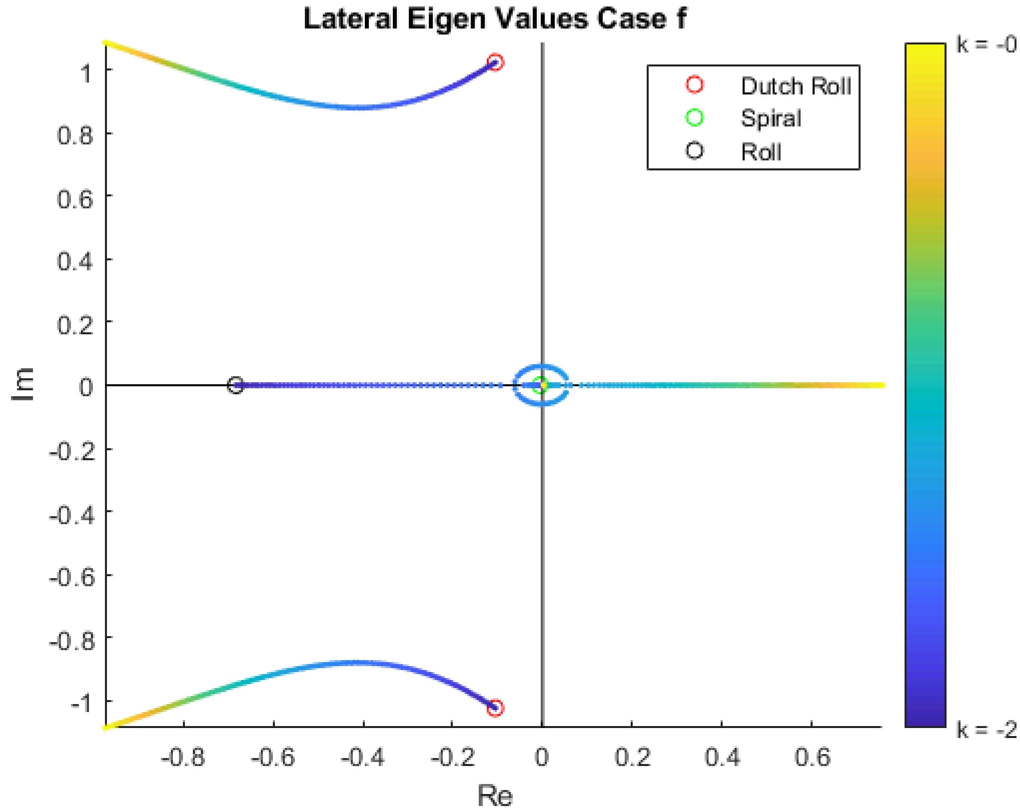
Lateral Eigen Values Case b
Dutch Roll
Spiral
Roll
k = -0
k = -10
Im
Re
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-0.6
-0.4
-0.2
0
0.2
0.4

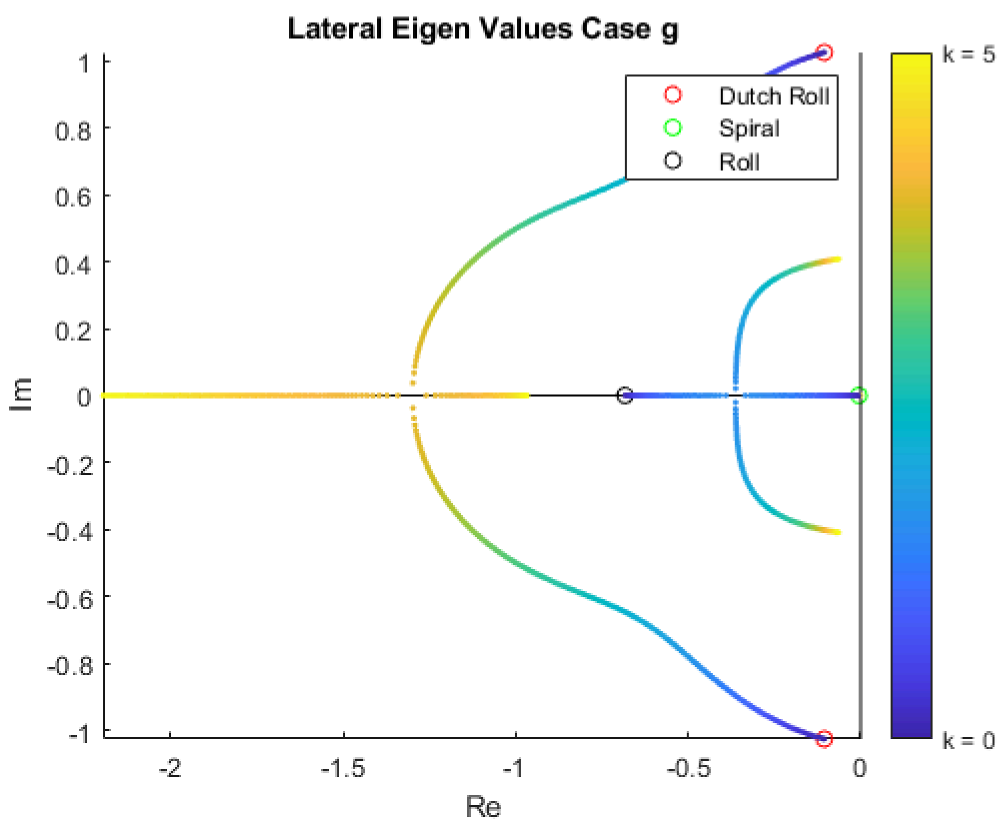


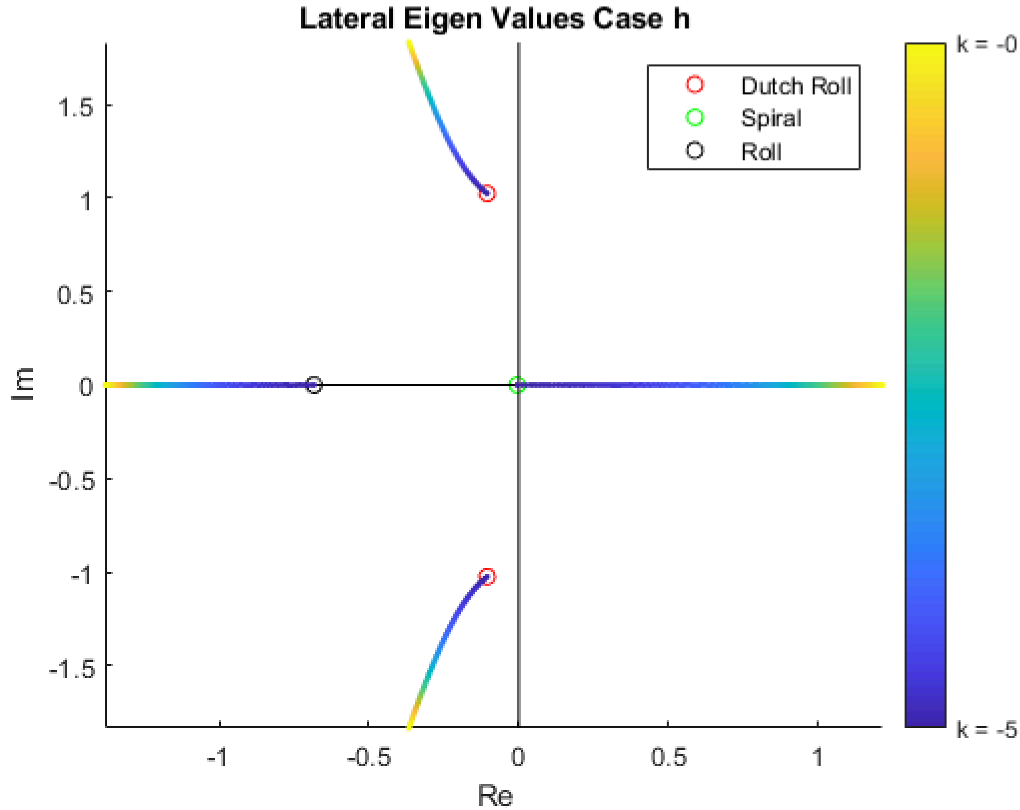
Lateral Eigen Values Case c
Dutch Roll
Spiral
Roll
k = -0
k = -10
Im
Re
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-0.7
-0.6
-0.5
-0.4
-0.3
-0.2
-0.1
0

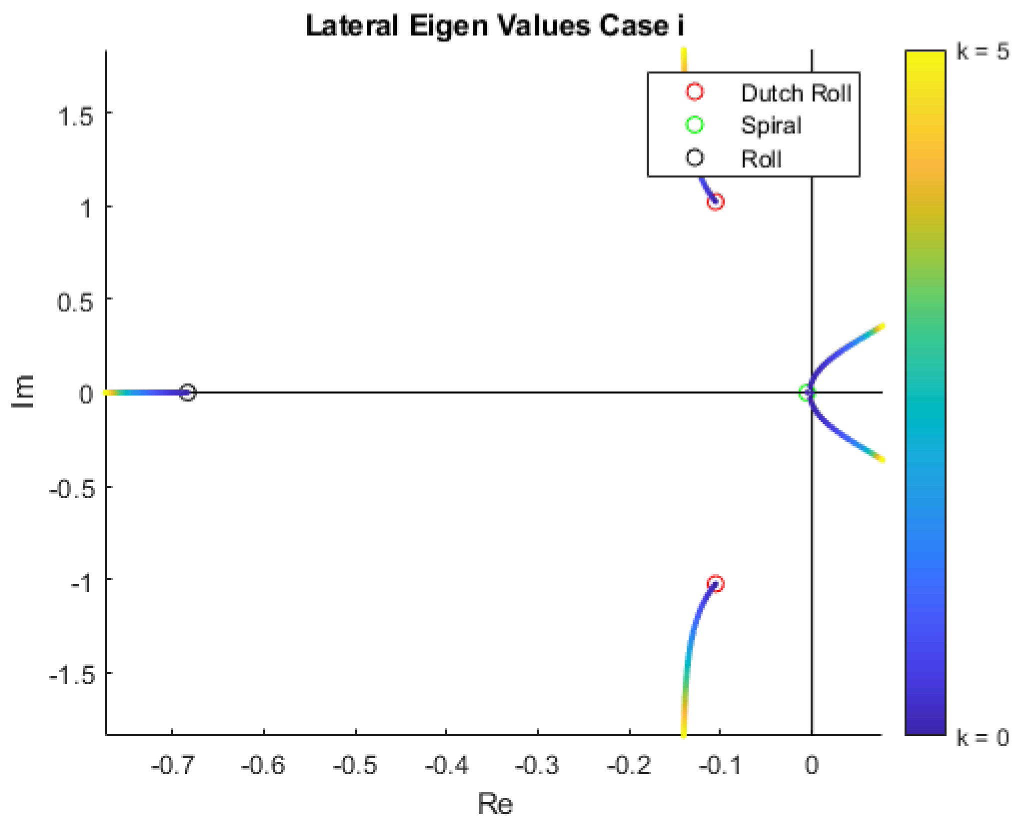

Lateral Eigen Values Case d
Dutch Roll
Spiral
Roll
k = 20
k = 0
Im
Re
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-0.8
-0.7
-0.6
-0.5
-0.4
-0.3
-0.2
-0.1
0
0.1


Lateral Eigen Values Case e
Dutch Roll
Spiral
Roll
k = -0
k = -0
Im
Re
2.5
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2.5
-0.6
-0.5
-0.4
-0.3
-0.2
-0.1
0


Lateral Eigen Values Case f
Dutch Roll
Spiral
Roll
k = -0
k = -2
Im
Re
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6


Lateral Eigen Values Case g
Dutch Roll
Spiral
Roll
k = 5
k = 0
Im
Re
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-2
-1.5
-1
-0.5
0


Lateral Eigen Values Case h
Dutch Roll
Spiral
Roll
Im
Re
k = -0
k = -5


Lateral Eigen Values Case i
Dutch Roll
Spiral
Roll
Im
Re
k = 5
k = 0

Functions Called

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

```
function PlotEig(A,B,Kin,name)
% PlotEig is used to generate the eigen values of a bunch of K values

% Creates the K matrix that will be used for math
K = zeros(size(Kin));
[r,c] = size(Kin);

% This loop finds the index of the changing K value
for i = 1:r
    for j = 1:c
        if numel(Kin{i,j}) > 1
            k = Kin{i,j};
            x = j;
            y = i;
            break
        end
    end
end
lam1 = zeros(size(k));
lam2 = zeros(size(k));
lam3 = zeros(size(k));
lam4 = zeros(size(k));
lam5 = zeros(size(k));
lam6 = zeros(size(k));

% This loop calculates the eigen values by iterating through each k
 value
% and creates a new K everytime to calculate the closed loop A matrix
for i = 1:numel(k)
    K(y,x) = k(i);
    Acl = A+B*K;
    [~,D]   = eig(Acl);
    lam1(i) = D(1,1);
    lam2(i) = D(2,2);
    lam3(i) = D(3,3);
    lam4(i) = D(4,4);
    lam5(i) = D(5,5);
    lam6(i) = D(6,6);
end

%% Plotting
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]));
figure
hold on
c = linspace(1,10,length(k));
```

```
s(1) = plot(real([lam3(1),lam4(1)]),imag([lam3(1),lam4(1)]),'ro');
s(2) = plot(real(lam5(1)),imag(lam5(1)),'go');
s(3) = plot(real(lam6(1)),imag(lam6(1)),'ko');
s(4) = plot([20,-20],[0,0],'k');
s(4) = plot([0,0],[20,-20],'k');
s(4) = scatter(real(lam1),imag(lam1),5,c,'filled');
s(5) = scatter(real(lam2),imag(lam2),5,c,'filled');
s(6) = scatter(real(lam3),imag(lam3),5,c,'filled');
s(7) = scatter(real(lam4),imag(lam4),5,c,'filled');
s(8) = scatter(real(lam5),imag(lam5),5,c,'filled');
s(9) = scatter(real(lam6),imag(lam6),5,c,'filled');
colorbar('Ticks',[1,10],'TickLabels',{sprintf('k =
 %.0f',min(k)),sprintf('k = %.0f',max(k))});
title(name)
xlabel('Re')
ylabel('Im')
title(['Lateral Eigen Values ',name])
legend(s([1, 2, 3]),'Dutch Roll','Spiral','Roll')
xlim([minX,maxX])
ylim([minY,maxY])
end
```

Published with MATLAB® R2019b