Table of Contents


```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%
% Author: Samuel Razumovskiy
% Date written: 10/24/19
% Date modified: 10/31/19
%
% Purpose: Finding the effect of adding control to a 747
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%

clear,clc,close all

% Defining all the variables
Xu = -712.6;
Xw = 22561;
Xq = 0;
Xwd = 0;
Xdel = 177653;
Zu = -19584;
Zw = -124932;
Zq = -561782;
Zwd = 4529.7;
Zdel = -1.48e6;
Mu = 36367;
Mw = -250289;
Mq = -1.89e7;
Mwd = -18406;
Mdel = -4.89e7;
exz = -6.8;
Ix = 2.468e7;
Iy = 4.488e7;
Iz = 6.738e7;
Izx = 1.315e6;
m = 6.366e5*4.448/9.81;
V = 157.9;
g = 9.81;
u0 = V*cosd(exz);
S = 510.9667;
c = 8.324;

% Defining equations using variables
Xup = Xu*cosd(exz)^2-(Xw+Zu)*sind(exz)*cosd(exz)+Zw*sind(exz)^2;
```

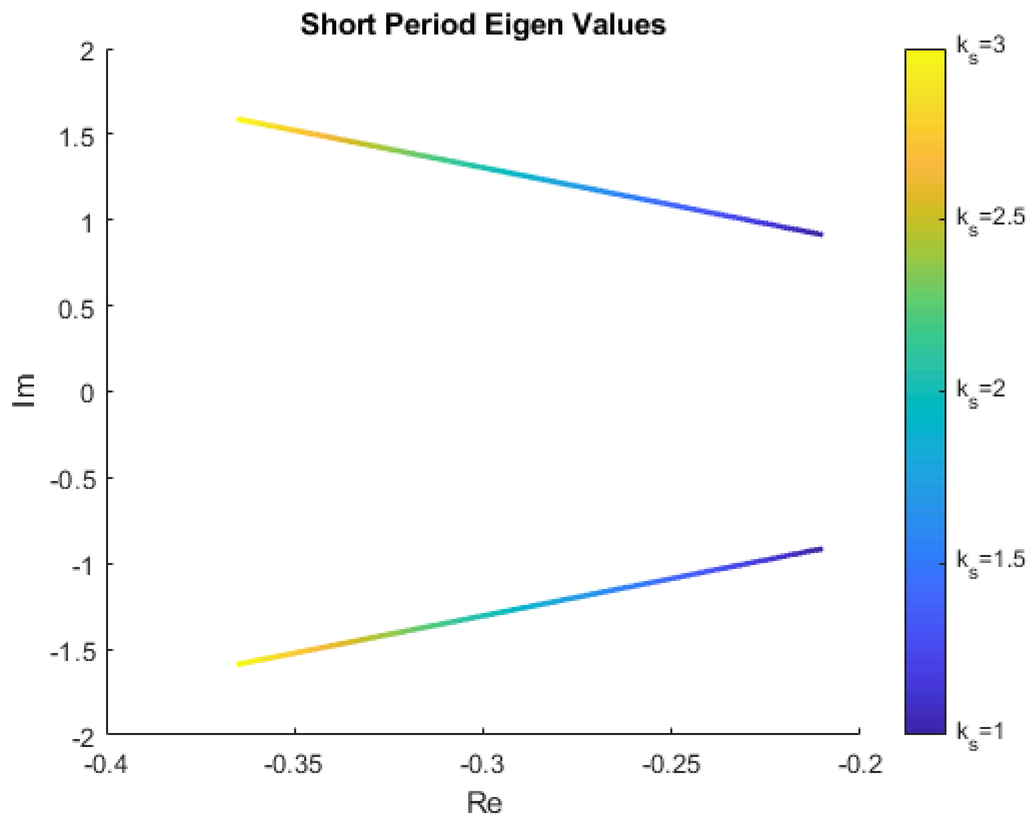
```
Xwp = Xw*cosd(exz)^2+(Xu-Zw)*sind(exz)*cosd(exz)-Zu*sind(exz)^2;
Xqp = Xq*cosd(exz)-Zq*sind(exz);
Xudp = Zwd*sin(exz)^2;
Xwdp = -Zwd*sind(exz)*cosd(exz);
Zup = Zu*cosd(exz)^2-(Zw-Xu)*sind(exz)*cosd(exz)-Xw*sind(exz)^2;
Zwp = Zw*cosd(exz)^2+(Zu+Xw)*sind(exz)*cosd(exz)+Xu*sind(exz)^2;
Zqp = Zq*cosd(exz)+Xq*sind(exz);
Zudp = -Zwd*sind(exz)*cosd(exz);
Zwdp = Zwd*cosd(exz)^2;
Mup = Mu*cosd(exz)-Mw*sind(exz);
Mwp = Mw*cosd(exz)+Mu*sind(exz);
Mqp = Mq;
Mudp = -Mwd*sind(exz);
Mwdp = Mwd*cosd(exz);

% Calculating the A matrix
A = [Xup/m,Xwp/m,0,-g;Zup/(m-Zwdp),Zwp/(m-Zwdp),(Zqp+m*u0)/(m-
Zwdp),0;...
    1/Iy*(Mup+(Mwdp*Zup/(m-Zwdp))),1/Iy*(Mwp+(Mwdp*Zwp/(m-Zwdp))),1/
Iy*(Mqp+(Mwdp*(Zup+m*u0)/(m-Zwdp))),0;
    0,0,1,0];
```

Problem 3a

Using coefficients of stability derivatives provided

```
Cxde = -3.818e-6;
Czde = -0.3648;
Cmde = -1.444;
rho = .65283;

Xde = Cxde*1/2*rho*u0^2*S;
Zde = Czde*1/2*rho*u0^2*S;
Mde = Cmde*1/2*rho*u0^2*S*c;

zeta0 = -Mqp/(Iy*2*sqrt(-u0*Mwp/Iy));
ks = 1:0.01:3;
for i = 1:numel(ks)
    k2 = (u0*Mwp*(1-ks(i)))/Mde;
    k1 = Mqp/Mde+Iy/Mde*2*zeta0*sqrt(-ks(i)*u0*Mwp/Iy);
    Ashort = [(Mqp-k1*Mde)/Iy,(u0*Mwp-k2*Mde)/Iy;1,0];
    [V,D] = eig(Ashort);
    lam1(i) = D(1);
    lam2(i) = D(2,2);
end
figure
hold on
c = linspace(1,10,length(ks));
scatter(real(lam1),imag(lam1),5,c,'filled')
scatter(real(lam2),imag(lam2),5,c,'filled')
colorbar('Ticks',linspace(1,10,5),'TickLabels',
{'k_s=1','k_s=1.5','k_s=2','k_s=2.5','k_s=3'});
xlabel('Re')
```

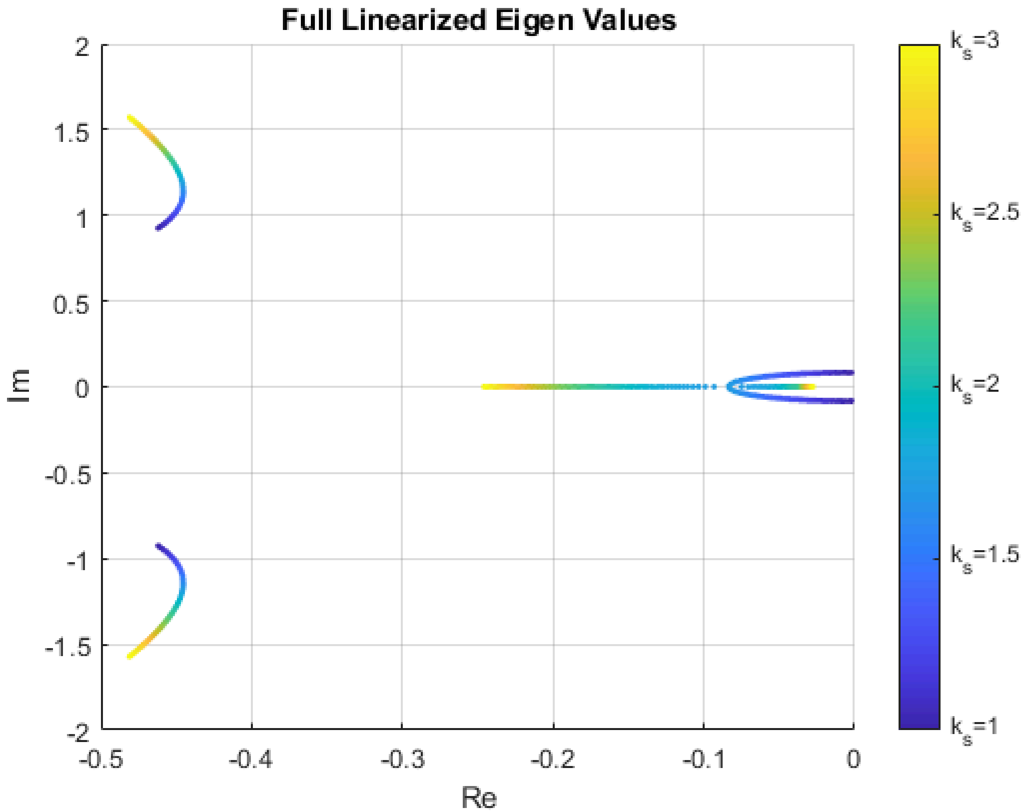
```
ylabel('Im')
title('Short Period Eigen Values')
```



Problem 3b

```
clear lam1 lam2
B = [Xde/m,0;Zde/(m-Zwdp),0;Mde/Iy+(Mwdp*Zde/(Iy*(m-Zwdp))),0;0,0];

for i = 1:numel(ks)
    k2 = (u0*Mwp*(1-ks(i)))/Mde;
    k1 = Mqp/Mde+Iy/Mde*2*zeta0*sqrt(-ks(i)*u0*Mwp/Iy);
    K = [0,0,-k1,-k2;0,0,0,0];
    FullLin = A+B*K;
    [V,D] = eig(FullLin);
    lam1(i) = D(1);
    lam2(i) = D(2,2);
    lam3(i) = D(3,3);
    lam4(i) = D(4,4);
end
figure
hold on
grid on
scatter(real(lam1),imag(lam1),5,c,'filled')
scatter(real(lam2),imag(lam2),5,c,'filled')
scatter(real(lam3),imag(lam3),5,c,'filled')
scatter(real(lam4),imag(lam4),5,c,'filled')
```

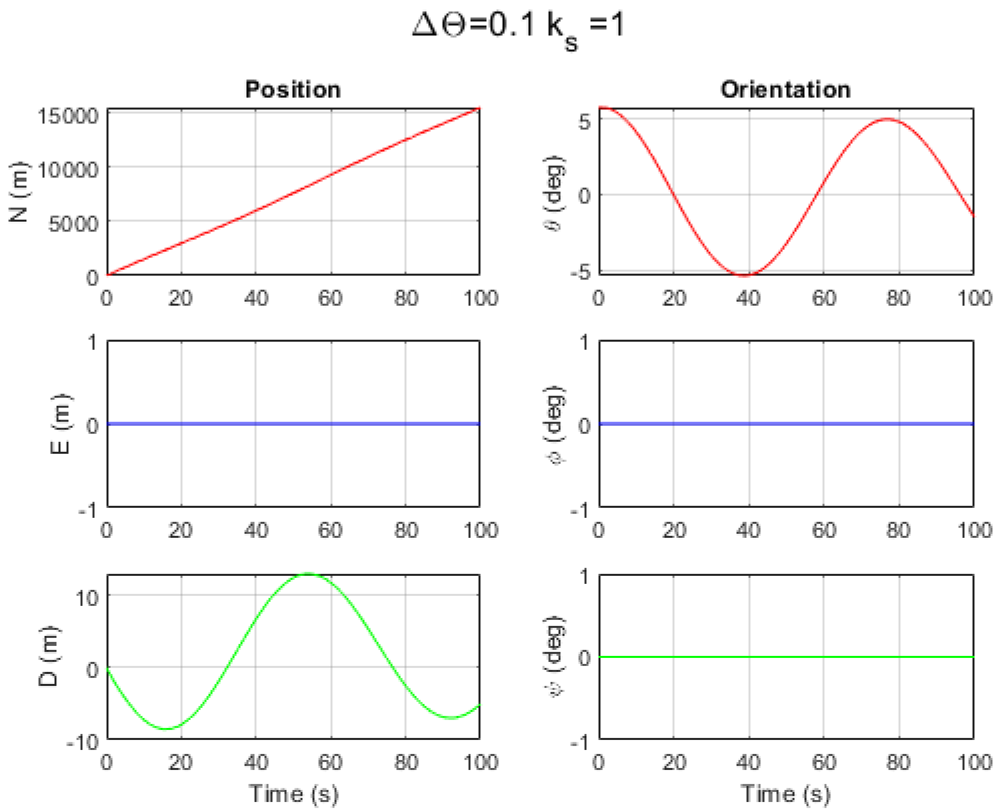
```
colorbar('Ticks',linspace(1,10,5),'TickLabels',
{'k_s=1','k_s=1.5','k_s=2','k_s=2.5','k_s=3'});
xlabel('Re')
ylabel('Im')
title('Full Linearized Eigen Values')
```

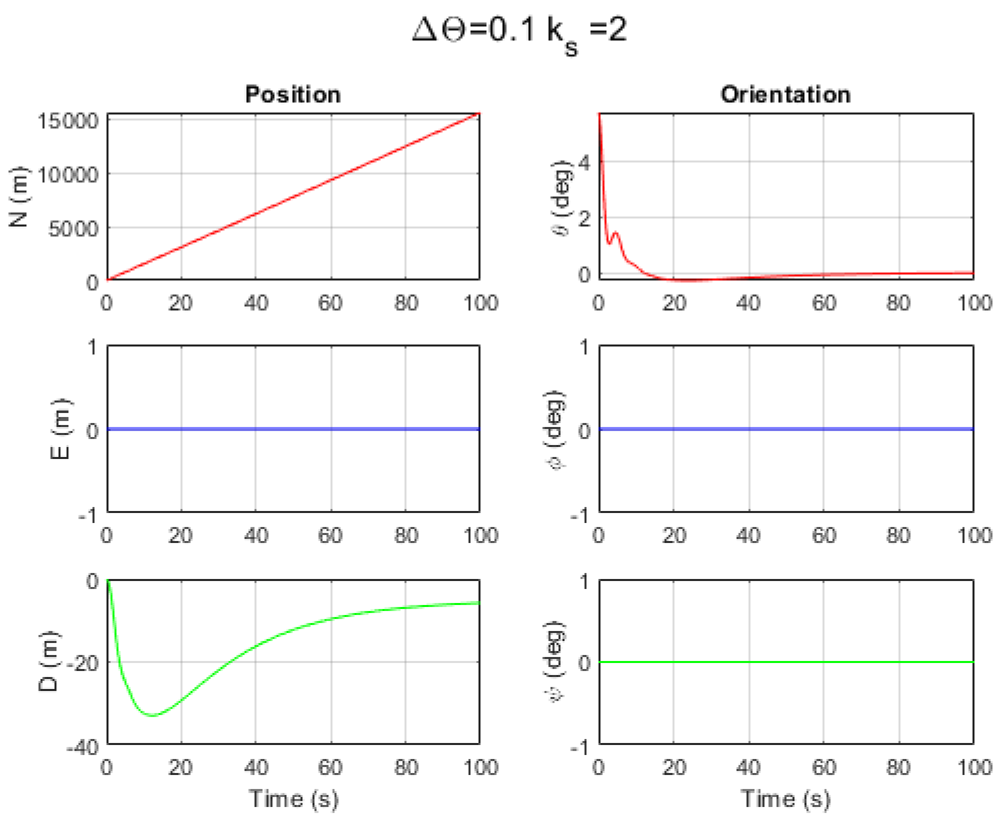


Problem 3c

```
ks = [1,2];
for ks = ks
    k2 = (u0*Mwp*(1-ks))/Mde;
    k1 = Mqp/Mde+Iy/Mde*2*zeta0*sqrt(-ks*u0*Mwp/Iy);
    K = [0,0,-k1,-k2;0,0,0,0];

    tspan = [0 100];

    pos_N = 0; pos_E = 0; pos_D = 0; % m E frame
    u = 0; v = 0; 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; q = 0; r = 0; % rad/s B frame
    y_rot = [psi; theta; phi; p; q; r];

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

```
    [t,pos] =
ode45(@(t,y)linearized_Aircraft_ODE(t,y,u0,A,B,K),tspan,y,opt);

    figure
    value = sprintf("%1.f",ks);
    sgtitle(strcat('{\Delta}{\Theta}=0.1 k_s = ',value))
    subplot(3,2,1)
    plot(t,pos(:,1),'r')
    title('Position')
    ylabel('N (m)')
    grid on

    subplot(3,2,3)
    plot(t,pos(:,2),'b')
    ylabel('E (m)')
    grid on

    subplot(3,2,5)
    plot(t,pos(:,3),'g')
    ylabel('D (m)')
    xlabel('Time (s)')
    grid on

    subplot(3,2,2)
    plot(t,pos(:,8).*180./pi,'r')
    title("Orientation")
    ylabel("\theta (deg)")
    grid on

    subplot(3,2,4)
    plot(t,pos(:,9).*180./pi,'b')
    ylabel("\phi (deg)")
    grid on

    subplot(3,2,6)
    plot(t,pos(:,7).*180./pi,'g')
    ylabel("\psi (deg)")
    xlabel('Time (s)')
    grid on
end
```


ΔΘ=0.1 k s =1
Position
Orientation
N (m)
E (m)
D (m)
θ (deg)
φ (deg)
ψ (deg)
Time (s)
Time (s)


ΔΘ=0.1 k s =2
Position
Orientation
N (m)
E (m)
D (m)
θ (deg)
φ (deg)
ψ (deg)
Time (s)
Time (s)

Functions Called

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

```
function [dydt] = linearized_Aircraft_ODE(t,y,u0,A,B,K)
% 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); w= y(6); % m/s
del_psi = y(7); del_theta = y(8); del_phi = y(9); % rad
del_p = y(10); q = y(11); del_r = y(12); % rad/s
del_VE_B = [del_u;del_v;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_u;w;q;del_theta];
del_U = K*del_y;

del_y_d = A*del_y+B*del_U;

del_u_d = del_y_d(1);
del_v_d = 0;
w_d = del_y_d(2);

% Velocity in E frame coordinates
VE_E = L_EB*del_VE_B;
vel_NE = u0+del_u;
vel_EE = VE_E(2);
vel_DE = w-10*del_theta;
```

```
% 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 = 0;
q_d = del_y_d(3);
del_r_d = 0;

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

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

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

Published with MATLAB® R2019b