ASEN 3128 Homework 7

Jack Lambert*
University of Colorado Boulder, ASEN 3128-012, Group 16
Submitted 3/19/2018

*SID: 104414093

A. Question 1:

Jack Lambert | ASEN3128 | HW #7

1.) $C_{m\alpha} = \underbrace{a(h - h_{wb})}_{=0 \text{ @ } h = h_{wb}} - a_t \bar{V}_H \left(1 - \frac{\partial\epsilon}{\partial\alpha}\right)$

$C_{m\alpha} = -a_t \bar{V}_H \left(1 - \frac{\partial\epsilon}{\partial\alpha}\right) \rightarrow$ Assume $\frac{\partial\epsilon}{\partial\alpha}$ is small

$= -a_t \underbrace{\bar{V}_H}_{\text{double}} = -a_t \left(\frac{\bar{l}_t S_t}{\bar{c} S}\right) \rightarrow \bar{V}_H = \underbrace{\frac{\bar{l}_t S_t}{\bar{c} S}}_{\text{double}}$

- To double pitch stiffness we must double $\bar{V}_H$, which correlates to the family of variables, $\left(\frac{\bar{l}_t S_t}{\bar{c} S}\right)$, collectively doubling in total.

Resulting Effect -

Characteristic Equation $\rightarrow \lambda^2 - \overbrace{\frac{M_q}{I_y}}^{2\omega_N \zeta} \lambda - \overbrace{\frac{U_0 M_w}{I_y}}^{\omega_N^2} = 0$

$M_q = -\frac{1}{2} \rho S \bar{c} U_0 a_t \bar{V}_H l_t$

$U_0 M_w = \frac{1}{2} \rho U_0^2 \bar{c} S C_{m\alpha}$

$\omega_N = \sqrt{\frac{\frac{1}{2} \rho U_0^2 \bar{c} S C_{m\alpha}}{I_y}} \rightarrow \omega_N$ would increase by $\sqrt{2}$ if $C_{m\alpha}$ is doubled

$\zeta = \frac{M_q}{2 I_y \omega_N} \rightarrow \frac{-\rho S \bar{c} U_0 a_t (\bar{V}_H l_t)}{4 \omega_N} \rightarrow$ changes by $2 \rightarrow 4$ times greater

- if l_t is doubled, 4 times greater
- if combination of l_t, S_t $\rightarrow 2 \rightarrow 4$, $\neq 2$
- if just S_t is doubled, 2 times greater

- Since constants remain near constant, the change in ζ would have a range of $\frac{2 \rightarrow 4}{\sqrt{2}}$, or an increase of $\boxed{(\zeta\sqrt{2}) \leq \zeta_{New} \leq (2\zeta\sqrt{2})}$ where larger increases of damping ratio correlate to more change in $\bar{l}_t$ rather than S_t.

Figure 1. Question 1

B. Question 2:

Using the values for the control derivatives found on page 229 of the Dynamics of Flight book and dimensionalizing them based on the flight conditions provided from case III in Appendix E for a Boeing 747 flying at 40,000 feet, the following B matrix was calculated using the derivations below and implementing them into MATLAB.

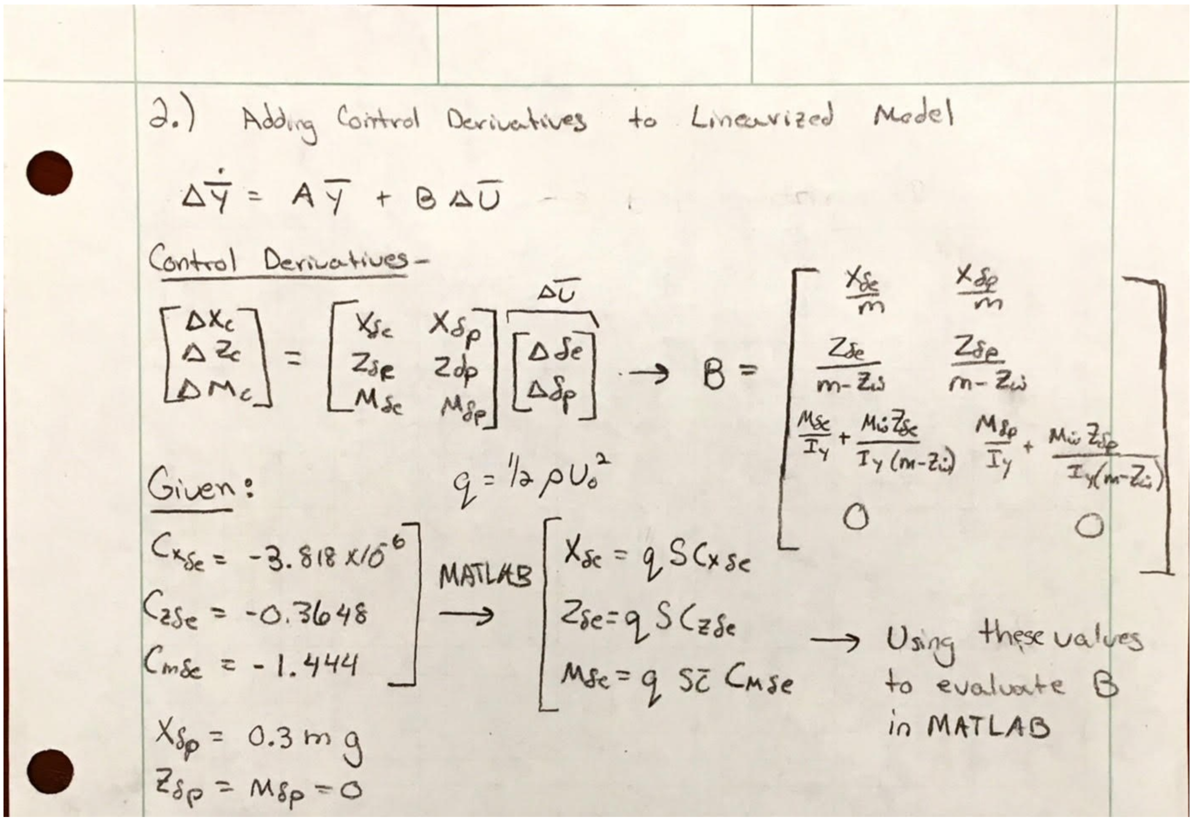


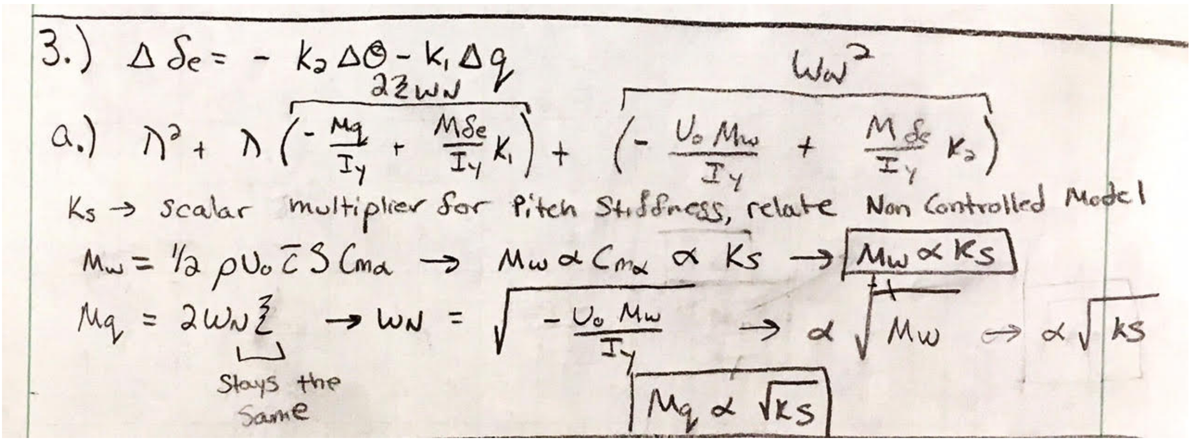
Figure 2. Question 2

$$B = \begin{vmatrix} -7.182 * 10^-5 & 2.943 \\ -6.9076 & 0 \\ -1.452 & 0 \\ 0 & 0 \end{vmatrix}$$

I. Question 3:

A. Part a:

To design for increased pitch stiffness in a short period mode, the reduced 2x2 model was used. To model varying pitch stiffness, a scalar multiplier, ks, was varied from 1 to 3, in steps of 0.01. This was done while retaining the original damping ratio. This was done by using the relations between pitch stiffness and the partials from the characteristic equations as displayed below. These relations were found for the uncontrolled case, where alternatively we could have solved for the gains k1 and k2 for the controlled model and implemented those values for the A matrix, however, we leave it as the uncontrolled case to show what will be needed in the A matrix for an increased pitch stiffness of $-C_m\alpha \cdot ks$.



3.) $\Delta\delta_e = -k_2\Delta\theta - k_1\Delta q$

a.) $\lambda^2 + \lambda\overbrace{\left(-\frac{M_q}{I_y} + \frac{M_{\delta e}}{I_y}k_1\right)}^{2\zeta\omega_N} + \overbrace{\left(-\frac{U_0 M_w}{I_y} + \frac{M_{\delta e}}{I_y}k_2\right)}^{\omega_N^2}$

$K_s \rightarrow$ scalar multiplier for Pitch Stiffness, relate Non Controlled Model

$M_w = \frac{1}{2}\rho U_0 \bar{c} S C_{m\alpha} \rightarrow M_w \propto C_{m\alpha} \propto K_s \rightarrow \boxed{M_w \propto K_s}$

$M_q = 2\omega_N\underbrace{\zeta}_{\text{Stays the Same}} \rightarrow \omega_N = \sqrt{-\frac{U_0 M_w}{I_y}} \rightarrow \propto \sqrt{M_w} \rightarrow \propto \sqrt{ks}$

$\boxed{M_q \propto \sqrt{ks}}$

Figure 3. Question 3, Part a, Derivations

Once the relationships between the scalar pitch stiffness multiplier, ks, is found between the corresponding components in the A matrix, the new scaled A matrix can then be used to find the eigenvalues. This was done over each discrete ks value and plotted below. The eigenvalues were found by the relation $|A - \lambda I| = 0$, where the real and imaginary parts of the eigenvalues are plotted on their corresponding axis. It is noticed that the real and imaginary parts increase in magnitude, where the increasing negative real part correlates to larger pitch stiffness as expected.

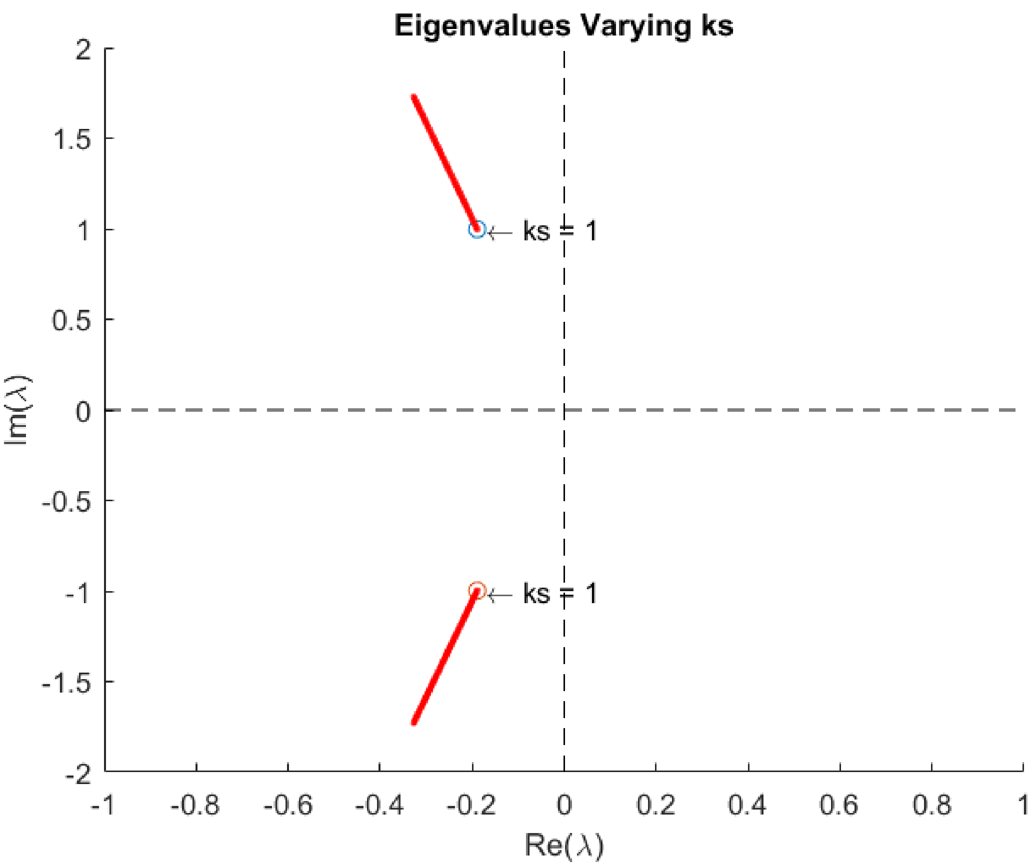


Figure 4. Question 3, Part a, Eigenvalues Plot

B. Part b:

Using the relations for the change in the components in the A matrix with changing pitch stiffness, we can derive control constants to give us these changes in pitch stiffness using the control law: $\Delta\delta_e = -k_2\Delta\theta - k_1\Delta q$. This is done by relating the components of the characteristic equation with the control law implemented to the the components of the characteristic equation we want after changing the pitch stiffness by a scalar ks, as done in part a. The derivations showing this are as such:

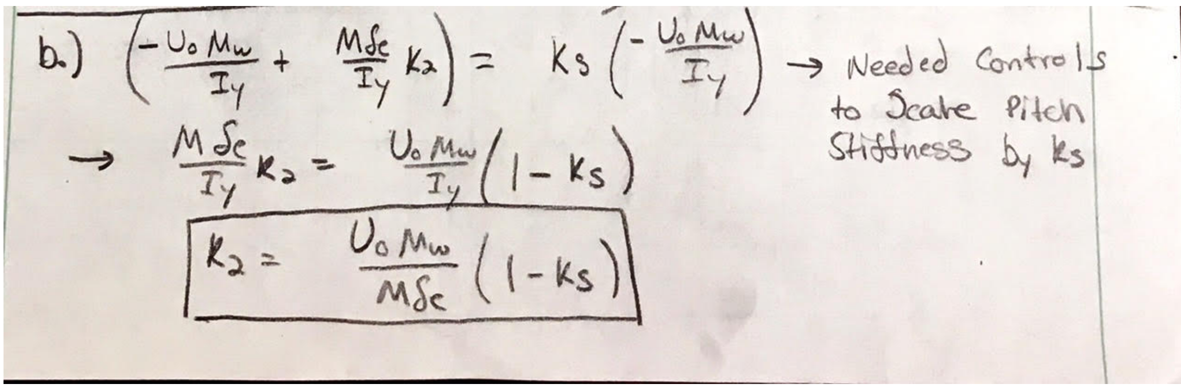


Figure 5. Question 3, Part b, Derivations

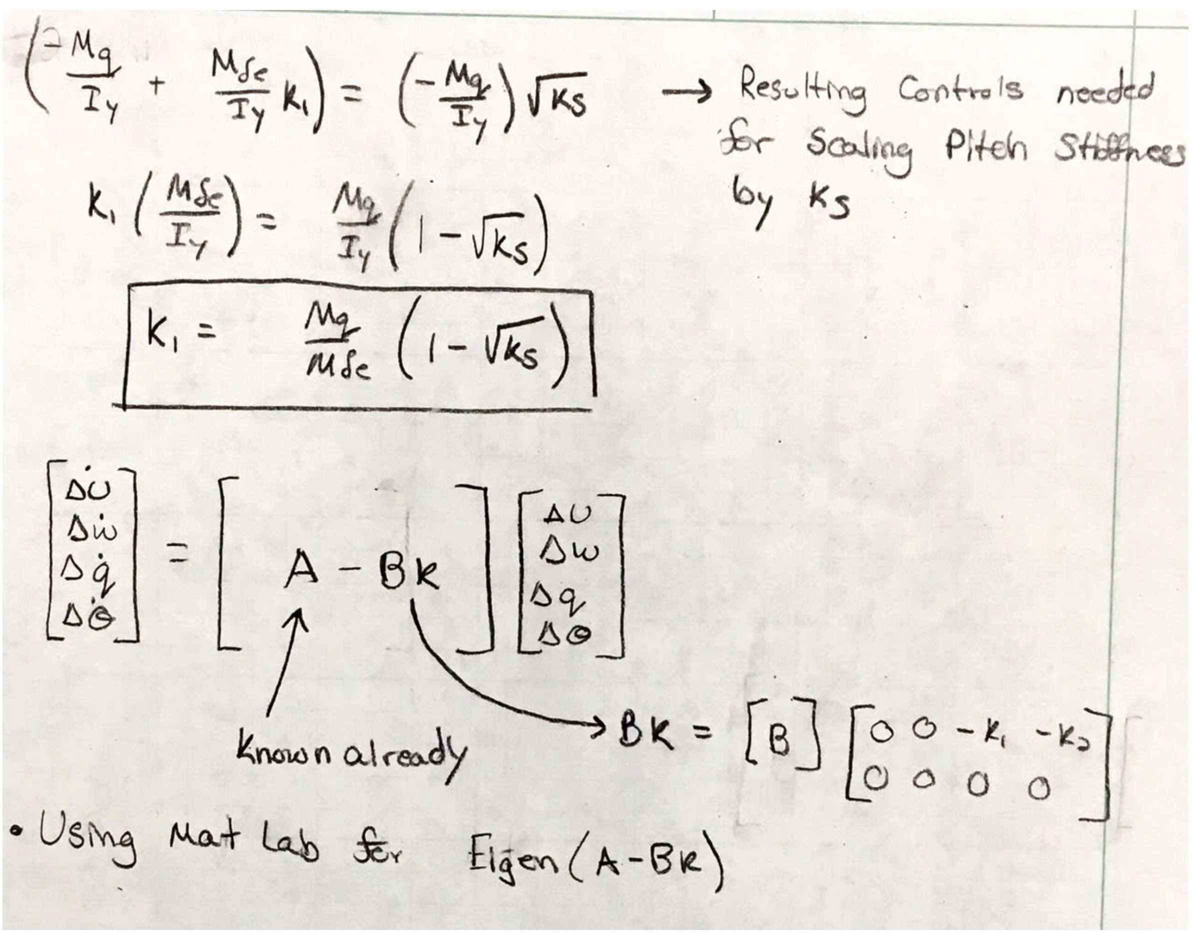


Figure 6. Question 3, Part b, Derivations

Now that we have the gain values k1 and k2 in terms of our knowns, we can use the results from the more simplified reduced 2x2 model for a short period approximation and apply it to the full linearized model. This is done by implementing the B Matrix we solved for in problem 2, so that we have a new closed loop state matrix, A - BK, as shown in the derivations. This Matrix gives us the total picture of the controls need to change the matrix A so that the pitch stiffness of the total system is changing by the scale factor ks. The resulting eigenvalues of this closed loop state matrix are plotted below, where the eigenvalues with smaller real and imaginary parts are for the eigenvalues of the phugoid mode and the larger real and imaginary eigenvalues are for the short period mode. We know this to be true since the more negative real parts have faster time constants.

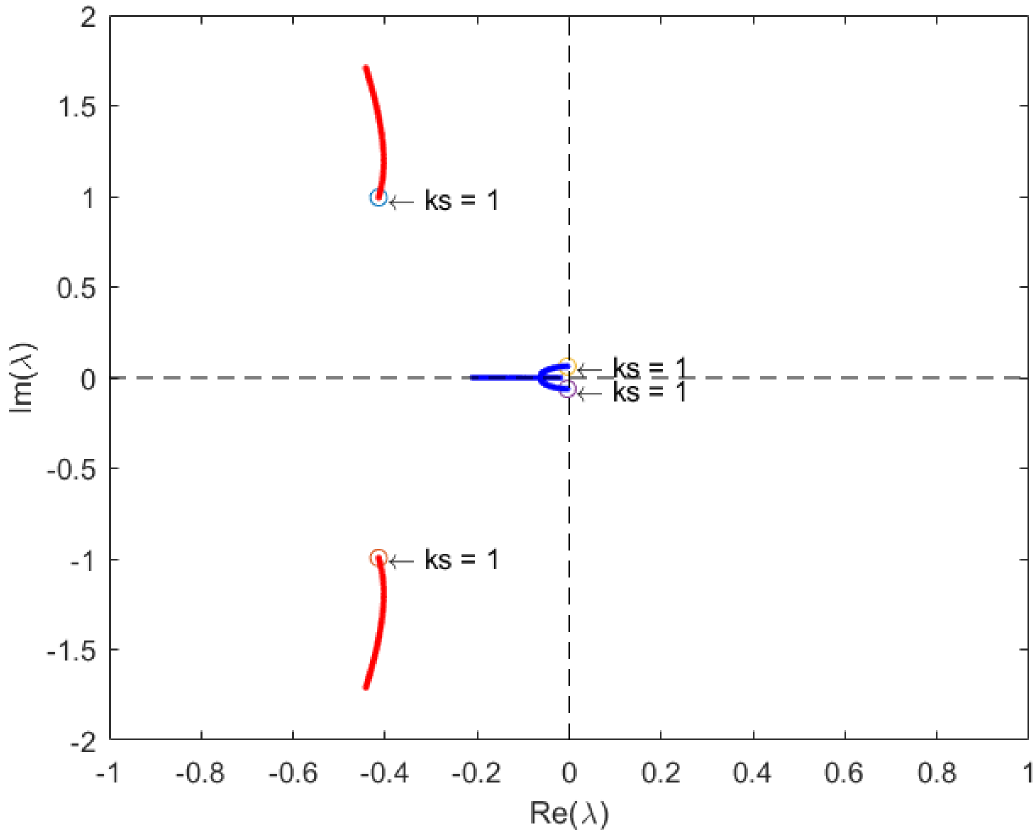


Figure 7. Question 3, Part b, Eigenvalues Plot

The general trend for the eigenvalues show that the short period mode eigenvalues stay near constant for the real parts and increase with ks for the imaginary parts. This is opposition to the trend of the phugoid mode, which is increasing in its eigenvalues real part and decreasing in the eigenvalues imaginary part. This shows that the short period mode is becoming more damped and the phugiod mode is becoming effectively damped.

C. Part c:

1. Simulation with ks = 1 (Unchanged)

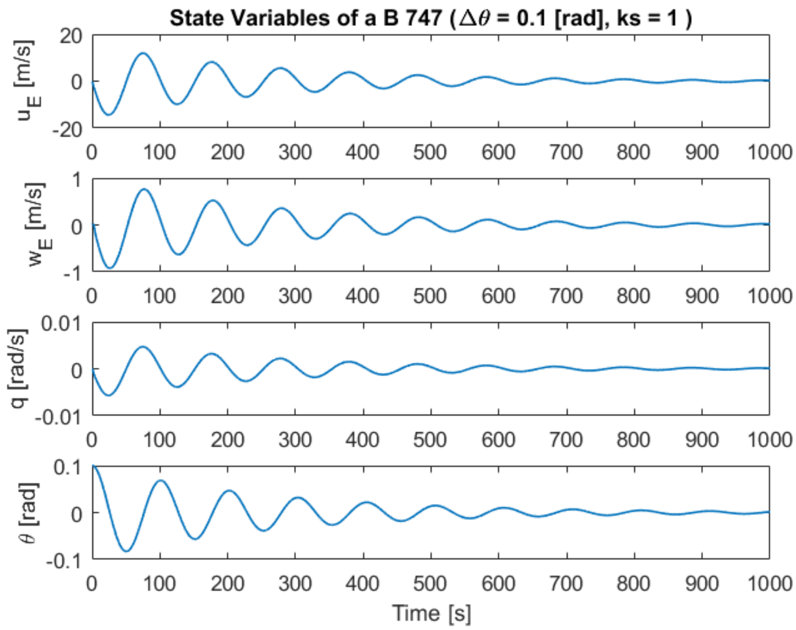


Figure 8. Question 3, Part c, ks = 1

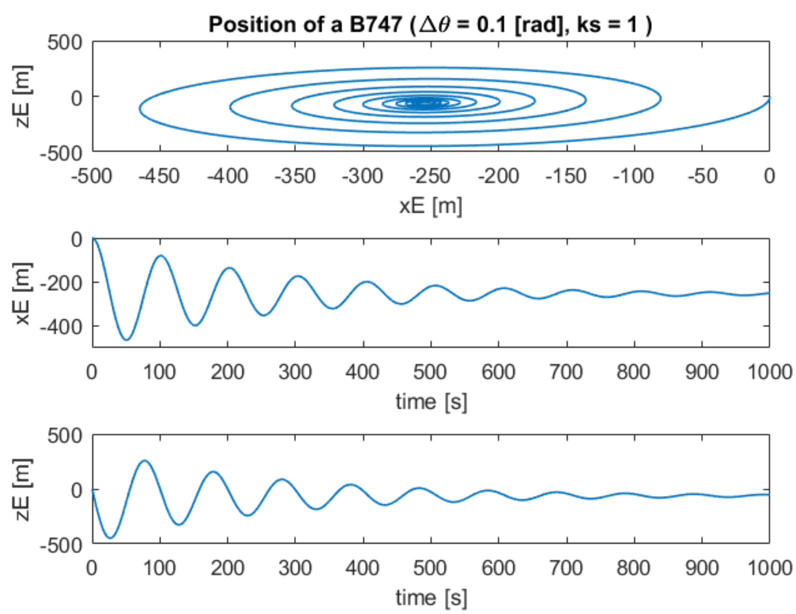


Figure 9. Question 3, Part c, ks = 1

2. Simulation with ks = 2

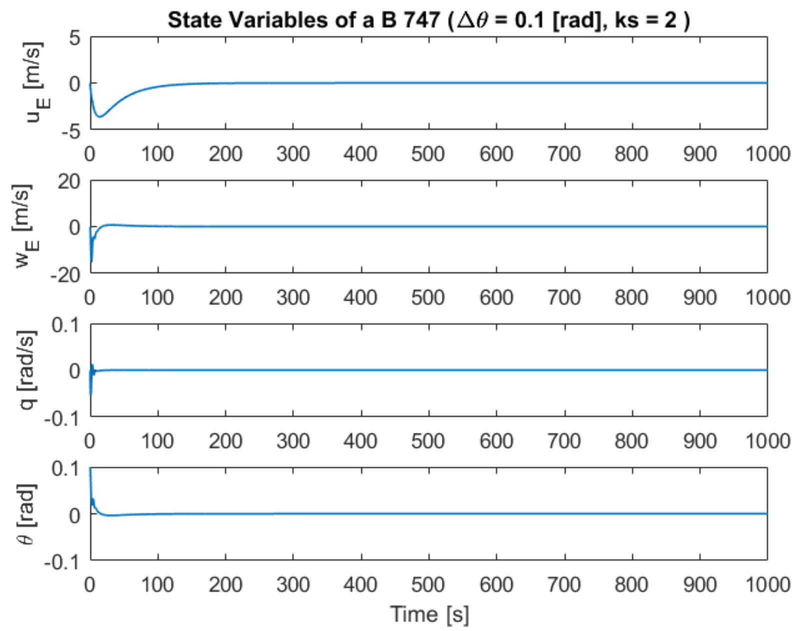


Figure 10. Question 3, Part c, ks = 2

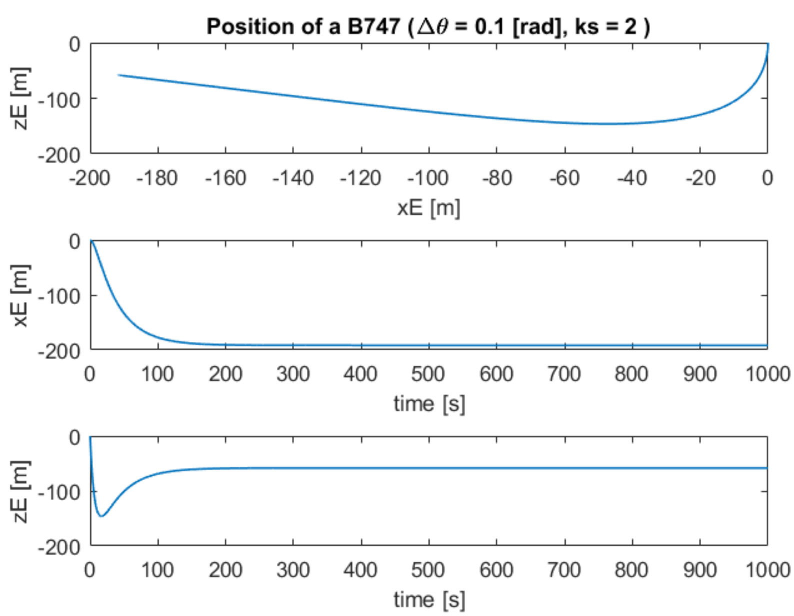


Figure 11. Question 3, Part c, ks = 2

As can be seen in the plots above the results from when ks = 1, where the controls for the short period mode are zero, have much different oscillations for the $\Delta\theta = 0.1$ pertubations, than the case where ks = 2 and the pitch stiffness is doubled. As expected the phugoid mode is driven to have nearly no oscillation in a much shorter time than the uncontrolled case. This implies that the phugoid mode changes when controls are implemented to change the short period mode, which is due to the coupled nature of state variables, where changing the pitch stiffness by a scale factor in the short period mode requires controls that will inherently effect the state variables and cause an induced effect on the phugoid mode. Since we are increasing pitch stiffness, the real parts of the modes increase slightly as the time constant gets reduced. The imaginary parts of the short period mode grow and imaginary parts of the phugoid mode approach zero. This is due to the oscillation of the phugoid mode being damped to zero as the pitch stiffness increases, which intuitively makes sense. The oscillation due to the imaginary parts of the short period mode increase as the controls are excited and the short period is heavily damped.

D. Part d:

- Our Control Matrix K is given by:

$$K = \begin{bmatrix} 0 & 0 & -K_1 & -K_2 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

- When multiplying by our B Matrix to get all terms of our Control in terms of the state variables $[\Delta u \; \Delta w \; \Delta q \; \Delta \theta]^T$, we notice that we have undesigned consequence for the controls K_1 & K_2 as seen:

$$\begin{bmatrix} \Delta \dot{u} \\ \Delta \dot{w} \\ \Delta \dot{q} \\ \Delta \dot{\theta} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} - B_{11} K_1 & a_{14} - B_{11} K_2 \\ a_{21} & a_{22} & a_{23} - B_{21} K_1 & a_{24} - B_{21} K_2 \\ a_{31} & a_{32} & a_{33} - B_{31} K_1 & a_{34} - B_{31} K_2 \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \longrightarrow \text{Undesigned Result!}$$

Figure 12. Question 3, Part d, Derivations

When analyzing the coupled effect of controls for the short period mode on the phugoid mode, there were are some key insights to why this happens. First, there is an undesigned effect that results when we implement controls for the short mode as can be seen in the image above. The upper half of the closed loop matrix ends up receiving the controls design for the short period. Second, there is also a natural coupling in the A matrix between the rotational and translational state variables. Since the four variables are coupled, they changed together, which has the inherent effect of changing how the system reacts. In this case the phugoid mode was changed due this natiral coupling between the state variables. This dependence is most likely the dominating contributor to the phugoid mode changing with the controls implemented to change the short period mode since the values from the changing B matrix are seen to small relative to the original A matrix.

E. Appendix A - MATLAB Code

1. Script that Answers Questions 2-3a,b

```
%% Author: Jack Lambert
% ASEN 3129
% Homework 7
% Problem 3 Parts a,b
% Purpose: To control the eigen values by creating gains that give us
% desired pitch stiffness. We will see trends in the changing eigenvalues
% for the phugoid and short period mode with changing the pitch stiffness
% by a scalar factor ks
%% Airplane Parameters
% Nondimensional Derivatives
% Table 6.1 -
Cx = [-.108, .2193, 0, 0];
Cz = [-.106, -4.92, -5.921, 5.896];
Cm = [.1043, -1.023, -23.92, -6.314];

% Nondimensional Elevator Derivatives (Page 229 in Etkin)
C_x_de = -3.818*10^-6;
C_z_de = -0.3648;
C_m_de = -1.444;

% Table E.1 B747 Case 3
Alt = 40000*(0.3048); % Altitude [ft] -> [m]
[T, a, P, rho] = atmosisa(Alt); % Standard Atmosphere Properties at Alt.
W = 6.366*10^5*4.44822; % Weight [lb]->[N]
Ix_PA = 1.82*10^7*1.35581795; % Moment of Interia x-PA [slug ft^2]-> [kg m^2]
Iy_PA = 3.31*10^7*1.35581795; % Moment of Interia y-PA [slug ft^2]-> [kg m^2]
Iz_PA = 4.97*10^7*1.35581795; % Moment of Interia z-PA [slug ft^2]-> [kg m^2]
Izx_PA = 9.70*10^5*1.35581795; % Moment of Interia zx-PA [slug ft^2]-> [kg m^2]
zeta = -2.4; % Angle between Stability Axis and PA [degrees]
I = [Ix_PA, 0,-Izx_PA;...
    0, Iy_PA,0;...
    -Izx_PA, 0, Iz_PA]; % Inertia Matrix in PA
Q_PA_SA = [cosd(zeta), 0, -sind(zeta);...
    0, 1, 0;...
    sind(zeta), 0, cosd(zeta)]; % Transformation Matrix [PA-SA]
I_SA = Q_PA_SA * I * Q_PA_SA'; % MOI in Stability axis Frame
Ix = I_SA(1,1); % Moment of Interia x-SA [kg m^2]
Iy = I_SA(2,2); % Moment of Interia y-SA [kg m^2]
Iz = I_SA(3,3); % Moment of Interia z-SA [kg m^2]
Izx = (1/2)*(Ix-Iz)*sind(2*zeta)+Izx_PA*...
    (sind(zeta)^2-cosd(zeta)^2); % Moment of Interia zx-SA [kg m^2]
CD = .043; % Coefficient of Drag
cbar = 27.31*(0.3048); % Mean Chord Length [ft]->[m]
S = 5500*(0.3048)^2; % Surface Area [ft^2]->[m^2]
g = 9.81; % Gravity Constant [m/s^2]
m = W/g; % Mass of Plane [kg]

%% Trim States
Vel = 871*(0.3048);% Velocity [ft/s] -> [m/s]
u0 = Vel; % Initial Velocity in x-coord - Stability Axis Frame (Trim State)
theta0 = 0; % Initial Pitch Angle [deg]
Cw0 = W/(.5*rho*S*u0^2);
%% Function that Computes Dimensional Derivatives from Non-Dimenional derivatives
[X, Z, M, X_c, Z_c, M_c ] = NonDimLong(rho,u0,S,W,theta0,Cx,Cz,Cm,cbar,C_x_de,C_z_de,C_m_de);

%% State Variable Matrix A
row1 = [X(1)/m, X(2)/m, 0, -g*cosd(theta0)];
row2 = [Z(1)/(m-Z(4)), Z(2)/(m-Z(4)), (Z(3)+m*u0)/(m-Z(4)), (-W*sind(theta0))/(m-Z(4))];
row3 = [(1/Iy)*(M(1) + ((M(4)*Z(1))/(m-Z(4))) ),...
        (1/Iy)*(M(2) + ((M(4)*Z(2))/(m-Z(4))) ),...
        (1/Iy)*(M(3) + ((M(4)*(Z(3)+m*u0))/(m-Z(4))) ),...
        -((M(4)*W*sind(theta0))/(Iy*(m-Z(4))))];
row4 = [0, 0, 1, 0];

A = [row1;row2;row3;row4];

%% Input Matrix B
% Dimensionalizing Elevator Derivative

% Compenents of B Matrix
```

```
row1_C = [X_c(1)/m, X_c(2)/m];
row2_C = [Z_c(1)/(m-Z(4)), Z_c(2)/(m-Z(4))];
row3_C = [M_c(1)/Iy + (M(4)*Z_c(1))/(Iy*(m-Z(4))), M_c(2)/Iy + (M(4)*Z_c(2))...
    /(Iy*(m-Z(4)))];
row4_C = [0, 0];

B = [row1_C;row2_C;row3_C;row4_C];

%% Part a
ks = 1:0.01:3;

figure
for i = 1:length(ks)
    A1 = [M(3)/Iy * sqrt(ks(i)), (u0*M(2)/Iy)*ks(i);
    1, 0];
    [eV1,eVal1] = eig(A1);
    modes = diag(eVal1);
    hold on
    plot(real(modes(1)),imag(modes(1)),'.r')
    plot(real(modes(2)),imag(modes(2)),'.r')
     if i == 1
         % To see initial state at ks = 1
         plot(real(modes(1)),imag(modes(1)),'-o')
         text(real(modes(1)),imag(modes(1)),' \leftarrow ks = 1')
         plot(real(modes(2)),imag(modes(2)),'-o')
         text(real(modes(2)),imag(modes(2)),' \leftarrow ks = 1')

     end

    plot([0,0],[-2,2],'--k')
    plot([-1,1],[0,0],'--k')

    title('Eigenvalues Varying ks')
    xlabel('Re(\lambda)')
    ylabel('Im(\lambda)')

end
hold off

%% Part b
figure
for i = 1:length(ks)
    k1 = (M(3)/M_c(1))*(1-ks(i)^(1/2));
    k2 = ((u0*M(2))/M_c(1))*(1-ks(i));
    K = [0, 0, k1, k2;
        0, 0, 0, 0];
    A_BK = A-B*K;

    [eV2,eVal2] = eig(A_BK);
    modes = diag(eVal2);
    plot(real(modes(1)),imag(modes(1)),'.r')
    hold on
    plot(real(modes(2)),imag(modes(2)),'.r')
    plot(real(modes(3)),imag(modes(3)),'.b')
    plot(real(modes(4)),imag(modes(4)),'.b')

    if i == 1
        % To see initial state at ks = 1
        plot(real(modes(1)),imag(modes(1)),'-o')
        text(real(modes(1)),imag(modes(1)),' \leftarrow ks = 1')
        plot(real(modes(2)),imag(modes(2)),'-o')
        text(real(modes(2)),imag(modes(2)),' \leftarrow ks = 1')
        plot(real(modes(3)),imag(modes(3)),'-o')
        text(real(modes(3)),imag(modes(3)),' \leftarrow ks = 1')
        plot(real(modes(4)),imag(modes(4)),'-o')
        text(real(modes(4)),imag(modes(4)),' \leftarrow ks = 1')
    end

    plot([0,0],[-2,2],'--k')
    plot([-1,1],[0,0],'--k')
```

```
    xlabel('Re(\lambda)')
    ylabel('Im(\lambda)')
    legend('Phigoid Mode', 'Short Period Mode')

end


check = 1;
```

2. *Non-Dimensional Coefficients to Dimensional Coefficients Function*

```
%% Author: Jack Lambert
% ASEN 3128
% Purpose: This function calculates the dimensional derivatives for the
% state matrix and the control matrix
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [X, Z, M, X_c, Z_c, M_c ] = NonDimLong(rho,u0,S,W,theta0,Cx,Cz,...
    Cm,cbar,C_x_de,C_z_de,C_m_de)

% Computing the Nondimesnional Inital Weight Derivative
Cw0 = W/ ((1/2)*rho*S*u0^2);

%% State Variable Derivatives
% X
Xu = rho*u0*S*Cw0*sind(theta0) + .5*rho*u0*S*Cx(1);
Xw = .5*rho*u0*S*Cx(2);
Xq = .25*rho*u0*cbar*S*Cx(3);
Xwdot = .25*rho*cbar*S*Cx(4);

X = [Xu, Xw, Xq, Xwdot]';

% Z
Zu = -rho*u0*S*Cw0*cosd(theta0) + .5*rho*u0*S*Cz(1);
Zw = .5*rho*u0*S*Cz(2);
Zq = .25*rho*u0*cbar*S*Cz(3);
Zwdot = .25*rho*cbar*S*Cz(4);

Z = [Zu, Zw, Zq, Zwdot]';

% M
Mu = .5*rho*u0*cbar*S*Cm(1);
Mw = .5*rho*u0*cbar*S*Cm(2);
Mq = .25*rho*u0*(cbar^2)*S*Cm(3);
Mwdot = .25*rho*(cbar^2)*S*Cm(4);

M = [Mu, Mw, Mq, Mwdot]';

%% Control Derivatives

% Elevator controls
X_c(1) = 1/2*rho*u0^2*S*C_x_de;
Z_c(1) = 1/2*rho*u0^2*S*C_z_de;
M_c(1) = 1/2*rho*u0^2*S*cbar*C_m_de;

% Thrust Controls
X_c(2) = 0;
Z_c(2) = 0;
M_c(2) = 0;
```

3. *Function that Computes the Closed Loop Matrix*

```
%% Author: Jack Lambert
% ASEN 3128
```

```
% Homework 7
% Purpose: To keep all constants in one function so they are not defined
% more than once and then Compute the xonstants for the state variable
% matrix A and the input matrix B. This function also provide the trim
% states
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [A_BK,theta0,u0] = Linearizedset(ks)
%% Airplane Parameters
% Nondimensional Derivatives
% Table 6.1 -
Cx = [-.108, .2193, 0, 0];
Cz = [-.106, -4.92, -5.921, 5.896];
Cm = [.1043, -1.023, -23.92, -6.314];

% Nondimensional Elevator Derivatives (Page 229 in Etkin)
C_x_de = -3.818*10^-6;
C_z_de = -0.3648;
C_m_de = -1.444;

% Table E.1 B747 Case 3
Alt = 40000*(0.3048); % Altitude [ft] -> [m]
[T, a, P, rho] = atmosisa(Alt); % Standard Atmosphere Properties at Alt.
W = 6.366*10^5*4.44822; % Weight [lb]->[N]
Ix_PA = 1.82*10^7*1.35581795; % Moment of Interia x-PA [slug ft^2]-> [kg m^2]
Iy_PA = 3.31*10^7*1.35581795; % Moment of Interia y-PA [slug ft^2]-> [kg m^2]
Iz_PA = 4.97*10^7*1.35581795; % Moment of Interia z-PA [slug ft^2]-> [kg m^2]
Izx_PA = 9.70*10^5*1.35581795; % Moment of Interia zx-PA [slug ft^2]-> [kg m^2]
zeta = -2.4; % Angle between Stability Axis and PA [degrees]
I = [Ix_PA, 0,-Izx_PA;...
    0, Iy_PA,0;...
    -Izx_PA, 0, Iz_PA]; % Inertia Matrix in PA
Q_PA_SA = [cosd(zeta), 0, -sind(zeta);...
    0, 1, 0;...
    sind(zeta), 0, cosd(zeta)]; % Transformation Matrix [PA-SA]
I_SA = Q_PA_SA * I * Q_PA_SA'; % MOI in Stability axis Frame
Ix = I_SA(1,1); % Moment of Interia x-SA [kg m^2]
Iy = I_SA(2,2); % Moment of Interia y-SA [kg m^2]
Iz = I_SA(3,3); % Moment of Interia z-SA [kg m^2]
Izx = (1/2)*(Ix-Iz)*sind(2*zeta)+Izx_PA*...
    (sind(zeta)^2-cosd(zeta)^2); % Moment of Interia zx-SA [kg m^2]
CD = .043; % Coefficient of Drag
cbar = 27.31*(0.3048); % Mean Chord Length [ft]->[m]
S = 5500*(0.3048)^2; % Surface Area [ft^2]->[m^2]
g = 9.81; % Gravity Constant [m/s^2]
m = W/g; % Mass of Plane [kg]

%% Trim States
Vel = 871*(0.3048);% Velocity [ft/s] -> [m/s]
u0 = Vel; % Initial Velocity in x-coord - Stability Axis Frame (Trim State)
theta0 = 0; % Initial Pitch Angle [deg]
Cw0 = W/(.5*rho*S*u0^2);
%% Function that Computes Dimensional Derivatives from Non-Dimenional derivatives
[X, Z, M, X_c, Z_c, M_c ] = NonDimLong(rho,u0,S,W,theta0,Cx,Cz,Cm,cbar,C_x_de,C_z_de,C_m_de);

%% State Variable Matrix A
row1 = [X(1)/m, X(2)/m, 0, -g*cosd(theta0)];
row2 = [Z(1)/(m-Z(4)), Z(2)/(m-Z(4)), (Z(3)+m*u0)/(m-Z(4)), (-W*sind(theta0))/(m-Z(4))];
row3 = [(1/Iy)*(M(1) + ((M(4)*Z(1))/(m-Z(4))) ),...
        (1/Iy)*(M(2) + ((M(4)*Z(2))/(m-Z(4))) ),...
        (1/Iy)*(M(3) + ((M(4)*(Z(3)+m*u0))/(m-Z(4))) ),...
        -((M(4)*W*sind(theta0))/(Iy*(m-Z(4))))];
row4 = [0, 0, 1, 0];

A = [row1;row2;row3;row4];

%% Input Matrix B
% Dimensionalizing Elevator Derivative

% Compenents of B Matrix
row1_C = [X_c(1)/m, X_c(2)/m];
```

```
row2_C = [Z_c(1)/(m-Z(4)), Z_c(2)/(m-Z(4))];
row3_C = [M_c(1)/Iy + (M(4)*Z_c(1))/(Iy*(m-Z(4))), M_c(2)/Iy + (M(4)*Z_c(2))...
    /(Iy*(m-Z(4)))];
row4_C = [0, 0];

B = [row1_C;row2_C;row3_C;row4_C];

% Conrols with Varying Stiffness (scales by ks)
k1 = (M(3)/M_c(1))*(1-ks^(1/2));
k2 = ((u0*M(2))/M_c(1))*(1-ks);
K = [0, 0, k1, k2;
    0, 0, 0, 0];
A_BK = A-B*K;


end
```

4. Main Function

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Author: Jack Lambert
% Dale Lawrence
% Aircraft Dynmaics Homework 7
% Problem 3
% Purpose: Sets Initial COnditions for each Pertubation Case and Calls ODE45
% to plot the State Variables vs time
% Date Modefied: 2/12/18
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ODE45 Variable Allocation
%                       X_E = z(1); % z-position, Inerital Frame
%                       Z_E = z(2); % z-position, Inerital Frame
%                       u_dot = z(3); % x-component of Velocity, Body Frame
%                       z_dot = z(4); % x-component of Velocity, Body Frame
%                       q_dot = z(6); % Angular Velocity about the y-axis [rad/s]
%                       theta_dot = z(5); % Pitch Angle
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Initial Conditions
c1 = 0; % xE: Location in Inertial Coordinates [m]
c2 = 0; % zE: Location in Inertial Coordinates [m]
c3 = 0; % Delta U: x-comp, BF Interial Velocity [m/s]
c4 = 0; % Delta W: z-comp, BF Interial Velocity [m/s]
c5 = 0; % Delta q: y-comp, BF Angular Velocity [rad/s]
c6 = 0.1; % Delta Theta: Pitch Angle

condition = [c1 c2 c3 c4 c5 c6];
ks = [1, 2]; % What we are scaling the pitch stiffness by

%% State Variables vs. Time
t = [0 200]; % Larger times to see phugoid mode, shorter for short period mode


string = ["ks = 1","ks = 2"]; % Title for Varying IC's
% Phugoid Response (Longer Time)
for i = 1:2
    % Calling ODE45
    [t,z] = ode45(@(t,y) ODEcall(t,y,ks(i)),t,condition);


    % U_E vs time
    figure
    subplot(4,1,1)
    plot(t ,z(:,3),'Linewidth',1)
    tit = sprintf('%s %s %s','State Variables of a B 747 (\Delta\theta = 0.1 [rad],',string(i),')');
    title(tit)
    ylabel('u_E [m/s]')
```

```
    % W_E vs time
    subplot(4,1,2)
    plot(t ,z(:,4),'Linewidth',1)
    ylabel('w_E [m/s]')

    % q vs time
    subplot(4,1,3)
    plot(t ,z(:,5),'Linewidth',1)
    ylabel('q [rad/s]')

    % Theta vs time
    subplot(4,1,4)
    plot(t ,z(:,6),'Linewidth',1)
    ylabel('\theta [rad]')
    xlabel('Time [s]')

end

%% Plotting Position
for i = 1:2
    [t,z] = ode45(@(t,y) ODEcall(t,y,ks(i)),t,condition);

    % xE vs zE
    figure
    subplot(3,1,1)
    plot(z(:,1) ,z(:,2),'Linewidth',1)
    tit = sprintf('%s %s %s','Position of a B747 (\Delta\theta = 0.1 [rad],',string(i),')');
    title(tit)
    xlabel('xE [m]')
    ylabel('zE [m]')

    % xE vs t
    subplot(3,1,2)
    plot(t ,z(:,1),'Linewidth',1)
    xlabel('time [s]')
    ylabel('xE [m]')

   % zE vs t
    subplot(3,1,3)
    plot(t ,z(:,2),'Linewidth',1)
    xlabel('time [s]')
    ylabel('zE [m]')

end
```

5. *Function ODE45 calls for the Linearized Longitudinal Set*

```
%% Author: Jack Lambert
% ASEN 3128
% Problem 3
% Purpose: Function for ODE45 to call to calculate the State variables xE,
% zE, u_dot, w_dot, q_dot, and theta_dot. This function uses the simplified
% assumptions for the Linearized Longitudinal Dynamics Set
% Last Edited: 3/11/2018
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [dydt] = ODEcall(t,y,ks)

X_E = y(1); % x-position, Inerital Frame
Z_E = y(2); % z-position, Inerital Frame
u_dot = y(3); % x-component of Velocity, Body Frame
w_dot = y(4); % z-component of Velocity, Body Frame
q_dot = y(5); % y-component of Angular Velocity, Body Frame
theta_dot = y(6); % Pitch Angle

%% State Variable Matrix for Linearized Longitudinal Set
```

```
[A_BK,theta0,u0] = Linearizedset(ks); % A matrix function based on plane and parameters
State = [u_dot, w_dot, q_dot, theta_dot]'; % Couple State Variables in Long. Set
var = A_BK*State; % Couple State Variables in Long. Set
%% Solving for Inertial Position
dydt(1) = u_dot*cosd(theta0) + w_dot*sind(theta0) - u0*theta_dot*sind(theta0); % xE
dydt(2) = -u_dot*sind(theta0) + w_dot*cosd(theta0)-u0*theta_dot*cosd(theta0); % zE
%% Solving for State Variables in the Linearized Longitudinal Set
dydt(3) = var(1); % uE
dydt(4) = var(2); % wE
dydt(5) = var(3); % q
dydt(6) = var(4); % theta

dydt = dydt'; % Inverts for ODE45
end
```