ASEN 3128 Homework 10

Jack Lambert*
University of Colorado Boulder, ASEN 3128-012, Group 15
Submitted 4/23/2018

I. Question 1 - Dimensional Stability Derivatives

Using the non-dimensional Stability coefficient data from Table 6.6 in the textbook, together with the data in Table E.1 in Appendix Em for case II of a Boeing 747 airplane, the following dimensional stability derivatives were found:

Table 1. Lateral Dimensional Derivatives

	Y (N)	L (Nm)	N (Nm)
v (m/s)	-23092	-4.39e5	3.06e5
p (rad/s)	0	-1.54e7	-1.91e6
r (rad/s)	0	1.42e7	-1.28e7

II. Question 2 - A Matrix:

A. Part A :

Implementing the conditions of a Boeing 747 flying at an altitude of 20,000 feet, using the values provided from Table E.1 for case II, the full A matrix is provided below:

$$A = \begin{vmatrix} -0.0800 & 0 & -157.9 & 9.81 \\ -0.01843 & -0.619 & 0.604 & 0 \\ 0.00558 & 0.00610 & -0.225 & 0 \\ 0 & 1 & 0 & 0 \end{vmatrix}$$

III. Question 3:

When taking the eigenvalues of the lateral dynamics A matrix, it was seen that there are three different modes, constituted of the roll, spiral, and dutch roll modes. To identify which eigenvalues corresponded to which modes, the use real and imaginary parts of the eigenvectors were used. Since the only mode that had imaginary parts and inherently, a complex conjugate pair of eigenvalues, was the Dutch roll. The roll mode was noticed to be the eigenvalues with the larger negative real parts and the spiral mode was the mode with the smaller real imaginary parts. using these classifications the following expressions were used to calculate the natural frequency, dampening ratio, and time constant:

$$\omega_n = \sqrt{\omega^2 + n^2}, \qquad \zeta = -\frac{n}{\omega_n}, \qquad \tau = -\frac{1}{n} \tag{1}$$

Where n and ω are the real and imaginary parts of the eigenvalues, respectively. The numerical values for the variables above for both the phugoid and short period mode are provided below:

*SID: 104414093

A. Dutch Roll:

Dutch Roll Mode:

$$Eigenvector_1 = \begin{vmatrix} 0.9997 + 0.0000i \\ -0.0104 - 0.0127i \\ 0.0009 + 0.0056i \\ 0.0139 - 0.0096i \end{vmatrix} \tag{2}$$

$$Eigenvector_2 = \begin{vmatrix} 0.9997 + 0.0000i \\ -0.0104 + 0.0127i \\ 0.0009 - 0.0056i \\ 0.0139 + 0.0096i \end{vmatrix}$$

$$\lambda_{1,2} : -0.0838 \pm 0.9709i \tag{3}$$

$$\omega_n : 0.9745[rad/s] \tag{4}$$

$$\zeta : 0.086 \tag{5}$$

$$\tau : 11.94[s] \tag{6}$$

Roll Mode:

$$Eigenvector_1 = \begin{vmatrix} -0.9441 \\ -0.1969 \\ 0.0124 \\ 0.2641 \end{vmatrix} \tag{7}$$

$$\lambda : -0.7455 \tag{8}$$

$$\tau : 1.341[s] \tag{9}$$

Spiral Mode:

$$Eigenvector_1 = \begin{vmatrix} 0.9201 \\ -0.0042 \\ 0.0239 \\ 0.3908 \end{vmatrix} \tag{10}$$

$$\lambda : -0.0107 \tag{11}$$

$$\tau : 93.71[s] \tag{12}$$

IV. Question 4:

Using the simplifications for the dutch roll dynamics approximation the following eigenvalues:

$$\lambda_{1,2} : -0.1524 \pm 0.9360i \tag{13}$$

When comparing the eigenvalues of the dutch roll approximation in comparison to the full lateral dynamics model it was noticed that the real part of the eigenvalues for the approximation had larger negative real parts. This meaning that the approximation would damp the system faster as the real part drives the envelope, which has larger negative parts, therefore, would return to trim much faster. The complex parts of the eigenvalues were relatively close, meaning that the time constants would be similar.

V. Question 5:

A. Part a:

Simulating the full linearized lateral dynamics model, the response of Boeing 747 at trim is plotted below:

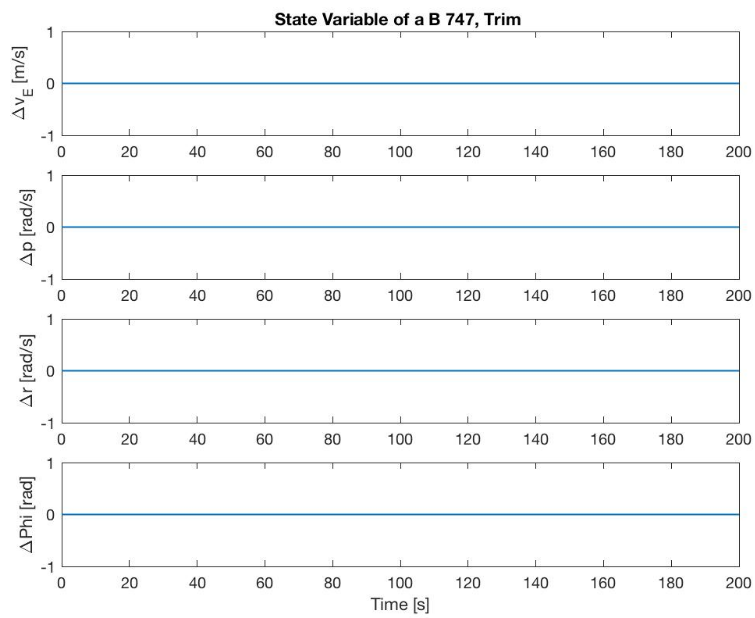

Figure 1. Trim

As can be see by the state variable response, the trim state is an equilibrium state as the state variable all stay at their nominal conditions of trim.

B. Part b:

Implementing the same full linearized lateral dynamics model as in part a, but this time with perturbations about trim the following results are plotted below. Perturbations are defined about the state vector in the following scenarios as - $y(0) = [\Delta v(0), \Delta p(0), \Delta r(0), \Delta\phi(0)]^T$ - with units in SI.

1. *Case I* - $y(0) = [10, 0, 0, 0]^T$

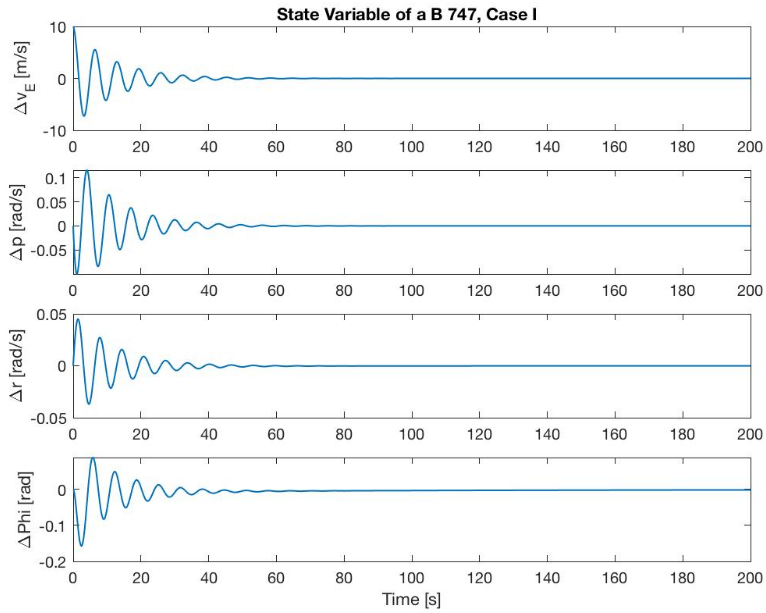

Figure 2. Case I

2. *Case II* - $y(0) = [0, 0.1, 0, 0]^T$

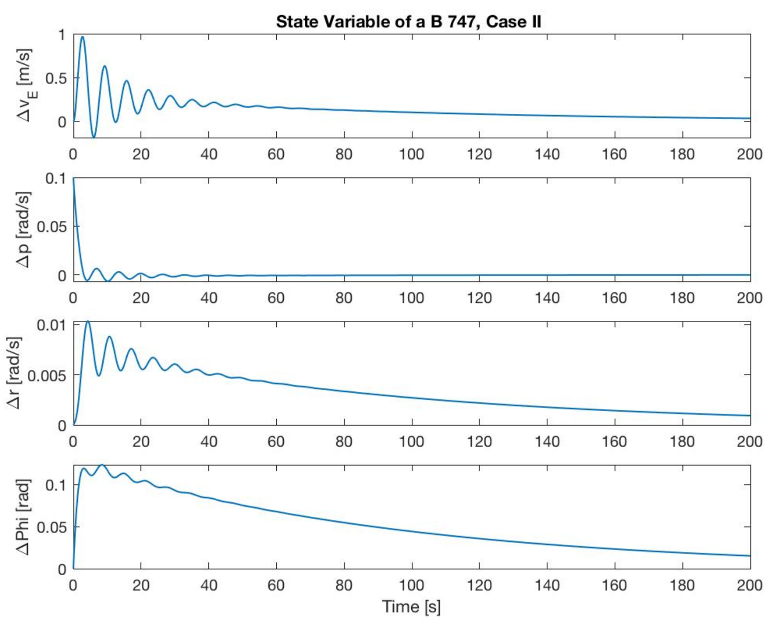

Figure 3. Case II

3. *Case III* - $y(0) = [-3.0747, -0.2556, 0.0038, 0.3015]^T$

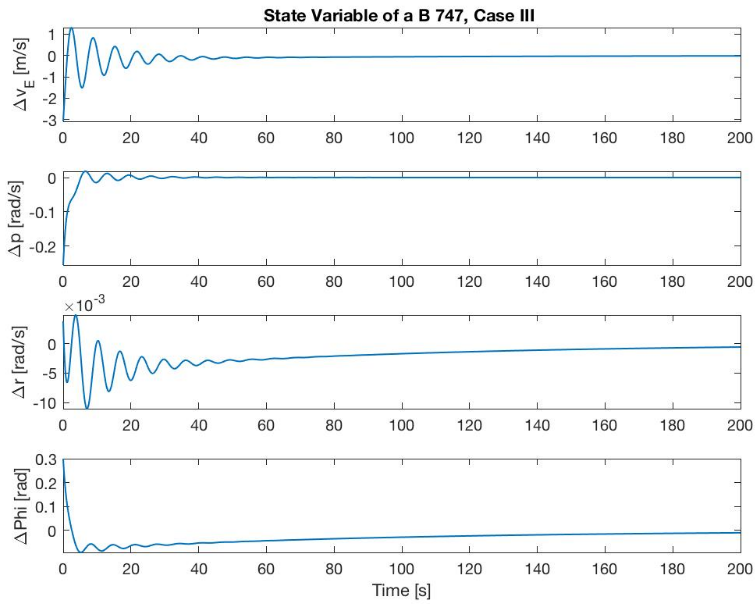

Figure 4. Case III

4. *Case IV* - $y(0) = [1.1039, -0.0051, 0.0287, 0.4697]^T$

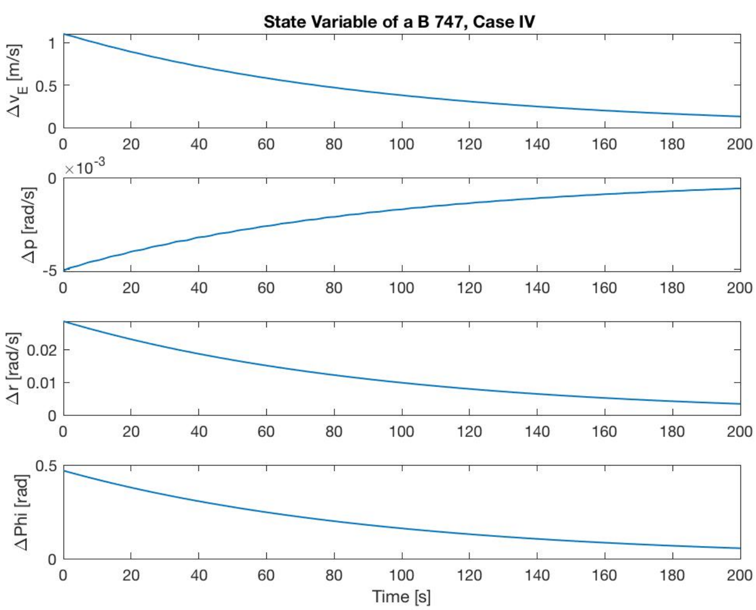

Figure 5. Case I

C. Part c

As can be seen by the response of the system, the spiral mode is excited by deviations in yaw rate, roll rate, and . The spiral mode can be easily seen as it is the least damped mode and therefore is represented in the plots by the slow return of the state variables to nominal. The roll mode has no complex parts, which is why it does not oscillate about the envelope. This makes sense since the side slip velocity, roll rate, and bank angle barely change in a spiral motion, where yaw is drastically changing. The roll mode is illustrated in the plots in a similar way to the spiral mode as it also has no complex parts, therefore has no oscillatory motion. The roll mode is more damped than the spiral mode, therefore returns to nominal conditions much faster than the spiral mode. This mode is seen to be excited by perturbations in roll. Lastly, the dutch roll mode is easy to identify as well since it is the only mode with complex conjugate pairs. This means that the response has oscillatory motion. This mode is seen to excited by deviations in side slip velocity, roll rate, and yaw rate. This mode is a cause of coupling in the state variable which is why it makes sense that all the state variables can excite it.

D. Appendix A - MATLAB Code

1. Calculates Modal Properties of the A Matrix and Dutch Roll Approximation A Matrix

```
%% Author: Jack Lambert
% ASEN 3128
% Homework 9
% Purpose: Find the dimensional derivatives given the none dimensional
% derivatives from pg.187 of Etkin for the the flight conditions of a
% Boeing 747, given on pg. 165 of Etkin. Also finds the changes in the
% y-component bofy force, roll moment, and yaw moment with changes in
% y-component of velocity, x-comp of angular velocity, and z-comp of
% angular velocity
% Date Modified: 4/15/2018
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Airplane Parameters
% Nondimensional Lateral Derivatives
% Table 6.6 -
Cy = [-.8771, 0, 0];
Cl = [-.2797, -.3295, .304];
Cn = [.1946, -.04073, -.2737];

% Table E.1 B747 Case 2
Alt = 20000*(0.3048); % Altitude [ft] -> [m]
[~, a, P, rho] = atmosisa(Alt); % Standard Atmosphere Properties at Alt.
W = 6.366*10^5*4.44822; % Weight [lb]->[N]
Ix_PA = 1.82e7*1.35581795; % Moment of Interia x-PA [slug ft^2]-> [kg m^2]
Iy_PA = 3.31e7*1.35581795; % Moment of Interia y-PA [slug ft^2]-> [kg m^2]
Iz_PA = 4.97e7*1.35581795; % Moment of Interia z-PA [slug ft^2]-> [kg m^2]
Izx_PA = 9.70e5*1.35581795; % Moment of Interia zx-PA [slug ft^2]-> [kg m^2]
zeta = -6.8; % 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 = -I_SA(1,3);
CD = .040; % Coefficient of Drag
cbar = 27.31*(0.3048); % Mean Chord Length [ft]->[m]
b = 195.68*(0.3048); % Span [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]

% Primed Inertias
Ix_lat = (Ix*Iz-Izx^2)/Iz; % [kg m^2]
Iz_lat = (Ix*Iz-Izx^2)/Ix; % [kg m^2]
Izx_lat = Izx/(Ix*Iz-Izx^2); % [kg m^2]
```

```matlab
%% Trim States
Vel = 518*(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]

%% State Variable Derivatives
% Y (N)
Yv = (1/2)*rho*u0*S*Cy(1);
Yp = (1/4)*rho*u0*b*S*Cy(2);
Yr = (1/4)*rho*u0*b*S*Cy(3);

Y = [Yv, Yp, Yr]';

% L (N*m)
Lv = (1/2)*rho*u0*b*S*Cl(1);
Lp = (1/4)*rho*u0*b^2*S*Cl(2);
Lr = (1/4)*rho*u0*b^2*S*Cl(3);

L = [Lv, Lp, Lr]';

% N (N*m)
Nv = (1/2)*rho*u0*b*S*Cn(1);
Np = (1/4)*rho*u0*b^2*S*Cn(2);
Nr = (1/4)*rho*u0*b^2*S*Cn(3);

N = [Nv, Np, Nr]';

T = table(Y,L,N);
T.Properties.VariableNames = {'Y' 'L' 'N'}

%% Lateral Dynamics A matrix

row1 = [Y(1)/m, Y(2)/m, (Y(3)/m-u0), g*cosd(theta0)];
row2 = [(L(1)/Ix_lat+Izx_lat*N(1)), (L(2)/Ix_lat+Izx_lat*N(2)),(L(3)/Ix_lat+Izx_lat*N(3)),0];
row3 = [(Izx_lat*L(1)+ N(1)/Iz_lat), (Izx_lat*L(2)+ N(2)/Iz_lat), (Izx_lat*L(3)+ N(3)/Iz_lat), 0];
row4 = [0, 1, tand(theta0),0];

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

T = table(A)


%% Computing the Eigenvalues of the A matrix
[eVA,eValA] = eig(A);

modesA = diag(eValA);

max_real = max(abs(real(modesA)));

%% Classifying Each Mode
 n = 1;
 j = 1;
 k = 1;
for i = 1:length(modesA)
    if logical(imag(modesA(i))) == 1
        DR_Mode(n) = modesA(i); % Dutch Roll Mode Eigenvalues
        DR_vec(:,n) = eVA(:,i); % Dutch Roll Mode Eigenvector
        n = n+1;
    elseif abs(real(modesA(i))) == max_real
        Roll_Mode(j) = modesA(i); % Roll Mode Eigenvalues
        Roll_vec(:,j) = eVA(:,i); % Roll Mode Eigenvector
        j = j+1;
    else
        Spiral_Mode(k) = modesA(i); % Spiral Mode Eigenvalues
        Spiral_vec(k,:) = eVA(:,i); % Spiral Mode Eigenvectors
        k = k+1;

    end
end
```

```
%% Calculating the Natural Frequency, Dampeing Ratio, and Time Constant

% Dutch Roll Mode

Wn_DR = ( real(DR_Mode(1))^2+imag(DR_Mode(1))^(2) )^(1/2); % Natural Frequency
zeta_DR = -real(DR_Mode(1))/Wn_DR; % Dampening Coefficient
TimeConst_DR = -1/real(DR_Mode(1)); % Time Constant (s)

% Roll Mode

TimeConst_Roll = -1/real(Roll_Mode); % Time Constant (s)

% Spiral Mode

TimeConst_Spiral = -1/real(Spiral_Mode); % Time Constant (s)

%% Dutch Roll Approximation

A_DR = [A(1,1),-u0;
        A(3,1),A(3,3)];

% Eigenvalues

eVA = eig(A_DR);


check = 1;
```

2. *Function to Output A Matrix*

```
%% Author: Jack Lambert
% ASEN 3128
% Homework 10
% Purpose: Find the dimensional derivatives given the none dimensional
% derivatives from pg.187 of Etkin for the the flight conditions of a
% Boeing 747, given on by Case 2 in table E.1. Also finds the changes in the
% y-component bofy force, roll moment, and yaw moment with changes in
% y-component of velocity, x-comp of angular velocity, and z-comp of
% angular velocity
% Date Modified: 4/20/2018
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [A] = Amat()
%% Airplane Parameters
% Nondimensional Lateral Derivatives
% Table 6.6 -
Cy = [-.8771, 0, 0];
Cl = [-.2797, -.3295, .304];
Cn = [.1946, -.04073, -.2737];

% Table E.1 B747 Case 2
Alt = 20000*(0.3048); % Altitude [ft] -> [m]
[~, ~, ~, rho] = atmosisa(Alt); % Standard Atmosphere Properties at Alt.
W = 6.366*10^5*4.44822; % Weight [lb]->[N]
Ix_PA = 1.82e7*1.35581795; % Moment of Interia x-PA [slug ft^2]-> [kg m^2]
Iy_PA = 3.31e7*1.35581795; % Moment of Interia y-PA [slug ft^2]-> [kg m^2]
Iz_PA = 4.97e7*1.35581795; % Moment of Interia z-PA [slug ft^2]-> [kg m^2]
Izx_PA = 9.70e5*1.35581795; % Moment of Interia zx-PA [slug ft^2]-> [kg m^2]
zeta = -6.8; % 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 = -I_SA(1,3);
CD = .040; % Coefficient of Drag
cbar = 27.31*(0.3048); % Mean Chord Length [ft]->[m]
b = 195.68*(0.3048); % Span [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]

% Primed Inertias
Ix_lat = (Ix*Iz-Izx^2)/Iz; % [kg m^2]
Iz_lat = (Ix*Iz-Izx^2)/Ix; % [kg m^2]
Izx_lat = Izx/(Ix*Iz-Izx^2); % [kg m^2]

%% Trim States
Vel = 518*(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]

%% State Variable Derivatives
% Y (N)
Yv = (1/2)*rho*u0*S*Cy(1);
Yp = (1/4)*rho*u0*b*S*Cy(2);
Yr = (1/4)*rho*u0*b*S*Cy(3);

Y = [Yv, Yp, Yr]';

% L (N*m)
Lv = (1/2)*rho*u0*b*S*Cl(1);
Lp = (1/4)*rho*u0*b^2*S*Cl(2);
Lr = (1/4)*rho*u0*b^2*S*Cl(3);

L = [Lv, Lp, Lr]';

% N (N*m)
Nv = (1/2)*rho*u0*b*S*Cn(1);
Np = (1/4)*rho*u0*b^2*S*Cn(2);
Nr = (1/4)*rho*u0*b^2*S*Cn(3);

N = [Nv, Np, Nr]';

%% Lateral Dynamics A matrix

row1 = [Y(1)/m, Y(2)/m, (Y(3)/m-u0), g*cosd(theta0)];
row2 = [(L(1)/Ix_lat+Izx_lat*N(1)), (L(2)/Ix_lat+Izx_lat*N(2)),(L(3)/Ix_lat+Izx_lat*N(3)),0];
row3 = [(Izx_lat*L(1)+ N(1)/Iz_lat), (Izx_lat*L(2)+ N(2)/Iz_lat), (Izx_lat*L(3)+ N(3)/Iz_lat), 0];
row4 = [0, 1, tand(theta0),0];

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

3. Function for ODE45 to Call

```
%% Author: Jack Lambert
% ASEN 3128
% Purpose: Function for ODE45 to call to calculate the State variables
% u_dot, w_dot, q_dot, and theta_dot for the PWD Approximation. This function
% uses the simplified assumptions for the Linearized Longitudinal Dynamics Set
% Last Edited: 4/9/2018
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [dydt] = ODEcall(t,y)

v_dot = y(1); % y-component of Velocity, Body Frame
p_dot = y(2); % roll-rate
r_dot = y(3); % yaw rate
```

```
theta_dot = y(4); % Pitch Angle

%% State Variable Matrix for Linearized Longitudinal Set
[A] = Amat(); % A matrix function based on plane and parameters
State = [v_dot, p_dot, r_dot, theta_dot]'; % Couple State Variables in Long. Set
var = A*State; % Couple State Variables in Long. Set
%% Solving for State Variables in the Linearized Longitudinal Set
dydt(1) = var(1); % v
dydt(2) = var(2); % p
dydt(3) = var(3); % v
dydt(4) = var(4); % p

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

2. *Main Function for Lateral Dynamics Model*

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Author: Jack Lambert
% Dale Lawrence
% Aircraft Dynmaics Homework 10
% Purpose: Sets Initial Conditions for each Pertubation Case and Calls ODE45
% to plot the State Variables vs time
% Date Modefied: 4/20/18
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ODE45 Variable Allocation
%                     v_dot = z(1); % y-component of Velocity, Body Frame
%                     p_dot = z(2); % Angular Velocity about the z-axis  [rad/s]
%                     r_dot = z(3); % Angular Velocity about the z-axis [rad/s]
%                     phi_dot = z(4); % Bank Angle [rad]
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Initial Conditions
c1 = [10, 0,-3.0747, 1.1039]; % y-component of Velocity, Body Frame
c2 = [0, 0.1,-0.2556,-0.0051]; % Angular Velocity about the z-axis  [rad/s]
c3 = [0, 0, 0.0038,0.3015]; % Angular Velocity about the z-axis [rad/s]
c4 = [0, 0, 0.0287,0.4697]; % Bank Angle [rad]
for i = 1:4
    condition{i}= [c1(i) c2(i) c3(i) c4(i)];
end
%% State Variables vs. Time
time = [0 200]; % Set the time to be integrated [s]

string = ["Case I ","Case II","Case III","Case IV"]; % Title for Varying IC's
% Phugoid Response (Longer Time)
for i = 1:4
    % Calling ODE45
    [t,z] = ode45('ODEcall_DR',time,condition{i});


    % V_E vs time
    figure
    subplot(4,1,1)
    plot(t ,z(:,1),'Linewidth',1)
    tit = sprintf('%s %s %s','State Variable of a B 747,',string(i));
    title(tit)
    ylabel('\Deltav_E [m/s]')


    % p vs time
    subplot(4,1,2)
    plot(t ,z(:,2),'Linewidth',1)
    ylabel('\Deltap [rad/s]')

    % r vs time
    subplot(4,1,3)
    plot(t ,z(:,3),'Linewidth',1)
```

```
    ylabel('\Deltar [rad/s]')

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

end
```