University of Colorado Boulder

ASEN 3128 - Aircraft Dynamics

Assignment 7 - Afternoon Section

Assignment 7

Author:
Benjiman Smith
105979656

Professor:
D Lawrence

27 February, 2020

College of Engineering & Applied Science
UNIVERSITY OF COLORADO BOULDER

The objective of this assignment was to utilize equations and knowledge derived in class to better understand the longitudional stability modes of fixed wing aircraft, along with how to simulate the longitudional stability dynamics of a fixed wing aircraft through utilization of Matlab

I. Nomenclature

M	=	Mach Number [$unitless$]
m	=	mass [kg]
V	=	velocity [m/s]
g	=	gravity [m/s^2]
MW	=	Weight [kN]
I_x	=	Moment of Inertia about the x axis [kgm^2]
I_y	=	Moment of Inertia about the y axis [kgm^2]
I_z	=	Moment of Inertia about the z axis [kgm^2]
I_zx	=	Moment of Inertia about the zx axis [kgm^2]
ξ	=	angle about the $\hat{y}$ axis between the body frame and the stability frame [$radians$]
C_D	=	Coefficient of Drag [$unitless$]
X	=	forces in the body $\hat{x}$ direction [N]
Z	=	forces in the body $\hat{z}$ direction [N]
M	=	Moments about the body $\hat{y}$ direction [Nm]
X'	=	forces in the stability frame $\hat{x_s}$ direction [N]
Z'	=	forces in the stability frame $\hat{z_s}$ direction [N]
M'	=	Moments about the stability $\hat{y_s}$ direction [Nm]
u	=	body $\hat{x}$ component of velocity [m/s]
w	=	body $\hat{z}$ component of velocity [m/s]
$\dot{w}$	=	body derivative of $\hat{z}$ component of velocity [m/s]
q	=	pitch rate [rad/s]
δ_e	=	elevator deflection [rad]
$(X_u)'$	=	partial derivative of u with respect to X [Ns/m]
$(X_w)'$	=	partial derivative of w with respect to X [Ns/m]
$(X_q)'$	=	partial derivative of q with respect to X [Ns/rad]
$(X_{\dot{w}})'$	=	partial derivative of $\dot{w}$ with respect to X [Ns/rad]
$(X_{\delta_e})'$	=	partial derivative of δ_e with respect to X [N/rad]
$(Z_u)'$	=	partial derivative of u with respect to Z [Ns/m]
$(Z_w)'$	=	partial derivative of w with respect to Z [Ns/m]
$(Z_q)'$	=	partial derivative of q with respect to Z [Ns/rad]
$(Z_{\dot{w}})'$	=	partial derivative of $\dot{w}$ with respect to Z [Ns/rad]
$(Z_{\delta_e})'$	=	partial derivative of δ_e with respect to Z [N/rad]
$(M_u)'$	=	partial derivative of u with respect to M [Ns/m]
$(M_w)'$	=	partial derivative of w with respect to M [Ns/m]
$(M_q)'$	=	partial derivative of q with respect to M [Ns/rad]
$(M_w)'$	=	partial derivative of $\dot{w}$ with respect to M [Ns/rad]
$(M_{\delta_e})'$	=	partial derivative of δ_e with respect to M [N/rad]
θ_0	=	initial θ value [rad]
Δq	=	pitch rate deviated from trim [$radian/s^2$]
$\Delta\dot{q}$	=	pitch rate derivative deviated from trim [$radian/s^2$]
Δu	=	u deviated from trim [$radian/s^2$]
$\Delta\dot{q}$	=	u derivative deviated from trim [$radian/s^2$]
Δw	=	w deviated from trim [$radian/s^2$]
$\Delta\dot{w}$	=	w derivative deviated from trim [$radian/s^2$]
$\Delta\theta$	=	θ deviated from trim [$radian/s^2$]
$\Delta\dot{\theta}$	=	θ derivative deviated from trim [$radian/s^2$]

II. Question 1

Using the dimensional stability derivatives for case II of the Boeing 747-100 aircraft found in the *Dynamics of Flight: Stability and Control* textbook utilized in class, the following SI-equivalent of the English Unit dimensional stability derivatives found in table E.3 of the textbook were calculated in MATLAB. The following conversion equations were utilized to perform this transformation, with equation 1 converting pound force to Newtons, equation 8 converting feet to meters, and 3 to convert the moments of inertia from English units to SI units.

$$1lbf = 4.45N \tag{1}$$

$$1ft = 0.3048m \tag{2}$$

$$1slug * ft^2 = 1.3558kg * m^2 \tag{3}$$

Prior to calculating the SI-Equivalent of the dimensional stability derivatives, the 2nd case of the Boeing 747-100 initial conditions found in Table E.1 of the textbook were converted to SI units.

Table 1 747-100 Case II in SI Units

DD	***Case II***
M	0.5
V (m/s)	157.8864
W (kN)	2832.87
I_X (kgm^2)	2.4676e7
I_Y (kgm^2)	4.4878e7
I_Z (kgm^2)	6.7384e7
I_{ZX} (kgm^2)	1.3151e6
ξ **(rad)**	-0.1187
C_D	0.04

After the initial conditions had been converted, the longitudinal stability derivatives were converted from English Units to SI Units Using Equations 1 and 8.

Table 2 SI Version of Dimensional Stability Derivatives

DD	$X(N)$	$Z(N)$	$M(Nm)$
u (m/s)	-712.9052	-1.9593e4	3.6383e4
w (m/s)	2.2571e4	-1.2499e5	-2.5040e5
q (rad/s)	0	-562035	-1.8908e7
$\dot{w}$ **(rad/s)**	0	4.5318e3	-1.8414e4
δ_e **(rad)**	177733	-1486745	-4.8937e7

III. Question 2

In order to analyze the stability of the 747-100, the frame in which the above measurements were taken needed to be converted into the stability frame, where θ_0 = 0, and ξ = -6.8 degrees, as this was the rotation about the $\hat{y}$ axis to get to the stability frame from the body frame. The following equations used to perform this transformation were from Table B.12,6 from the aforementioned textbook.

Longitudinal

$$
\begin{aligned}
(X_u)' &= X_u \cos^2 \xi - (X_w + Z_u) \sin \xi \cos \xi + Z_w \sin^2 \xi \\
(X_w)' &= X_w \cos^2 \xi + (X_u - Z_w) \sin \xi \cos \xi - Z_u \sin^2 \xi \\
(X_q)' &= X_q \cos \xi - Z_q \sin \xi \\
(X_{\dot{u}})' &= Z_{\dot{w}} \sin^2 \xi \quad (1) \\
(X_{\dot{w}})' &= -Z_{\dot{w}} \sin \xi \cos \xi \quad (1) \\
(Z_u)' &= Z_u \cos^2 \xi - (Z_w - X_u) \sin \xi \cos \xi - X_w \sin^2 \xi \\
(Z_w)' &= Z_w \cos^2 \xi + (Z_u + X_w) \sin \xi \cos \xi + X_u \sin^2 \xi \\
(Z_q)' &= Z_q \cos \xi + X_q \sin \xi \\
(Z_{\dot{u}})' &= -Z_{\dot{w}} \sin \xi \cos \xi \quad (1) \\
(Z_{\dot{w}})' &= Z_{\dot{w}} \cos^2 \xi \\
(M_u)' &= M_u \cos \xi - M_w \sin \xi \\
(M_w)' &= M_w \cos \xi + M_u \sin \xi \\
(M_q)' &= M_q \\
(M_{\dot{u}})' &= -M_{\dot{w}} \sin \xi \quad (1) \\
(M_{\dot{w}})' &= M_{\dot{w}} \cos \xi \quad (1)
\end{aligned}
\qquad \text{(B.12,6)}
$$

Fig. 1

As seen from these equations, there are not any equations relating δ_e from the body frame to the stability frame. This is due to the fact that in the stability frame, δ_e has the same values as that of the body frame. Using the equations from B 12,6, along with the SI-converted dimensional stability derivative values in the body frame found in table 2, the dimensional stability derivatives in the stability frame were calculated.

Table 3 Dimensional Stability Derivatives in the Stability Frame, SI Units

DD	$X'(N)$	$Z'(N)$	$M'(Nm)$
$u(m/s)$	-2.1050e3	-3.4413e4	6.4787e3
$w(m/s)$	7.9182e3	-1.2360e5	-2.5295e5
$q(rad/s)$	-6.6547e4	-5.5808e5	-1.8908e7
$\dot{w}(rad/s)$	532.8036	4.4682e3	-1.8285e4
δ_e **(rad)**	177733	-1486745	-4.8937e7

IV. Question 3

Utilizing table 3 along with Eq. 4.9,18, Longitudinal Equations, Eq. (4.9,18):

$$\begin{bmatrix} \Delta\dot{u} \\ \dot{w} \\ \dot{q} \\ \Delta\dot{\theta} \end{bmatrix} = \begin{bmatrix} \dfrac{X_u}{m} & \dfrac{X_w}{m} & 0 & -g\cos\theta_o \\ \dfrac{Z_u}{m - Z_{\dot{w}}} & \dfrac{Z_w}{m - Z_{\dot{w}}} & \dfrac{Z_q + mu_o}{m - Z_{\dot{w}}} & \dfrac{-mg\sin\theta_o}{m - Z_{\dot{w}}} \\ \dfrac{1}{I_y}\left[M_u + \dfrac{M_{\dot{w}} Z_u}{(m - Z_{\dot{w}})}\right] & \dfrac{1}{I_y}\left[M_w + \dfrac{M_{\dot{w}} Z_w}{(m - Z_{\dot{w}})}\right] & \dfrac{1}{I_y}\left[M_q + \dfrac{M_{\dot{w}}(Z_q + mu_o)}{(m - Z_{\dot{w}})}\right] & -\dfrac{M_{\dot{w}}\, mg\sin\theta_o}{I_y(m - Z_{\dot{w}})} \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \Delta u \\ w \\ q \\ \Delta\theta \end{bmatrix} + \begin{bmatrix} \dfrac{\Delta X_c}{m} \\ \dfrac{\Delta Z_c}{m - Z_{\dot{w}}} \\ \dfrac{\Delta M_c}{I_y} + \dfrac{M_{\dot{w}}}{I_y}\dfrac{\Delta Z_c}{(m - Z_{\dot{w}})} \\ 0 \end{bmatrix}$$

$$\Delta\dot{x}_E = \Delta u\cos\theta_o + w\sin\theta_o - u_o\,\Delta\theta\sin\theta_o$$

$$\Delta\dot{z}_E = -\Delta u\sin\theta_o + w\cos\theta_o - u_o\,\Delta\theta\cos\theta_o$$

Fig. 2

The A matrix was derived for the linearized longitudinal dynamics in SI-units.

$$\begin{bmatrix} -0.0073 & 0.0274 & 0 & -9.81 \\ -0.1210 & -0.4347 & 158.4048 & 0 \\ 0.0002 & -0.0055 & -0.4859 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Question 4

After the derivation of the A matrix for the linearized longitudional dynamics in SI units, the eigenvectors and eigenvalues were found through the use of the *eig* function, along with hand derivations for further understanding.

Problem 3, Assignment 7

Wednesday, March 11, 2020 7:23 AM

Longitudinal Equations, Eq. (4.9,18):

$$\begin{bmatrix} \Delta\dot{u} \\ \dot{w} \\ \dot{q} \\ \Delta\dot{\theta} \end{bmatrix} = \overbrace{\begin{bmatrix} \frac{X_u}{m} & \frac{X_w}{m} & 0 & -g\cos\theta_o \\ \frac{Z_u}{m - Z_{\dot{w}}} & \frac{Z_w}{m - Z_{\dot{w}}} & \frac{Z_q + mu_o}{m - Z_{\dot{w}}} & \frac{-mg\sin\theta_o}{m - Z_{\dot{w}}} \\ \frac{1}{I_y}\left[M_u + \frac{M_{\dot{w}}Z_u}{(m - Z_{\dot{w}})}\right] & \frac{1}{I_y}\left[M_w + \frac{M_{\dot{w}}Z_w}{(m - Z_{\dot{w}})}\right] & \frac{1}{I_y}\left[M_q + \frac{M_{\dot{w}}(Z_q + mu_o)}{(m - Z_{\dot{w}})}\right] & -\frac{M_{\dot{w}}\, mg\sin\theta_o}{I_y(m - Z_{\dot{w}})} \\ 0 & 0 & 1 & 0 \end{bmatrix}}^{A} \begin{bmatrix} \Delta u \\ w \\ q \\ \Delta\theta \end{bmatrix} + \begin{bmatrix} \frac{\Delta X_c}{m} \\ \frac{\Delta Z_c}{m - Z_{\dot{w}}} \\ \frac{\Delta M_c}{I_y} + \frac{M_{\dot{w}}}{I_y}\frac{\Delta Z_c}{(m - Z_{\dot{w}})} \\ 0 \end{bmatrix}$$

$$\Delta\dot{x}_E = \Delta u\cos\theta_o + w\sin\theta_o - u_o\,\Delta\theta\sin\theta_o$$

$$\Delta\dot{z}_E = -\Delta u\sin\theta_o + w\cos\theta_o - u_o\,\Delta\theta\cos\theta_o$$

LET $AX = \lambda \cdot X$, WHERE λ IS A SCALAR

THEREFORE $(A - \lambda \underset{\substack{\uparrow \\ \text{IDENTITY} \\ \text{MATRIX}}}{I})X = 0$

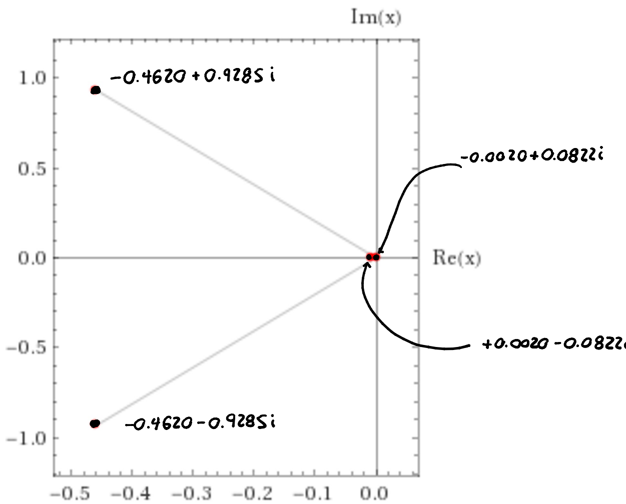
SINCE WE KNOW $X \neq 0$, AS WE HAVE REAL Δq & $\Delta\theta$ VALUES, $\det(A - \lambda I) = 0$ IS TRUE, THEREFORE:

$$\begin{bmatrix} \frac{X_u}{m} - \lambda & \frac{X_w}{m} & 0 & -g\cos\theta_o \\ \frac{Z_u}{m - Z_{\dot{w}}} & \frac{Z_u}{m - Z_{\dot{w}}} - \lambda & \frac{Z_q + mu_o}{m - Z_{\dot{w}}} & -\frac{mg\sin\theta_o}{m - Z_{\dot{w}}} \\ \frac{1}{I_y}\left[M_u + \frac{M_{\dot{w}}Z_u}{(m - Z_{\dot{w}})}\right] & \frac{1}{I_y}\left[M_w + \frac{M_{\dot{w}}Z_w}{(m - Z_{\dot{w}})}\right] & \frac{1}{I_y}\left[M_q + \frac{M_{\dot{w}}(Z_q + mu_o)}{(m - Z_{\dot{w}})}\right] - \lambda & \frac{M_{\dot{w}}\, mg\sin\theta_o}{I_y(m - Z_{\dot{w}})} \\ 0 & 0 & 1 & -\lambda \end{bmatrix} = A - \lambda I$$

$$= \begin{bmatrix} -0.0073 - \lambda & 0.0274 & 0 & -9.81 \\ -0.1210 & -0.4347 - \lambda & 158.4048 & 0 \\ 0.0002 & -0.0055 & -0.4859 - \lambda & 0 \\ 0 & 0 & 1 & -\lambda \end{bmatrix} = A - \lambda I$$

$$\mathrm{DET}(A - \lambda I) = \mathrm{DET}\begin{bmatrix} -0.0073 - \lambda & 0.0274 & 0 & -9.81 \\ -0.1210 & -0.4347 - \lambda & 158.4048 & 0 \\ 0.0002 & -0.0055 & -0.4859 - \lambda & 0 \\ 0 & 0 & 1 & -\lambda \end{bmatrix} = \lambda^4 + 0.9279\lambda^3 + 1.09248\lambda^2 + 0.00864476\lambda$$

EIGEN VALUES ARE DIFFICULT TO FIND FOR THIS POLYNOMIAL BY HAND, SO SOFTWARE WAS UTILIZED



$$\Rightarrow \boxed{\text{MODES: } \begin{cases} -0.0020 \pm 0.0822i \\ -0.4620 \pm 0.9285i \end{cases}}$$

the eigenvectors output by the *eig* function were as follows.

$$\begin{bmatrix} 0.0301 \pm 0.0099\mathrm{i} \\ 0.9995 \\ -0.0001 \pm 0.0059\mathrm{i} \\ -0.0051 \pm 0.0024\mathrm{i} \end{bmatrix}$$

$$\begin{bmatrix} 0.9966 \\ -0.0264 \pm 0.0074\mathrm{i} \\ 6.918e-4 \pm 3.3953\text{e-5i} \\ -0.0051 \pm 0.0024\mathrm{i} \\ -0.0006 \pm 0.0084\mathrm{i} \end{bmatrix}$$

and the eigenvalues output by the *eig* function were two coupled sets of values, one couple consisting of the Short Period mode, and the other consisting of the Phugoid mode.

Table 4 modes of the linearized longitudional dynamics

Phugoid mode	$-0.0020 \pm 0.0822i$
Short Period mode	$-0.4620 \pm 0.9285i$

The differentiation between the Short period mode and the Phugoid mode can be seen by the following work.

Problem 4, Assignment 7

Wednesday, March 11, 2020 7:52 AM

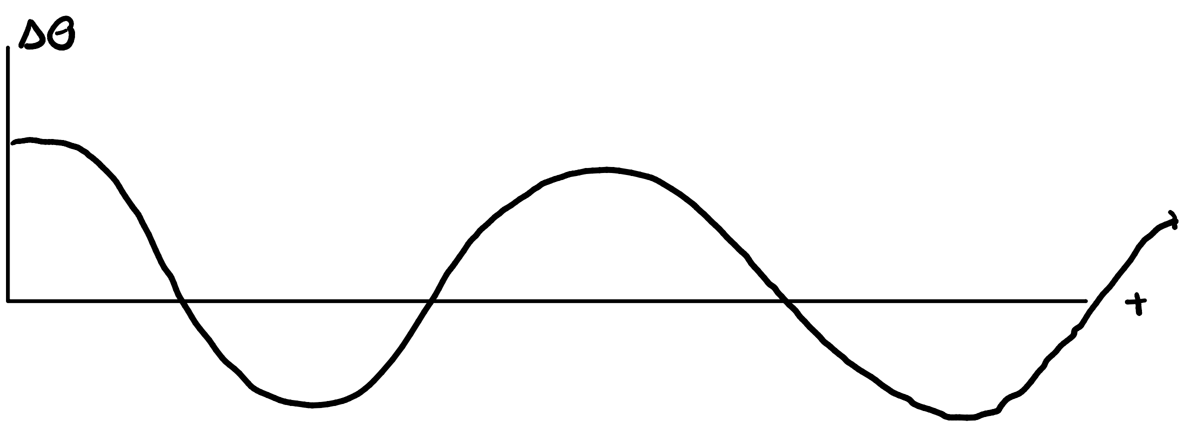
Phugoid mode	$-0.0020 \pm 0.0822i$
Short Period mode	$-0.4620 \pm 0.9285i$

SHORT PERIOD MODE IS CHARACTERIZED BY HEAVY DAMPING, AND A HIGH FREQUENCY, AS SEEN ON THE PLOT BELOW.

§ SHORT PERIOD RESPONSE

THE PHUGOID MODE IS CHARACTERIZED BY LOW DAMPING VALUES & A SMALL FREQUENCY, AS SEEN BY THE PLOT BELOW.

§ PHUGOID RESPONSE



TAKING THIS INTO ACCOUNT, AND LOOKING INTO THE EIGENVALUES CALCULATED IN MATLAB, IT IS SEEN THAT THE $-0.4620 \pm 0.9285i$ HAS A LARGE DAMPING TERM (-0.4620), & A QUICK FREQUENCY (0.9285), SHOWING THIS EIGENVALUE COUPLE REPRESENTS THE SHORT PERIOD MODE.

IT IS SEEN THAT THE $-0.0020 \pm 0.0822i$ HAS A SMALL DAMPING TERM (0.0020), & A SLOW FREQUENCY (0.0822), SHOWING THIS EIGENVALUE COUPLE REPRESENTS THE PHUGOID MODE

$$\Rightarrow \begin{cases} -0.0020 \pm 0.0822i & \Rightarrow \text{PHUGOID MODE} \\ -0.4620 \pm 0.9285i & \Rightarrow \text{SHORT PERIOD MODE} \end{cases}$$

Using the found modes, the corresponding modal damping ratios and modal natural frequencies were calculated using the following equations.

$$\omega_n = \sqrt{\omega^2 + n^2} \tag{4}$$

$$\xi = \frac{-n}{\omega_n} \tag{5}$$

where the eigenvalues are given as

$$n \pm \omega i \tag{6}$$

Table 5 modal damping ratios and modal natural frequencies for phugoid and short period modes

DD	**Phugoid mode**	**Short Period Mode**
ω_n	0.0822	1.0371
ξ	0.0241	0.4454

Question 5

In class, a simplified longitudinal dynamics model was developed, as shown in the figure below.

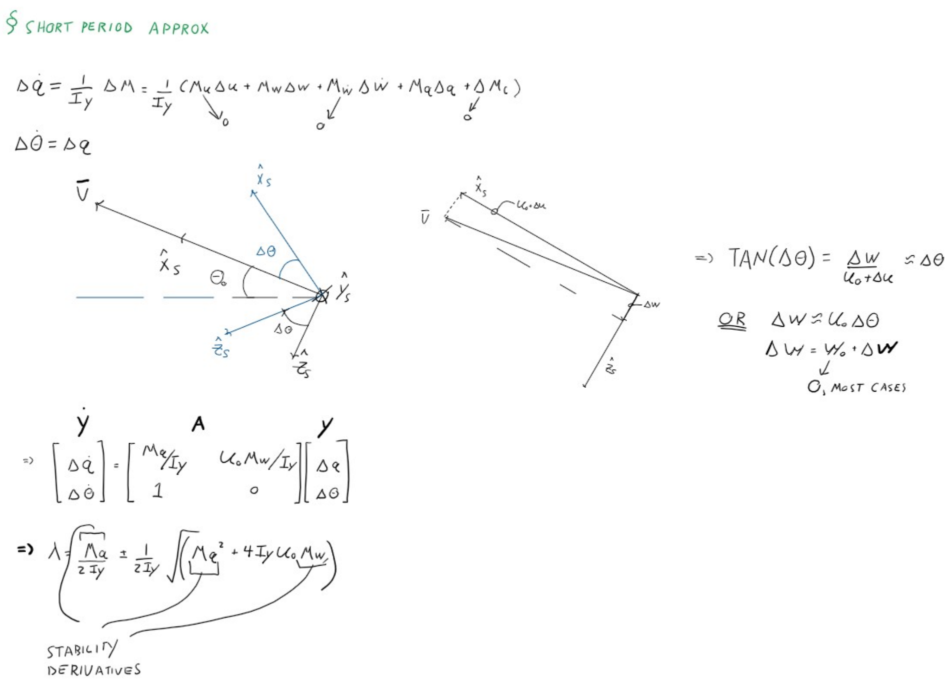


Fig. 3

Finding the eigenvalues of this A matrix using the *eig* function in Matlab, coupled with hand calculations for further understanding yielded the following eigenvalue couple.

$$-0.2107 \pm 0.9195i \tag{7}$$

Comparing this eigenvalue couple to the eigenvalue couple developed for the Short Period Mode, several key elements can be seen. Due to the assumptions made for the in-class aproximation developement seen in figure 3, the ΔM term has several terms go to zero, including Δu, $\Delta\dot{w}$, as these values were assumed to be negligible, and ΔM_c, as there was no moment due to control effects. With Δu, $\Delta \dot{w}$ terms set to zero, there was only one eigenvalue couple represented, as the A matrix was reduced to a 2x2. Along with this assumption, $tan(\Delta\theta) \approx \Delta\theta$ was assumed, which is true for values of $\Delta\theta \leq 10$. These assumptions rendered the discrepancies between the short period mode and the in-class approximation, with a 54 % difference between the n-values, and a 0.969 % difference between the ω-values.

The oscillation period of the Phugoid mode above was directly compared to the Lanchester approximation found in the textbook as

$$T = 0.138u_0 \tag{8}$$

Rendering the following results:

Table 6 Oscillation Period of Phugoid Mode and Lanchester Approximation

DD	$Lanchester Approx.$	$Phugoidmode$
T (s)	71.4840	76.4157

The 6.454 % difference in values between the oscillation period of the Phugoid Mode and the Lanchester Aprroximation is due to the assumptions made in the approximation.

In deriving Lanchester's Approximation, it was assumed that $\Delta\alpha$ and α_T were equal to zero. If $\Delta\alpha$ is zero, it means the aircraft's stability frame is the same as the body frame, which is untrue for the linearized longitudional dynamics derrivation of the Phugoid Mode. This shows that the assumption of $\delta\alpha = 0$ does not hold in the derivation of the phugoid Mode as ξ = -6.8 degrees. ξ and $\Delta\alpha$ are related due to their common body frame reference.

Quesrion 6

In order to simulate the linearized longitudinal dynamics derrived in Question 3, an *ODE45* function was written that takes in the A-matrix derived in Question 3, along with initial conditions for Δu, Δw, Δq, $\Delta\theta$, and returns the change in these initial conditions over time.

Using this *ODE45* function, simulations were run for different initial conditions to model the response of the 747-100 to differing disturbances. The first simulation ran was to verify that the trim state of the aircraft was in fact an equilibrium.

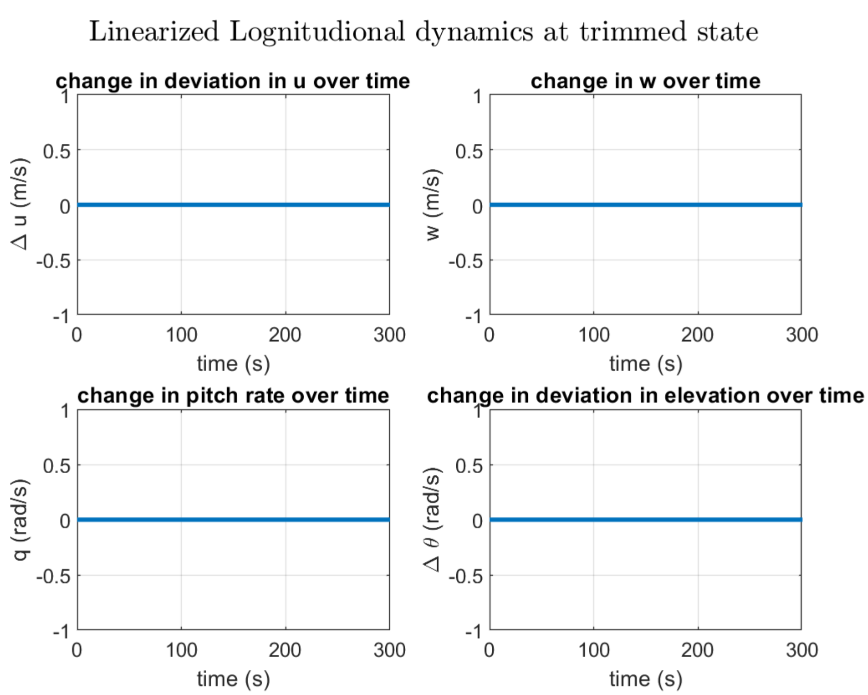


Fig. 4 Trimmed State

The results seen in figure 4 are reasonable, showing that the values for Δu, Δw, Δq, and $\Delta\theta$ are at a constant zero value. Due to the consistency of each initial condition's zero value over time, it is seen that the aircraft is in equilibrium.

For an initial condition case where $\Delta u^E = 10\ \frac{m}{s}$, the simulation output the following results.

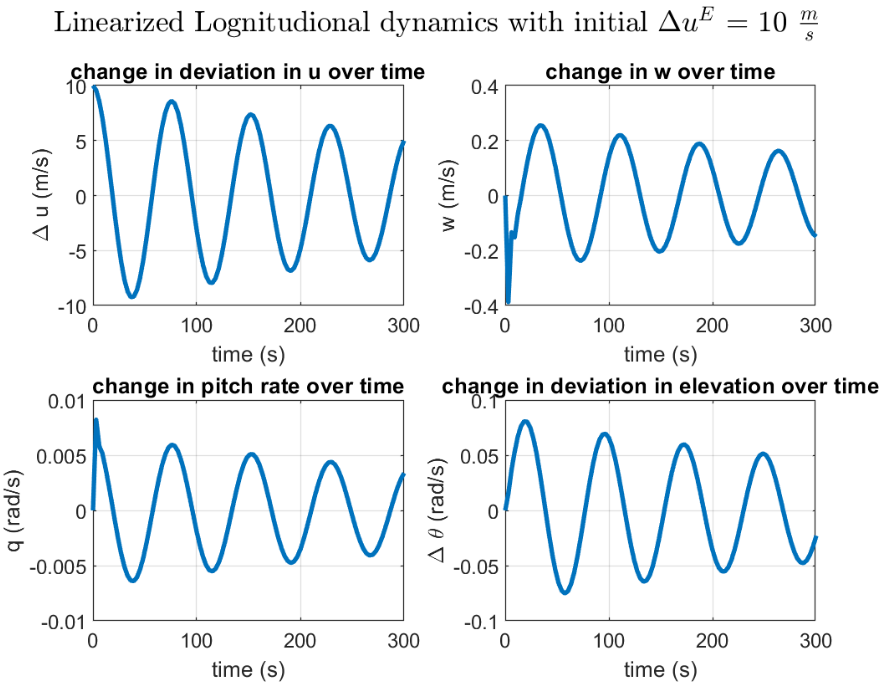


Fig. 5 $\Delta u^E = 10\ \frac{m}{s}$

As seen from figure 5, Δu initially starts at a value of 10 m/s, and as Δu progresses with time, the damping factor slowly damps the amplitude of the oscillation in Δu m/s at a slow frequency, showing the main mode utilized for u over time is the phugoid mode. Similar results are seen for deviation in elevation over time, as the short period mode isn't effecting the elevation over time.

As seen from Δw progression over time, both the short period mode and the phugoid mode are present. This is due to the initial upset of the aircraft due to the initial condition of $\Delta u^E = 10\ \frac{m}{s}$ causing a quick spike in the negative $\hat{z_s}$ velocity direction, as the trimmed angle of the aircraft ξ is a negative value. This spike relates to the short period mode, as there is heavy damping for this period of time, coupled with a very fast frequency. After this initial spike, the aircraft begins to recover, as the oscillations have a long frequency with a small damping factor, relating to the phugoid mode.

As seen from change in pitch rate over time, both the short period mode and the phugoid mode are present. This is due to the initial upset of the aircraft due to the initial condition of $\Delta u^E = 10\ \frac{m}{s}$ causing a quick spike in the positive pitch rate direction, as elevation angle quickly increases initially due to the positive Δu^E. This spike relates to the short period mode, as there is heavy damping for this period of time, coupled with a very fast frequency. After this initial spike, the aircraft begins to recover, as the oscillations have a long frequency with a small damping factor, relating to the phugoid mode.

For an initial condition case where $\Delta w^E = 10\ \frac{m}{s}$, the simulation output the following results.

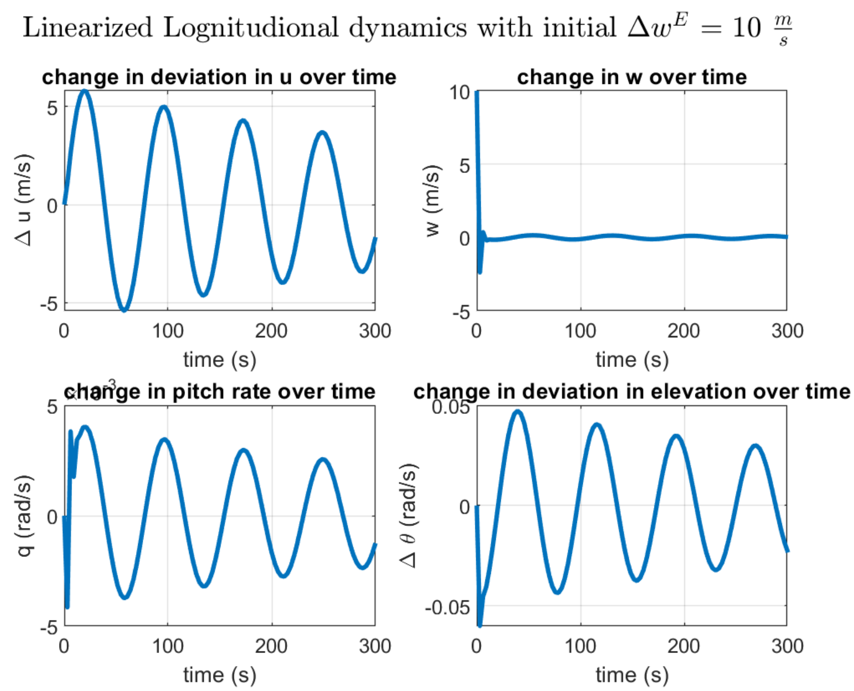


Fig. 6 Δw^E **= 10** $\frac{m}{s}$

As seen from figure 5, as Δu progresses with time, the damping factor slowly damps the amplitude of the oscillation in Δu m/s at a slow frequency, showing the main mode effecting the u velocity is the phugoid mode.

As seen from Δw progression over time, both the short period mode and the phugoid mode are present. This is due to the initial upset of the aircraft due to the initial condition of $\Delta w^E = 10\ \frac{m}{s}$ causing a quick drop in the negative $\hat{z}$ direction to correct the initial Δw^E. This spike relates to the short period mode, as there is heavy damping for this period of time, coupled with a very fast frequency. After this initial spike, the aircraft begins to recover, as the oscillations have a long frequency with a small damping factor, relating to the phugoid mode.

As seen from change in pitch rate over time, both the short period mode and the phugoid mode are present. This is due to the initial upset of the aircraft due to the initial condition of $\Delta w^E = 10\ \frac{m}{s}$ causing a quick spike in the negative pitch rate direction, as elevation angle quickly increases initially due to the initial positive Δw^E. This spike relates to the short period mode, as there is heavy damping for this period of time, coupled with a very fast frequency. After this initial spike, the aircraft begins to recover, as the oscillations have a long frequency with a small damping factor, relating to the phugoid mode.

With the deviation in elevation over time plot, it is seen that the short period mode effects the aircraft's stability initially due to the initial positive Δw^E, as there was a quick spike in the negative elevation direction to counteract the initial deviation. Due to the high damping of this short period mode, the deviation in elevation over time plot has a phugoid mode response after this initial spike.

For an initial condition case where $\Delta q = 0.1\ \frac{rad}{s}$, the simulation output the following results.

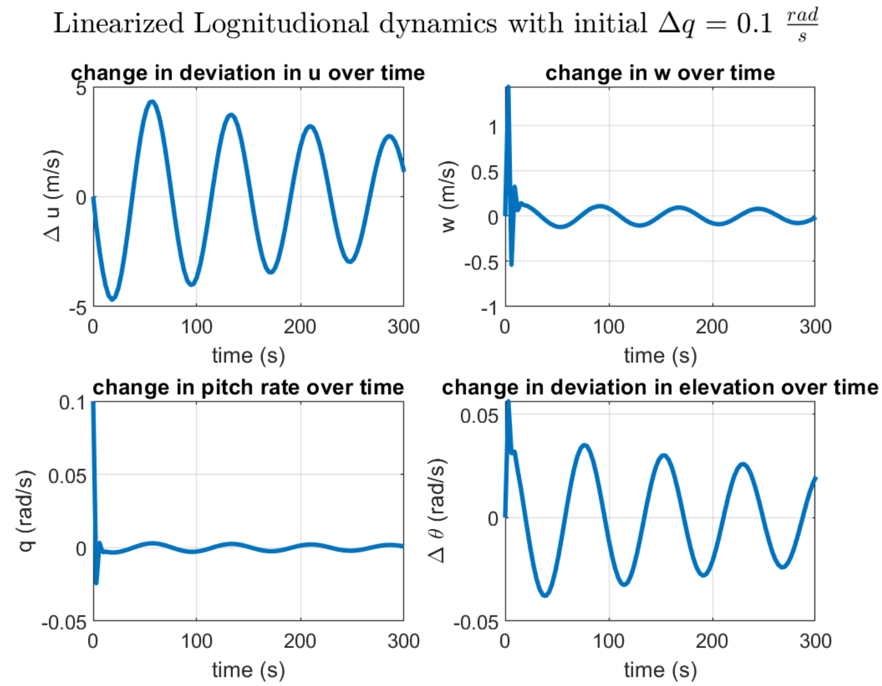


Fig. 7 Δq **= 0.1** $\frac{rad}{s}$

As seen from figure 5, as Δu progresses with time, the damping factor slowly damps the amplitude of the oscillation in u at a slow frequency, showing the main mode effecting the u velocity is the Phugoid mode.

As seen from Δw progression over time, both the short period mode and the phugoid mode are present. This is due to the initial upset of the aircraft due to the initial condition of $\Delta q^E = 0.1\ \frac{rad}{s}$ causing a quick spike in the positive w velocity. This spike relates to the short period mode, as there is heavy damping for this period of time, coupled with a very fast frequency. Due to the pitch rate being deviated, the short period mode has a longer period of influence on Δw over time when compared to the aforementioned deviations Δu^E and Δw^E , as the change in elevation increasing at a positive rate causes a large spike in w, causing a need for several oscillations to correct the behavior. After the prolonged short period mode, the aircraft begins to recover, as the oscillations have a long frequency with a small damping factor, relating to the phugoid mode.

As seen from change in pitch rate over time, both the short period mode and the phugoid mode are present. This is due to the initial upset of the aircraft due to the initial condition of $\Delta q^E = 0.1\ \frac{rad}{s}$ causing a quick spike in the negative pitch rate direction to correct the initial positive pitch rate. This spike correction relates to the short period mode, as there is heavy damping for this period of time, coupled with a very fast frequency. After this initial spike, the aircraft begins to recover, as the oscillations have a long frequency with a small damping factor, relating to the phugoid mode.

For an initial condition case where $\Delta\theta = 0.1$ rad, the simulation output the following results.

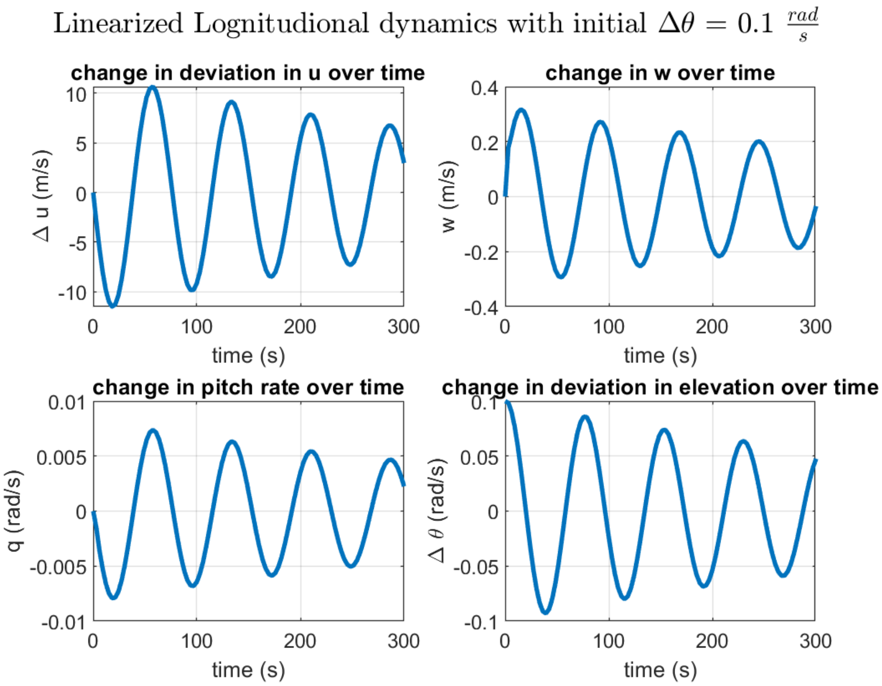


Fig. 8 $\Delta\theta$ **= 0.1** rad

As seen from figure 5, as Δu progresses with time, the damping factor slowly damps the amplitude of the oscillation in Δu m/s at a slow frequency, showing the main mode in the u velocity is the phugoid mode. Similar results are seen for deviation in elevation over time, showing that the short period mode isn't drastically effecting elevation angle.

As seen from Δw progression over time, the phugoid mode is the only mode present. This is due to the aircraft's behavoir due to an initial angle deviation from trim is handled purely by the phugoid mode, as no extreme action in needed from the short period mode to bring the aircraft back to equilibrium.

As seen from change in pitch rate over time, the phugoid mode is the only mode present. This is due to the aircraft's behavoir due to an initial angle deviation from trim is handled purely by the phugoid mode, as no extreme action in needed from the short period mode to bring the aircraft back to equilibrium. The same results can be seen for the elevation deviation over time. Starting at 0.1 rad, the behavior of the deviation has a low damping rate coupled with a slow frequency, showing the main presence of the phugoid mode.

From the analysis of the simulations with the different sets of initial conditions, conclusions can be drawn about which initial deviations excite which modes. As seen from the $\Delta u^E = 10\ \frac{m}{s}$ case in figure 5, The Phugoid mode Short period modes were excited. This result was expected, as an initial $\hat{x}$ direction velocity would cause some slight oscillation from trim, as the stability frame of the aircraft is slightly off axis from the body frame, rotated negatively about the body $\hat{y}$ axis. This is clearly seen in the figure. There are slight short period mode responses in both the $\hat{w}$ and q directions, but this is due to the difference in coordinates from the body frame to the stability frame. This the short period mode response is also due to the fact that with a rate of change/rotation, the aircraft must respond in a more critical manner in order to keep further deviation from trim from occurring.

The case in which $\Delta w^E = 10\ \frac{m}{s}$, both the phugoid and Short period modes were excited. This makes sense, as an initial deviation in the body $\hat{z}$ direction would require a large amplitude initial modal response to recover the aircraft to equilibrium. This occurs in real scenarios, as when aircraft drop/gain altitude due to turbulence, there is a sharp initial response done by the inherent design of the plane to try to return the plane to equilibrium. The phugoid mode response was expected as well, as the short period mode utilized by the aircraft first doesn't remove all of the oscillation present from the initial disturbance. This the short period mode response is also due to the fact that with a rate of change/rotation, the aircraft must respond more quickly in order to keep further deviation from trim from occurring.

When $\Delta q = 0.1\ \frac{rad}{s}$, it was clearly seen that the short period mode was easily excited. This makes sense physically as well, as when a commercial aircraft developes an initial pitch rate, the response is to quickly damp out the changing elevation angle to return the aircraft to equilibrium. This the short period mode response is also due to the fact that with a rate of change/rotation, the aircraft must respond more urgently in order to keep further deviation from trim from occurring.

For the case in which $\Delta\theta = 0.1$ rad, the phugoid mode was clearly the main modal response utilized by the aircraft. This makes sense as well, as this value isn't changing as a rate, therefore the response can be of a low frequency with a low damping factor.

The modern commercial aircraft are designed to return to equilibrium in the most comfortable, safe, and quick manner possible. This is seen from all four initial condition cases.

V. MATLAB Code

Question 1 Script

```
% This script answers question 1 of assignment 7 by converting tables E1
% and E3 from the textbook into SI units
%   Author: Benjiman Smith
%   Collaborators: E. Owen, I. Quezada
%   Date: 3/6/2020
%
close all
clear all
%% Table E1, Case II
Altitude = 20000; % ft
Altitude = Altitude*0.3048; % m
M = 0.5;
V = 518; % ft/s
V = V*0.3048; % m/s
W = 6.366e5; % lbf
W = 4.45*W; %N
Ix = 1.82e7; % slugs ft^2
Ix = 1.3558179619*Ix; % kg m^2
Iy = 3.31e7; % slugs ft^2
Iy = 1.3558179619*Iy; % kg m^2
Iz = 4.97e7; % slugs ft^2
Iz = 1.3558179619*Iz; % kg m^2
Izx = 9.70e5; % slugs ft^2
Izx = 1.3558179619*Izx; % kg m^2
zeta = deg2rad(-6.8); %rad
Cd = .040;
CaseII = [Altitude, M, V, W, Ix, Iy, Iz, Izx, zeta, Cd]; % define Case II SI unit equivalent of
    table E.1
%% Table E3, Longitudional
%1st column
Xu = -4.883e1; % lb*s/ft
Xu = Xu*(4.45/0.3048); % N*s/m
Xw = 1.546e3; % lb*s/ft
Xw = Xw*(4.45/0.3048); % N*s/m
Xq = 0; % lb*s/rad
Xwdot = 0; % lb*s^2/ft
Xdele = 3.994e4; % lb/rad
Xdele = Xdele*4.45; % N/rad
% 2nd column
Zu = -1.342e3; % lb*s/ft
Zu = Zu*(4.45/0.3048); % N*s/m
Zw = -8.561e3; % lb*s/ft
Zw = Zw*(4.45/0.3048); % N*s/m
Zq = -1.263e5; % lb*s/rad
Zq = Zq*4.45; % N*s/rad
Zwdot = 3.104e2; % lb*s^2/ft
Zwdot = Zwdot*(4.45/0.3048); % N*s^2/m
Zdele = -3.341e5; % lb/rad
Zdele = Zdele*4.45; % N/rad
% 3rd column
Mu = 8.176e3; % lb*s
Mu = Mu*4.45; % N/s
Mw = -5.627e4; % lb*s
Mw = Mw*4.45; % N/s
Mq = -1.394e7; % lb*s*ft/rad
Mq = Mq*4.45*0.3048; %N*s*m/rad
Mwdot = -4.138e3; % lb*s^2
Mwdot = Mwdot*4.45;
Mdele = -3.608e7; % lb*ft/rad
Mdele = Mdele*4.45*0.3048; % Nm/rad;
Converted = [Xu, Zu, Mu; Xw, Zw, Mw; Xq, Zq, Mq; Xwdot, Zwdot, Mwdot; Xdele, Zdele, Mdele]; %
    Table E.3 converted to SI units
```

Question 2 Script

```
% This script answers question 2 by converting the body frame stability
% derrivative table derived in Question1.m to the stability frame
%   Author: Benjiman Smith
%   Collaborators: E. Owen, I. Quezada
%   Date: 3/6/2020
%
close all
clear all
%% Table E1, Case II
Altitude = 20000; % ft
Altitude = Altitude*0.3048; % m
M = 0.5;
V = 518; % ft/s
V = V*0.3048; % m/s
W = 6.366e5; % lbf
W = 4.45*W; %N
Ix = 1.82e7; % slugs ft^2
Ix = 1.3558179619*Ix; % kg m^2
Iy = 3.31e7; % slugs ft^2
Iy = 1.3558179619*Iy; % kg m^2
Iz = 4.97e7; % slugs ft^2
Iz = 1.3558179619*Iz; % kg m^2
Izx = 9.70e5; % slugs ft^2
Izx = 1.3558179619*Izx; % kg m^2
zeta = deg2rad(-6.8); %rad
Cd = .040;
CaseII = [Altitude, M, V, W, Ix, Iy, Iz, Izx, zeta, Cd]; % define Case II SI unit equivalent of table E.1
%% Table E2, Longitudional
%1st column
Xu = -4.883e1; % lb*s/ft
Xu = Xu*(4.45/0.3048); % N*s/m
Xw = 1.546e3; % lb*s/ft
Xw = Xw*(4.45/0.3048); % N*s/m
Xq = 0; % lb*s/rad
Xwdot = 0; % lb*s^2/ft
Xdele = 3.994e4; % lb/rad
Xdele = Xdele*4.45; % N/rad
% 2nd column
Zu = -1.342e3; % lb*s/ft
Zu = Zu*(4.45/0.3048); % N*s/m
Zw = -8.561e3; % lb*s/ft
Zw = Zw*(4.45/0.3048); % N*s/m
Zq = -1.263e5; % lb*s/rad
Zq = Zq*4.45; % N*s/rad
Zwdot = 3.104e2; % lb*s^2/ft
Zwdot = Zwdot*(4.45/0.3048); % N*s^2/m
Zdele = -3.341e5; % lb/rad
Zdele = Zdele*4.45; % N/rad
% 3rd column
Mu = 8.176e3; % lb*s
Mu = Mu*4.45; % N/s
Mw = -5.627e4; % lb*s
Mw = Mw*4.45; % N/s
Mq = -1.394e7; % lb*s*ft/rad
Mq = Mq*4.45*0.3048; %N*s*m/rad
Mwdot = -4.138e3; % lb*s^2
Mwdot = Mwdot*4.45;
Mdele = -3.608e7; % lb*ft/rad
Mdele = Mdele*4.45*0.3048; % Nm/rad;
Converted = [Xu, Zu, Mu; Xw, Zw, Mw; Xq, Zq, Mq; Xwdot, Zwdot, Mwdot; Xdele, Zdele, Mdele]; % Table E.3 converted to SI units
%% Body frame to Stability frame
Xudot = Xu*((cos(zeta)^2)) - ((Xw + Zu)*sin(zeta)*cos(zeta)) + Zw*(sin(zeta)^2); % B.12,6
Xwprime = Xw*((cos(zeta)^2)) + ((Xu - Zw)*sin(zeta)*cos(zeta)) - Zu*(sin(zeta)^2); % B.12,6
Xqdot = (Xq*cos(zeta)) - (Zq*sin(zeta)); % B.12,6
Xudotdot = Zwdot*((sin(zeta))^2); % B.12,6
Xwdotdot = -Zwdot*sin(zeta)*cos(zeta); % B.12,6
```

```
Zudot = Zu*((cos(zeta)^2)) - ((Zw - Xu)*sin(zeta)*cos(zeta)) - Xw*(sin(zeta)^2); % B.12,6
Zwprime = Zw*((cos(zeta)^2)) + ((Zu + Xw)*sin(zeta)*cos(zeta)) + Xu*(sin(zeta)^2); % B.12,6
Zqdot = (Zq*cos(zeta)) + (Xq*sin(zeta)); % B.12,6
Zudotdot = -Zwdot*sin(zeta)*cos(zeta); % B.12,6
Zwdotdot = Zwdot*((cos(zeta))^2); % B.12,6

Mudot = (Mu*cos(zeta))-(Mw*sin(zeta)); % B.12,6
Mwprime = (Mw*cos(zeta)) + (Mu*sin(zeta)); % B.12,6
Mqdot = Mq; % B.12,6
Mudotdot = -Mwdot*sin(zeta); % B.12,6
Mwdotdot = Mwdot*cos(zeta); % B.12,6
Xcolumn = [Xudot, Xwprime, Xqdot, Xwdotdot]'; % define column of x values
Zcolumn = [Zudot, Zwprime, Zqdot, Zwdotdot]'; % define column of z values
Mcolumn = [Mudot, Mwprime, Mqdot, Mwdotdot]'; % define column of m values
DimDeriv = [Xcolumn, Zcolumn, Mcolumn]; % Matrix of stability derivatives in the stability frame
```

Question 3 Script

```
% This script answers question 3 by building the 4x4 A matrix found in
% table 4.9 18 in the textbook.
%   Author: Benjiman Smith
%   Collaborators: E. Owen, I. Quezada
%   Date: 3/6/2020
%
clear all
close all
W = 6.366e5;
W = 4.45*W; %N
g = 9.81; % m/s
m = W/g;

%% Table E1, Case II
Altitude = 20000; % ft
Altitude = Altitude*0.3048; % m
M = 0.5;
V = 518; % ft/s
V = V*0.3048; % m/s
W = 6.366e5; % lbf
W = 4.45*W; %N
Ix = 1.82e7; % slugs ft^2
Ix = 1.3558179619*Ix; % kg m^2
Iy = 3.31e7; % slugs ft^2
Iy = 1.3558179619*Iy; % kg m^2
Iz = 4.97e7; % slugs ft^2
Iz = 1.3558179619*Iz; % kg m^2
Izx = 9.70e5; % slugs ft^2
Izx = 1.3558179619*Izx; % kg m^2
zeta = deg2rad(-6.8); %rad
Cd = .040;
CaseII = [Altitude, M, V, W, Ix, Iy, Iz, Izx, zeta, Cd]; % define Case II SI unit equivalent of
    table E.1
%% Table E2, Longitudional
%1st column
Xu = -4.883e1; % lb*s/ft
Xu = Xu*(4.45/0.3048); % N*s/m
Xw = 1.546e3; % lb*s/ft
Xw = Xw*(4.45/0.3048); % N*s/m
Xq = 0; % lb*s/rad
Xwdot = 0; % lb*s^2/ft
Xdele = 3.994e4; % lb/rad
Xdele = Xdele*4.45; % N/rad
% 2nd column
Zu = -1.342e3; % lb*s/ft
Zu = Zu*(4.45/0.3048); % N*s/m
Zw = -8.561e3; % lb*s/ft
Zw = Zw*(4.45/0.3048); % N*s/m
```

```
Zq = -1.263e5; % lb*s/rad
Zq = Zq*4.45; % N*s/rad
Zwdot = 3.104e2; % lb*s^2/ft
Zwdot = Zwdot*(4.45/0.3048); % N*s^2/m
Zdele = -3.341e5; % lb/rad
Zdele = Zdele*4.45; % N/rad
% 3rd column
Mu = 8.176e3; % lb*s
Mu = Mu*4.45; % N/s
Mw = -5.627e4; % lb*s
Mw = Mw*4.45; % N/s
Mq = -1.394e7; % lb*s*ft/rad
Mq = Mq*4.45*0.3048; %N*s*m/rad
Mwdot = -4.138e3; % lb*s^2
Mwdot = Mwdot*4.45;
Mdele = -3.608e7; % lb*ft/rad
Mdele = Mdele*4.45*0.3048; % Nm/rad;
Converted = [Xu, Zu, Mu; Xw, Zw, Mw; Xq, Zq, Mq; Xwdot, Zwdot, Mwdot; Xdele, Zdele, Mdele]; % Table E.3 converted to SI units
%% Body frame to Stability frame
Xudot = Xu*((cos(zeta)^2)) - ((Xw + Zu)*sin(zeta)*cos(zeta)) + Zw*(sin(zeta)^2); % B.12,6
Xwprime = Xw*((cos(zeta)^2)) + ((Xu - Zw)*sin(zeta)*cos(zeta)) - Zu*(sin(zeta)^2); % B.12,6
Xqdot = (Xq*cos(zeta)) - (Zq*sin(zeta)); % B.12,6
Xudotdot = Zwdot*((sin(zeta))^2); % B.12,6
Xwdotdot = -Zwdot*sin(zeta)*cos(zeta); % B.12,6

Zudot = Zu*((cos(zeta)^2)) - ((Zw - Xu)*sin(zeta)*cos(zeta)) - Xw*(sin(zeta)^2); % B.12,6
Zwprime = Zw*((cos(zeta)^2)) + ((Zu + Xw)*sin(zeta)*cos(zeta)) + Xu*(sin(zeta)^2); % B.12,6
Zqdot = (Zq*cos(zeta)) + (Xq*sin(zeta)); % B.12,6
Zudotdot = -Zwdot*sin(zeta)*cos(zeta); % B.12,6
Zwdotdot = Zwdot*((cos(zeta))^2); % B.12,6

Mudot = (Mu*cos(zeta))-(Mw*sin(zeta)); % B.12,6
Mwprime = (Mw*cos(zeta)) + (Mu*sin(zeta)); % B.12,6
Mqdot = Mq; % B.12,6
Mudotdot = -Mwdot*sin(zeta); % B.12,6
Mwdotdot = Mwdot*cos(zeta); % B.12,6
Xcolumn = [Xudot, Xwprime, Xqdot, Xwdotdot]'; % define column of x values
Zcolumn = [Zudot, Zwprime, Zqdot, Zwdotdot]'; % define column of z values
Mcolumn = [Mudot, Mwprime, Mqdot, Mwdotdot]'; % define column of m values
DimDeriv = [Xcolumn, Zcolumn, Mcolumn]; % Matrix of stability derivatives in the stability frame
%% Question 3

A = [Xudot/m, Xwprime/m ,0, -g*cos(0);...
    Zudot/(m-Zwdotdot), Zwprime/(m-Zwdotdot),  (Zqdot+m*V)/(m-Zwdotdot), -m*g*sin(0)/(m-Zwdotdot);...
    (Mudot +(Mwdotdot*Zudot/(m-Zwdotdot)))/Iy, (Mwprime +(Mwdotdot*Zwprime/(m-Zwdotdot)))/Iy, (Mqdot +(Mwdotdot*(Zqdot +m*V)/(m-Zwdotdot)))/Iy, -Mwdotdot*m*g*sin(0)/(Iy*(m-Zwdotdot));...
    0, 0, 1, 0]
```

Question 4 Script

```
% This script answers question 4 by finding the eigenvectors and
% eigenvalues of the 4x4 A matrix derrived in Question4.m, along with their
% correspoding damping ratios and natural frequencies
%   Author: Benjiman Smith
%   Collaborators: E. Owen, I. Quezada
%   Date: 2/20/2020
%
clear all
close all
W = 6.366e5;
W = 4.45*W; %N
g = 9.81; % m/s
m = W/g;

```

```
%% Table E1, Case II
Altitude = 20000; % ft
Altitude = Altitude*0.3048; % m
M = 0.5;
V = 518; % ft/s
V = V*0.3048; % m/s
W = 6.366e5; % lbf
W = 4.45*W; %N
Ix = 1.82e7; % slugs ft^2
Ix = 1.3558179619*Ix; % kg m^2
Iy = 3.31e7; % slugs ft^2
Iy = 1.3558179619*Iy; % kg m^2
Iz = 4.97e7; % slugs ft^2
Iz = 1.3558179619*Iz; % kg m^2
Izx = 9.70e5; % slugs ft^2
Izx = 1.3558179619*Izx; % kg m^2
zeta = deg2rad(-6.8); %rad
Cd = .040;
CaseII = [Altitude, M, V, W, Ix, Iy, Iz, Izx, zeta, Cd]; % define Case II SI unit equivalent of table E.1
%% Table E2, Longitudional
%1st column
Xu = -4.883e1; % lb*s/ft
Xu = Xu*(4.45/0.3048); % N*s/m
Xw = 1.546e3; % lb*s/ft
Xw = Xw*(4.45/0.3048); % N*s/m
Xq = 0; % lb*s/rad
Xwdot = 0; % lb*s^2/ft
Xdele = 3.994e4; % lb/rad
Xdele = Xdele*4.45; % N/rad
% 2nd column
Zu = -1.342e3; % lb*s/ft
Zu = Zu*(4.45/0.3048); % N*s/m
Zw = -8.561e3; % lb*s/ft
Zw = Zw*(4.45/0.3048); % N*s/m
Zq = -1.263e5; % lb*s/rad
Zq = Zq*4.45; % N*s/rad
Zwdot = 3.104e2; % lb*s^2/ft
Zwdot = Zwdot*(4.45/0.3048); % N*s^2/m
Zdele = -3.341e5; % lb/rad
Zdele = Zdele*4.45; % N/rad
% 3rd column
Mu = 8.176e3; % lb*s
Mu = Mu*4.45; % N/s
Mw = -5.627e4; % lb*s
Mw = Mw*4.45; % N/s
Mq = -1.394e7; % lb*s*ft/rad
Mq = Mq*4.45*0.3048; %N*s*m/rad
Mwdot = -4.138e3; % lb*s^2
Mwdot = Mwdot*4.45;
Mdele = -3.608e7; % lb*ft/rad
Mdele = Mdele*4.45*0.3048; % Nm/rad;
Converted = [Xu, Zu, Mu; Xw, Zw, Mw; Xq, Zq, Mq; Xwdot, Zwdot, Mwdot; Xdele, Zdele, Mdele]; % Table E.3 converted to SI units
%% Body frame to Stability frame
Xudot = Xu*((cos(zeta)^2)) - ((Xw + Zu)*sin(zeta)*cos(zeta)) + Zw*(sin(zeta)^2); % B.12,6
Xwprime = Xw*((cos(zeta)^2)) + ((Xu - Zw)*sin(zeta)*cos(zeta)) - Zu*(sin(zeta)^2); % B.12,6
Xqdot = (Xq*cos(zeta)) - (Zq*sin(zeta)); % B.12,6
Xudotdot = Zwdot*((sin(zeta))^2); % B.12,6
Xwdotdot = -Zwdot*sin(zeta)*cos(zeta); % B.12,6

Zudot = Zu*((cos(zeta)^2)) - ((Zw - Xu)*sin(zeta)*cos(zeta)) - Xw*(sin(zeta)^2); % B.12,6
Zwprime = Zw*((cos(zeta)^2)) + ((Zu + Xw)*sin(zeta)*cos(zeta)) + Xu*(sin(zeta)^2); % B.12,6
Zqdot = (Zq*cos(zeta)) + (Xq*sin(zeta)); % B.12,6
Zudotdot = -Zwdot*sin(zeta)*cos(zeta); % B.12,6
Zwdotdot = Zwdot*((cos(zeta))^2); % B.12,6

Mudot = (Mu*cos(zeta))-(Mw*sin(zeta)); % B.12,6
```

```
Mwprime = (Mw*cos(zeta)) + (Mu*sin(zeta)); % B.12,6
Mqdot = Mq; % B.12,6
Mudotdot = -Mwdot*sin(zeta); % B.12,6
Mwdotdot = Mwdot*cos(zeta); % B.12,6
Xcolumn = [Xudot, Xwprime, Xqdot, Xwdotdot]'; % define column of x values
Zcolumn = [Zudot, Zwprime, Zqdot, Zwdotdot]'; % define column of z values
Mcolumn = [Mudot, Mwprime, Mqdot, Mwdotdot]'; % define column of m values
DimDeriv = [Xcolumn, Zcolumn, Mcolumn]; % Matrix of stability derivatives in the stability frame

%% Question 3

A = [Xudot/m, Xwprime/m ,0, -g*cos(0);...
    Zudot/(m-Zwdotdot), Zwprime/(m-Zwdotdot),  (Zqdot+m*V)/(m-Zwdotdot), -m*g*sin(0)/(m-Zwdotdot);...
    (Mudot +(Mwdotdot*Zudot/(m-Zwdotdot)))/Iy, (Mwprime +(Mwdotdot*Zwprime/(m-Zwdotdot)))/Iy, (Mqdot +(Mwdotdot*(Zqdot +m*V)/(m-Zwdotdot)))/Iy, -Mwdotdot*m*g*sin(0)/(Iy*(m-Zwdotdot));...
    0, 0, 1, 0];

%% Question 4

[eigvect, eigval] =  eig(A); % call eig function to get eigenvalues
shortperiod = eigval(1:2); % short period eigenvalues, large damping, not alot of oscillation
Phusoid = eigval(3:4); % Phusoid eigenvalues, lots of oscillation.
n = real(shortperiod(1)); % get real portion of eigenvalue
freq = imag(shortperiod(1)); % imaginary portion of eigenvalue
natfreqshort = sqrt(freq^2 + n^2); % natural frequency of short period
dampingratioshort = -n/natfreqshort; % damping ratio of short period

Phusoid = eigval(3:4); % Phusoid eigenvalues, lots of oscillation.
n = real(Phusoid(1)); % get real portion of eigenvalue
freq = imag(Phusoid(1)); % imaginary portion of eigenvalue
natfreqPhu = sqrt(freq^2 + n^2); % natural frequency of short period
dampingratioPhu = -n/natfreqPhu; % damping ratio of short period
```

Question 5 Script

```
% This script answers question 5 by outputting the results of the in class
% aproximation to be compared to the short period mode of the 4x4 A matrix
% found in Question4.m. This script also compares the oscillation period of
% the Phugoid mode found in Question4.m to the Lanchester approximation in the textbook
%   Author: Benjiman Smith
%   Collaborators: E. Owen, I. Quezada
%   Date: 3/7/2020
%
clear all
close all
W = 6.366e5;
W = 4.45*W; %N
g = 9.81; % m/s
m = W/g;

%% Table E1, Case II
Altitude = 20000; % ft
Altitude = Altitude*0.3048; % m
M = 0.5;
V = 518; % ft/s
V = V*0.3048; % m/s
W = 6.366e5; % lbf
W = 4.45*W; %N
Ix = 1.82e7; % slugs ft^2
Ix = 1.3558179619*Ix; % kg m^2
Iy = 3.31e7; % slugs ft^2
Iy = 1.3558179619*Iy; % kg m^2
Iz = 4.97e7; % slugs ft^2
Iz = 1.3558179619*Iz; % kg m^2
Izx = 9.70e5; % slugs ft^2
```

```
Izx = 1.3558179619*Izx; % kg m^2
zeta = deg2rad(-6.8); %rad
Cd = .040;
CaseII = [Altitude, M, V, W, Ix, Iy, Iz, Izx, zeta, Cd]; % define Case II SI unit equivalent of table E.1
%% Table E2, Longitudional
%1st column
Xu = -4.883e1; % lb*s/ft
Xu = Xu*(4.45/0.3048); % N*s/m
Xw = 1.546e3; % lb*s/ft
Xw = Xw*(4.45/0.3048); % N*s/m
Xq = 0; % lb*s/rad
Xwdot = 0; % lb*s^2/ft
Xdele = 3.994e4; % lb/rad
Xdele = Xdele*4.45; % N/rad
% 2nd column
Zu = -1.342e3; % lb*s/ft
Zu = Zu*(4.45/0.3048); % N*s/m
Zw = -8.561e3; % lb*s/ft
Zw = Zw*(4.45/0.3048); % N*s/m
Zq = -1.263e5; % lb*s/rad
Zq = Zq*4.45; % N*s/rad
Zwdot = 3.104e2; % lb*s^2/ft
Zwdot = Zwdot*(4.45/0.3048); % N*s^2/m
Zdele = -3.341e5; % lb/rad
Zdele = Zdele*4.45; % N/rad
% 3rd column
Mu = 8.176e3; % lb*s
Mu = Mu*4.45; % N/s
Mw = -5.627e4; % lb*s
Mw = Mw*4.45; % N/s
Mq = -1.394e7; % lb*s*ft/rad
Mq = Mq*4.45*0.3048; %N*s*m/rad
Mwdot = -4.138e3; % lb*s^2
Mwdot = Mwdot*4.45;
Mdele = -3.608e7; % lb*ft/rad
Mdele = Mdele*4.45*0.3048; % Nm/rad;
Converted = [Xu, Zu, Mu; Xw, Zw, Mw; Xq, Zq, Mq; Xwdot, Zwdot, Mwdot; Xdele, Zdele, Mdele]; % Table E.3 converted to SI units
%% Body frame to Stability frame
Xudot = Xu*((cos(zeta)^2)) - ((Xw + Zu)*sin(zeta)*cos(zeta)) + Zw*(sin(zeta)^2); % B.12,6
Xwprime = Xw*((cos(zeta)^2)) + ((Xu - Zw)*sin(zeta)*cos(zeta)) - Zu*(sin(zeta)^2); % B.12,6
Xqdot = (Xq*cos(zeta)) - (Zq*sin(zeta)); % B.12,6
Xudotdot = Zwdot*((sin(zeta))^2); % B.12,6
Xwdotdot = -Zwdot*sin(zeta)*cos(zeta); % B.12,6

Zudot = Zu*((cos(zeta)^2)) - ((Zw - Xu)*sin(zeta)*cos(zeta)) - Xw*(sin(zeta)^2); % B.12,6
Zwprime = Zw*((cos(zeta)^2)) + ((Zu + Xw)*sin(zeta)*cos(zeta)) + Xu*(sin(zeta)^2); % B.12,6
Zqdot = (Zq*cos(zeta)) + (Xq*sin(zeta)); % B.12,6
Zudotdot = -Zwdot*sin(zeta)*cos(zeta); % B.12,6
Zwdotdot = Zwdot*((cos(zeta))^2); % B.12,6

Mudot = (Mu*cos(zeta))-(Mw*sin(zeta)); % B.12,6
Mwprime = (Mw*cos(zeta)) + (Mu*sin(zeta)); % B.12,6
Mqdot = Mq; % B.12,6
Mudotdot = -Mwdot*sin(zeta); % B.12,6
Mwdotdot = Mwdot*cos(zeta); % B.12,6
Xcolumn = [Xudot, Xwprime, Xqdot, Xwdotdot]'; % define column of x values
Zcolumn = [Zudot, Zwprime, Zqdot, Zwdotdot]'; % define column of z values
Mcolumn = [Mudot, Mwprime, Mqdot, Mwdotdot]'; % define column of m values
DimDeriv = [Xcolumn, Zcolumn, Mcolumn]; % Matrix of stability derivatives in the stability frame


A = [Xudot/m, Xwprime/m ,0, -g*cos(0);...
    Zudot/(m-Zwdotdot), Zwprime/(m-Zwdotdot),  (Zqdot+m*V)/(m-Zwdotdot), -m*g*sin(0)/(m-Zwdotdot);...
    (Mudot +(Mwdotdot*Zudot/(m-Zwdotdot)))/Iy, (Mwprime +(Mwdotdot*Zwprime/(m-Zwdotdot)))/Iy, (Mqdot +(Mwdotdot*(Zqdot +m*V)/(m-Zwdotdot)))/Iy, -Mwdotdot*m*g*sin(0)/(Iy*(m-Zwdotdot))
```

```
        ;...
    0, 0, 1, 0];


[eigvect, eigval] =  eig(A); % call eig function to get eigenvalues
shortperiod = eigval(1:2); % short period eigenvalues, large damping, not alot of oscillation
Phusoid = eigval(3:4); % Phusoid eigenvalues, lots of oscillation.
n = real(shortperiod(1)); % get real portion of eigenvalue
freq = imag(shortperiod(1)); % imaginary portion of eigenvalue
natfreqshort = sqrt(freq^2 + n^2); % natural frequency of short period
dampingratioshort = -n/natfreqshort; % damping ratio of short period

Phusoid = eigval(3:4); % Phusoid eigenvalues, lots of oscillation.
n = real(Phusoid(1)); % get real portion of eigenvalue
freq = imag(Phusoid(1)); % imaginary portion of eigenvalue
natfreqPhu = sqrt(freq^2 + n^2); % natural frequency of short period
dampingratioPhu = -n/natfreqPhu; % damping ratio of short period

%% Question 5
B = [Mqdot/Iy, V*Mwprime/Iy; ...
    1, 0];
eigval2 = eig(B)
T_Phu = 2*pi/ freq
T_Lanchester = 0.138*V
```

Question 6 Script

```
% This script answers question 6 by simulating the longitudional dynamics
% of the 747-100 based off of the 4x4 A matrix derrived in Question3.m
% Author: Benjiman Smith
% Collaborators: E. Owen, I. Quezada
% Date: 3/06/2020
%
clear all
close all
W = 6.366e5;
W = 4.45*W; %N
g = 9.81; % m/s
m = W/g;

%% Table E1, Case II
Altitude = 20000; % ft
Altitude = Altitude*0.3048; % m
M = 0.5;
V = 518; % ft/s
V = V*0.3048; % m/s
W = 6.366e5; % lbf
W = 4.45*W; %N
Ix = 1.82e7; % slugs ft^2
Ix = 1.3558179619*Ix; % kg m^2
Iy = 3.31e7; % slugs ft^2
Iy = 1.3558179619*Iy; % kg m^2
Iz = 4.97e7; % slugs ft^2
Iz = 1.3558179619*Iz; % kg m^2
Izx = 9.70e5; % slugs ft^2
Izx = 1.3558179619*Izx; % kg m^2
zeta = deg2rad(-6.8); %rad
Cd = .040;
CaseII = [Altitude, M, V, W, Ix, Iy, Iz, Izx, zeta, Cd]; % define Case II SI unit equivalent of table E.1
%% Table E2, Longitudional
%1st column
Xu = -4.883e1; % lb*s/ft
Xu = Xu*(4.45/0.3048); % N*s/m
Xw = 1.546e3; % lb*s/ft
Xw = Xw*(4.45/0.3048); % N*s/m
Xq = 0; % lb*s/rad
```

```
Xwdot = 0; % lb*s^2/ft
Xdele = 3.994e4; % lb/rad
Xdele = Xdele*4.45; % N/rad
% 2nd column
Zu = -1.342e3; % lb*s/ft
Zu = Zu*(4.45/0.3048); % N*s/m
Zw = -8.561e3; % lb*s/ft
Zw = Zw*(4.45/0.3048); % N*s/m
Zq = -1.263e5; % lb*s/rad
Zq = Zq*4.45; % N*s/rad
Zwdot = 3.104e2; % lb*s^2/ft
Zwdot = Zwdot*(4.45/0.3048); % N*s^2/m
Zdele = -3.341e5; % lb/rad
Zdele = Zdele*4.45; % N/rad
% 3rd column
Mu = 8.176e3; % lb*s
Mu = Mu*4.45; % N/s
Mw = -5.627e4; % lb*s
Mw = Mw*4.45; % N/s
Mq = -1.394e7; % lb*s*ft/rad
Mq = Mq*4.45*0.3048; %N*s*m/rad
Mwdot = -4.138e3; % lb*s^2
Mwdot = Mwdot*4.45;
Mdele = -3.608e7; % lb*ft/rad
Mdele = Mdele*4.45*0.3048; % Nm/rad;
Converted = [Xu, Zu, Mu; Xw, Zw, Mw; Xq, Zq, Mq; Xwdot, Zwdot, Mwdot; Xdele, Zdele, Mdele]; % Table E.3 converted to SI units
%% Body frame to Stability frame
Xudot = Xu*((cos(zeta)^2)) - ((Xw + Zu)*sin(zeta)*cos(zeta)) + Zw*(sin(zeta)^2); % B.12,6
Xwprime = Xw*((cos(zeta)^2)) + ((Xu - Zw)*sin(zeta)*cos(zeta)) - Zu*(sin(zeta)^2); % B.12,6
Xqdot = (Xq*cos(zeta)) - (Zq*sin(zeta)); % B.12,6
Xudotdot = Zwdot*((sin(zeta))^2); % B.12,6
Xwdotdot = -Zwdot*sin(zeta)*cos(zeta); % B.12,6

Zudot = Zu*((cos(zeta)^2)) - ((Zw - Xu)*sin(zeta)*cos(zeta)) - Xw*(sin(zeta)^2); % B.12,6
Zwprime = Zw*((cos(zeta)^2)) + ((Zu + Xw)*sin(zeta)*cos(zeta)) + Xu*(sin(zeta)^2); % B.12,6
Zqdot = (Zq*cos(zeta)) + (Xq*sin(zeta)); % B.12,6
Zudotdot = -Zwdot*sin(zeta)*cos(zeta); % B.12,6
Zwdotdot = Zwdot*((cos(zeta))^2); % B.12,6

Mudot = (Mu*cos(zeta))-(Mw*sin(zeta)); % B.12,6
Mwprime = (Mw*cos(zeta)) + (Mu*sin(zeta)); % B.12,6
Mqdot = Mq; % B.12,6
Mudotdot = -Mwdot*sin(zeta); % B.12,6
Mwdotdot = Mwdot*cos(zeta); % B.12,6
Xcolumn = [Xudot, Xwprime, Xqdot, Xwdotdot]'; % define column of x values
Zcolumn = [Zudot, Zwprime, Zqdot, Zwdotdot]'; % define column of z values
Mcolumn = [Mudot, Mwprime, Mqdot, Mwdotdot]'; % define column of m values
DimDeriv = [Xcolumn, Zcolumn, Mcolumn]; % Matrix of stability derivatives in the stability frame


A = [Xudot/m, Xwprime/m ,0, -g*cos(0);...
    Zudot/(m-Zwdotdot), Zwprime/(m-Zwdotdot),  (Zqdot+m*V)/(m-Zwdotdot), -m*g*sin(0)/(m-Zwdotdot);...
    (Mudot +(Mwdotdot*Zudot/(m-Zwdotdot)))/Iy, (Mwprime +(Mwdotdot*Zwprime/(m-Zwdotdot)))/Iy, (Mqdot +(Mwdotdot*(Zqdot +m*V)/(m-Zwdotdot)))/Iy, -Mwdotdot*m*g*sin(0)/(Iy*(m-Zwdotdot));...
    0, 0, 1, 0];

%% Question 6

conditions = [0,0,0,0]'; % ?u^E is 10 m/s
tspan = linspace(0,300); % time vector from 0 300 seconds
[t1, X] = ode45(@(t, F)Specs3LB5LC(t, F, A), tspan, conditions); % nonlinear ODE

%sublotting of all 8 variables
figure()
```

```
sgtitle('Linearized Lognitudional dynamics at trimmed state', 'interpreter', 'latex');
subplot(2,2,1);
plot(t1, X(:,1),'linewidth', 2);
grid on
xlabel('time (s)')
ylabel('\Delta u (m/s)')
title('change in deviation in u over time')

subplot(2,2,2);
plot(t1, X(:,2),'linewidth', 2);
grid on
xlabel('time (s)')
ylabel('\Delta w (m/s)')
title('change in w over time')

subplot(2,2,3);
plot(t1, X(:,3),'linewidth', 2);
grid on
xlabel('time (s)')
ylabel('q (rad/s)')
title('change in pitch rate over time')

subplot(2,2,4);
plot(t1, X(:,4),'linewidth', 2);
grid on
xlabel('time (s)')
ylabel('\Delta \theta (rad)')
title('change in deviation in elevation over time')



% options = odeset('Events', @StopFnct, 'RelTol', 1e-8); % stop function that ends ODE when a
    tolerance of 1e-8 is met
conditions = [10,0,0,0]'; % ?u^E is 10 m/s
tspan = linspace(0,300); % time vector from 0 300 seconds
[t1, X] = ode45(@(t, F)Specs3LB5LC(t, F, A), tspan, conditions); % nonlinear ODE

%sublotting of all 8 variables
figure()

sgtitle('Linearized Lognitudional dynamics with initial $\Delta u^{E}$ = 10 $\frac{m}{s}$', '
    interpreter', 'latex');
subplot(2,2,1);
plot(t1, X(:,1),'linewidth', 2);
grid on
xlabel('time (s)')
ylabel('\Delta u (m/s)')
title('change in deviation in u over time')

subplot(2,2,2);
plot(t1, X(:,2),'linewidth', 2);
grid on
xlabel('time (s)')
ylabel('\Delta w (m/s)')
title('change in w over time')

subplot(2,2,3);
plot(t1, X(:,3),'linewidth', 2);
grid on
xlabel('time (s)')
ylabel('q (rad/s)')
title('change in pitch rate over time')

subplot(2,2,4);
plot(t1, X(:,4),'linewidth', 2);
grid on
xlabel('time (s)')
ylabel('\Delta \theta (rad)')
```

```
title('change in deviation in elevation over time')

conditions = [0,10,0,0]'; % ?w^E is 10 m/s
tspan = linspace(0,300); % time vector from 0 300 seconds
[t1, X] = ode45(@(t, F)Specs3LB5LC(t, F, A), tspan, conditions); % nonlinear ODE


figure()

sgtitle('Linearized Lognitudional dynamics with initial $\Delta w^{E}$ = 10 $\frac{m}{s}$', 'interpreter', 'latex');
subplot(2,2,1);
plot(t1, X(:,1),'linewidth', 2);
grid on
xlabel('time (s)')
ylabel('\Delta u (m/s)')
title('change in deviation in u over time')

subplot(2,2,2);
plot(t1, X(:,2),'linewidth', 2);
grid on
xlabel('time (s)')
ylabel('\Delta w (m/s)')
title('change in w over time')

subplot(2,2,3);
plot(t1, X(:,3),'linewidth', 2);
grid on
xlabel('time (s)')
ylabel('q (rad/s)')
title('change in pitch rate over time')

subplot(2,2,4);
plot(t1, X(:,4),'linewidth', 2);
grid on
xlabel('time (s)')
ylabel('\Delta \theta (rad)')
title('change in deviation in elevation over time')
conditions = [0,0,0.1,0]'; % q^E is 0.1 rad/s
tspan = linspace(0,300); % time vector from 0 300 seconds
[t1, X] = ode45(@(t, F)Specs3LB5LC(t, F, A), tspan, conditions); % nonlinear ODE

figure()

sgtitle('Linearized Lognitudional dynamics with initial $\Delta q$ = 0.1 $\frac{rad}{s}$', 'interpreter', 'latex');
subplot(2,2,1);
plot(t1, X(:,1),'linewidth', 2);
grid on
xlabel('time (s)')
ylabel('\Delta u (m/s)')
title('change in deviation in u over time')

subplot(2,2,2);
plot(t1, X(:,2),'linewidth', 2);
grid on
xlabel('time (s)')
ylabel('\Delta w (m/s)')
title('change in w over time')

subplot(2,2,3);
plot(t1, X(:,3),'linewidth', 2);
grid on
xlabel('time (s)')
ylabel('q (rad/s)')
title('change in pitch rate over time')

subplot(2,2,4);
```

```
plot(t1, X(:,4),'linewidth', 2);
grid on
xlabel('time (s)')
ylabel('\Delta \theta (rad)')
title('change in deviation in elevation over time')

conditions = [0,0,0,0.1]'; % q is 0.1 rad/s
tspan = linspace(0,300); % time vector from 0 300 seconds
[t1, X] = ode45(@(t, F)Specs3LB5LC(t, F, A), tspan, conditions); % nonlinear ODE

figure()

sgtitle('Linearized Lognitudional dynamics with initial $\Delta \theta$ = 0.1 $rad$', '
    interpreter', 'latex');
subplot(2,2,1);
plot(t1, X(:,1),'linewidth', 2);
grid on
xlabel('time (s)')
ylabel('\Delta u (m/s)')
title('change in deviation in u over time')

subplot(2,2,2);
plot(t1, X(:,2),'linewidth', 2);
grid on
xlabel('time (s)')
ylabel('\Delta w (m/s)')
title('change in w over time')

subplot(2,2,3);
plot(t1, X(:,3),'linewidth', 2);
grid on
xlabel('time (s)')
ylabel('q (rad/s)')
title('change in pitch rate over time')

subplot(2,2,4);
plot(t1, X(:,4),'linewidth', 2);
grid on
xlabel('time (s)')
ylabel('\Delta \theta (rad)')
title('change in deviation in elevation over time')
```

ODE Function

```
% This Function is called in the ODE function to show the changes in u, w,
% p, and theta over time.
%   Author: Benjiman Smith
%   Collaborators: E. Owen, I. Quezada
%   Date: 2/20/2020
%
function dydt = Specs2LB4NLC(t, Conditions, A) % function start
    dydt = A*Conditions; % Utilize A matrix to find diferential results

end % end
```