ASEN 3128 - Assignment 1

Connor Ott

Group 25 - Station 13a
March 11, 2018
University of Colorado - Boulder

Problem 1

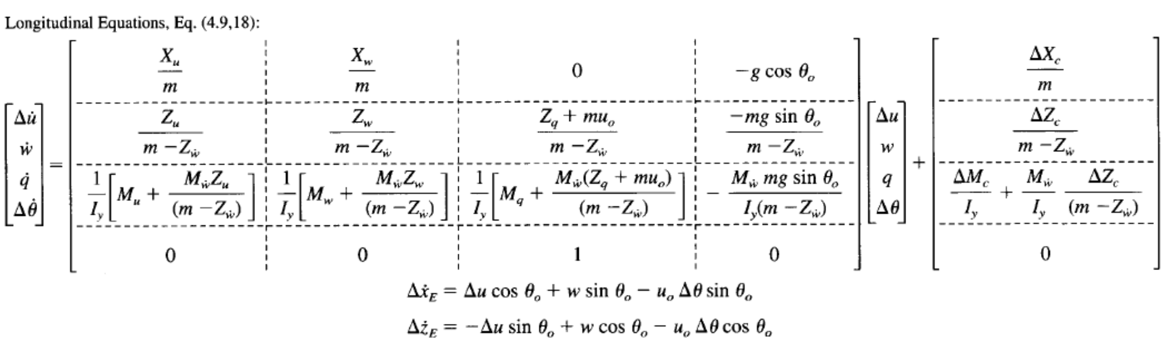
For this problem, in preparation for making the A matrix, all the longitudinal derivatives were computed. These are shown in in 1.

	X	Y	M
u	-2.21e+03	-2.35e+04	1.78e+04
w	4.49e+03	-1.01e+05	-1.74e+05
q	0.00e+00	-5.04e+05	-1.70e+07
$\dot{w}$	0.00e+00	1.89e+03	-1.69e+04

Table 1: Aerodynamic Derivatives

Problem 2

With the longitudinal derivatives, the A matrix was constructed according to the table taken from *Dynamics of Flight* on page 112. This is shown in Figure 1.



Longitudinal Equations, Eq. (4.9,18):

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

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

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

Figure 1

The A matrix itself is shown below.

$$A = \begin{bmatrix} -7.65e-03 & 1.55e-02 & 0.00e+00 & -9.81e+00 \\ -8.20e-02 & -3.51e-01 & 2.65e+02 & 0.00e+00 \\ 4.26e-04 & -3.75e-03 & -4.77e-01 & 0.00e+00 \\ 0.00e+00 & 0.00e+00 & 1.00e+00 & 0.00e+00 \end{bmatrix}$$

where,

$$\begin{bmatrix} \dot{\Delta u} \\ \dot{\Delta w} \\ \dot{\Delta q} \\ \dot{\Delta \theta} \end{bmatrix} = A \begin{bmatrix} \Delta u \\ \Delta w \\ \Delta q \\ \Delta \theta \end{bmatrix}$$

Problem 3

The eigenvalues of the above A matrix can be used to predict the plane behavior for both the short period and phugoid modes. Becuase there are two mode of behavior, two conjugate pairs of eigenvalues appear. These pairs are λ = -0.414 $\pm$ 0.994i and λ = -0.0038 $\pm$ 0.062i. The first corresponds to the short period mode, as indicated by it's more negative real part and greater imaginary part. The natural frequency and damping ratio of these modes can be determined with the following equations.

$$\omega_n = \sqrt{Re(\lambda)^2 + Im(\lambda)^2}$$
$$\zeta = -Re(\lambda)/\omega_n$$

The natural frequency and damping ratio of the short period mode are 1.076 rad/s and 0.385. For the phugoid mode, these are 0.062 rad/s and 0.061.

Problem 4

Using the short period approximation of the longitudinal dynamics, the following eigenvalue pair was found. λ = -0.1888 $\pm$ 0.9971i. The short period mode using the full longitudinal dynamics yielded the pair λ = -0.414 $\pm$ 0.994i. This tells us that the approximation holds extremely well for the imaginary part of the eigenvalues. The real portion, though on the same order of magnitude, is not approximated as well.

The oscillation period of the phugoid mode can also be compared to an approximation. *Dynamics of Flight* uses the Lanchester approximation to look at the phugoid mode period. This equation is shown below.

$$T = \frac{\pi\sqrt{2}u_0}{g} \tag{1}$$

This can then be compared to the full dynamics simulation. The oscillation frequency is equal to the imaginary part of the eigenvalue, this is used to compute the period. The approximate period was found to be 120.23 seconds. The full dynamic period was found to be 101.28 seconds.

Problem 5

The full dynamics were simulated with different perturbations in the initial state vector. The results of these are shown in the following figures.

i. Part a

To start, the simulation was verified by checking that the trim state was in equilibrium.

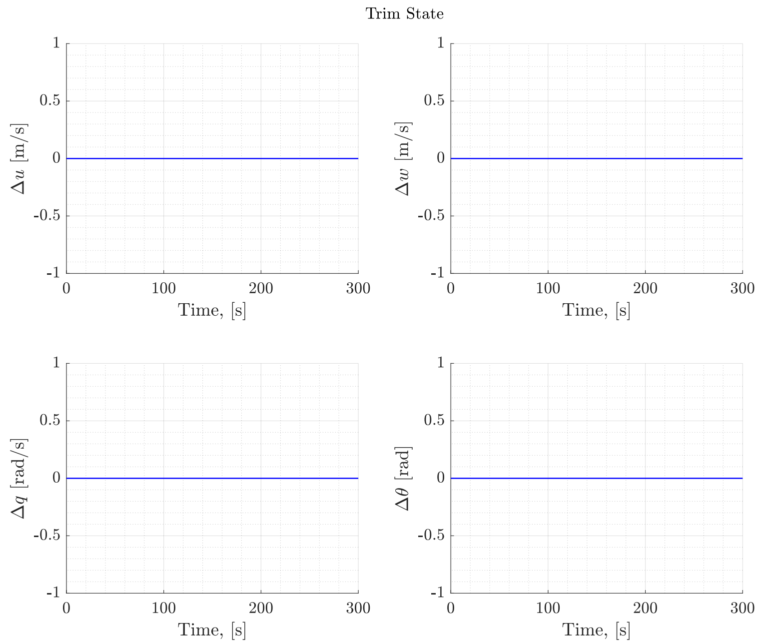


Figure 2: Trim state verification.

ii. Part b

Next, the trim state was perturbed to see how the short period and phugoid modes were excited.

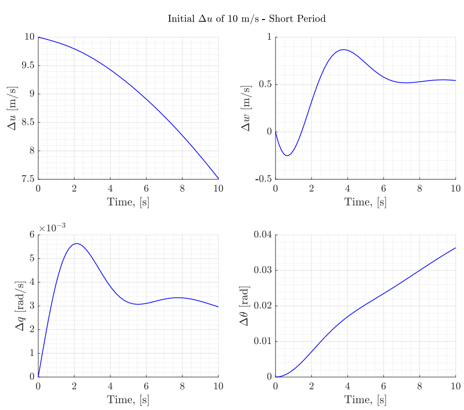


Figure 3: Initial deviation in u - short period mode.

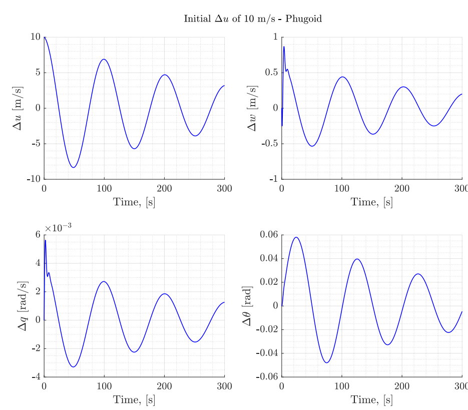


Figure 4: Initial deviation in u - phugoid mode.

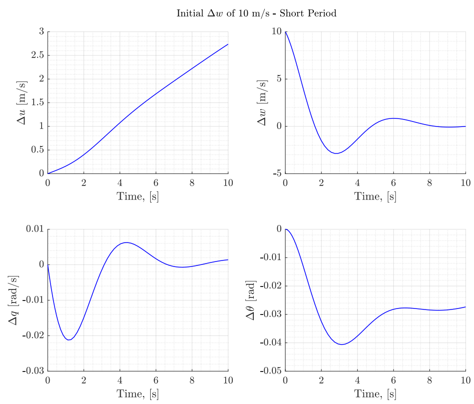


Figure 5: Initial deviation in w - short period mode.

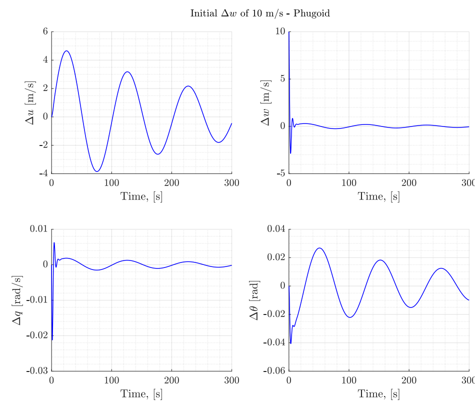


Figure 6: Initial deviation in w - phugoid mode.

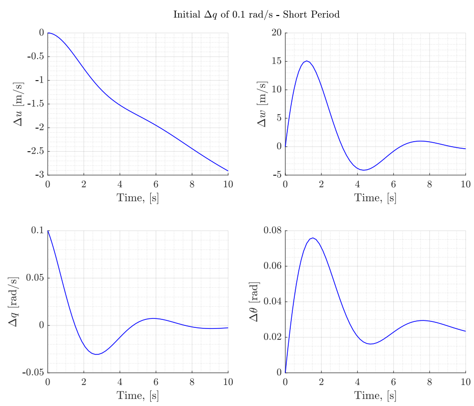


Figure 7: Initial deviation in q - short period mode.

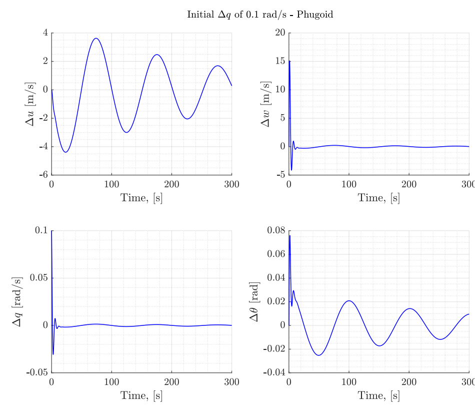


Figure 8: Initial deviation in q - phugoid mode.

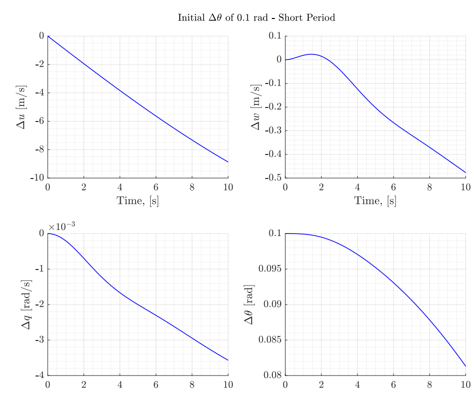


Figure 9: Initial deviation in θ - short period mode.

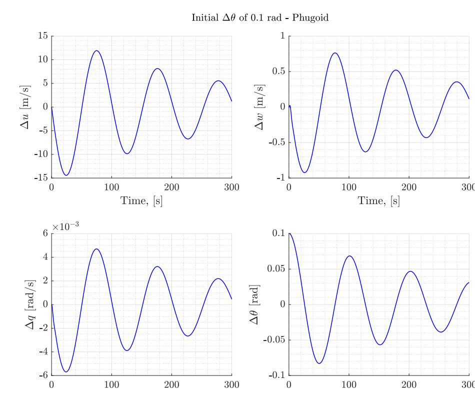


Figure 10: Initial deviation in θ - phugoid mode.

iii. Part c

It can be gathered from these that deviation in q and θ are best at exiting the short term mode, while deviations in u and w are best at exciting changes in the phugoid mode.

MATLAB Code

```
%------------------------------------------------------------------------
% Main function for calculations for Assignment 6 problems
%
% Created: 3/7/18 - Connor Ott
%------------------------------------------------------------------------

clc; clear; close all;
mpf = 0.3048;     % meters per foot
h = 40000 * mpf; % [ft]
[~, ~, ~, rho] = atmosisa(h);

%% Problem 1 - Aerodynamic Derivatives
% Values from table 6.1
C_X_u = -0.108;
C_xa = 0.2193;
C_xq = 0;
C_xahat = 0;

C_zu = -0.106;
C_za = -4.920;
C_zq = -5.921;
C_zahat = 5.896;

C_mu = 0.1043;
C_ma = -1.023;
C_mq = -23.92;
C_mahat = -6.314;

% Values from table E.1
theta_0 = 0;             % [deg]
u_0 = 265.48;            % [m/s]
w_0 = 0;                 % [m/s]
S = 5500 * mpf^2;        % [m^2]
cbar = 27.31 * mpf;      % [m]
W = 6.366e5 * 4.44822; % [N]
C_w0 = W / (0.5*rho*u_0^2*S);

% X derivatives
X_u = rho*u_0*S*C_w0*sind(theta_0) + 0.5*rho*u_0*S*C_X_u;
X_w = 0.5*rho*u_0*S*C_xa;
X_q = 0.25*rho*u_0*cbar*S*C_xq;
X_wdot = 0.25*rho*cbar*S*C_xahat;

% Z derivatives
Z_u = -rho*u_0*S*C_w0*cosd(theta_0) + 0.5*rho*u_0*S*C_zu;
Z_w = 0.5*rho*u_0*S*C_za;
Z_q = 0.25*rho*u_0*cbar*S*C_zq;
Z_wdot = 0.25*rho*cbar*S*C_zahat;

% Moment derivative
M_u = 0.5*rho*u_0*cbar*S*C_mu;
M_w = 0.5*rho*u_0*cbar*S*C_ma;
M_q = 0.25*rho*u_0*cbar^2*S*C_mq;
```

```matlab
M_wdot = 0.25*rho*cbar^2*S*C_mahat;

% Final Longitudinal Dimensional Derivatives
Xcol = [X_u, X_w, X_q, X_wdot]';
Zcol = [Z_u, Z_w, Z_q, Z_wdot]';
Mcol = [M_u, M_w, M_q, M_wdot]';
dataMat = [Xcol, Zcol, Mcol];
% disp(T)

% Creating LaTex table
input.data = dataMat;
input.tableColLabels = {'X', 'Y', 'M'};
input.tableRowLabels = {'u', 'w', 'q', '\dot{w}'};
input.dataFormat = {'%.2e', 2, '%.2e', 1};
input.tableColumnAlignment = 'c';
input.dataNanString = '-';
input.tableBorders = 0;
input.booktabs = 1;
input.tableCaption = 'Aerodynamic Derivatives';
input.tableLabel = 'aeroDer';
% call latexTable:
latex = latexTable(input); % Maybe don't output this every time



%% Problem 2 - A Matrix
W = 2.83176e6; % [N]
g = 9.81; % [m/s^2]
m = W/g; % [kg]
Ix = 0.247e8;
Iy = 0.449e8;
Iz = 0.673e8;


A = [X_u/m, X_w/m, 0, -g*cos(theta_0); ...
     Z_u/(m-Z_wdot), Z_w/(m-Z_wdot), (Z_q+m*u_0)/(m-Z_wdot), -m*g*sin(theta_0)/(m-Z_wdot); ...
     (1/Iy)*(M_u + M_wdot*Z_u/(m-Z_wdot)), (1/Iy)*(M_w + M_wdot*Z_w/(m-Z_wdot)), ...
     (1/Iy)*(M_q + M_wdot*(Z_q+m*u_0)/(m-Z_wdot)), M_wdot*m*g*sin(theta_0)/...
     (Iy*(m-Z_wdot)); ...
     0, 0, 1, 0];

[vecs, vals] = eig(A, 'vector');

% Creating LaTex table
input.data = A;
input.tableColLabels = {'', '', '', ''};
input.tableRowLabels = {'', '', '', ''};
input.dataFormat = {'%.2e', 3, '%.2e', 1};
input.tableColumnAlignment = 'c';
input.dataNanString = '-';
input.tableBorders = 0;
input.booktabs = 1;
input.tableCaption = '';
```

```matlab
input.tableLabel = '';
% call latexTable:
latex = latexTable(input); % Maybe don't output this every time


%% Problem 3 - eigs, zeta, omega_n
phuVal = vals(3:4);
shortVal = vals(1:2);

% Natural Frequency
w_nPhu = abs(phuVal(1));
w_nShort = abs(shortVal(1));

zetaPhu = -real(phuVal(1))/w_nPhu;
zetaShort = -real(shortVal(1))/w_nShort;


%% Problem 4 - Approximations vs. Full A Matrix
lamSaprx(:, 1) = [ (M_q/(2*Iy) + 1/(2*Iy)*sqrt(M_q^2 + 4*Iy*u_0*M_w));
                   (M_q/(2*Iy) - 1/(2*Iy)*sqrt(M_q^2 + 4*Iy*u_0*M_w)) ];

TPhu_aprx =  pi * sqrt(2) * u_0 / g
TPhu = 2*pi/imag(phuVal(1))

%% Problem 5 - ODE Sim
initCondsMat = diag([10, 10, 0.1, 0.1]);
tSpanLong = [0, 300];
tSpanShort = [0, 10];
titles = {'Initial $\Delta u$ of 10 m/s', ...
          'Initial $\Delta w$ of 10 m/s', ...
          'Initial $\Delta q$ of 0.1 rad/s', ...
          'Initial $\Delta \theta$ of 0.1 rad'};
yLabels = {'$\Delta u$ [m/s]', '$\Delta w$ [m/s]', ...
           '$\Delta q$ [rad/s]', '$\Delta \theta$ [rad]'};
printTitles = {'deltaU', 'deltaW','deltaQ', 'deltaTheta'};

for i = 1:length(initCondsMat)
    [t_s, F_s] = ode45(@(t, F)longSimODE(t, F, A), tSpanShort, ...
                                        initCondsMat(i, :));
    [t_p, F_p] = ode45(@(t, F)longSimODE(t, F, A), tSpanLong, ...
                                        initCondsMat(i, :));
   % Short Period Plots
    plotPlots(t_s, F_s, titles{i}, yLabels, printTitles{i})

   % Long Period Plots
    plotPlots(t_p, F_p, titles{i}, yLabels, printTitles{i})
end
```

```matlab
function dfdt = longSimODE(t, F, A)

% Lol
dfdt = A*F;

end
```

```matlab
function [] = plotPlots(t, F, title, yLabels, pTitle)
```

```
% Plot my subplots for the longitudinal dynamics simulation
set(0, 'defaulttextinterpreter', 'latex');
figure
set(gcf, 'units', 'normalized', ...
    'pos', [0.1, 0.1, 0.75, 0.75])

subplot(2, 2, 1)
hold on; grid on; grid minor;
plot(t, F(:, 1), 'b-', 'linewidth', 1.1)
xlabel('Time, [s]')
ylabel(yLabels{1})
set(gca, 'ticklabelinterpreter', 'latex', ...
         'fontsize', 15)

subplot(2, 2, 2)
hold on; grid on; grid minor;
plot(t, F(:, 2), 'b-', 'linewidth', 1.1)
xlabel('Time, [s]')
ylabel(yLabels{2})
set(gca, 'ticklabelinterpreter', 'latex', ...
         'fontsize', 15)

subplot(2, 2, 3)
hold on; grid on; grid minor;
plot(t, F(:, 3), 'b-', 'linewidth', 1.1)
xlabel('Time, [s]')
ylabel(yLabels{3})
set(gca, 'ticklabelinterpreter', 'latex', ...
         'fontsize', 15)

subplot(2, 2, 4)
hold on; grid on; grid minor;
plot(t, F(:, 4), 'b-', 'linewidth', 1.1)
xlabel('Time, [s]')
ylabel(yLabels{4})
set(gca, 'ticklabelinterpreter', 'latex', ...
         'fontsize', 15)

[~, t] = suplabel(title, 't');
set(t, 'interpreter', 'latex', ...
        'fontsize', 15)

currentPath = pwd;
if exist(['./Figures/', pTitle, '.pdf'], 'file') ~= 2
    set(gcf, 'PaperOrientation', 'landscape');
    set(gcf, 'PaperUnits', 'normalized');
    set(gcf, 'PaperPosition', [0 0 1 1]);
    print(gcf, '-dpdf', [currentPath, '/Figures/', pTitle]);
end

end
```