$$\begin{bmatrix} \dot{V_T} \\ \dot{\alpha} \\ \dot{q} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} -0.0131 & -4.52 & -0.188 & -32.2 \\ -0.000162 & -0.846 & 0.995 & 0 \\ -3.77e-11 & -3.32 & -0.793 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} V_T \\ \alpha \\ q \\ \theta \end{bmatrix} + \begin{bmatrix} 3.89 \\ -0.129 \\ -9.19 \\ 0 \end{bmatrix} \begin{bmatrix} \delta e \end{bmatrix} \tag{1}$$

The short period eigenvalues are -0.00561 and -0.00561, natural frequencies 0.0661 and 0.0661, damping ratios 0.0849 and 0.0849.
The phugoid eigenvalues are -0.82 and -0.82, natural frequencies 1.99 and 1.99, damping ratios 0.411 and 0.411. The system is stable because the eigenvalues are all negative.
The pitch rate transfer function is:

$$\frac{q(s)}{\delta e(s)} = \frac{0s^4 + -527s^3 + -428s^2 + -5.02s + 5.68e-17}{1s^4 + 1.65s^3 + 4s^2 + 0.0518s + 0.0174} \tag{2}$$

The angle of attack transfer function is:

$$\frac{\alpha(s)}{\delta e(s)} = \frac{0s^4 + -7.39s^3 + -530s^2 + -7.01s + -2.75}{1s^4 + 1.65s^3 + 4s^2 + 0.0518s + 0.0174} \tag{3}$$

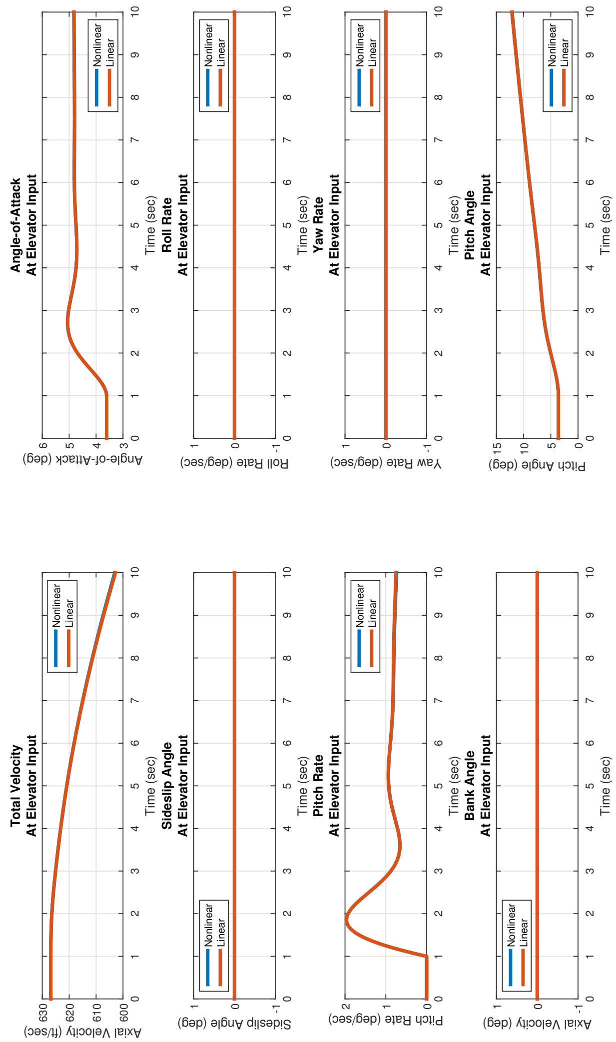

Figure 1: Elevator Input Simulation Graphical Solution

$$\begin{bmatrix} \dot{\beta} \\ \dot{p} \\ \dot{r} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} -0.232 & 0.0633 & -0.996 & 0.0512 \\ -29.5 & -3.02 & 0.0201 & 0 \\ 6.23 & -0.0274 & -0.417 & 0 \\ 0 & 1 & 0.0631 & 0 \end{bmatrix} \begin{bmatrix} V_T \\ \alpha \\ q \\ \theta \end{bmatrix} + \begin{bmatrix} 0.00521 & 0.031 \\ -36.5 & 8.11 \\ -0.492 & -2.83 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \delta a \\ \delta r \end{bmatrix} \tag{4}$$

The dutch roll mode eigenvalues are -0.408 and -0.408, natural frequencies 2.78 and 2.78, damping ratios 0.147 and 0.147. The roll mode eigenvalue is -2.82, and time constant 0.354. The spiral mode eigenvalue is -0.0257, and time constant 39. The system is stable because the eigenvalues are all negative.

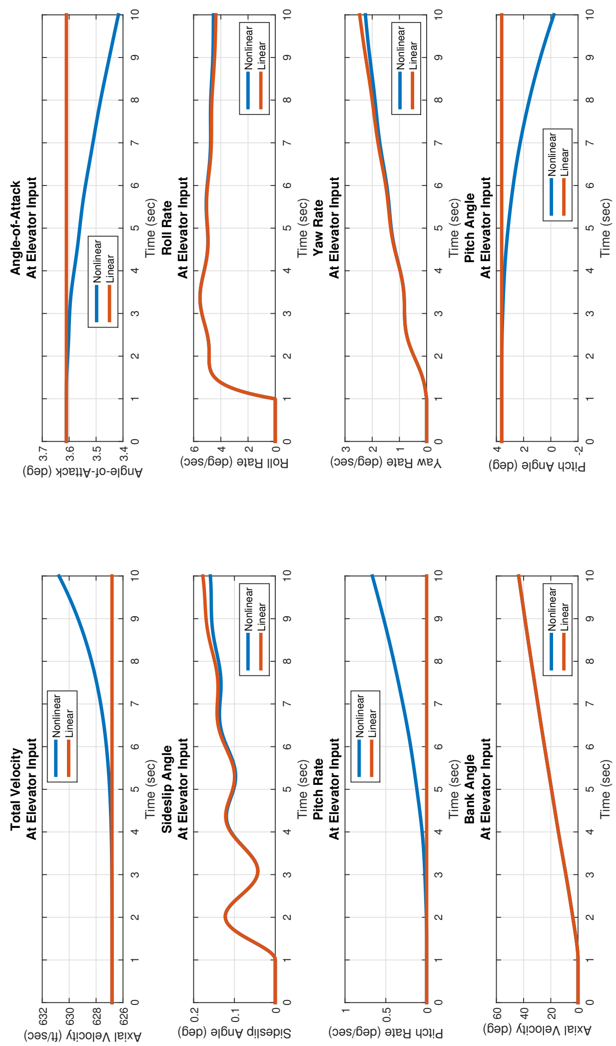

Figure 2: Aileron Input Simulation Graphical Solution

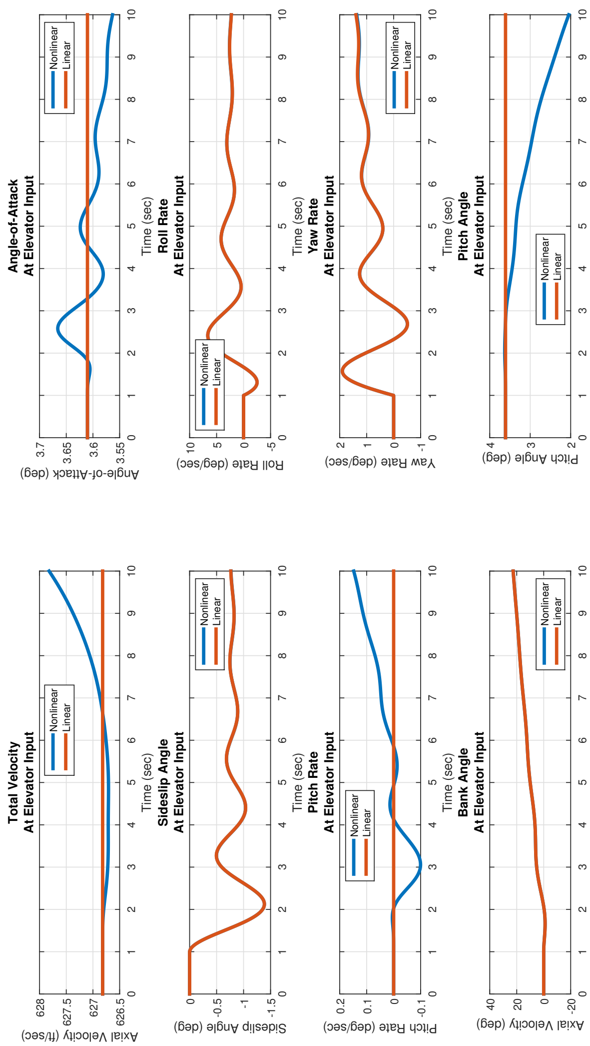

Figure 3: Rudder Input Simulation Graphical Solution

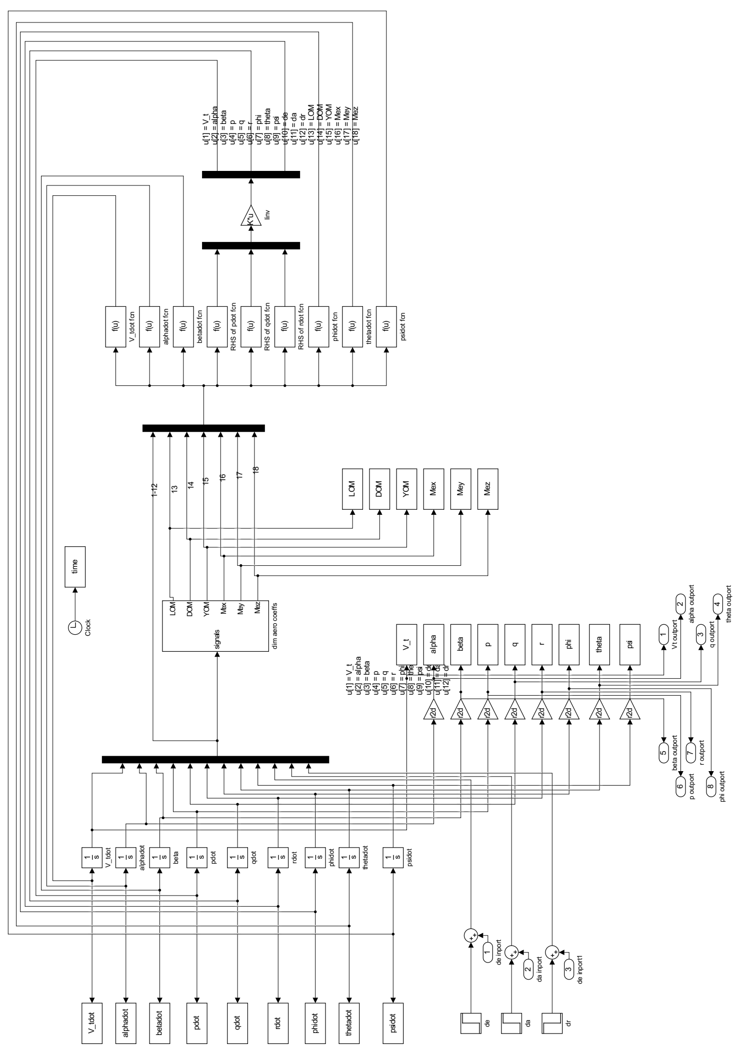

Figure 4: Simulink Diagram

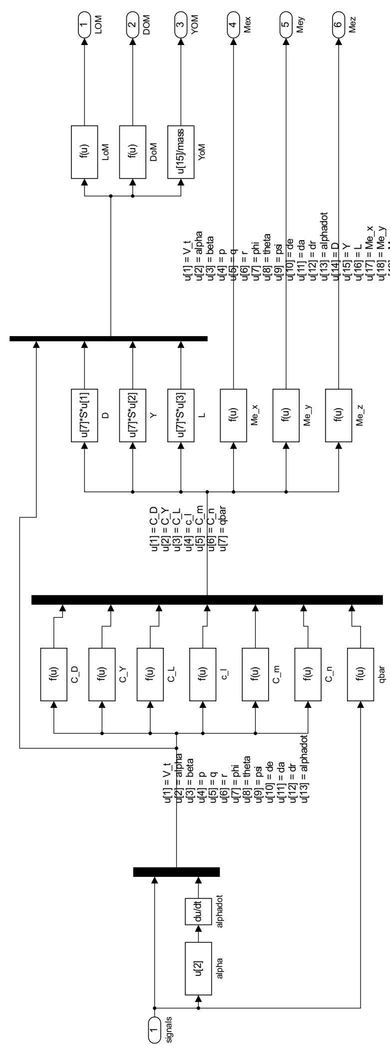

Figure 5: Simulink Diagram

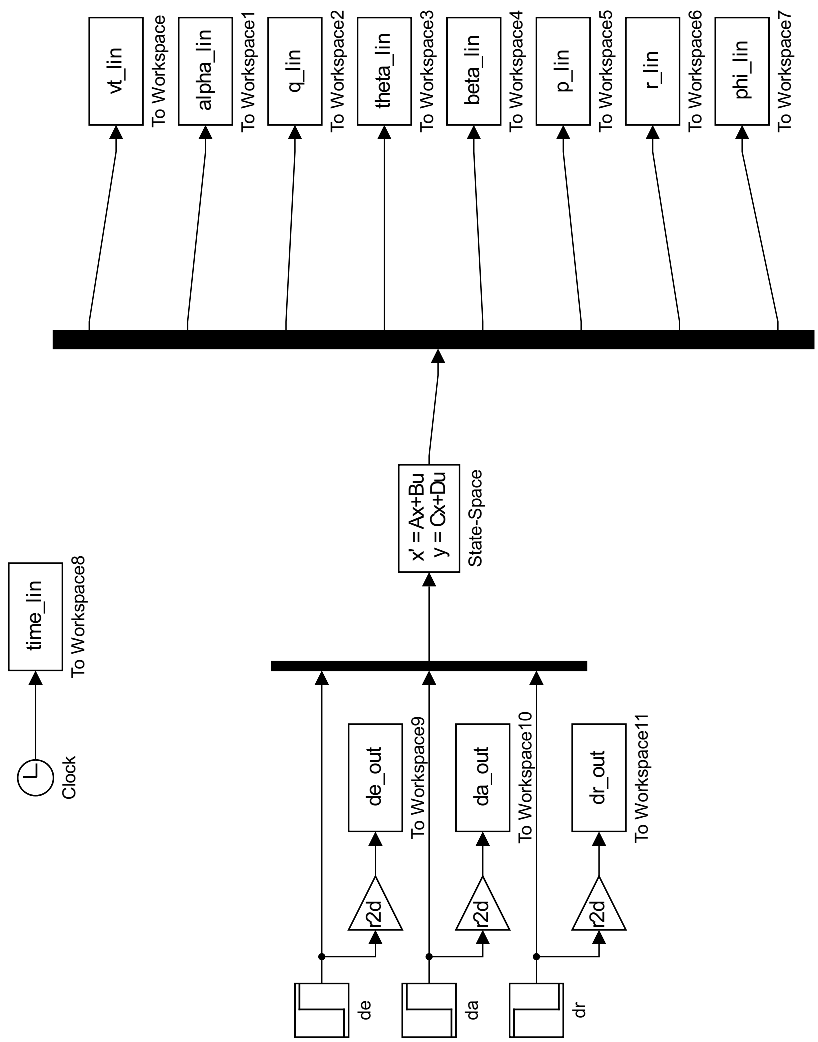

Figure 6: Simulink Diagram

```
% Project_5_m.m
% Clair Cunningham

clear all; close all; clc
fid = fopen('Project_5.txt','w+');

% Set and Obtain Screen Size
set(0,'Units','pixels')
Pix_SS = get(0,'screensize');
%Position Center At 90 percent screen in vector form :[left bottom width height]
width = Pix_SS(3)*.90;
height = Pix_SS(4)*.90;
taskbar = 40;
left = (Pix_SS(3)-width-Pix_SS(1))/2;
bottom = (Pix_SS(4)-height-Pix_SS(2))/2+taskbar;
pos_size = [left bottom width height-taskbar];

%% Numbers
% Conversion factor from radians to degrees
r2d = 180/pi;

%Conversion factor from degrees to radians
d2r = pi/180;
pd2pr = 180/pi;

% Aircraft Properties
i_xx = 8890.63; %slug-ft^2
i_yy = 71973.5; %slug-ft^2
i_zz = 77141.1; %slug-ft^2
i_xz = 181.119; %slug-ft^2
i_xy = 0.0; %slug-ft^2
i_yz = 0.0; %slug-ft^2
I = [i_xx -i_xy -i_xz; -i_xy i_yy -i_yz; -i_xz -i_yz i_zz];
Iinv = inv(I);
S = 300; % ft^2
cbar = 11.31; % ft
```

```
b = 30; % ft
mass = 762.8447; % slugs


% Properties of Air
rho = 0.0014962376; % slugs/ft^2
g = 32.17561865; % ft/sec^2
% qbar = 0.5*rho*V_t^2; % lbf/ft^2

%++++++++++++++++++++++++ Degree Format
    ++++++++++++++++++++++++
% Initial "Trim" Conditions
alpha_0_d = 3.6102915; % deg
beta_0_d = 0; % deg
p_0_d = 0; % deg
q_0_d = 0; % deg
r_0_d = 0; % deg
phi_0_d = 0; % deg
theta_0_d = 3.6102915; % deg
psi_0_d = 0; % deg
de_0_d = -3.03804303; % deg
da_0_d = 0; % deg
dr_0_d = 0; % deg
df_0_d = 1.5; % deg

% Coefficient of Lift
c_L_0 = 0.004608463; %      confirmed
c_L_alpha_d = 0.0794655; %1/deg
c_L_q_d = 0.0508476; %1/deg
c_L_alphadot_d = 0.0; %1/deg
c_L_de_d = 0.0121988; %1/deg
c_L_df_d = 0.0144389; %1/deg

% Coefficient of Drag
c_D_0 = 0.01192128;
c_D_alpha_d = 0.00550063; %1/deg
c_D_q_d = 0.00315057; %1/deg
c_D_alphadot_d = 0.0; %1/deg
```

```
c_D_de_d = -0.000587647; %1/deg
c_d_df_d = 0.00136385; %1/deg

% Coefficient of Side Force
c_y_0 = 0.0;
c_y_beta_d = -0.0219309; %1/deg
c_y_p_d = 0.00133787; %1/deg
c_y_r_d = 0.0094053; %1/deg
c_y_da_d = 0.00049355; %1/deg
c_y_dr_d = 0.00293048; %1/deg

% Coefficient of Rolling Moment
c_l_0 = 0.0;
c_l_beta_d = -0.00173748; %1/deg
c_l_p_d = -0.00739342; %1/deg
c_l_r_d = 0.0000699792; %1/deg
c_l_da_d = -0.00213984; %1/deg
c_l_dr_d = 0.000479021; %1/deg

% Coefficient of Pitching Moment
c_m_0 = -0.02092347;
c_m_alpha_d = -0.0041873; %1/deg
c_m_q_d = -0.110661; %1/deg
c_m_alphadot_d = 0.0; %1/deg
c_m_de_d = -0.0115767; %1/deg
c_m_df_d = 0.000580220; %1/deg

% Coefficient of Yawing Moment
c_n_0 = 0.0;
c_n_beta_d = 0.00320831; %1/deg
c_n_p_d = -0.000432575; %1/deg
c_n_r_d = -0.00886783; %1/deg
c_n_da_d = -0.000206591; %1/deg
c_n_dr_d = -0.00144865; %1/deg

%++++++++++++++++++++++++ Radian Format ++++++++++++++++++++++++
% Initial "Trim" Conditions
```

```
V_t_0 = 626.81863; % ft/sec
T_0 = 3146.482666; % lb

% Conversion factor from radians to degrees
% r2d = 180/pi;

%Conversion factor from degrees to radians
% d2r = pi/180;
% pd2pr = 180/pi;

alpha_0 = alpha_0_d*d2r;%
beta_0 = beta_0_d*d2r; %
p_0 = p_0_d*d2r; %
q_0 = q_0_d*d2r; %
r_0 = r_0_d*d2r; %
phi_0 = phi_0_d*d2r; %
theta_0 = theta_0_d*d2r; %
psi_0 = psi_0_d*d2r; %
de_0 = de_0_d*d2r; %
da_0 = da_0_d*d2r; %
dr_0 = dr_0_d*d2r; %
df_0 = df_0_d*d2r; %

% Coefficient of Life
c_L_0 = 0.004608463;
c_L_alpha = c_L_alpha_d*pd2pr; %1/rad
c_L_q = c_L_q_d*pd2pr; %1/rad
c_L_alphadot = c_L_alphadot_d*pd2pr; %1/rad
c_L_de = c_L_de_d*pd2pr; %1/rad
c_L_df = c_L_df_d*pd2pr; %1/rad

% Coefficient of Drag
c_D_0 = 0.01192128;
c_D_alpha = c_D_alpha_d*pd2pr; %1/rad
c_D_q = c_D_q_d*pd2pr; %1/rad
c_D_alphadot = c_D_alphadot_d*pd2pr; %1/rad
c_D_de = c_D_de_d*pd2pr; %1/rad
c_D_df = c_d_df_d*pd2pr; %1/rad
```

```
% Coefficient of Side Force
c_y_0 = 0.0;
c_y_beta = c_y_beta_d*pd2pr; %1/rad
c_y_p = c_y_p_d*pd2pr; %1/rad
c_y_r = c_y_r_d*pd2pr; %1/rad
c_y_da = c_y_da_d*pd2pr; %1/rad
c_y_dr = c_y_dr_d*pd2pr; %1/rad

% Coefficient of Rolling Moment
c_l_0 = 0.0;
c_l_beta = c_l_beta_d*pd2pr; %1/rad
c_l_p = c_l_p_d*pd2pr; %1/rad
c_l_r = c_l_r_d*pd2pr; %1/rad
c_l_da = c_l_da_d*pd2pr; %1/rad
c_l_dr = c_l_dr_d*pd2pr; %1/rad

% Coefficient of Pitching Moment
c_m_0 = -0.02092347;
c_m_alpha = c_m_alpha_d*pd2pr; %1/rad
c_m_q = c_m_q_d*pd2pr; %1/rad
c_m_alphadot = c_m_alphadot_d*pd2pr; %1/rad
c_m_de = c_m_de_d*pd2pr; %1/rad
c_m_df = c_m_df_d*pd2pr; %1/rad

% Coefficient of Yawing Moment
c_n_0 = 0.0;
c_n_beta = c_n_beta_d*pd2pr; %1/rad
c_n_p = c_n_p_d*pd2pr; %1/rad
c_n_r = c_n_r_d*pd2pr; %1/rad
c_n_da = c_n_da_d*pd2pr; %1/rad
c_n_dr = c_n_dr_d*pd2pr; %1/rad

% Quaternion Initial Conditions
e1_0 = cos(psi_0/2)*cos(theta_0/2)*cos(phi_0/2)+sin(
    psi_0/2)*sin(theta_0/2)*sin(phi_0/2);
e2_0 = cos(psi_0/2)*cos(theta_0/2)*sin(phi_0/2)-sin(
    psi_0/2)*sin(theta_0/2)*cos(phi_0/2);
```

```
e3_0 = cos(psi_0/2)*sin(theta_0/2)*cos(phi_0/2)+sin(
   psi_0/2)*cos(theta_0/2)*sin(phi_0/2);
e4_0 = sin(psi_0/2)*cos(theta_0/2)*cos(phi_0/2)-cos(
   psi_0/2)*sin(theta_0/2)*sin(phi_0/2);
%% Model Linearization
fprintf(fid,'\\sectionmark{Project \\# 5\\hspace*{\\
   fill} Clair Cunningham \\hspace*{\\fill} Problem
   1}\n');
%Inputs to Elevator, Aileron, or Rudder
    % Degs
    de_d = 0;
    da_d = 0;
    df_d = 0;
    dr_d = 0;
    % Radians
    de = de_d*d2r;
    da = da_d*d2r;
    df = df_d*d2r;
    dr = dr_d*d2r;

% Get Linear Model
[A,B,C,D]=linmod('Project_5_s_7_5');

%get eigenstructure
Alon = A(1:4,1:4);
Blon = B(1:4,1);
Clon = C(1:4,1:4);
Dlon = D(1:4,1);

fprintf(fid,'\\begin{equation}\\begin{bmatrix}\\dot{
   V_T}\\\\ \\dot{\\alpha} \\\\ \\dot{q} \\\\ \\dot
   {\\theta} \\end{bmatrix}=\n\\begin{bmatrix}%.3g &
    %.3g & %.3g & %.3g \\\\ %.3g & %.3g & %.3g & %.3
   g \\\\ %.3g & %.3g & %.3g & %.3g \\\\ %.3g & %.3g
    & %.3g & %.3g \\end{bmatrix}\n\\begin{bmatrix}
   V_T\\\\ \\alpha \\\\ q \\\\ \\theta \\end{bmatrix
   }+\n\\begin{bmatrix}%.3g \\\\ %.3g \\\\ %.3g \\\\
    %.3g\\end{bmatrix}\\begin{bmatrix} \\delta e\\
```

```
end{bmatrix}\\end{equation}\\\\',Alon',Blon)

[Wn_lon,Z_lon,P_lon] = damp(Alon);

fprintf(fid,'The short period eigenvalues are %.3g and %.3g, natural frequencies %.3g and %.3g, damping ratios %.3g and %.3g. \\\\ \n',P_lon(3:4),Wn_lon(3:4),Z_lon(3:4))
fprintf(fid,'The phugoid eigenvalues are %.3g and %.3g, natural frequencies %.3g and %.3g, damping ratios %.3g and %.3g. \n',P_lon(1:2),Wn_lon(1:2),Z_lon(1:2))
fprintf(fid,'The system is stable because the eigenvalues are all negative.\\\\ \n')

[nume,den] = ss2tf(Alon,Blon,Clon,Dlon);

fprintf(fid,'The pitch rate transfer function is: \\begin{equation}\\frac{q(s)}{\\delta e(s)} = \\frac{%.3g s^4 + %.3g s^3 + %.3g s^2 + %.3g s + %.3g}{%.3g s^4 + %.3g s^3 + %.3g s^2 + %.3g s + %.3g}\\end{equation}',nume(3,:),den)
fprintf(fid,'The angle of attack transfer function is: \\begin{equation}\\frac{\\alpha(s)}{\\delta e(s)} = \\frac{%.3g s^4 + %.3g s^3 + %.3g s^2 + %.3g s + %.3g}{%.3g s^4 + %.3g s^3 + %.3g s^2 + %.3g s + %.3g}\\end{equation}',nume(2,:),den)

%% Elevator Input Simulation
num = 1;
%Inputs to Elevator, Aileron, or Rudder
    % Degs
    de_d = -0.5;
    da_d = 0;
    df_d = 0;
    dr_d = 0;
    % Radians
    de = de_d*d2r;
```

```
    da = da_d*d2r;
    df = df_d*d2r;
    dr = dr_d*d2r;

%simulate nonlinear system
sim('Project_5_s_7_5')

%define ICs for linear system
xo=[0 0 0 0 0 0 0 0 0];

%simulate linear system
sim('Project_5_lin_s_7_5')

%Call up next figure
fig = figure('OuterPosition',pos_size,'PaperPositionMode','auto');
fig.Name = 'Elevator Input Simulation';
%Changes Paper Print Orientation to landscape on a per figure basic
orient landscape
%Output plots to called figure
subplot(4,2,1)
plot(time,V_t,time,vt_lin+V_t_0,'LineWidth',3.0); title({'Total Velocity','At Elevator Input'}); xlabel('Time (sec)');ylabel('Axial Velocity (ft/sec)'); grid on; legend('Nonlinear','Linear','Location','Best')
subplot(4,2,2)
plot(time,alpha,time,alpha_lin+alpha_0_d,'LineWidth',3.0); title({'Angle-of-Attack','At Elevator Input'});xlabel('Time (sec)');ylabel('Angle-of-Attack (deg)'); grid on; legend('Nonlinear','Linear','Location','Best')
subplot(4,2,3)
plot(time,beta,time,beta_lin+beta_0_d,'LineWidth',3.0); title({'Sideslip Angle','At Elevator Input'});xlabel('Time (sec)');ylabel('Sideslip Angle (deg)'); grid on; legend('Nonlinear','Linear','
```

```
Location','Best')
subplot(4,2,4)
plot(time,p,time,p_lin+p_0,'LineWidth',3.0); title({'Roll Rate','At Elevator Input'});xlabel('Time (sec)');ylabel('Roll Rate (deg/sec)'); grid on; legend('Nonlinear','Linear','Location','Best')
subplot(4,2,5)
plot(time,q,time,q_lin+q_0,'LineWidth',3.0); title({'Pitch Rate','At Elevator Input'});xlabel('Time (sec)');ylabel('Pitch Rate (deg/sec)'); grid on; legend('Nonlinear','Linear','Location','Best')
subplot(4,2,6)
plot(time,r,time,r_lin+r_0,'LineWidth',3.0); title({'Yaw Rate','At Elevator Input'});xlabel('Time (sec)');ylabel('Yaw Rate (deg/sec)'); grid on; legend('Nonlinear','Linear','Location','Best')
subplot(4,2,7)
plot(time,phi,time,phi_lin+phi_0_d,'LineWidth',3.0); title({'Bank Angle','At Elevator Input'});xlabel('Time (sec)');ylabel('Axial Velocity (deg)'); grid on; legend('Nonlinear','Linear','Location','Best')
subplot(4,2,8)
plot(time,theta,time,theta_lin+theta_0_d,'LineWidth',3.0); title({'Pitch Angle','At Elevator Input'});xlabel('Time (sec)');ylabel('Pitch Angle (deg)'); grid on; legend('Nonlinear','Linear','Location','Best')
% %'Paperposition', [left bottom width height]
%  set(fig,'PaperPositionMode', 'manual', 'PaperUnits','Inches', 'Paperposition',[0.0 0.0 11 8.5])
%  print('-P\\meprint2\gle-2120-pr02c',fig)
name = [strrep(fig.Name,' ','_')];
print(figure(num),'-depsc','-noui','-painters',name);
fprintf(fid,'\\sectionmark{Project \\# 5\\hspace*{\\fill} Clair Cunningham \\hspace*{\\fill} Problem
```

```
%d}\n',num);
fprintf(fid,'\n\\vspace*{\\fill}\\begin{figure}[H]\\centering\\includegraphics[keepaspectratio=true,height=1\\textheight,width=1\\textwidth,angle=90]{%s.eps}\n \\caption{%s Graphical Solution}\\end{figure}\\vspace*{\\fill}\n\\newpage\n',name,fig.Name);
%% Problem 2
fprintf(fid,'\\sectionmark{Project \\# 5\\hspace*{\\fill} Clair Cunningham \\hspace*{\\fill} Problem 2}\n');
Alat = A(5:8,5:8);
Blat = B(5:8,2:3);
Clat = C(5:8,5:8);
Dlat = D(5:8,2:3);

[Wn_lat,Z_lat,P_lat] = damp(Alat);
fprintf(fid,'\\begin{equation}\\begin{bmatrix}\\dot{\\beta}\\\\ \\dot{p} \\\\ \\dot{r} \\\\ \\dot{\\phi} \\end{bmatrix}=\n\\begin{bmatrix}%.3g & %.3g & %.3g & %.3g \\\\ %.3g & %.3g & %.3g & %.3g \\\\ %.3g & %.3g & %.3g & %.3g \\\\ %.3g & %.3g & %.3g & %.3g \\end{bmatrix}\n\\begin{bmatrix}V_T \\\\ \\alpha \\\\ q \\\\ \\theta \\end{bmatrix}+\n\\begin{bmatrix}%.3g & %.3g \\\\ %.3g & %.3g \\\\ %.3g & %.3g \\\\ %.3g & %.3g\\end{bmatrix}\\begin{bmatrix} \\delta a \\\\ \\delta r \\end{bmatrix}\\end{equation}',Alat',Blat')

fprintf(fid,'The dutch roll mode eigenvalues are %.3g and %.3g, natural frequencies %.3g and %.3g, damping ratios %.3g and %.3g.\n',P_lat(1:2),Wn_lat(1:2),Z_lat(1:2))
t_roll = -1/P_lat(3);
t_spiral = -1/P_lat(4);
fprintf(fid,'The roll mode eigenvalue is %.3g, and time constant %.3g. \n',P_lat(3),t_roll)
fprintf(fid,'The spiral mode eigenvalue is %.3g, and
```

```
    time constant %.3g. \n',P_lat(4),t_spiral)
fprintf(fid,'The system is stable because the
    eigenvalues are all negative. \\\\ \n')

%% Aileron Input Simulation
num = num + 1;
%Inputs to Elevator, Aileron, or Rudder
    % Degs
    de_d = 0;
    da_d = -0.5;
    df_d = 0;
    dr_d = 0; % deg
    % Radians
    de = de_d*d2r;
    da = da_d*d2r;
    df = df_d*d2r;
    dr = dr_d*d2r;

%simulate nonlinear system
sim('Project_5_s_7_5')

%define ICs for linear system
xo=[0 0 0 0 0 0 0 0 0];

%simulate linear system
sim('Project_5_lin_s_7_5')

%Call up next figure
fig = figure('OuterPosition',pos_size,'
    PaperPositionMode','auto');
fig.Name = 'Aileron Input Simulation';
%Changes Paper Print Orientation to landscape on a
    per figure basic
orient landscape
%Output plots to called figure
subplot(4,2,1)
plot(time,V_t,time,vt_lin+V_t_0,'LineWidth',3.0);
    title({'Total Velocity','At Elevator Input'});
```

```
    xlabel('Time (sec)');ylabel('Axial Velocity (ft/
    sec)'); grid on; legend('Nonlinear','Linear','
    Location','Best')
subplot(4,2,2)
plot(time,alpha,time,alpha_lin+alpha_0_d,'LineWidth'
    ,3.0); title({'Angle-of-Attack','At Elevator
    Input'});xlabel('Time (sec)');ylabel('Angle-of-
    Attack (deg)'); grid on; legend('Nonlinear','
    Linear','Location','Best')
subplot(4,2,3)
plot(time,beta,time,beta_lin+beta_0_d,'LineWidth'
    ,3.0); title({'Sideslip Angle','At Elevator Input
    '});xlabel('Time (sec)');ylabel('Sideslip Angle (
    deg)'); grid on; legend('Nonlinear','Linear','
    Location','Best')
subplot(4,2,4)
plot(time,p,time,p_lin+p_0,'LineWidth',3.0); title({
    'Roll Rate','At Elevator Input'});xlabel('Time (
    sec)');ylabel('Roll Rate (deg/sec)'); grid on;
    legend('Nonlinear','Linear','Location','Best')
subplot(4,2,5)
plot(time,q,time,q_lin+q_0,'LineWidth',3.0); title({
    'Pitch Rate','At Elevator Input'});xlabel('Time (
    sec)');ylabel('Pitch Rate (deg/sec)'); grid on;
    legend('Nonlinear','Linear','Location','Best')
subplot(4,2,6)
plot(time,r,time,r_lin+r_0,'LineWidth',3.0); title({
    'Yaw Rate','At Elevator Input'});xlabel('Time (
    sec)');ylabel('Yaw Rate (deg/sec)'); grid on;
    legend('Nonlinear','Linear','Location','Best')
subplot(4,2,7)
plot(time,phi,time,phi_lin+phi_0_d,'LineWidth',3.0);
     title({'Bank Angle','At Elevator Input'});xlabel
    ('Time (sec)');ylabel('Axial Velocity (deg)');
    grid on; legend('Nonlinear','Linear','Location','
    Best')
subplot(4,2,8)
plot(time,theta,time,theta_lin+theta_0_d,'LineWidth'
```

```
,3.0); title({'Pitch Angle','At Elevator Input'});xlabel('Time (sec)');ylabel('Pitch Angle (deg)'); grid on; legend('Nonlinear','Linear','Location','Best')
% %'Paperposition', [left bottom width height]
%  set(fig,'PaperPositionMode', 'manual', 'PaperUnits','Inches', 'Paperposition',[0.0 0.0 11 8.5])
%  print('-P\\meprint2\gle-2120-pr02c',fig)
name = [strrep(fig.Name,' ','_')];
print(figure(num),'-depsc','-noui','-painters',name);
fprintf(fid,'\\sectionmark{Project \\# 5\\hspace*{\\fill} Clair Cunningham \\hspace*{\\fill} Problem %d}\n',num);
fprintf(fid,'\n\\vspace*{\\fill}\\begin{figure}[H]\\centering\\includegraphics[keepaspectratio=true,height=0.99\\textheight,width=1\\textwidth,angle=90]{%s.eps}\n \\caption{%s Graphical Solution}\\end{figure}\\vspace*{\\fill}\n',name,fig.Name);
%% Rudder Input Simulation
num = num +1;
%Inputs to Elevator, Aileron, or Rudder
    % Degs
    de_d = 0;
    da_d = 0;
    df_d = 0;
    dr_d = -2.0; % deg
    % Radians
    de = de_d*d2r;
    da = da_d*d2r;
    df = df_d*d2r;
    dr = dr_d*d2r;

%simulate nonlinear system
sim('Project_5_s_7_5')

%define ICs for linear system
```

```
xo=[0 0 0 0 0 0 0 0 0];

%simulate linear system
sim('Project_5_lin_s_7_5')

%Call up next figure
fig = figure('OuterPosition',pos_size,'PaperPositionMode','auto');
fig.Name = 'Rudder Input Simulation';
%Changes Paper Print Orientation to landscape on a per figure basic
orient landscape
%Output plots to called figure
subplot(4,2,1)
plot(time,V_t,time,vt_lin+V_t_0,'LineWidth',3.0); title({'Total Velocity','At Elevator Input'}); xlabel('Time (sec)');ylabel('Axial Velocity (ft/sec)'); grid on; legend('Nonlinear','Linear','Location','Best')
subplot(4,2,2)
plot(time,alpha,time,alpha_lin+alpha_0_d,'LineWidth',3.0); title({'Angle-of-Attack','At Elevator Input'});xlabel('Time (sec)');ylabel('Angle-of-Attack (deg)'); grid on; legend('Nonlinear','Linear','Location','Best')
subplot(4,2,3)
plot(time,beta,time,beta_lin+beta_0_d,'LineWidth',3.0); title({'Sideslip Angle','At Elevator Input'});xlabel('Time (sec)');ylabel('Sideslip Angle (deg)'); grid on; legend('Nonlinear','Linear','Location','Best')
subplot(4,2,4)
plot(time,p,time,p_lin+p_0,'LineWidth',3.0); title({'Roll Rate','At Elevator Input'});xlabel('Time (sec)');ylabel('Roll Rate (deg/sec)'); grid on; legend('Nonlinear','Linear','Location','Best')
subplot(4,2,5)
plot(time,q,time,q_lin+q_0,'LineWidth',3.0); title({
```

```
'Pitch Rate','At Elevator Input'});xlabel('Time (sec)');ylabel('Pitch Rate (deg/sec)'); grid on; legend('Nonlinear','Linear','Location','Best')
subplot(4,2,6)
plot(time,r,time,r_lin+r_0,'LineWidth',3.0); title({'Yaw Rate','At Elevator Input'});xlabel('Time (sec)');ylabel('Yaw Rate (deg/sec)'); grid on; legend('Nonlinear','Linear','Location','Best')
subplot(4,2,7)
plot(time,phi,time,phi_lin+phi_0_d,'LineWidth',3.0); title({'Bank Angle','At Elevator Input'});xlabel('Time (sec)');ylabel('Axial Velocity (deg)'); grid on; legend('Nonlinear','Linear','Location','Best')
subplot(4,2,8)
plot(time,theta,time,theta_lin+theta_0_d,'LineWidth',3.0); title({'Pitch Angle','At Elevator Input'});xlabel('Time (sec)');ylabel('Pitch Angle (deg)'); grid on; legend('Nonlinear','Linear','Location','Best')
% %'Paperposition', [left bottom width height]
%  set(fig,'PaperPositionMode', 'manual', 'PaperUnits','Inches', 'Paperposition',[0.0 0.0 11 8.5])
%  print('-P\\meprint2\gle-2120-pr02c',fig)
name = [strrep(fig.Name,' ','_')];
print(figure(num),'-depsc','-noui','-painters',name);
fprintf(fid,'\\sectionmark{Project \\# 5\\hspace*{\\fill} Clair Cunningham \\hspace*{\\fill} Problem %d}\n',num);
fprintf(fid,'\\vspace*{\\fill}\\begin{figure}[H]\\centering\\includegraphics[keepaspectratio=true,height=0.99\\textheight,width=1\\textwidth,angle=90]{%s.eps}\n \\caption{%s Graphical Solution}\\end{figure}\\vspace*{\\fill}\n',name,fig.Name);
%% End of File Commands
name = 'Project_5_s_7_5';
```

```
load_system(name);
%modelhandle = get_param('name', 'Handle')
handles = find_system(name, 'FindAll', 'On', 'SearchDepth', 10, ...
    'regexp', 'on', 'blocktype', 'port');
list = get(handles,'Path');
if ~iscell(list)
    list = {list};
end
list = unique(list);
if ~iscell(handles)
    handle = {handles};
end
% % add main model
% list{end+1} = name;
% linear model
list{end+1} = 'Project_5_lin_s_7_5';
% GEt only last part of path, that is, after the last /
[r1, r2] = regexpi(list, '[^/]+$', 'tokens', 'match');

% Convert to usable format
names = [r2{:}]';

% Cells of printNames.
% Just rename every non-alphanumeric char to _, all space to ''
printNames = regexprep(names', {'\s', '\W'}, {'', '_'});

for i = 1 : length(list)
    item = char(list(i));
modelhandle = get_param(item, 'Handle');
set(modelhandle,'PaperPositionMode', 'manual', 'PaperUnits','Inches', 'Paperposition',[0.0 0.0 11 8.5])
print(['-s' item],'-depsc','-noui','-painters',
```

```
printNames{i});
fprintf(fid,'\\sectionmark{Project \\# 5\\hspace*{\\fill} Clair Cunningham \\hspace*{\\fill} %s}\n',strrep(printNames{i},'_','\_'));
fprintf(fid,'\\vspace*{\\fill}\\begin{figure}[H]\\centering\\includegraphics[keepaspectratio=true,height=0.99\\textheight,width=1\\textwidth,angle=90]{%s.eps}\n \\caption{Simulink Diagram}\\end{figure}\\vspace*{\\fill}\n',printNames{i});
end
fclose(fid);
```