

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

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:

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

The angle of attack transfer function is:

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

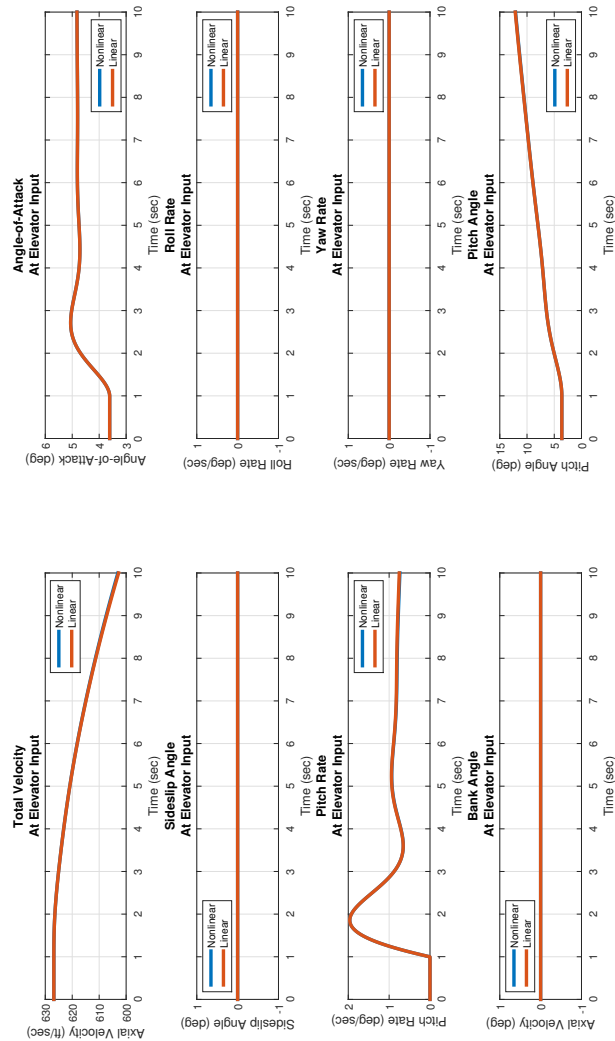


Figure 1: Elevator Input Simulation Graphical Solution

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

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.

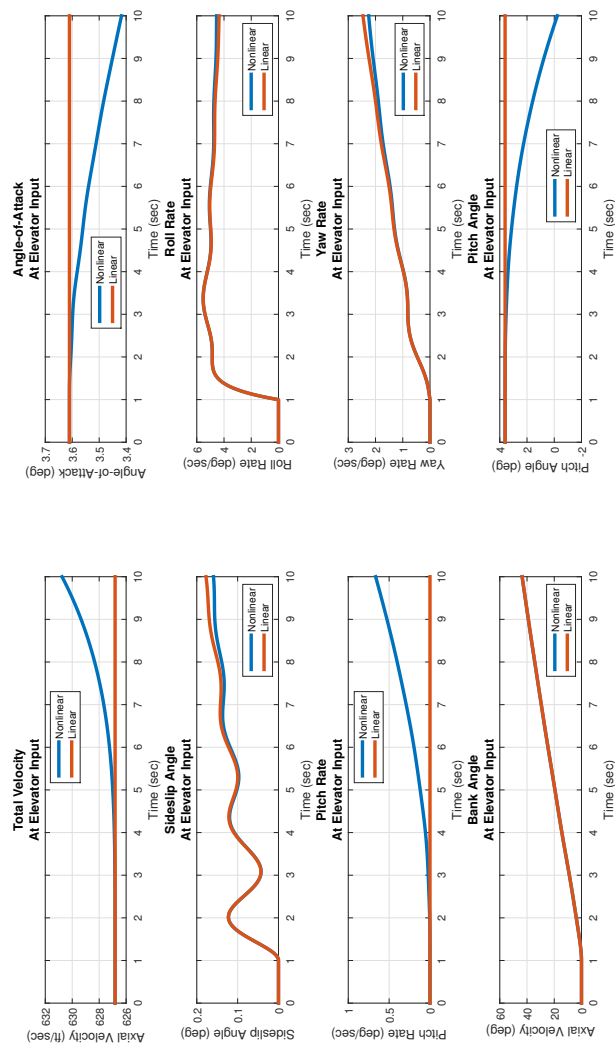


Figure 2: Aileron Input Simulation Graphical Solution

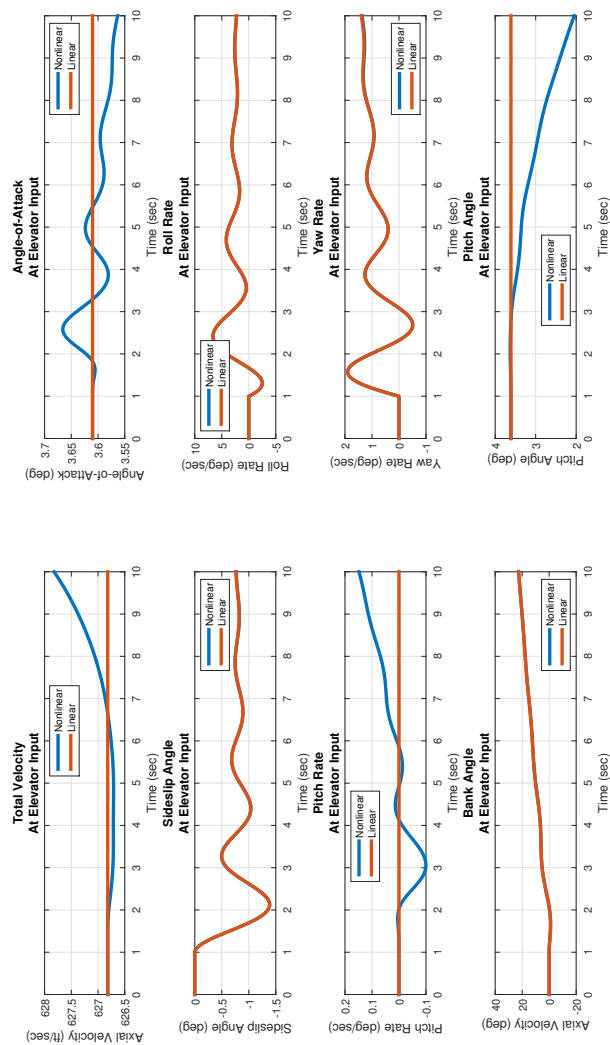
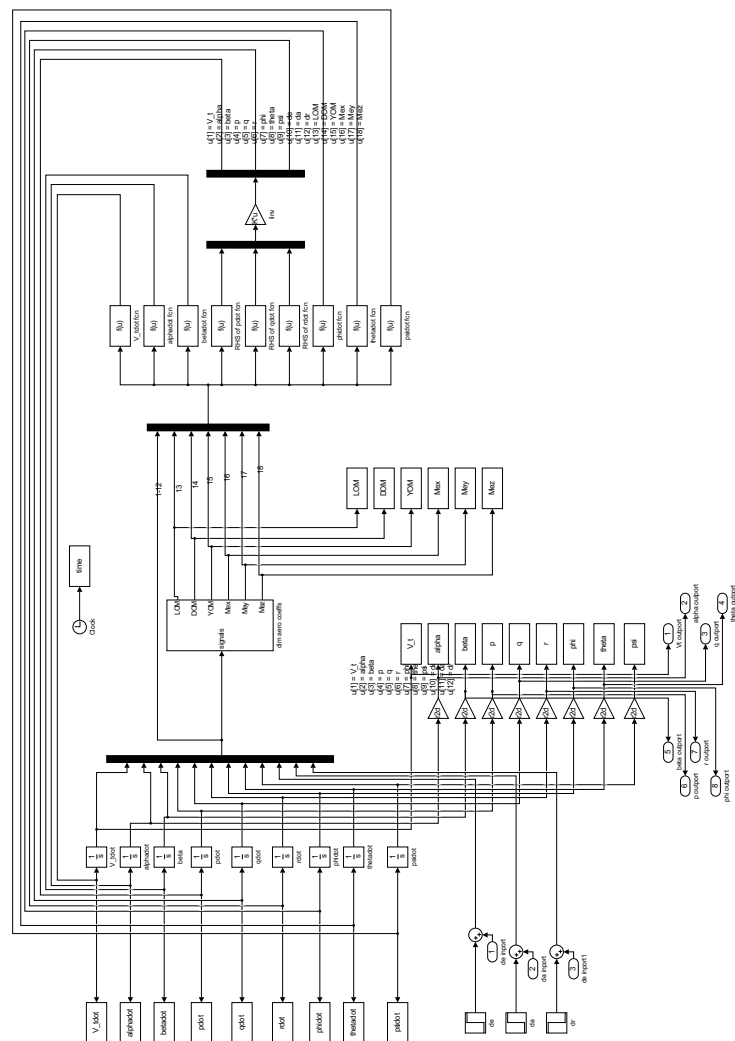


Figure 3: Rudder Input Simulation Graphical Solution



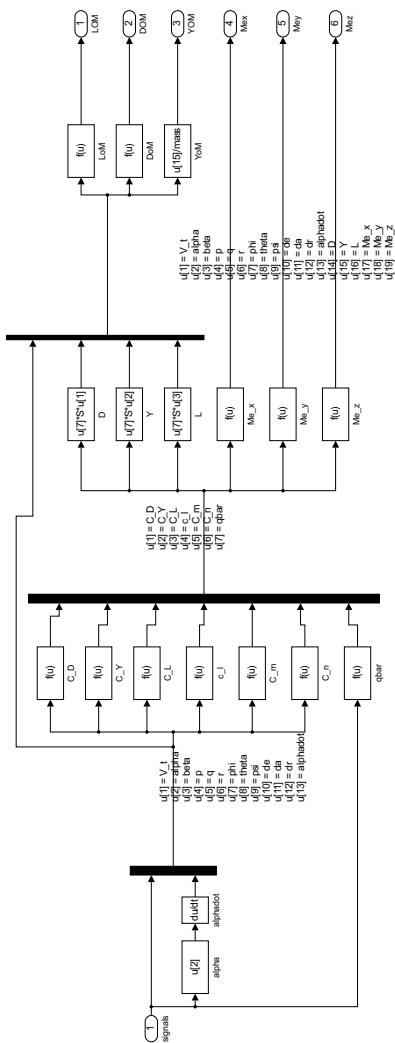


Figure 5: Simulink Diagram

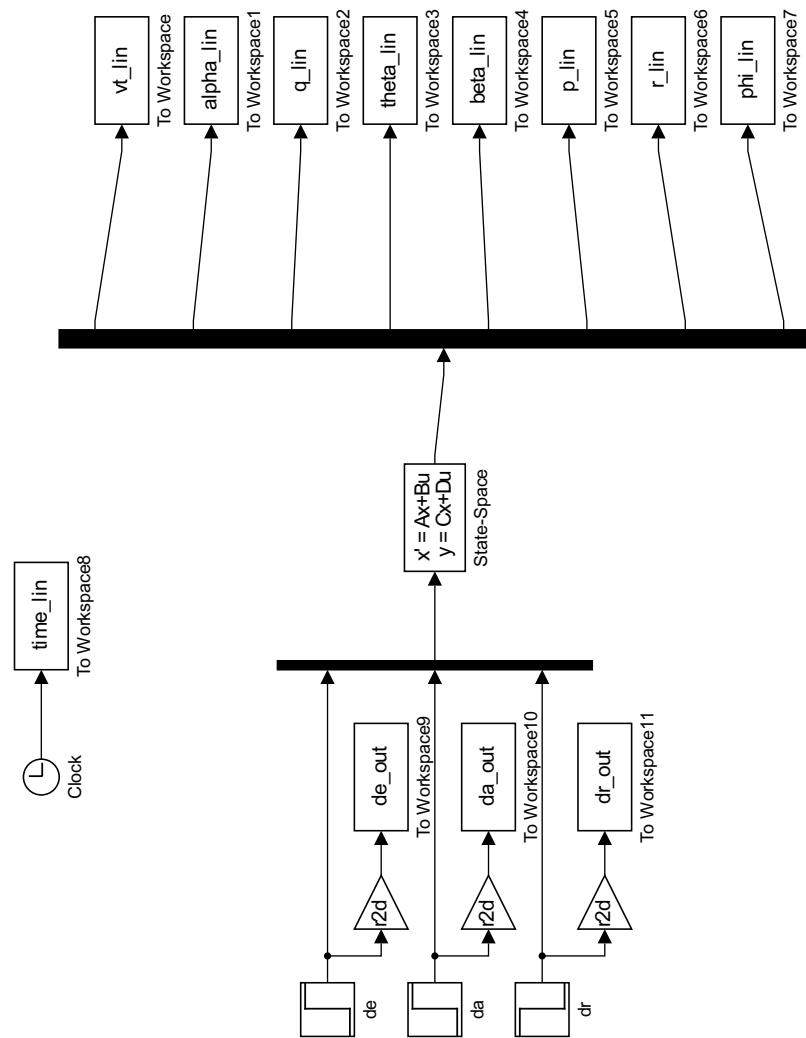


Figure 6: Simulink Diagram



```
1 % Project_5_m.m
2 % Clair Cunningham
3
4 clear all; close all; clc
5 fid = fopen('Project_5.txt','w+');
6
7 % Set and Obtain Screen Size
8 set(0,'Units','pixels')
9 Pix_SS = get(0,'screensize');
10 %Position Center At 90 percent screen in vector form
    :[left bottom width height]
11 width = Pix_SS(3)*.90;
12 height = Pix_SS(4)*.90;
13 taskbar = 40;
14 left = (Pix_SS(3)-width-Pix_SS(1))/2;
15 bottom = (Pix_SS(4)-height-Pix_SS(2))/2+taskbar;
16 pos_size = [left bottom width height-taskbar];
17
18 %% Numbers
19 % Conversion factor from radians to degrees
20 r2d = 180/pi;
21
22 %Conversion factor from degrees to radians
23 d2r = pi/180;
24 pd2pr = 180/pi;
25
26 % Aircraft Properties
27 i_xx = 8890.63; %slug-ft^2
28 i_yy = 71973.5; %slug-ft^2
29 i_zz = 77141.1; %slug-ft^2
30 i_xz = 181.119; %slug-ft^2
31 i_xy = 0.0; %slug-ft^2
32 i_yz = 0.0; %slug-ft^2
33 I = [i_xx -i_xy -i_xz; -i_xy i_yy -i_yz; -i_xz -i_yz
    i_zz];
34 Iinv = inv(I);
35 S = 300; % ft^2
36 cbar = 11.31; % ft
```

```
37 b = 30; % ft
38 mass = 762.8447; % slugs
39
40
41 % Properties of Air
42 rho = 0.0014962376; % slugs/ft^2
43 g = 32.17561865; % ft/sec^2
44 % qbar = 0.5*rho*V_t^2; % lbf/ft^2
45
46 %+++++ Degree Format
    ++++++
47 % Initial "Trim" Conditions
48 alpha_0_d = 3.6102915; % deg
49 beta_0_d = 0; % deg
50 p_0_d = 0; % deg
51 q_0_d = 0; % deg
52 r_0_d = 0; % deg
53 phi_0_d = 0; % deg
54 theta_0_d = 3.6102915; % deg
55 psi_0_d = 0; % deg
56 de_0_d = -3.03804303; % deg
57 da_0_d = 0; % deg
58 dr_0_d = 0; % deg
59 df_0_d = 1.5; % deg
60
61 % Coefficient of Lift
62 c_L_0 = 0.004608463; % confirmed
63 c_L_alpha_d = 0.0794655; %1/deg
64 c_L_q_d = 0.0508476; %1/deg
65 c_L_alphadot_d = 0.0; %1/deg
66 c_L_de_d = 0.0121988; %1/deg
67 c_L_df_d = 0.0144389; %1/deg
68
69 % Coefficient of Drag
70 c_D_0 = 0.01192128;
71 c_D_alpha_d = 0.00550063; %1/deg
72 c_D_q_d = 0.00315057; %1/deg
73 c_D_alphadot_d = 0.0; %1/deg
```

```
74 c_D_de_d = -0.000587647; %1/deg
75 c_d_df_d = 0.00136385; %1/deg
76
77 % Coefficient of Side Force
78 c_y_0 = 0.0;
79 c_y_beta_d = -0.0219309; %1/deg
80 c_y_p_d = 0.00133787; %1/deg
81 c_y_r_d = 0.0094053; %1/deg
82 c_y_da_d = 0.00049355; %1/deg
83 c_y_dr_d = 0.00293048; %1/deg
84
85 % Coefficient of Rolling Moment
86 c_l_0 = 0.0;
87 c_l_beta_d = -0.00173748; %1/deg
88 c_l_p_d = -0.00739342; %1/deg
89 c_l_r_d = 0.0000699792; %1/deg
90 c_l_da_d = -0.00213984; %1/deg
91 c_l_dr_d = 0.000479021; %1/deg
92
93 % Coefficient of Pitching Moment
94 c_m_0 = -0.02092347;
95 c_m_alpha_d = -0.0041873; %1/deg
96 c_m_q_d = -0.110661; %1/deg
97 c_m_alphadot_d = 0.0; %1/deg
98 c_m_de_d = -0.0115767; %1/deg
99 c_m_df_d = 0.000580220; %1/deg
100
101 % Coefficient of Yawing Moment
102 c_n_0 = 0.0;
103 c_n_beta_d = 0.00320831; %1/deg
104 c_n_p_d = -0.000432575; %1/deg
105 c_n_r_d = -0.00886783; %1/deg
106 c_n_da_d = -0.000206591; %1/deg
107 c_n_dr_d = -0.00144865; %1/deg
108
109 %+++++ Radian Format
    ++++++
110 % Initial "Trim" Conditions
```

```
111 V_t_0 = 626.81863; % ft/sec
112 T_0 = 3146.482666; % lb
113
114 % Conversion factor from radians to degrees
115 % r2d = 180/pi;
116
117 %Conversion factor from degrees to radians
118 % d2r = pi/180;
119 % pd2pr = 180/pi;
120
121 alpha_0 = alpha_0_d*d2r;%
122 beta_0 = beta_0_d*d2r; %
123 p_0 = p_0_d*d2r; %
124 q_0 = q_0_d*d2r; %
125 r_0 = r_0_d*d2r; %
126 phi_0 = phi_0_d*d2r; %
127 theta_0 = theta_0_d*d2r; %
128 psi_0 = psi_0_d*d2r; %
129 de_0 = de_0_d*d2r; %
130 da_0 = da_0_d*d2r; %
131 dr_0 = dr_0_d*d2r; %
132 df_0 = df_0_d*d2r; %
133
134 % Coefficient of Life
135 c_L_0 = 0.004608463;
136 c_L_alpha = c_L_alpha_d*pd2pr; %1/rad
137 c_L_q = c_L_q_d*pd2pr; %1/rad
138 c_L_alphadot = c_L_alphadot_d*pd2pr; %1/rad
139 c_L_de = c_L_de_d*pd2pr; %1/rad
140 c_L_df = c_L_df_d*pd2pr; %1/rad
141
142 % Coefficient of Drag
143 c_D_0 = 0.01192128;
144 c_D_alpha = c_D_alpha_d*pd2pr; %1/rad
145 c_D_q = c_D_q_d*pd2pr; %1/rad
146 c_D_alphadot = c_D_alphadot_d*pd2pr; %1/rad
147 c_D_de = c_D_de_d*pd2pr; %1/rad
148 c_D_df = c_D_df_d*pd2pr; %1/rad
```

```
149
150 % Coefficient of Side Force
151 c_y_0 = 0.0;
152 c_y_beta = c_y_beta_d*pd2pr; %1/rad
153 c_y_p = c_y_p_d*pd2pr; %1/rad
154 c_y_r = c_y_r_d*pd2pr; %1/rad
155 c_y_da = c_y_da_d*pd2pr; %1/rad
156 c_y_dr = c_y_dr_d*pd2pr; %1/rad
157
158 % Coefficient of Rolling Moment
159 c_l_0 = 0.0;
160 c_l_beta = c_l_beta_d*pd2pr; %1/rad
161 c_l_p = c_l_p_d*pd2pr; %1/rad
162 c_l_r = c_l_r_d*pd2pr; %1/rad
163 c_l_da = c_l_da_d*pd2pr; %1/rad
164 c_l_dr = c_l_dr_d*pd2pr; %1/rad
165
166 % Coefficient of Pitching Moment
167 c_m_0 = -0.02092347;
168 c_m_alpha = c_m_alpha_d*pd2pr; %1/rad
169 c_m_q = c_m_q_d*pd2pr; %1/rad
170 c_m_alphadot = c_m_alphadot_d*pd2pr; %1/rad
171 c_m_de = c_m_de_d*pd2pr; %1/rad
172 c_m_df = c_m_df_d*pd2pr; %1/rad
173
174 % Coefficient of Yawing Moment
175 c_n_0 = 0.0;
176 c_n_beta = c_n_beta_d*pd2pr; %1/rad
177 c_n_p = c_n_p_d*pd2pr; %1/rad
178 c_n_r = c_n_r_d*pd2pr; %1/rad
179 c_n_da = c_n_da_d*pd2pr; %1/rad
180 c_n_dr = c_n_dr_d*pd2pr; %1/rad
181
182 % Quaternion Initial Conditions
183 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);
184 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);
```

Page 14

```

    end{bmatrix}\\end{equation}\\\\',Alon',Blon)
211
212 [Wn_lon,Z_lon,P_lon] = damp(Alon);
213
214 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))
215 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))
216 fprintf(fid,'The system is stable because the
    eigenvalues are all negative.\\\\ \\n')
217
218 [nume,den] = ss2tf(Alon,Blon,Clon,Dlon);
219
220 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)
221 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)
222
223 %% Elevator Input Simulation
224 num = 1;
225 %Inputs to Elevator, Aileron, or Rudder
226 % Degr
227 de_d = -0.5;
228 da_d = 0;
229 df_d = 0;
230 dr_d = 0;
231 % Radians
232 de = de_d*d2r;

```

```
233     da = da_d*d2r;
234     df = df_d*d2r;
235     dr = dr_d*d2r;
236
237 %simulate nonlinear system
238 sim('Project_5_s_7_5')
239
240 %define ICs for linear system
241 xo=[0 0 0 0 0 0 0 0 0];
242
243 %simulate linear system
244 sim('Project_5_lin_s_7_5')
245
246 %Call up next figure
247 fig = figure('OuterPosition',pos_size,'
    PaperPositionMode','auto');
248 fig.Name = 'Elevator Input Simulation';
249 %Changes Paper Print Orientation to landscape on a
    per figure basis
250 orient landscape
251 %Output plots to called figure
252 subplot(4,2,1)
253 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')
254 subplot(4,2,2)
255 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')
256 subplot(4,2,3)
257 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')
258 subplot(4,2,4)
259 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')
260 subplot(4,2,5)
261 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')
262 subplot(4,2,6)
263 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')
264 subplot(4,2,7)
265 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')
266 subplot(4,2,8)
267 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')
268 % %'Paperposition', [left bottom width height]
269 % set(fig,'PaperPositionMode', 'manual', '
    PaperUnits','Inches', 'Paperposition',[0.0 0.0 11
    8.5])
270 % print('-P\\mepprint2\\gle-2120-pr02c',fig)
271 name = [strrep(fig.Name,' ','_')];
272 print(ffigure(num),'-depsc','-noui','-painters',name)
    ;
273 fprintf(fid,'\\sectionmark{Project \\# 5\\hspace*{\\
    fill} Clair Cunningham \\hspace*{\\fill} Problem

```

```

    %d}\n',num);
274 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);
275 %% Problem 2
276 fprintf(fid, '\\sectionmark{Project \\# 5\\hspace*{\\
    fill} Clair Cunningham \\hspace*{\\fill} Problem
    2}\\n');
277 Alat = A(5:8,5:8);
278 Blat = B(5:8,2:3);
279 Clat = C(5:8,5:8);
280 Dlat = D(5:8,2:3);
281
282 [Wn_lat,Z_lat,P_lat] = damp(Alat);
283 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')
284
285 fprintf(fid, 'The dutch roll mode eigenvalues are %.3
    g 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))
286 t_roll = -1/P_lat(3);
287 t_spiral = -1/P_lat(4);
288 fprintf(fid, 'The roll mode eigenvalue is %.3g, and
    time constant %.3g. \\n',P_lat(3),t_roll)
289 fprintf(fid, 'The spiral mode eigenvalue is %.3g, and

```

```
        time constant %.3g. \n',P_lat(4),t_spiral)
290 fprintf(fid,'The system is stable because the
        eigenvalues are all negative. \\\n')
291
292 %% Aileron Input Simulation
293 num = num + 1;
294 %Inputs to Elevator, Aileron, or Rudder
295     % Degr
296     de_d = 0;
297     da_d = -0.5;
298     df_d = 0;
299     dr_d = 0; % deg
300     % Radians
301     de = de_d*d2r;
302     da = da_d*d2r;
303     df = df_d*d2r;
304     dr = dr_d*d2r;
305
306 %simulate nonlinear system
307 sim('Project_5_s_7_5')
308
309 %define ICs for linear system
310 xo=[0 0 0 0 0 0 0 0 0];
311
312 %simulate linear system
313 sim('Project_5_lin_s_7_5')
314
315 %Call up next figure
316 fig = figure('OuterPosition',pos_size,'
        PaperPositionMode','auto');
317 fig.Name = 'Aileron Input Simulation';
318 %Changes Paper Print Orientation to landscape on a
        per figure basic
319 orient landscape
320 %Output plots to called figure
321 subplot(4,2,1)
322 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')
323 subplot(4,2,2)
324 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')
325 subplot(4,2,3)
326 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')
327 subplot(4,2,4)
328 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')
329 subplot(4,2,5)
330 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')
331 subplot(4,2,6)
332 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')
333 subplot(4,2,7)
334 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')
335 subplot(4,2,8)
336 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')
337 % '%Paperposition', [left bottom width height]
338 % set(fig,'PaperPositionMode', 'manual', '
        PaperUnits','Inches', 'Paperposition',[0.0 0.0 11
        8.5])
339 % print('-P\\mepprint2\\gle-2120-pr02c',fig)
340 name = [strrep(fig.Name,' ','_')];
341 print(ffigure(num),'-depsc','-noui','-painters',name)
        ;
342 fprintf(fid,'\\sectionmark{Project \\# 5\\hspace*{\\
        fill} Clair Cunningham \\hspace*{\\fill} Problem
        %d}\\n',num);
343 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);
344 %% Rudder Input Simulation
345 num = num +1;
346 %Inputs to Elevator, Aileron, or Rudder
347     % Degr
348     de_d = 0;
349     da_d = 0;
350     df_d = 0;
351     dr_d = -2.0; % deg
352     % Radians
353     de = de_d*d2r;
354     da = da_d*d2r;
355     df = df_d*d2r;
356     dr = dr_d*d2r;
357
358 %simulate nonlinear system
359 sim('Project_5_s_7_5')
360
361 %define ICs for linear system

```

```
362 xo=[0 0 0 0 0 0 0 0 0];
363
364 %simulate linear system
365 sim('Project_5_lin_s_7_5')
366
367 %Call up next figure
368 fig = figure('OuterPosition',pos_size,'
    PaperPositionMode','auto');
369 fig.Name = 'Rudder Input Simulation';
370 %Changes Paper Print Orientation to landscape on a
    per figure basis
371 orient landscape
372 %Output plots to called figure
373 subplot(4,2,1)
374 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')
375 subplot(4,2,2)
376 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')
377 subplot(4,2,3)
378 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')
379 subplot(4,2,4)
380 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')
381 subplot(4,2,5)
382 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')
383 subplot(4,2,6)
384 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')
385 subplot(4,2,7)
386 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')
387 subplot(4,2,8)
388 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')
389 % '%Paperposition', [left bottom width height]
390 % set(fig,'PaperPositionMode', 'manual', '
        PaperUnits','Inches', 'Paperposition',[0.0 0.0 11
        8.5])
391 % print('-P\\meprint2\\gle-2120-pr02c',fig)
392 name = [strrep(fig.Name,' ','_')];
393 print(ffigure(num),'-depsc','-noui','-painters',name)
        ;
394 fprintf(fid,'\\sectionmark{Project \\# 5\\hspace*{\\
        fill} Clair Cunningham \\hspace*{\\fill} Problem
        %d}\\n',num);
395 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);
396 %% End of File Commands
397 name = 'Project_5_s_7_5';

```

```
398 load_system(name);
399 %modelhandle = get_param('name', 'Handle')
400 handles = find_system(name, 'FindAll', 'On', '
    SearchDepth', 10, ...
401     'regex', 'on', 'blocktype', 'port');
402 list = get(handles, 'Path');
403 if ~iscell(list)
404     list = {list};
405 end
406 list = unique(list);
407 if ~iscell(handles)
408     handle = {handles};
409 end
410 % % add main model
411 % list{end+1} = name;
412 % linear model
413 list{end+1} = 'Project_5_lin_s_7_5';
414 % GEt only last part of path, that is, after the
    last /
415 [r1, r2] = regexp(list, '[^/]+$ ', 'tokens', 'match'
    );
416
417 % Convert to usable format
418 names = [r2{:}]';
419
420 % Cells of printNames.
421 % Just rename every non-alphanumeric char to _, all
    space to ' '
422 printNames = regexprep(names', {'\s', '\W'}, {' ', '_
    '});
423
424 for i = 1 : length(list)
425     item = char(list(i));
426 modelhandle = get_param(item, 'Handle');
427 set(modelhandle, 'PaperPositionMode', 'manual', '
    PaperUnits', 'Inches', 'Paperposition', [0.0 0.0 11
    8.5])
428 print(['-s' item], '-depsc', '-noui', '-painters',
```



```
    printNames{i});  
429 fprintf(fid, '\\sectionmark{Project \\# 5\\hspace*{\\  
    fill} Clair Cunningham \\hspace*{\\fill} %s}\\n',  
    strep(printNames{i}, '_', '\\_'));  
430 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});  
431 end  
432 fclose(fid);
```