

The figure below while not the exact same as the solution given has variance in the 10^{-6} place only, a small number. Even if the model output is not exactly the same it still remains within a very close trim condition given no input. The rest of the solutions for the different problems follow with no variance from the solution given.

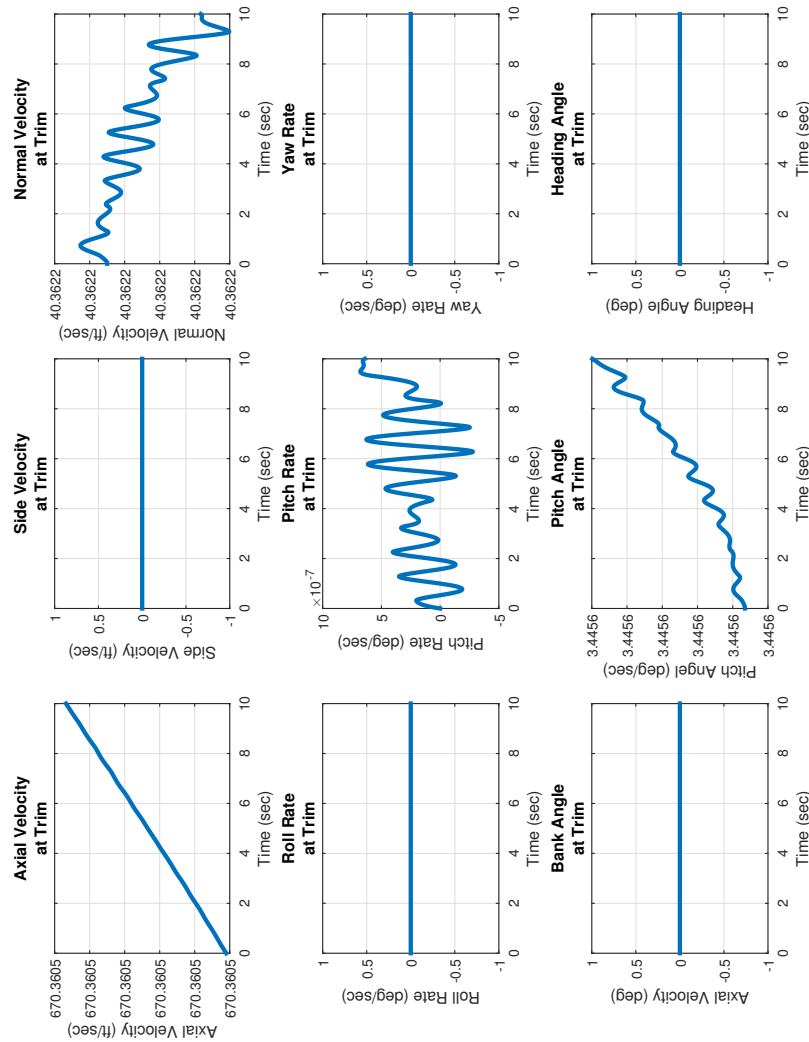


Figure 1: Problem 1 Graphical Solution

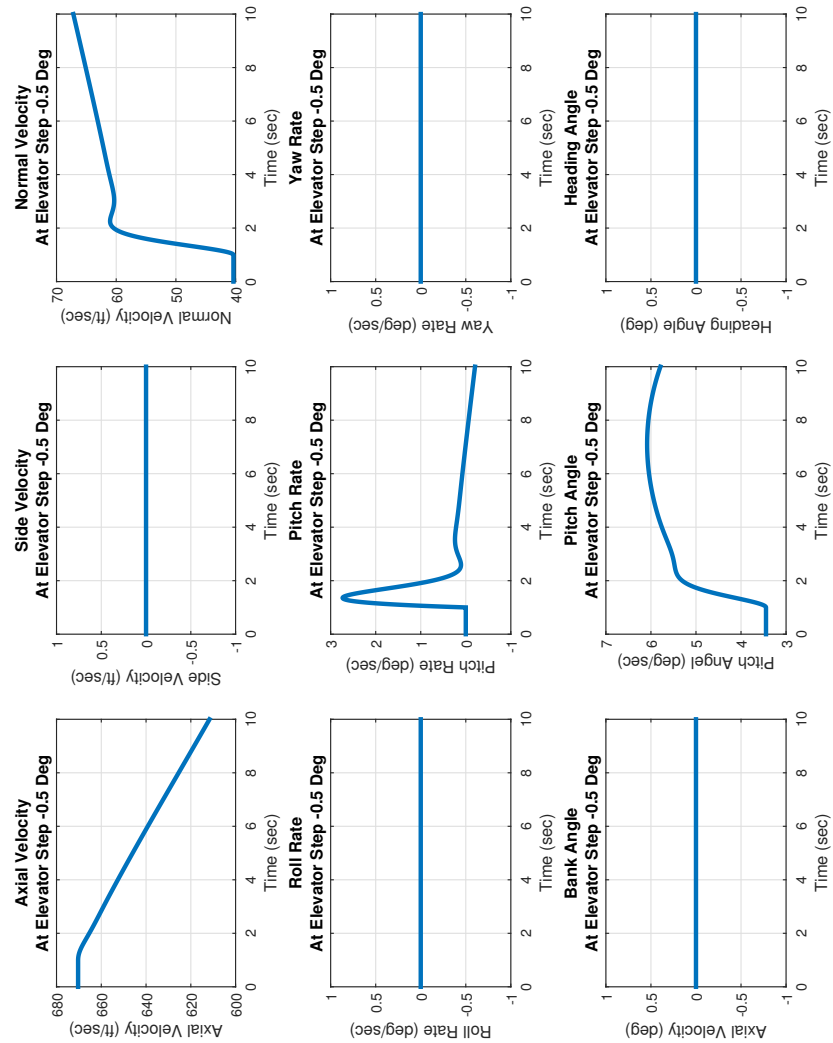


Figure 2: Problem 2 Graphical Solution

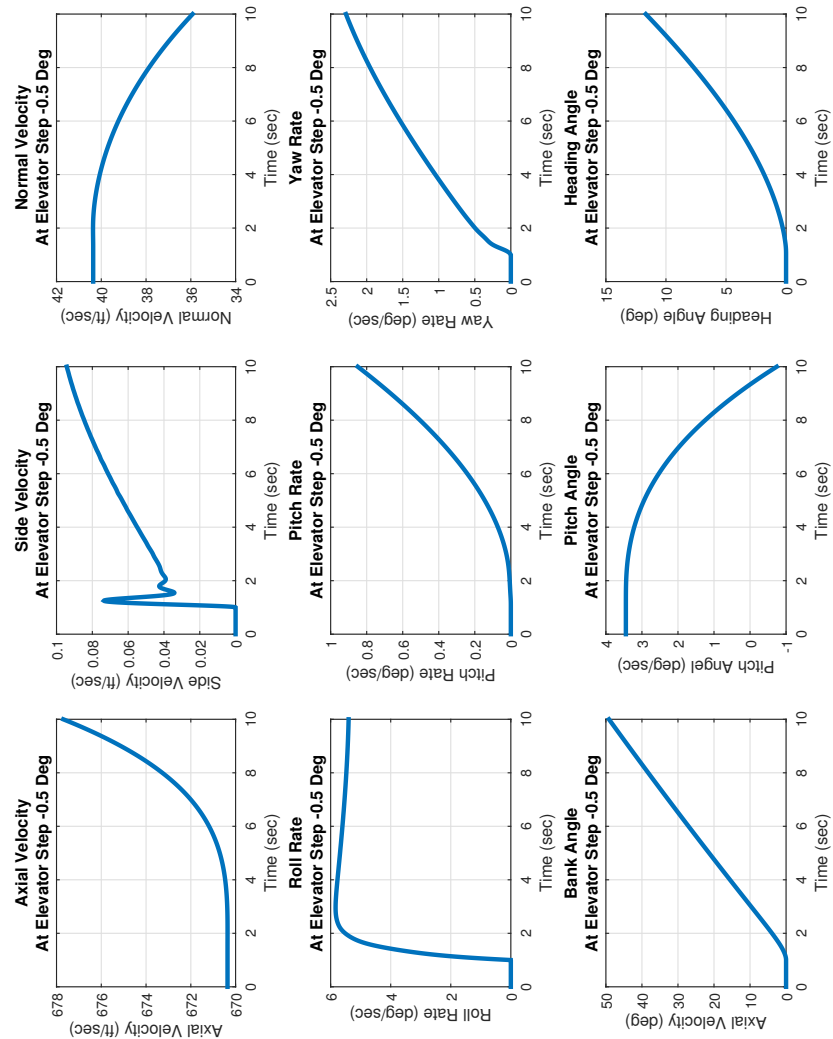


Figure 3: Problem 3 Graphical Solution

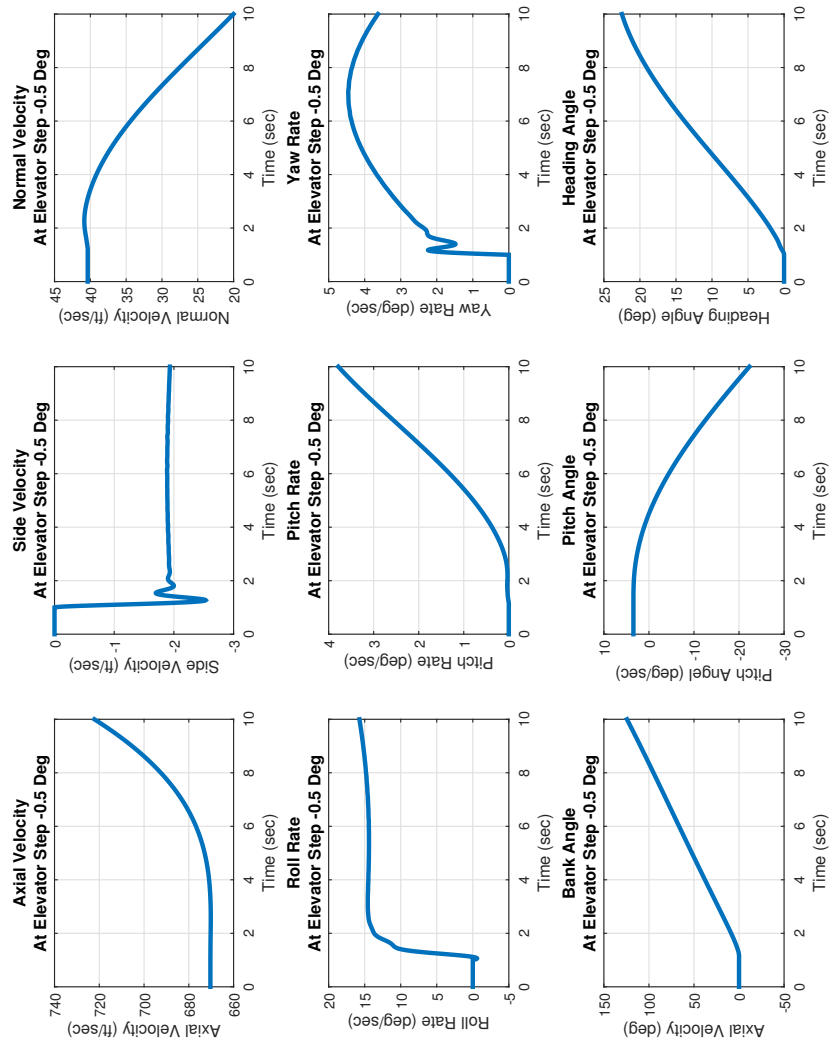


Figure 4: Problem 4 Graphical Solution

```
1 % Project 3
2 % Clair Cunningham
3
4 clear all; close all; clc
5 fid = fopen('Project_3.txt','w+');
6
7 % %Sets the units of your root object (screen) to
   pixels
8 set(0,'units','pixels');
9 %
10 % %Obtains this pixel information
11 Pix_SS = get(0,'screensize');
12 %
13 % %Sets the units of your root object (screen) to
   inches
14 set(0,'units','inches');
15 %
16 % %Obtains this inch information
17 Inch_SS = get(0,'screensize');
18 %
19 % %Calculates the resolution (pixels per inch)
20 Res = Pix_SS./Inch_SS;
21
22
23 % initial trim conditions
24 u_0 = 670.360471; % ft/sec
25 v_0 = 0; % ft/sec
26 w_0 = 40.362171; % ft/sec
27 p_0 = 0; % ft/sec
28 q_0 = 0; % ft/sec
29 r_0 = 0; % radians
30 phi_0 = 0; % radians
31 theta_0 = atan(w_0/u_0); % radians
32 psi_0 = 0; % radians
33
34 % Initial Conditions and Time Step
35 IC = [u_0, v_0, w_0, p_0, q_0, r_0, phi_0, theta_0,
        psi_0];
```

```
36 time = [0:0.01:10]';
37
38 % get solution for trim
39 [time_out,x_out] = ode45(@Project_3_f,time,IC);
40 u = x_out(:,1);
41 v = x_out(:,2);
42 w = x_out(:,3);
43 p = x_out(:,4)*180/pi;
44 q = x_out(:,5)*180/pi;
45 r = x_out(:,6)*180/pi;
46 phi = x_out(:,7)*180/pi;
47 theta = x_out(:,8)*180/pi;
48 psi = x_out(:,9)*180/pi;
49 fig = figure('OuterPosition',[0 0 Pix_SS(3)*.90
    Pix_SS(4)*.90], 'PaperPositionMode','auto');
50 subplot(3,3,1)
51 plot(time_out,u,'LineWidth',3.0); title({'Axial
    Velocity','at Trim'});xlabel('Time (sec)');ylabel
    ('Axial Velocity (ft/sec)'); grid on
52 %set(gca, 'YTickLabel', num2str(get(gca,'YTick'),'','%
    d'))
53 subplot(3,3,2)
54 plot(time_out,v,'LineWidth',3.0); title({'Side
    Velocity','at Trim'});xlabel('Time (sec)');ylabel
    ('Side Velocity (ft/sec)'); grid on
55 subplot(3,3,3)
56 plot(time_out,w,'LineWidth',3.0); title({'Normal
    Velocity','at Trim'});xlabel('Time (sec)');ylabel
    ('Normal Velocity (ft/sec)'); grid on
57 subplot(3,3,4)
58 plot(time_out,p,'LineWidth',3.0); title({'Roll Rate'
    , 'at Trim'});xlabel('Time (sec)');ylabel('Roll
    Rate (deg/sec)'); grid on
59 subplot(3,3,5)
60 plot(time_out,q,'LineWidth',3.0); title({'Pitch Rate
    ', 'at Trim'});xlabel('Time (sec)');ylabel('Pitch
    Rate (deg/sec)'); grid on
61 subplot(3,3,6)
```

```
62 plot(time_out,r,'LineWidth',3.0); title({'Yaw Rate',
    'at Trim'});xlabel('Time (sec)');ylabel('Yaw Rate
    (deg/sec)'); grid on
63 subplot(3,3,7)
64 plot(time_out,phi,'LineWidth',3.0); title({'Bank
    Angle','at Trim'});xlabel('Time (sec)');ylabel('
    Axial Velocity (deg)'); grid on
65 subplot(3,3,8)
66 plot(time_out,theta,'LineWidth',3.0); title({'Pitch
    Angle','at Trim'});xlabel('Time (sec)');ylabel('
    Pitch Angel (deg/sec)'); grid on
67 subplot(3,3,9)
68 plot(time_out,psi,'LineWidth',3.0); title({'Heading
    Angle','at Trim'});xlabel('Time (sec)');ylabel('
    Heading Angle (deg)'); grid on
69
70 name = ['Figure_' num2str(fig.Number)];
71 print(fig,'-depsc','-noui','-painters',name);
72 fprintf(fid,'\\sectionmark{Project \\# 3}\\hspace*{\\
    fill} Clair Cunningham \\hspace*{\\fill} Problem
    %d}\\n',fig.Number);
73 fprintf(fid,'The figure below while not the exact
    same as the solution given has variance in the
    10$^{-6}$ place only, a small number. Even if the
    model output is not exactly the same it still
    remains within a very close trim condition given
    no input. The rest of the solutions for the
    different problems follow with no variance from
    the solution given. ');
74 fprintf(fid,'\\vspace*{\\fill}\\begin{figure}[H]\\
    centering\\includegraphics[keepaspectratio=true,
    height=1\\textheight,width=1\\textwidth,angle
    =90]{%s.eps}\\n \\caption{Problem %d Graphical
    Solution}\\end{figure}\\vspace*{\\fill}\\n\\
    newpage\\n',name,fig.Number);
75
76 % get solution for step input of -0.5 deg to the
    elevator
```

```
77 [time_out,x_out] = ode45(@Project_3_de_f,time,IC);
78 u = x_out(:,1);
79 v = x_out(:,2);
80 w = x_out(:,3);
81 p = x_out(:,4)*180/pi;
82 q = x_out(:,5)*180/pi;
83 r = x_out(:,6)*180/pi;
84 phi = x_out(:,7)*180/pi;
85 theta = x_out(:,8)*180/pi;
86 psi = x_out(:,9)*180/pi;
87 fig = figure('OuterPosition',[0 0 Pix_SS(3)*.90
    Pix_SS(4)*.90], 'PaperPositionMode','auto');
88 subplot(3,3,1)
89 plot(time_out,u,'LineWidth',3.0); title({'Axial
    Velocity','At Elevator Step -0.5 Deg'});xlabel('
    Time (sec)');ylabel('Axial Velocity (ft/sec)');
    grid on
90 subplot(3,3,2)
91 plot(time_out,v,'LineWidth',3.0); title({'Side
    Velocity','At Elevator Step -0.5 Deg'});xlabel('
    Time (sec)');ylabel('Side Velocity (ft/sec)');
    grid on
92 subplot(3,3,3)
93 plot(time_out,w,'LineWidth',3.0); title({'Normal
    Velocity','At Elevator Step -0.5 Deg'});xlabel('
    Time (sec)');ylabel('Normal Velocity (ft/sec)');
    grid on
94 subplot(3,3,4)
95 plot(time_out,p,'LineWidth',3.0); title({'Roll Rate'
    , 'At Elevator Step -0.5 Deg'});xlabel('Time (sec)
    ');ylabel('Roll Rate (deg/sec)'); grid on
96 subplot(3,3,5)
97 plot(time_out,q,'LineWidth',3.0); title({'Pitch Rate
    ', 'At Elevator Step -0.5 Deg'});xlabel('Time (sec)
    ');ylabel('Pitch Rate (deg/sec)'); grid on
98 subplot(3,3,6)
99 plot(time_out,r,'LineWidth',3.0); title({'Yaw Rate',
    'At Elevator Step -0.5 Deg'});xlabel('Time (sec)')
```



```
);ylabel('Yaw Rate (deg/sec)'); grid on
100 subplot(3,3,7)
101 plot(time_out,phi,'LineWidth',3.0); title({'Bank
      Angle','At Elevator Step -0.5 Deg'});xlabel('Time
      (sec)');ylabel('Axial Velocity (deg)'); grid on
102 subplot(3,3,8)
103 plot(time_out,theta,'LineWidth',3.0); title({'Pitch
      Angle','At Elevator Step -0.5 Deg'});xlabel('Time
      (sec)');ylabel('Pitch Angel (deg/sec)'); grid on
104 subplot(3,3,9)
105 plot(time_out,psi,'LineWidth',3.0); title({'Heading
      Angle','At Elevator Step -0.5 Deg'});xlabel('Time
      (sec)');ylabel('Heading Angle (deg)'); grid on
106
107 name = ['Figure_' num2str(fig.Number)];
108 print(fig,'-depsc','-noui','-painters',name);
109 fprintf(fid,'\\sectionmark{Project \\# 3\\hspace*{\\
      fill} Clair Cunningham \\hspace*{\\fill} Problem
      %d}\\n',fig.Number);
110 fprintf(fid,'\\vspace*{\\fill}\\begin{figure}[H]\\
      centering\\includegraphics[keepaspectratio=true,
      height=1\\textheight,width=1\\textwidth,angle
      =90]{%s.eps}\\n \\caption{Problem %d Graphical
      Solution}\\end{figure}\\vspace*{\\fill}\\n\\
      newpage\\n',name,fig.Number);
111
112 % get solution for step input of -0.5 deg to the
      aileron
113 [time_out,x_out] = ode45(@Project_3_da_f,time,IC);
114 u = x_out(:,1);
115 v = x_out(:,2);
116 w = x_out(:,3);
117 p = x_out(:,4)*180/pi;
118 q = x_out(:,5)*180/pi;
119 r = x_out(:,6)*180/pi;
120 phi = x_out(:,7)*180/pi;
121 theta = x_out(:,8)*180/pi;
122 psi = x_out(:,9)*180/pi;
```

```
123 fig = figure('OuterPosition',[0 0 Pix_SS(3)*.90
    Pix_SS(4)*.90], 'PaperPositionMode', 'auto');
124 subplot(3,3,1)
125 plot(time_out,u,'LineWidth',3.0); title({'Axial
    Velocity','At Elevator Step -0.5 Deg'});xlabel('
    Time (sec)');ylabel('Axial Velocity (ft/sec)');
    grid on
126 subplot(3,3,2)
127 plot(time_out,v,'LineWidth',3.0); title({'Side
    Velocity','At Elevator Step -0.5 Deg'});xlabel('
    Time (sec)');ylabel('Side Velocity (ft/sec)');
    grid on
128 subplot(3,3,3)
129 plot(time_out,w,'LineWidth',3.0); title({'Normal
    Velocity','At Elevator Step -0.5 Deg'});xlabel('
    Time (sec)');ylabel('Normal Velocity (ft/sec)');
    grid on
130 subplot(3,3,4)
131 plot(time_out,p,'LineWidth',3.0); title({'Roll Rate'
    , 'At Elevator Step -0.5 Deg'});xlabel('Time (sec)
    ');ylabel('Roll Rate (deg/sec)'); grid on
132 subplot(3,3,5)
133 plot(time_out,q,'LineWidth',3.0); title({'Pitch Rate
    ', 'At Elevator Step -0.5 Deg'});xlabel('Time (sec
    ');ylabel('Pitch Rate (deg/sec)'); grid on
134 subplot(3,3,6)
135 plot(time_out,r,'LineWidth',3.0); title({'Yaw Rate',
    'At Elevator Step -0.5 Deg'});xlabel('Time (sec)'
    );ylabel('Yaw Rate (deg/sec)'); grid on
136 subplot(3,3,7)
137 plot(time_out,phi,'LineWidth',3.0); title({'Bank
    Angle','At Elevator Step -0.5 Deg'});xlabel('Time
    (sec)');ylabel('Axial Velocity (deg)'); grid on
138 subplot(3,3,8)
139 plot(time_out,theta,'LineWidth',3.0); title({'Pitch
    Angle','At Elevator Step -0.5 Deg'});xlabel('Time
    (sec)');ylabel('Pitch Angel (deg/sec)'); grid on
140 subplot(3,3,9)
```

```
141 plot(time_out,psi,'LineWidth',3.0); title({'Heading  
    Angle','At Elevator Step -0.5 Deg'});xlabel('Time  
    (sec)');ylabel('Heading Angle (deg)'); grid on
142
143 name = ['Figure_' num2str(fig.Number)];
144 print(fig,'-depsc','-noui','-painters',name);
145 fprintf(fid,'\\sectionmark{Project \\# 3\\hspace*{\\fill} Clair Cunningham \\hspace*{\\fill} Problem  
    %d}\\n',fig.Number);
146 fprintf(fid,'\\vspace*{\\fill}\\begin{figure}[H]\\centering\\includegraphics[keepaspectratio=true,  
    height=1\\textheight,width=1\\textwidth,angle  
    =90]{%s.eps}\\n \\caption{Problem %d Graphical  
    Solution}\\end{figure}\\vspace*{\\fill}\\n\\newpage\\n',name,fig.Number);
147
148 % get solution for step input of -2.0 deg to the  
    rudder
149 [time_out,x_out] = ode45(@Project_3_dr_f,time,IC);
150 u = x_out(:,1);
151 v = x_out(:,2);
152 w = x_out(:,3);
153 p = x_out(:,4)*180/pi;
154 q = x_out(:,5)*180/pi;
155 r = x_out(:,6)*180/pi;
156 phi = x_out(:,7)*180/pi;
157 theta = x_out(:,8)*180/pi;
158 psi = x_out(:,9)*180/pi;
159 fig = figure('OuterPosition',[0 0 Pix_SS(3)*.90  
    Pix_SS(4)*.90],'PaperPositionMode','auto');
160 subplot(3,3,1)
161 plot(time_out,u,'LineWidth',3.0); title({'Axial  
    Velocity','At Elevator Step -0.5 Deg'});xlabel('Time (sec)');ylabel('Axial Velocity (ft/sec)');  
    grid on
162 subplot(3,3,2)
163 plot(time_out,v,'LineWidth',3.0); title({'Side  
    Velocity','At Elevator Step -0.5 Deg'});xlabel('Time (sec)');
```

```

        Time (sec)');ylabel('Side Velocity (ft/sec)');
        grid on
164 subplot(3,3,3)
165 plot(time_out,w,'LineWidth',3.0); title({'Normal
        Velocity','At Elevator Step -0.5 Deg'});xlabel('
        Time (sec)');ylabel('Normal Velocity (ft/sec)');
        grid on
166 subplot(3,3,4)
167 plot(time_out,p,'LineWidth',3.0); title({'Roll Rate'
        , 'At Elevator Step -0.5 Deg'});xlabel('Time (sec)
        ');ylabel('Roll Rate (deg/sec)'); grid on
168 subplot(3,3,5)
169 plot(time_out,q,'LineWidth',3.0); title({'Pitch Rate
        ', 'At Elevator Step -0.5 Deg'});xlabel('Time (sec
        ');ylabel('Pitch Rate (deg/sec)'); grid on
170 subplot(3,3,6)
171 plot(time_out,r,'LineWidth',3.0); title({'Yaw Rate',
        'At Elevator Step -0.5 Deg'});xlabel('Time (sec)'
        );ylabel('Yaw Rate (deg/sec)'); grid on
172 subplot(3,3,7)
173 plot(time_out,phi,'LineWidth',3.0); title({'Bank
        Angle','At Elevator Step -0.5 Deg'});xlabel('Time
        (sec)');ylabel('Axial Velocity (deg)'); grid on
174 subplot(3,3,8)
175 plot(time_out,theta,'LineWidth',3.0); title({'Pitch
        Angle','At Elevator Step -0.5 Deg'});xlabel('Time
        (sec)');ylabel('Pitch Angel (deg/sec)'); grid on
176 subplot(3,3,9)
177 plot(time_out,psi,'LineWidth',3.0); title({'Heading
        Angle','At Elevator Step -0.5 Deg'});xlabel('Time
        (sec)');ylabel('Heading Angle (deg)'); grid on
178
179 name = ['Figure_' num2str(fig.Number)];
180 print(fig,'-depsc','-noui','-painters',name);
181 fprintf(fid,'\\sectionmark{Project \\# 3\\hspace*{\\
        fill} Clair Cunningham \\hspace*{\\fill} Problem
        %d}\\n',fig.Number);
182 fprintf(fid,'\\vspace*{\\fill}\\begin{figure}[H]\\

```

```
        centering\\includegraphics[keepaspectratio=true,
        height=1\\textheight,width=1\\textwidth,angle
        =90]{%s.eps}\\n \\caption{Problem %d Graphical
        Solution}\\end{figure}\\vspace*{\\fill}\\n\\
        newpage\\n',name,fig.Number);
183
184 fclose(fid);
```

```
1 %Project_3_f.m
2 function dx = Project_3_f(t,x)
3 % set up variables to incoming conditions
4 u = x(1); % ft/sec
5 v = x(2); % ft/sec
6 w = x(3); % ft/sec
7 p = x(4); % ft/sec
8 q = x(5); % ft/sec
9 r = x(6); % radians
10 phi = x(7); % radians
11 theta = x(8); % radians
12 psi = x(9); % radians
13
14 % % % command troubleshooting
15 % % u = u_0;
16 % % v = v_0;
17 % % w = w_0;
18 % % p = p_0;
19 % % q = q_0;
20 % % r = r_0;
21 % % phi = phi_0;
22 % % theta = theta_0;
23 % % psi = psi_0;
24
25 % Initial Trim Conditions
26 T = 3767.207337; % thrust in lbs
27 de = -2.9846046; % deg
28 da = 0; % deg
29 dr = 0; % deg
30
31 % aircraft properties
32 i_xx = 8691.46164;
33 i_yy = 70668.585;
34 i_zz = 70418.67355;
35 i_xz = 151.43836;
36 i_xy = 0;
37 i_yz = 0;
38
```

```
39 s = 300; % ft^2
40 cbar = 11.32; % ft
41 b = 30; % ft
42 m = 756.5262463; % mass in slugs
43 rho = 0.0012669984; % slugs/ft^3
44 g = 32.17561865; % ft/sec^2
45
46 % other variables
47 qbar = 0.5*rho*u^2;
48
49 % intitial aerodynamic conditions; where except for
    "not" units (/deg);
50 c_x_0 = -2.13536e-02;
51 c_x_u = 1.289018e-04;
52 c_x_w = -2.17775e-03;
53 c_x_q = 2.1928052e-04;
54 c_x_de = 1.386632e-03;
55 c_y_0 = 0;
56 c_y_v = -6.4490425e-02;
57 c_y_p = 1.33481e-03;
58 c_y_r = 9.401418e-03;
59 c_y_da = 4.618436e-04;
60 c_y_dr = 2.991717e-03;
61 c_z_0 = 5.092263e-02;
62 c_z_u = -4.3444023e-04;
63 c_z_w = -1.9946051e-03;
64 c_z_q = -5.3473522e-02;
65 c_z_de = -1.2167892e-02;
66 c_l_0 = 0;
67 c_l_v = -1.75539e-03;
68 c_l_p = -7.392626e-03;
69 c_l_r = 5.910111e-05;
70 c_l_da = -2.089358e-03;
71 c_l_dr = 4.7651867e-04;
72 c_m_0 = -1.39985e-02;
73 c_m_u = -1.15335756e-04;
74 c_m_w = -1.16313463e-03;
75 c_m_q = -6.08086182e-01;
```

```
76 c_m_de = -4.632495451e-02;
77 c_n_0 = 0;
78 c_n_v = 5.1988574e-03;
79 c_n_p = -4.294548e-04;
80 c_n_r = -8.6047784e-03;
81 c_n_da = -1.95539e-04;
82 c_n_dr = -5.50282873e-03;
83
84 % aerodynamic model; where u,v,w have units ft/sec;
85 c_x = c_x_0 + c_x_u*u + c_x_w*w + (cbar/(2*u))*c_x_q
      *q*(180/pi) + c_x_de*de;
86 c_y = c_y_0 + c_y_v*v + (b/(2*u))*(c_y_p*p + c_y_r*r
     )*(180/pi) + c_y_da*da + c_y_dr*dr;
87 c_z = c_z_0 + c_z_u*u + c_z_w*w + (cbar/(2*u))*c_z_q
      *q*(180/pi) + c_z_de*de;
88 c_l = c_l_0 + c_l_v*v + (b/(2*u))*(c_l_p*p + c_l_r*r
     )*(180/pi) + c_l_da*da + c_l_dr*dr;
89 c_m = c_m_0 + c_m_u*u + c_m_w*w + (cbar/(2*u))*c_m_q
      *q*(180/pi) + c_m_de*de;
90 c_n = c_n_0 + c_n_v*v + (b/(2*u))*(c_n_p*p + c_n_r*r
     )*(180/pi) + c_n_da*da + c_n_dr*dr;
91
92 % external forces
93 f_a_x = qbar*s*c_x;
94 f_a_y = qbar*s*c_y;
95 f_a_z = qbar*s*c_z;
96 f_T_x = T;
97 f_T_y = 0;
98 f_T_z = 0;
99
100 % external moments
101 m_e_x = qbar*s*b*c_l;
102 m_e_y = qbar*s*cbar*c_m;
103 m_e_z = qbar*s*b*c_n;
104
105 % Solves for the rate change of velocity from force
    equations
106 udot = -g*sin(theta)+(f_a_x+f_T_x)/m+v*r-w*q;
```



```

107 vdot = g*sin(phi)*cos(theta)+(f_a_y+f_T_y)/m+w*p-u*r
      ;
108 wdot = g*cos(phi)*cos(theta)+(f_a_z+f_T_z)/m-v*p+u*q
      ;
109 UVW = [udot;vdot;wdot];
110 % Solves for rate change of moments
111 I = [i_xx -i_xy -i_xz; -i_xy i_yy -i_yz; -i_xz -i_yz
      i_zz]; % The "inertia" matrix
112 B = [q*r*(i_yy-i_zz)+(q^2-r^2)*i_yz-p*r*i_xy+p*q*
      i_xz+m_e_x; p*r*(i_zz-i_xx)+(r^2-p^2)*i_xz-p*q*
      i_yz+q*r*i_xy+m_e_y; p*q*(i_xx-i_yy)+(p^2-q^2)*
      i_xy-q*r*i_xz+p*r*i_yz+m_e_z];
113 A = inv(I)*B; % rate of change matrix
114 % Solves for rate change of angles
115 phidot = p+q*sin(phi)*tan(theta)+r*cos(phi)*tan(
      theta);
116 thetadot = q*cos(phi)-r*sin(phi);
117 psidot = (q*sin(phi)+r*cos(phi))*sec(theta);
118 POW = [phidot;thetadot;psidot];
119
120 C = [UVW;B;POW];
121 D = [1 0 0 0 0 0 0 0 0
      0 1 0 0 0 0 0 0 0
      0 0 1 0 0 0 0 0 0
      0 0 0 i_xx -i_xy -i_xz 0 0 0
      0 0 0 -i_xy i_yy -i_yz 0 0 0
      0 0 0 -i_xz -i_yz i_zz 0 0 0
      0 0 0 0 0 0 1 0 0
      0 0 0 0 0 0 0 1 0
      0 0 0 0 0 0 0 0 1];
130 E = inv(D)*C;
131 % Spits out derivatives
132 dx = E; % don't forget to add the semi-colon back
      after checking to make sure all rate changes are
      zero or close enough to the millionth
133 % pause
134 end

```

```
1 %Project_3_da_f.m
2 function dx = Project_3_da_f(t,x)
3 % set up variables to incoming conditions
4 u = x(1); % ft/sec
5 v = x(2); % ft/sec
6 w = x(3); % ft/sec
7 p = x(4); % ft/sec
8 q = x(5); % ft/sec
9 r = x(6); % radians
10 phi = x(7); % radians
11 theta = x(8); % radians
12 psi = x(9); % radians
13
14 % % command troubleshooting
15 % u = u_0;
16 % v = v_0;
17 % w = w_0;
18 % p = p_0;
19 % q = q_0;
20 % r = r_0;
21 % phi = phi_0;
22 % theta = theta_0;
23 % psi = psi_0;
24
25 % Initial Trim Conditions
26 T = 3767.207337; % thrust in lbs
27 de = -2.9846046; % deg
28 da = 0; % deg
29 dr = 0; % deg
30
31 if t >= 1
32     da = da + -0.5; % deg
33 end
34
35 % aircraft properties
36 i_xx = 8691.46164;
37 i_yy = 70668.585;
38 i_zz = 70418.67355;
```

```
39 i_xz = 151.43836;
40 i_xy = 0;
41 i_yz = 0;
42
43 s = 300; % ft^2
44 cbar = 11.32; % ft
45 b = 30; % ft
46 m = 756.5262463; % mass in slugs
47 rho = 0.0012669984; % slugs/ft^3
48 g = 32.17561865; % ft/sec^2
49
50 % other variables
51 qbar = 0.5*rho*u^2;
52
53 % intitial aerodynamic conditions; where except for
    "not" units (/deg);
54 c_x_0 = -2.13536e-02;
55 c_x_u = 1.289018e-04;
56 c_x_w = -2.17775e-03;
57 c_x_q = 2.1928052e-04;
58 c_x_de = 1.386632e-03;
59 c_y_0 = 0;
60 c_y_v = -6.4490425e-02;
61 c_y_p = 1.33481e-03;
62 c_y_r = 9.401418e-03;
63 c_y_da = 4.618436e-04;
64 c_y_dr = 2.991717e-03;
65 c_z_0 = 5.092263e-02;
66 c_z_u = -4.3444023e-04;
67 c_z_w = -1.9946051e-03;
68 c_z_q = -5.3473522e-02;
69 c_z_de = -1.2167892e-02;
70 c_l_0 = 0;
71 c_l_v = -1.75539e-03;
72 c_l_p = -7.392626e-03;
73 c_l_r = 5.910111e-05;
74 c_l_da = -2.089358e-03;
75 c_l_dr = 4.7651867e-04;
```

```
76 c_m_0 = -1.39985e-02;
77 c_m_u = -1.15335756e-04;
78 c_m_w = -1.16313463e-03;
79 c_m_q = -6.08086182e-01;
80 c_m_de = -4.632495451e-02;
81 c_n_0 = 0;
82 c_n_v = 5.1988574e-03;
83 c_n_p = -4.294548e-04;
84 c_n_r = -8.6047784e-03;
85 c_n_da = -1.95539e-04;
86 c_n_dr = -5.50282873e-03;
87
88 % aerodynamic model; where u,v,w have units ft/sec;
89 c_x = c_x_0 + c_x_u*u + c_x_w*w + (cbar/(2*u))*c_x_q
      *q*(180/pi) + c_x_de*de;
90 c_y = c_y_0 + c_y_v*v + (b/(2*u))*(c_y_p*p + c_y_r*r
     )*(180/pi) + c_y_da*da + c_y_dr*dr;
91 c_z = c_z_0 + c_z_u*u + c_z_w*w + (cbar/(2*u))*c_z_q
      *q*(180/pi) + c_z_de*de;
92 c_l = c_l_0 + c_l_v*v + (b/(2*u))*(c_l_p*p + c_l_r*r
     )*(180/pi) + c_l_da*da + c_l_dr*dr;
93 c_m = c_m_0 + c_m_u*u + c_m_w*w + (cbar/(2*u))*c_m_q
      *q*(180/pi) + c_m_de*de;
94 c_n = c_n_0 + c_n_v*v + (b/(2*u))*(c_n_p*p + c_n_r*r
     )*(180/pi) + c_n_da*da + c_n_dr*dr;
95
96 % external forces
97 f_a_x = qbar*s*c_x;
98 f_a_y = qbar*s*c_y;
99 f_a_z = qbar*s*c_z;
100 f_T_x = T;
101 f_T_y = 0;
102 f_T_z = 0;
103
104 % external moments
105 m_e_x = qbar*s*b*c_l;
106 m_e_y = qbar*s*cbar*c_m;
107 m_e_z = qbar*s*b*c_n;
```

```
108
109 % Solves for the rate change of velocity from force
    equations
110 udot = -g*sin(theta)+(f_a_x+f_T_x)/m+v*r-w*q;
111 vdot = g*sin(phi)*cos(theta)+(f_a_y+f_T_y)/m+w*p-u*r
    ;
112 wdot = g*cos(phi)*cos(theta)+(f_a_z+f_T_z)/m-v*p+u*q
    ;
113 UVW = [udot;vdot;wdot];
114 % Solves for rate change of moments
115 I = [i_xx -i_xy -i_xz; -i_xy i_yy -i_yz; -i_xz -i_yz
    i_zz]; % The "inertia" matrix
116 B = [q*r*(i_yy-i_zz)+(q^2-r^2)*i_yz-p*r*i_xy+p*q*
    i_xz+m_e_x; p*r*(i_zz-i_xx)+(r^2-p^2)*i_xz-p*q*
    i_yz+q*r*i_xy+m_e_y; p*q*(i_xx-i_yy)+(p^2-q^2)*
    i_xy-q*r*i_xz+p*r*i_yz+m_e_z];
117 A = inv(I)*B; % rate of change matrix
118 % Solves for rate change of angles
119 phidot = p+q*sin(phi)*tan(theta)+r*cos(phi)*tan(
    theta);
120 thetadot = q*cos(phi)-r*sin(phi);
121 psidot = (q*sin(phi)+r*cos(phi))*sec(theta);
122 POW = [phidot;thetadot;psidot];
123
124 % Spits out derivatives
125 dx = [UVW;A;POW]; % don't forget to add the semi-
    colon back after checking to make sure all rate
    changes are zero or close enough to the millionth
126 % pause
127 end
```

```
1 %Project_3_de_f.m
2 function dx = Project_3_de_f(t,x)
3 % set up variables to incoming conditions
4 u = x(1); % ft/sec
5 v = x(2); % ft/sec
6 w = x(3); % ft/sec
7 p = x(4); % ft/sec
8 q = x(5); % ft/sec
9 r = x(6); % radians
10 phi = x(7); % radians
11 theta = x(8); % radians
12 psi = x(9); % radians
13
14 % % command troubleshooting
15 % u = u_0;
16 % v = v_0;
17 % w = w_0;
18 % p = p_0;
19 % q = q_0;
20 % r = r_0;
21 % phi = phi_0;
22 % theta = theta_0;
23 % psi = psi_0;
24
25 % Initial Trim Conditions
26 T = 3767.207337; % thrust in lbs
27 de = -2.9846046; % deg
28 da = 0; % deg
29 dr = 0; % deg
30
31 if t >= 1
32     de = de + -0.5; % deg
33 end
34
35 % aircraft properties
36 i_xx = 8691.46164;
37 i_yy = 70668.585;
38 i_zz = 70418.67355;
```

```
39 i_xz = 151.43836;
40 i_xy = 0;
41 i_yz = 0;
42
43 s = 300; % ft^2
44 cbar = 11.32; % ft
45 b = 30; % ft
46 m = 756.5262463; % mass in slugs
47 rho = 0.0012669984; % slugs/ft^3
48 g = 32.17561865; % ft/sec^2
49
50 % other variables
51 qbar = 0.5*rho*u^2;
52
53 % initial aerodynamic conditions; where except for
    "not" units (/deg);
54 c_x_0 = -2.13536e-02;
55 c_x_u = 1.289018e-04;
56 c_x_w = -2.17775e-03;
57 c_x_q = 2.1928052e-04;
58 c_x_de = 1.386632e-03;
59 c_y_0 = 0;
60 c_y_v = -6.4490425e-02;
61 c_y_p = 1.33481e-03;
62 c_y_r = 9.401418e-03;
63 c_y_da = 4.618436e-04;
64 c_y_dr = 2.991717e-03;
65 c_z_0 = 5.092263e-02;
66 c_z_u = -4.3444023e-04;
67 c_z_w = -1.9946051e-03;
68 c_z_q = -5.3473522e-02;
69 c_z_de = -1.2167892e-02;
70 c_l_0 = 0;
71 c_l_v = -1.75539e-03;
72 c_l_p = -7.392626e-03;
73 c_l_r = 5.910111e-05;
74 c_l_da = -2.089358e-03;
75 c_l_dr = 4.7651867e-04;
```

```
76 c_m_0 = -1.39985e-02;
77 c_m_u = -1.15335756e-04;
78 c_m_w = -1.16313463e-03;
79 c_m_q = -6.08086182e-01;
80 c_m_de = -4.632495451e-02;
81 c_n_0 = 0;
82 c_n_v = 5.1988574e-03;
83 c_n_p = -4.294548e-04;
84 c_n_r = -8.6047784e-03;
85 c_n_da = -1.95539e-04;
86 c_n_dr = -5.50282873e-03;
87
88 % aerodynamic model; where u,v,w have units ft/sec;
89 c_x = c_x_0 + c_x_u*u + c_x_w*w + (cbar/(2*u))*c_x_q
      *q*(180/pi) + c_x_de*de;
90 c_y = c_y_0 + c_y_v*v + (b/(2*u))*(c_y_p*p + c_y_r*r
     )*(180/pi) + c_y_da*da + c_y_dr*dr;
91 c_z = c_z_0 + c_z_u*u + c_z_w*w + (cbar/(2*u))*c_z_q
      *q*(180/pi) + c_z_de*de;
92 c_l = c_l_0 + c_l_v*v + (b/(2*u))*(c_l_p*p + c_l_r*r
     )*(180/pi) + c_l_da*da + c_l_dr*dr;
93 c_m = c_m_0 + c_m_u*u + c_m_w*w + (cbar/(2*u))*c_m_q
      *q*(180/pi) + c_m_de*de;
94 c_n = c_n_0 + c_n_v*v + (b/(2*u))*(c_n_p*p + c_n_r*r
     )*(180/pi) + c_n_da*da + c_n_dr*dr;
95
96 % external forces
97 f_a_x = qbar*s*c_x;
98 f_a_y = qbar*s*c_y;
99 f_a_z = qbar*s*c_z;
100 f_T_x = T;
101 f_T_y = 0;
102 f_T_z = 0;
103
104 % external moments
105 m_e_x = qbar*s*b*c_l;
106 m_e_y = qbar*s*cbar*c_m;
107 m_e_z = qbar*s*b*c_n;
```



```
108
109 % Solves for the rate change of velocity from force
    equations
110 udot = -g*sin(theta)+(f_a_x+f_T_x)/m+v*r-w*q;
111 vdot = g*sin(phi)*cos(theta)+(f_a_y+f_T_y)/m+w*p-u*r
    ;
112 wdot = g*cos(phi)*cos(theta)+(f_a_z+f_T_z)/m-v*p+u*q
    ;
113 UVW = [udot;vdot;wdot];
114 % Solves for rate change of moments
115 I = [i_xx -i_xy -i_xz; -i_xy i_yy -i_yz; -i_xz -i_yz
    i_zz]; % The "inertia" matrix
116 B = [q*r*(i_yy-i_zz)+(q^2-r^2)*i_yz-p*r*i_xy+p*q*
    i_xz+m_e_x; p*r*(i_zz-i_xx)+(r^2-p^2)*i_xz-p*q*
    i_yz+q*r*i_xy+m_e_y; p*q*(i_xx-i_yy)+(p^2-q^2)*
    i_xy-q*r*i_xz+p*r*i_yz+m_e_z];
117 A = inv(I)*B; % rate of change matrix
118 % Solves for rate change of angles
119 phidot = p+q*sin(phi)*tan(theta)+r*cos(phi)*tan(
    theta);
120 thetadot = q*cos(phi)-r*sin(phi);
121 psidot = (q*sin(phi)+r*cos(phi))*sec(theta);
122 POW = [phidot;thetadot;psidot];
123
124 % Spits out derivatives
125 dx = [UVW;A;POW]; % don't forget to add the semi-
    colon back after checking to make sure all rate
    changes are zero or close enough to the millionth
126 % pause
127 end
```

```
1 %Project_3_dr_f.m
2 function dx = Project_3_dr_f(t,x)
3 % set up variables to incoming conditions
4 u = x(1); % ft/sec
5 v = x(2); % ft/sec
6 w = x(3); % ft/sec
7 p = x(4); % ft/sec
8 q = x(5); % ft/sec
9 r = x(6); % radians
10 phi = x(7); % radians
11 theta = x(8); % radians
12 psi = x(9); % radians
13
14 % % command troubleshooting
15 % u = u_0;
16 % v = v_0;
17 % w = w_0;
18 % p = p_0;
19 % q = q_0;
20 % r = r_0;
21 % phi = phi_0;
22 % theta = theta_0;
23 % psi = psi_0;
24
25 % Initial Trim Conditions
26 T = 3767.207337; % thrust in lbs
27 de = -2.9846046; % deg
28 da = 0; % deg
29 dr = 0; % deg
30
31 if t >= 1
32     dr = dr + -2.0; % deg
33 end
34
35 % aircraft properties
36 i_xx = 8691.46164;
37 i_yy = 70668.585;
38 i_zz = 70418.67355;
```

```
39 i_xz = 151.43836;
40 i_xy = 0;
41 i_yz = 0;
42
43 s = 300; % ft^2
44 cbar = 11.32; % ft
45 b = 30; % ft
46 m = 756.5262463; % mass in slugs
47 rho = 0.0012669984; % slugs/ft^3
48 g = 32.17561865; % ft/sec^2
49
50 % other variables
51 qbar = 0.5*rho*u^2;
52
53 % intitial aerodynamic conditions; where except for
    "not" units (/deg);
54 c_x_0 = -2.13536e-02;
55 c_x_u = 1.289018e-04;
56 c_x_w = -2.17775e-03;
57 c_x_q = 2.1928052e-04;
58 c_x_de = 1.386632e-03;
59 c_y_0 = 0;
60 c_y_v = -6.4490425e-02;
61 c_y_p = 1.33481e-03;
62 c_y_r = 9.401418e-03;
63 c_y_da = 4.618436e-04;
64 c_y_dr = 2.991717e-03;
65 c_z_0 = 5.092263e-02;
66 c_z_u = -4.3444023e-04;
67 c_z_w = -1.9946051e-03;
68 c_z_q = -5.3473522e-02;
69 c_z_de = -1.2167892e-02;
70 c_l_0 = 0;
71 c_l_v = -1.75539e-03;
72 c_l_p = -7.392626e-03;
73 c_l_r = 5.910111e-05;
74 c_l_da = -2.089358e-03;
75 c_l_dr = 4.7651867e-04;
```

```
76 c_m_0 = -1.39985e-02;
77 c_m_u = -1.15335756e-04;
78 c_m_w = -1.16313463e-03;
79 c_m_q = -6.08086182e-01;
80 c_m_de = -4.632495451e-02;
81 c_n_0 = 0;
82 c_n_v = 5.1988574e-03;
83 c_n_p = -4.294548e-04;
84 c_n_r = -8.6047784e-03;
85 c_n_da = -1.95539e-04;
86 c_n_dr = -5.50282873e-03;
87
88 % aerodynamic model; where u,v,w have units ft/sec;
89 c_x = c_x_0 + c_x_u*u + c_x_w*w + (cbar/(2*u))*c_x_q
      *q*(180/pi) + c_x_de*de;
90 c_y = c_y_0 + c_y_v*v + (b/(2*u))*(c_y_p*p + c_y_r*r
     )*(180/pi) + c_y_da*da + c_y_dr*dr;
91 c_z = c_z_0 + c_z_u*u + c_z_w*w + (cbar/(2*u))*c_z_q
      *q*(180/pi) + c_z_de*de;
92 c_l = c_l_0 + c_l_v*v + (b/(2*u))*(c_l_p*p + c_l_r*r
     )*(180/pi) + c_l_da*da + c_l_dr*dr;
93 c_m = c_m_0 + c_m_u*u + c_m_w*w + (cbar/(2*u))*c_m_q
      *q*(180/pi) + c_m_de*de;
94 c_n = c_n_0 + c_n_v*v + (b/(2*u))*(c_n_p*p + c_n_r*r
     )*(180/pi) + c_n_da*da + c_n_dr*dr;
95
96 % external forces
97 f_a_x = qbar*s*c_x;
98 f_a_y = qbar*s*c_y;
99 f_a_z = qbar*s*c_z;
100 f_T_x = T;
101 f_T_y = 0;
102 f_T_z = 0;
103
104 % external moments
105 m_e_x = qbar*s*b*c_l;
106 m_e_y = qbar*s*cbar*c_m;
107 m_e_z = qbar*s*b*c_n;
```

```
108
109 % Solves for the rate change of velocity from force
    equations
110 udot = -g*sin(theta)+(f_a_x+f_T_x)/m+v*r-w*q;
111 vdot = g*sin(phi)*cos(theta)+(f_a_y+f_T_y)/m+w*p-u*r
    ;
112 wdot = g*cos(phi)*cos(theta)+(f_a_z+f_T_z)/m-v*p+u*q
    ;
113 UVW = [udot;vdot;wdot];
114 % Solves for rate change of moments
115 I = [i_xx -i_xy -i_xz; -i_xy i_yy -i_yz; -i_xz -i_yz
    i_zz]; % The "inertia" matrix
116 B = [q*r*(i_yy-i_zz)+(q^2-r^2)*i_yz-p*r*i_xy+p*q*
    i_xz+m_e_x; p*r*(i_zz-i_xx)+(r^2-p^2)*i_xz-p*q*
    i_yz+q*r*i_xy+m_e_y; p*q*(i_xx-i_yy)+(p^2-q^2)*
    i_xy-q*r*i_xz+p*r*i_yz+m_e_z];
117 A = inv(I)*B; % rate of change matrix
118 % Solves for rate change of angles
119 phidot = p+q*sin(phi)*tan(theta)+r*cos(phi)*tan(
    theta);
120 thetadot = q*cos(phi)-r*sin(phi);
121 psidot = (q*sin(phi)+r*cos(phi))*sec(theta);
122 POW = [phidot;thetadot;psidot];
123
124 % Spits out derivatives
125 dx = [UVW;A;POW]; % don't forget to add the semi-
    colon back after checking to make sure all rate
    changes are zero or close enough to the millionth
126 % pause
127 end
```