

The matrix product of A and B is: $\begin{bmatrix} 102 & 53 & 39 & -23 \\ 48 & 8 & 41 & 11 \\ 116 & 26 & 95 & -59 \\ 12 & -23 & 76 & 79 \end{bmatrix}$ (a)

The matrix sum of A and B is: $\begin{bmatrix} 5 & 5 & 7 & 8 \\ 6 & 7 & 3 & 3 \\ 20 & -4 & -6 & 10 \\ 26 & 12 & 9 & -13 \end{bmatrix}$ (a)

The matrix difference of B from A is: $\begin{bmatrix} 1 & 9 & -9 & 2 \\ 2 & -1 & 1 & -1 \\ 4 & -2 & -10 & 8 \\ -10 & 0 & 5 & 5 \end{bmatrix}$ (a)

The matrix transpose of A is: $\begin{bmatrix} 3 & 4 & 12 & 8 \\ 7 & 3 & -3 & 6 \\ -1 & 2 & -8 & 7 \\ 5 & 1 & 9 & -4 \end{bmatrix}$ (b)

The matrix concatenation of of and A B \Rightarrow C is: $\begin{bmatrix} 3 & 7 & -1 & 5 & 2 & -2 & 8 & 3 \\ 4 & 3 & 2 & 1 & 2 & 4 & 1 & 2 \\ 12 & -3 & -8 & 9 & 8 & -1 & 2 & 1 \\ 8 & 6 & 7 & -4 & 18 & 6 & 2 & -9 \end{bmatrix}$ (c)

The matrix inverse of A is: $\begin{bmatrix} -0.0132159 & -0.0910426 & 0.0572687 & 0.0895742 \\ 0.193833 & -0.386931 & -0.00660793 & 0.13069 \\ -0.211454 & 0.876652 & -0.0837004 & -0.23348 \\ -0.105727 & 0.771659 & -0.0418502 & -0.283407 \end{bmatrix}$ (d)

The determinant of A is -1362. (e)

The eigenvectors of A are :

$\begin{bmatrix} -0.0132159 \\ 0.193833 \\ -0.211454 \\ -0.105727 \end{bmatrix}, \begin{bmatrix} -0.0910426 \\ -0.386931 \\ 0.876652 \\ 0.771659 \end{bmatrix}, \begin{bmatrix} 0.0572687 \\ -0.00660793 \\ -0.0837004 \\ -0.0418502 \end{bmatrix}, \begin{bmatrix} 0.0895742 \\ 0.13069 \\ -0.23348 \\ -0.283407 \end{bmatrix}$ (f)

The eigenvalues of A are: $\begin{bmatrix} 12.7825 & 0 & 0 & 0 \\ 0 & -12.0303 & 0 & 0 \\ 0 & 0 & -4.97017 & 0 \\ 0 & 0 & 0 & -1.78202 \end{bmatrix}$ (f)

The rank of matrix A is: 4

(g)

The matrix exponential of A is:

$$\begin{bmatrix} 148948 & 139551 & 29303.1 & 68405.4 \\ 107642 & 100851 & 21176.8 & 49435.4 \\ 143864 & 134788 & 28303 & 66071 \\ 169491 & 158797 & 33344.6 & 77840.2 \end{bmatrix}$$

(h)

The \log_{10} of C(1,2) is: 0.845098

(i)

The matrix X that satisfies $AX = B$ is:

$$\begin{bmatrix} 1.86197 & 0.142438 & 0.0969163 & -0.970631 \\ 1.91336 & -1.14464 & 1.41189 & -1.37518 \\ -3.54185 & 2.61233 & -1.44934 & 3.13656 \\ -4.10426 & 1.6395 & -0.72467 & 3.73495 \end{bmatrix}$$

(j)

The removal of negatives from a results in:

$$\begin{bmatrix} 3 & 7 & 0 & 5 \\ 4 & 3 & 2 & 1 \\ 12 & 0 & 0 & 9 \\ 8 & 6 & 7 & 0 \end{bmatrix}$$

(k)

The transfer functions of the system are

$$\frac{0.25s^2 + 0.25s + 2.5}{1s^2 + 1s + 6}, \frac{0.1s^2 + 1.1s + 0.6}{1s^2 + 1s + 6}, \frac{0s^2 + 1s + 2}{1s^2 + 1s + 6}, \text{ and } \frac{2s^2 + 3s + 6}{1s^2 + 1s + 6}. \quad \textcircled{a}$$

The eigenvalues of the system are $-0.5+2.3979i$, and $-0.5-2.3979i$. \textcircled{b}

The natural frequencies of the system are 2.449, and 2.449. \textcircled{b}

The damping ratios of the system are 0.2041, 0.2041. \textcircled{b}

The 1st poles ($-0.5+2.3979i, -0.5-2.3979i$) are the same as the 2nd poles ($-0.5+2.3979i, -0.5-2.3979i$).

The 1st poles ($-0.5+2.3979i, -0.5-2.3979i$) are the same as the 3rd poles ($-0.5+2.3979i, -0.5-2.3979i$).

The 1st poles ($-0.5+2.3979i, -0.5-2.3979i$) are the same as the 4th poles ($-0.5+2.3979i, -0.5-2.3979i$).

The transfer function poles ($-0.5+2.3979i, -0.5-2.3979i$) are the same as the eigenvalues ($-0.5+2.3979i, -0.5-2.3979i$). \textcircled{c}

The poles of the longitudinal system are $-0.0045932+0.070993i$, $-0.0045932-0.070993i$, $-0.75016+1.2508i$, and $-0.75016-1.2508i$.

The natural frequencies of the longitudinal system are 0.07114, 0.07114, 1.458, and 1.45847.

The damping ratios of the longitudinal system are 0.06456, 0.06456.

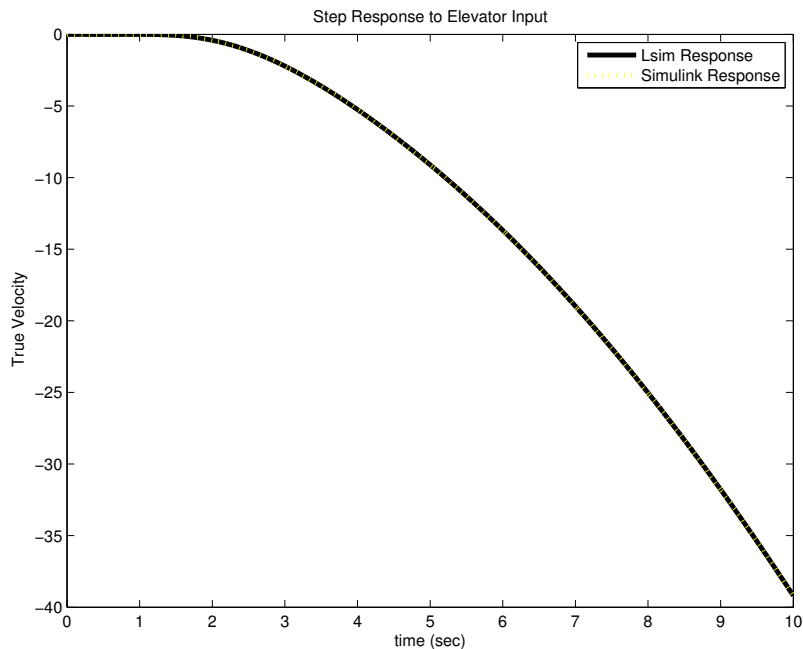
The damping ratios of the longitudinal system are 0.5143, 0.5143.

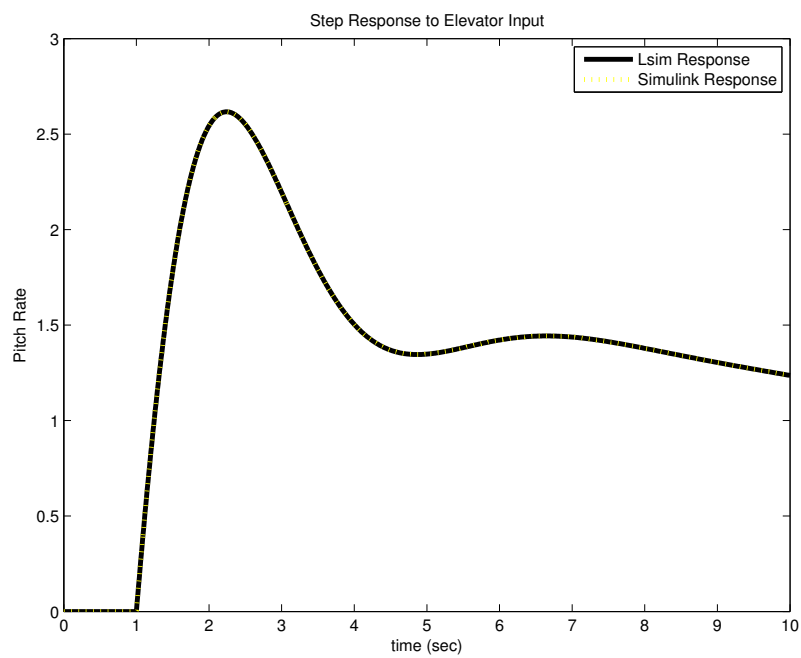
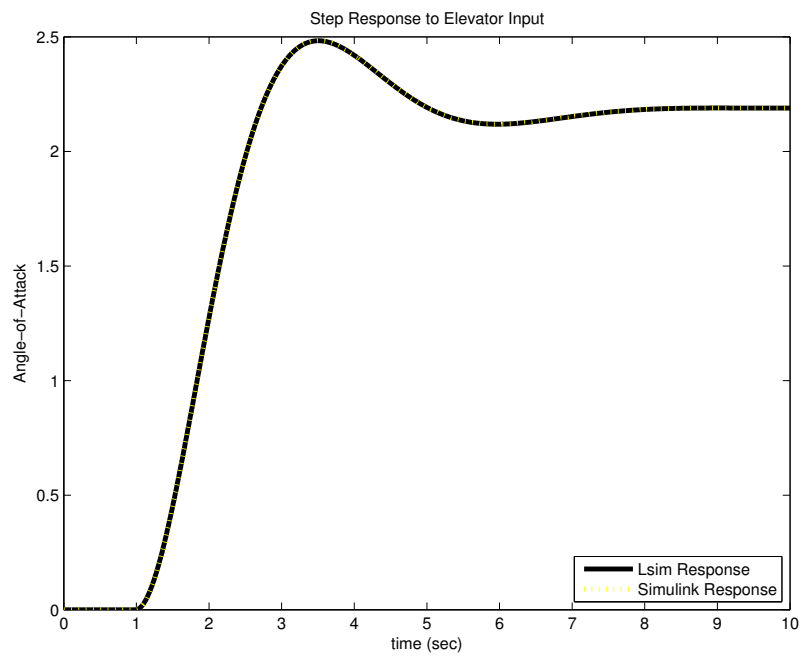
The poles of the lateral system are -0.025551 , $-0.40833+2.75i$, $-0.40833-2.75i$, and -2.8232 .

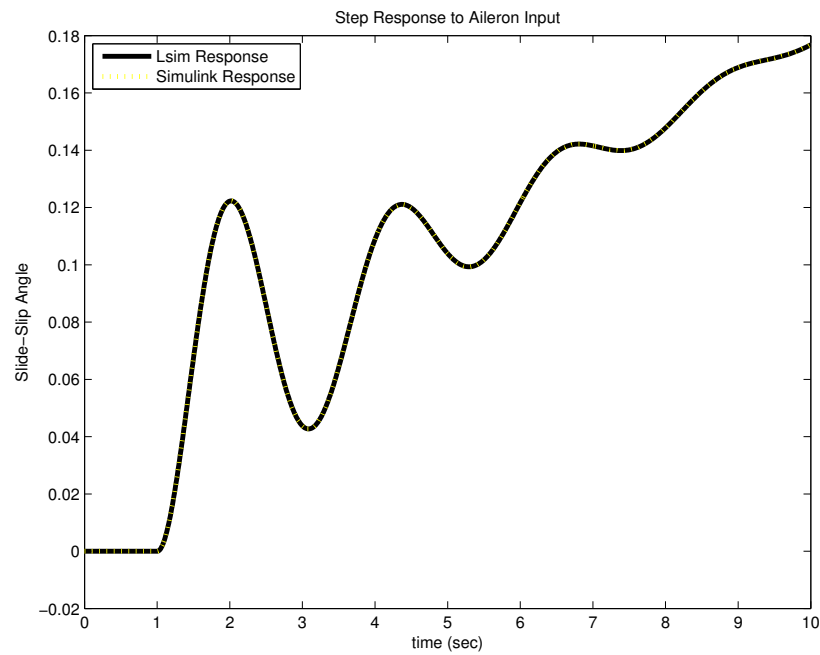
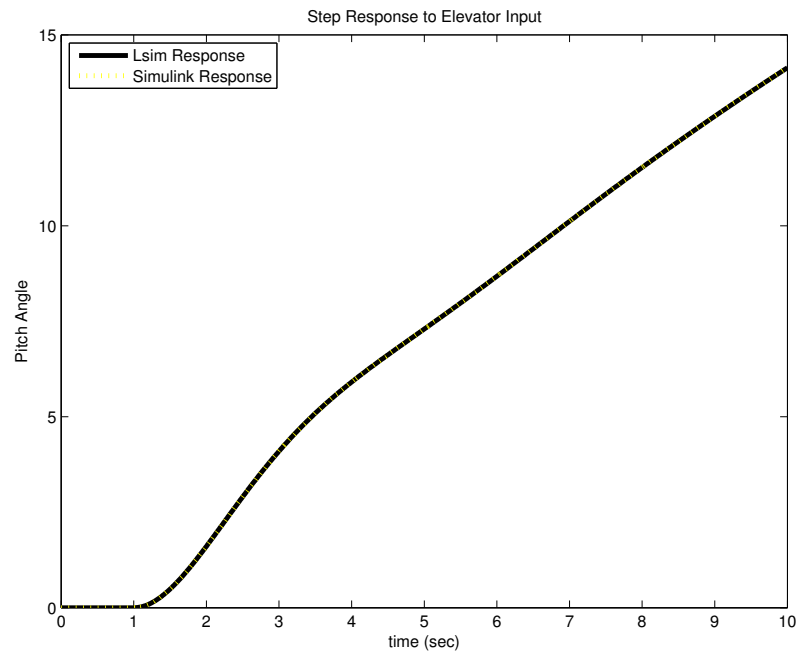
The natural frequencies of the lateral system are 0.02555, 2.78, 2.78, and 2.82318.

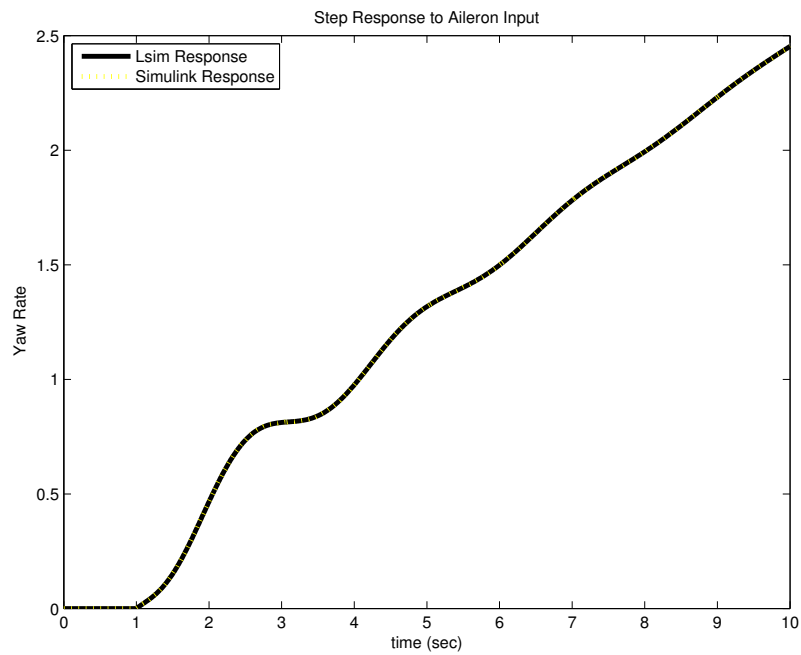
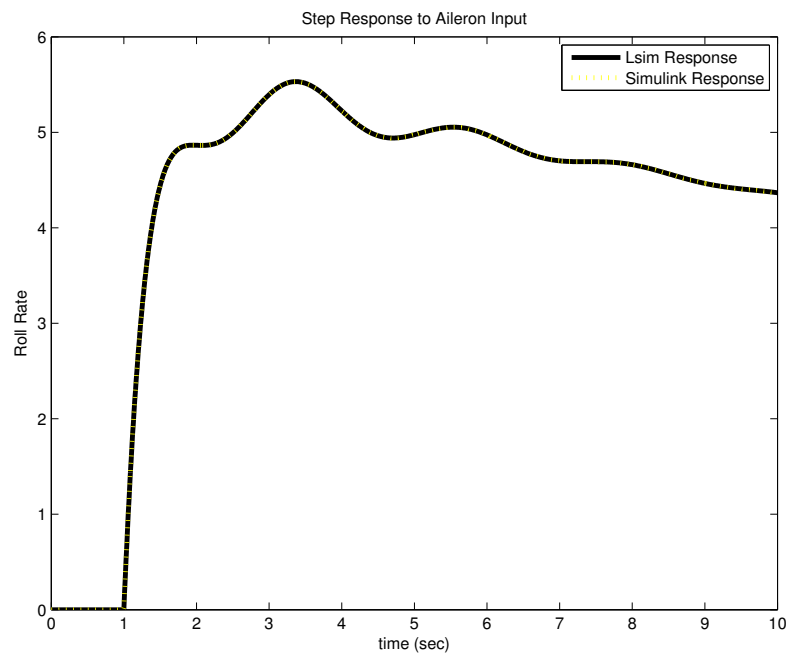
The damping ratios of the lateral system are 1, 0.1469, 0.1469, and 1. (b)

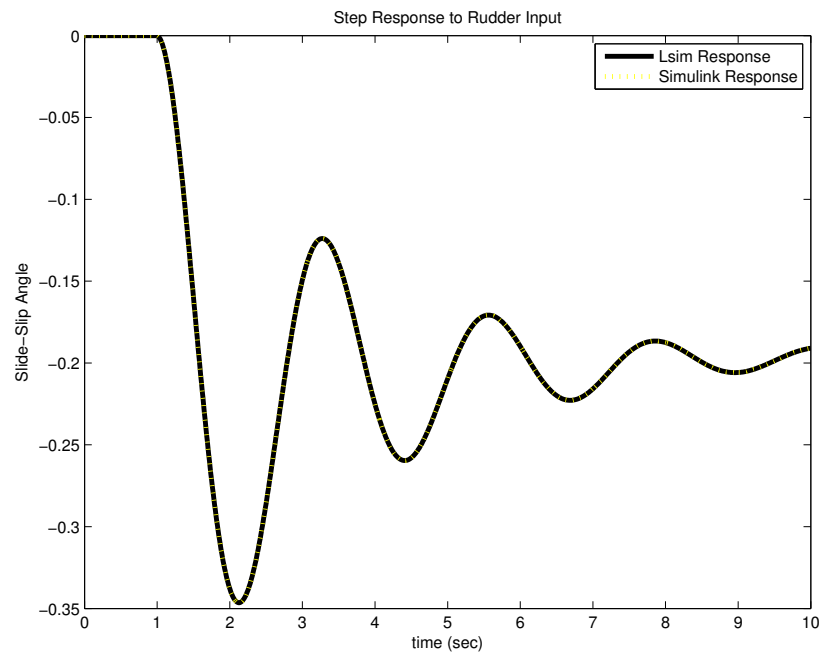
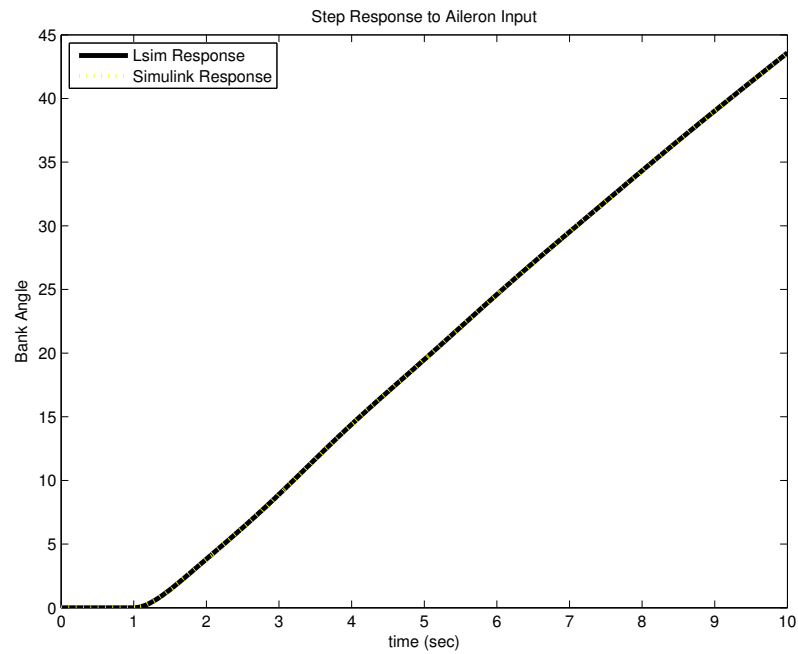
The following all show the same story, that the the lsim results and the simulink results **MATCH!**

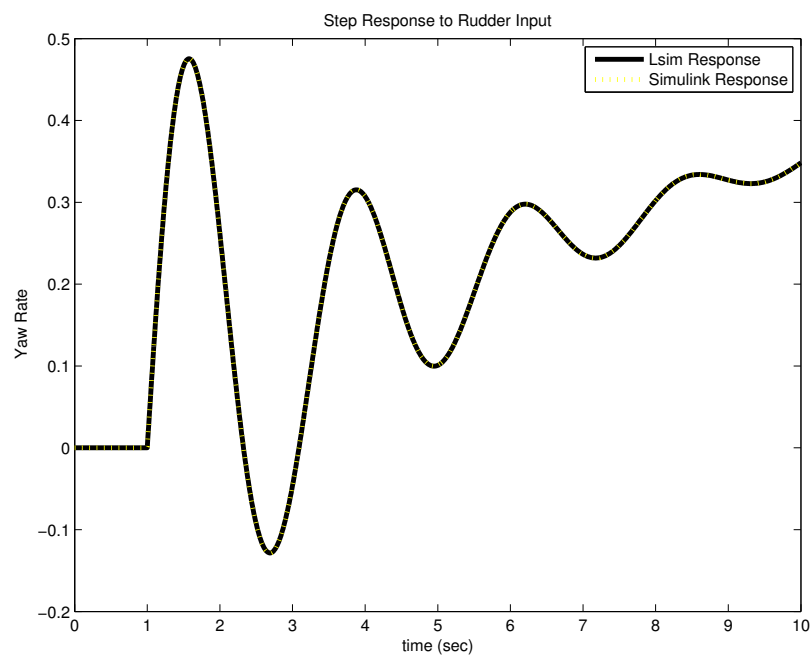
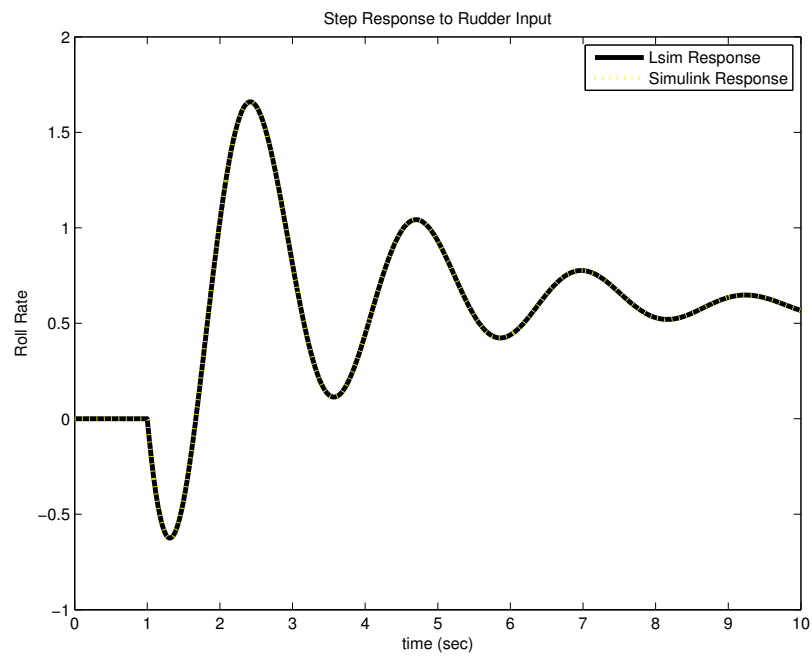












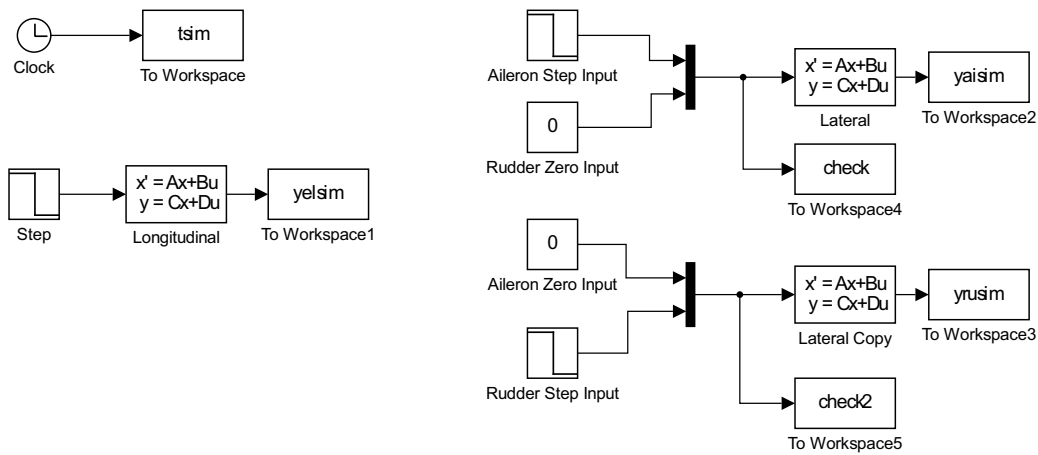
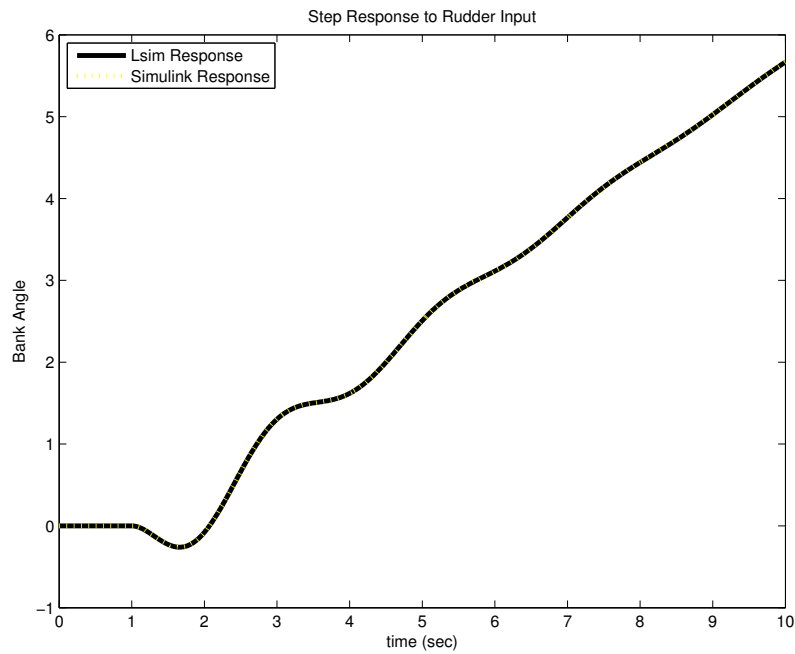


Figure 1: Simulink Diagram

```
1 clear all; close all; clc
2 %Project 1
3 %Problem 1
4 fid = fopen('Problem1.txt','w+');
5 A = [3 7 -1 5; 4 3 2 1; 12 -3 -8 9; 8 6 7 -4];
6 B = [ 2 -2 8 3; 2 4 1 2; 8 -1 2 1; 18 6 2 -9];
7
8 %Part a
9 AxB = A*B;
10 [row,col] = size(AxB);
11 fprintf(fid,'\\noindent The matrix product of A and
    B is: $\\begin{bmatrix}\\n');
12 for i=1:row
13     for j=1:col
14         if j==col
15             if i == row
16                 fprintf(fid,'%g', AxB(i,j));
17             else
18                 fprintf(fid,'%g \\\\' , AxB(i,j));
19             end
20         else
21             fprintf(fid,'%g & ', AxB(i,j));
22         end
23     end
24 end
25 fprintf(fid,'\\n\\end{bmatrix}\\hspace*{\\fill} \\
    circled{a}$\\\\\\n\\\\\\n');
26
27 AplusB = A+B;
28 fprintf(fid,'The matrix sum of A and B is: $\\begin{
    bmatrix}\\n');
29 [row,col] = size(AplusB);
30 for i=1:row
31     for j=1:col
32         if j==col
33             if i == row
34                 fprintf(fid,'%g', AplusB(i,j));
35             else
```

```

36             fprintf(fid, '%g \\\\' , AplusB(i,j));
37         end
38     else
39         fprintf(fid, '%g & ', AplusB(i,j));
40     end
41 end
42 end
43 fprintf(fid, '\\n\\end{bmatrix}\\hspace*{\\fill} \\
    circled{a}$\\\\\\\\n\\\\\\\\n');
44
45 AminusB = A-B;
46 fprintf(fid, 'The matrix difference of B from A is:
    $\\begin{bmatrix}\\n');
47 [row,col] = size(AminusB);
48 for i=1:row
49     for j=1:col
50         if j==col
51             if i == row
52                 fprintf(fid, '%g', AminusB(i,j));
53             else
54                 fprintf(fid, '%g \\\\' , AminusB(i,j))
55                 ;
56             end
57         else
58             fprintf(fid, '%g & ', AminusB(i,j));
59         end
60     end
61 fprintf(fid, '\\n\\end{bmatrix}\\hspace*{\\fill} \\
    circled{a}$\\\\\\\\n\\\\\\\\n');
62
63 %Part b
64 Atrans = transpose(A);
65 fprintf(fid, 'The matrix transpose of A is: $\\begin{
    bmatrix}\\n');
66 [row,col] = size(Atrans);
67 for i=1:row
68     for j=1:col

```

```

69         if j==col
70             if i == row
71                 fprintf(fid,'%g', Atrans(i,j));
72             else
73                 fprintf(fid,'%g \\\\'', Atrans(i,j));
74             end
75         else
76             fprintf(fid,'%g & ', Atrans(i,j));
77         end
78     end
79 end
80 fprintf(fid, '\n\\end{bmatrix}\\hspace*{\\fill} \\
    circled{b}$\\\\\\n\\\\\\n');
81
82 %Part c
83 C = cat(2, A, B);
84 fprintf(fid, 'The matrix concatenation of of and A B
    $\\rightarrow$ C is: $\\begin{bmatrix}\\n');
85 [row,col] = size(C);
86 for i=1:row
87     for j=1:col
88         if j==col
89             if i == row
90                 fprintf(fid,'%g', C(i,j));
91             else
92                 fprintf(fid,'%g \\\\'', C(i,j));
93             end
94         else
95             fprintf(fid,'%g & ', C(i,j));
96         end
97     end
98 end
99 fprintf(fid, '\n\\end{bmatrix}\\hspace*{\\fill} \\
    circled{c}$\\\\\\n\\\\\\n');
100
101 %Part d
102 Ainv = inv(A);
103 fprintf(fid, 'The matrix inverse of A is: $\\begin{

```

```

        bmatrix}\n');
104 [row,col] = size(Ainv);
105 for i=1:row
106     for j=1:col
107         if j==col
108             if i == row
109                 fprintf(fid,'%g', Ainv(i,j));
110             else
111                 fprintf(fid,'%g \\\\'', Ainv(i,j));
112             end
113         else
114             fprintf(fid,'%g & ', Ainv(i,j));
115         end
116     end
117 end
118 fprintf(fid,'\n\\end{bmatrix}\\hspace*{\\fill} \\
        circled{d}$\\\\\\n\\\\\\n');
119 %Part e
120 Adet = det(A);
121 fprintf(fid,'The determinant of A is %g.\\hspace*{\\
        fill} \\circled{e}\\\\\\n\\\\\\n',Adet);
122 %Part f
123 [V,D] = eig(A);
124
125 fprintf(fid,'The eigenvectors of A are :\\\\\\n\\\\\\n
        ');
126 [row,col] = size(V);
127 for j=1:col
128     fprintf(fid,'$\\begin{bmatrix}');
129     for i=1:row
130         if i == row
131             fprintf(fid,'%g', Ainv(i,j));
132         else
133             fprintf(fid,'%g \\\\'', Ainv(i,j));
134         end
135     end
136     if j==col
137         fprintf(fid,'\\end{bmatrix}\\hspace*{\\fill}

```

```

        \\circled{f}$\\\\\\n\\\\\\n ' ');
138     else
139         fprintf(fid, '\\end{bmatrix}$,\\n ');
140     end
141 end
142
143 fprintf(fid, 'The eigenvalues of A are: $\\begin{
    bmatrix}\\n');
144 [row,col] = size(D);
145 for i=1:row
146     for j=1:col
147         if j==col
148             if i == row
149                 fprintf(fid, '%g', D(i,j));
150             else
151                 fprintf(fid, '%g \\\\ ', D(i,j));
152             end
153         else
154             fprintf(fid, '%g & ', D(i,j));
155         end
156     end
157 end
158 fprintf(fid, '\\n\\end{bmatrix}\\hspace*{\\fill} \\
    circled{f}$\\\\\\n\\\\\\n');
159
160 %Part g
161 Arank = rank(A);
162 fprintf(fid, 'The rank of matrix A is: %g\\hspace*{\\
    fill} \\circled{g}\\\\\\n\\\\\\n', Arank);
163
164 %Part h
165 Aexpm = expm(A);
166 fprintf(fid, 'The matrix exponential of A is: $\\
    begin{bmatrix}\\n');
167 [row,col] = size(Aexpm);
168 for i=1:row
169     for j=1:col
170         if j==col

```

```

171         if i == row
172             fprintf(fid, '%g', Aexpm(i,j));
173         else
174             fprintf(fid, '%g \\\\'', Aexpm(i,j));
175         end
176     else
177         fprintf(fid, '%g & ', Aexpm(i,j));
178     end
179 end
180 end
181 fprintf(fid, '\n\\end{bmatrix}\\hspace*{\\fill} \\
        circled{h}$\\\\\\n\\\\\\n');
182
183 %Part i
184 Clog10 = log10(C(1,2));
185 fprintf(fid, 'The log$_{10}$ of C(1,2) is: %g\\hspace
        *{\\fill} \\circled{i}\\\\\\n\\\\\\n', Clog10);
186
187 %Part j
188 X = A\\B;
189 fprintf(fid, 'The matrix X that satisfies AX = B is:
        \\\\'\\n$\\begin{bmatrix}\\n');
190 [row,col] = size(X);
191 for i=1:row
192     for j=1:col
193         if j==col
194             if i == row
195                 fprintf(fid, '%g', X(i,j));
196             else
197                 fprintf(fid, '%g \\\\'', X(i,j));
198             end
199         else
200             fprintf(fid, '%g & ', X(i,j));
201         end
202     end
203 end
204 fprintf(fid, '\n\\end{bmatrix}\\hspace*{\\fill} \\
        circled{j}$\\\\\\n\\\\\\n');

```



```

205
206 %Part k
207 Ar = A;
208 for i = 1:4
209     for j = 1:4
210         if A(i,j)< 0
211             Ar(i,j) = 0;
212         end
213     end
214 end
215 fprintf(fid,'The removal of negatives from a results
        in: $\\begin{bmatrix}\\n');
216 [row,col] = size(Ar);
217 for i=1:row
218     for j=1:col
219         if j==col
220             if i == row
221                 fprintf(fid,'%g', Ar(i,j));
222             else
223                 fprintf(fid,'%g \\\\'', Ar(i,j));
224             end
225         else
226             fprintf(fid,'%g & ', Ar(i,j));
227         end
228     end
229 end
230 fprintf(fid,'\\n\\end{bmatrix}\\hspace*{\\fill} \\
        circled{k}$\\\\\\n\\\\\\n');
231 fclose(fid);
232
233
234 %Problem 2
235 fid = fopen('Problem2.txt','w+');
236 A2 = [0 1; -6 -1]; B2 = [0 1; 1 1];
237 C2 = [1 0; 0 1]; D2 = [0.25 0; 0.1 2];
238 [num1,den] = ss2tf(A2,B2,C2,D2,1);
239 [num2,den] = ss2tf(A2,B2,C2,D2,2);
240 fprintf(fid,'The transfer functions of the system

```

```

are\\\\\\n\\\\\\ $\\dfrac{\\%gs^2+\\%gs+\\%g}{\\%gs^2+\\%gs+\\%g}
\\$, $\\dfrac{\\%gs^2+\\%gs+\\%g}{\\%gs^2+\\%gs+\\%g}$, $\\dfrac{\\%gs^2+\\%gs+\\%g}{\\%gs^2+\\%gs+\\%g}$, and $\\dfrac{\\%gs^2+\\%gs+\\%g}{\\%gs^2+\\%gs+\\%g}$.\\\hspace*{\\fill}
\\circled{a}\\\\\\n\\\\\\n', num1(1,:), den, num1(2,:),
, den, num2(1,:), den, num2(2,:), den);
241 E = eig(A2);
242 [Wn,Z] = damp(ss(A2,B2,C2,D2));
243
244 fprintf(fid,'The eigenvalues of the system are %s,
and %s.\\\hspace*{\\fill} \\circled{b}\\\\\\n The
natural frequencies of the system are %.4g, and
%.4g.\\\hspace*{\\fill} \\circled{b}\\\\\\n The
damping ratios of the system are %.4g, %.4g.\\\
hspace*{\\fill} \\circled{b}\\\\\\n', num2str(E(1))
, num2str(E(2)), Wn, Z);
245 p1(:,1) = pole(tf(num1(1,:),den));
246 p1(:,2) = pole(tf(num1(2,:),den));
247 p1(:,3) = pole(tf(num2(1,:),den));
248 p1(:,4) = pole(tf(num2(2,:),den));
249 nome = ['st'; 'nd'; 'rd'; 'th'];
250 n = 0;
251 for i = 1:4
252     for j = 2:4
253         if p1(:,i) == p1(:,j)
254             fprintf(fid,'The %g%s poles (%s,%s) are
the same as the %g%s poles (%s,%s)
.\\\\\n', i, nome(i,:), num2str(p1(1,1))
, num2str(p1(2,1)), j, nome(j,:), num2str
(p1(1,2)), num2str(p1(2,2)));
255             n = n+1;
256         end
257         if n == 3
258             fprintf(fid,'The transfer function poles
(%s,%s) are the same as the
eigenvalues (%s,%s).\\\hspace*{\\fill}
\\circled{c}\\\\\\n', num2str(p1(1,1))
, num2str(p1(2,1)), num2str(E(1)),

```

```
                num2str(E(2)));
259             break;
260         end
261     end
262     if n == 3
263         break;
264     end
265 end
266 fclose(fid);
267
268 %Problem 3
269 fid = fopen('Problem3.txt','w+'); %creates/opens and
    overwrites text if file exists already
270 ALo = [-0.0111 -0.0788 -0.0033 -0.5615; -0.0092
    -0.7531 0.9951 0; 0.0062 -1.5765 -0.7453 0; 0 0 1
    0];
271 BLo = [0.0721; -0.1178; -9.0991; 0];
272 CLo = eye(4); %Identity Matrix
273 DLo = zeros(4,1); %Zero Column Vector
274
275 ALa = [-0.2316 0.0633 -0.9956 0.0510; -29.4924
    -3.0169 0.0201 0; 6.2346 -0.0274 -0.4169 0; 0 1
    0.0631 0];
276 BLa = [0.0052 0.0310; -36.4909 8.1090; -0.4916
    -2.8274; 0 0];
277 CLa = eye(4); %Identity Matrix
278 DLa = zeros(4,2);
279
280 %Part a
281 [WnLo,ZLo,PLo] = damp(ss(ALo,BLo,CLo,DLo));
282 fprintf(fid,'The poles of the longitudinal system
    are %s, %s, %s, and %s.\\n',num2str(PLo(1)),
    num2str(PLo(2)),num2str(PLo(3)),num2str(PLo(4)));
283 fprintf(fid,'The natural frequencies of the
    longitudinal system are %.4g, %.4g, %.4g, and %4
    g.\\n',WnLo);
284 fprintf(fid,'The damping ratios of the longitudinal
    system are %.4g, %.4g.\\n',ZLo);
```

```
285 [WnLa,ZLa,PLa] = damp(ss(ALa,BLa,CLa,DLa));
286 fprintf(fid,'The poles of the lateral system are %s,
      %s, %s, and %s.\\n',num2str(PLa(1)),num2str(
      PLa(2)),num2str(PLa(3)),num2str(PLa(4)));
287 fprintf(fid,'The natural frequencies of the lateral
      system are %.4g, %.4g, %.4g, and %4g.\\n',
      WnLa);
288 fprintf(fid,'The damping ratios of the lateral
      system are %.4g, %.4g, %.4g, and %.4g.\\hspace
      *{\\fill} \\circled{b}\\n',ZLa);
289
290 t0 = 0:0.01:10; %time sample length and step size
291 uel = cat(1,zeros(100,1),-0.5*ones(901,1)); %-0.5
      degree elevator step input at 1 second
292 uai = cat(2,uel,zeros(1001,1)); %-0.5 degree aileron
      step input at 1 second
293 uru = cat(2,zeros(1001,1),uel); %-0.5 degree rudder
      step input at 1 second
294
295 [yel,t] = lsim(ss(ALo,BLo,CLo,DLo),uel,t0);
296 %Where the First Column is True Velocity,
297       %Second Column is Angle-of-Attack
298       %Third Column is Pitch Rate
299       %Fourth Column is Pitch Angle
300
301 [yai,t] = lsim(ss(ALa,BLa,CLa,DLa),uai,t0);
302 [yru,t] = lsim(ss(ALa,BLa,CLa,DLa),uru,t0);
303 %Where the First Column is Slide-Slip Angle,
304       %Second Column is Roll Rate
305       %Third Column is Yaw Rate
306       %Fourth Column is Bank Angle
307
308 sim('Problem3.slx')%Run the Simulink Version
309 q = 1;
310 figure(q)
311 plot(t,yel(:,1),'k',tsim,yelsim(:,1),':y','LineWidth
      ',3.0)
312 title('Step Response to Elevator Input');
```

```
313 ylabel('True Velocity');
314 xlabel('time (sec)');
315 legend('Lsim Response','Simulink Response','Location
      ','NorthEast');
316 q = q + 1;
317 figure(q)
318 plot(t,yel(:,2),'k',tsim,yelsim(:,2),':y','LineWidth
      ',3.0)
319 title('Step Response to Elevator Input');
320 ylabel('Angle-of-Attack');
321 xlabel('time (sec)');
322 legend('Lsim Response','Simulink Response','Location
      ','SouthEast');
323 q = q + 1;
324 figure(q)
325 plot(t,yel(:,3),'k',tsim,yelsim(:,3),':y','LineWidth
      ',3.0)
326 title('Step Response to Elevator Input');
327 ylabel('Pitch Rate');
328 xlabel('time (sec)');
329 legend('Lsim Response','Simulink Response','Location
      ','NorthEast');
330 q = q + 1;
331 figure(q)
332 plot(t,yel(:,4),'k',tsim,yelsim(:,4),':y','LineWidth
      ',3.0)
333 title('Step Response to Elevator Input');
334 ylabel('Pitch Angle');
335 xlabel('time (sec)');
336 legend('Lsim Response','Simulink Response','Location
      ','NorthWest');
337 q = q + 1;
338
339 figure(q)
340 plot(t,yai(:,1),'k',tsim,yaisim(:,1),':y','LineWidth
      ',3.0)
341 title('Step Response to Aileron Input');
342 ylabel('Slide-Slip Angle');
```

```
343 xlabel('time (sec)');
344 legend('Lsim Response','Simulink Response','Location
      ','NorthWest');
345 q = q + 1;
346 figure(q)
347 plot(t,yai(:,2),'k',tsim,yaisim(:,2),':y','LineWidth
      ',3.0)
348 title('Step Response to Aileron Input');
349 ylabel('Roll Rate');
350 xlabel('time (sec)');
351 legend('Lsim Response','Simulink Response','Location
      ','NorthEast');
352 q = q + 1;
353 figure(q)
354 plot(t,yai(:,3),'k',tsim,yaisim(:,3),':y','LineWidth
      ',3.0)
355 title('Step Response to Aileron Input');
356 ylabel('Yaw Rate');
357 xlabel('time (sec)');
358 legend('Lsim Response','Simulink Response','Location
      ','NorthWest');
359 q = q + 1;
360 figure(q)
361 plot(t,yai(:,4),'k',tsim,yaisim(:,4),':y','LineWidth
      ',3.0)
362 title('Step Response to Aileron Input');
363 ylabel('Bank Angle');
364 xlabel('time (sec)');
365 legend('Lsim Response','Simulink Response','Location
      ','NorthWest');
366 q = q + 1;
367
368 figure(q)
369 plot(t,yru(:,1),'k',tsim,yrusim(:,1),':y','LineWidth
      ',3.0)
370 title('Step Response to Rudder Input');
371 ylabel('Slide-Slip Angle');
372 xlabel('time (sec)');
```

```

373 legend('Lsim Response','Simulink Response','Location
      ','NorthEast');
374 q = q + 1;
375 figure(q)
376 plot(t,yru(:,2),'k',tsim,yrusim(:,2),':y','LineWidth
      ',3.0)
377 title('Step Response to Rudder Input');
378 ylabel('Roll Rate');
379 xlabel('time (sec)');
380 legend('Lsim Response','Simulink Response','Location
      ','NorthEast');
381 q = q + 1;
382 figure(q)
383 plot(t,yru(:,3),'k',tsim,yrusim(:,3),':y','LineWidth
      ',3.0)
384 title('Step Response to Rudder Input');
385 ylabel('Yaw Rate');
386 xlabel('time (sec)');
387 legend('Lsim Response','Simulink Response','Location
      ','NorthEast');
388 q = q + 1;
389 figure(12)
390 plot(t,yru(:,4),'k',tsim,yrusim(:,4),':y','LineWidth
      ',3.0)
391 title('Step Response to Rudder Input');
392 ylabel('Bank Angle');
393 xlabel('time (sec)');
394 legend('Lsim Response','Simulink Response','Location
      ','NorthWest');
395 fprintf(fid,'\\n\\n\\n\\nThe following all show the same
      story, that the the lsim results and the simulink
      results \\textbf{MATCH!}');
396 for i=1:q
397     name = cat(2,'Figure_',num2str(i));
398     print(figure(i),'-depsc',name);
399     fprintf(fid,'\\begin{figure}[H]\\ncentering\\
        includegraphics[keepaspectratio=true,height
            =.45\\textheight,width=1\\textwidth]{%s.eps

```

```
        }\\end{figure}\\n',name);
400 end
401 print('-sProblem3','-depsc','simdiag');
402 fprintf(fid,'\\begin{figure}[H]\\centering\\
        includegraphics[keepaspectratio=true,height=.45\\
        textheight,width=1\\textwidth]{simdiag.eps}\\\\\\
        caption{Simulink Diagram}\\end{figure}\\n');
403 fclose(fid); %close open file with fileID 'fid'
404 %Report Generator
405 % import mlreportgen.dom.*;
406 % report = Document('today','docx');
407 % append(report, ['Today is ', date, '.']);
408 % Paragraph('Chapter 1. ');
409 % close(report);
410 % rptview(report.OutputPath);
```