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

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

The eigenvalues of the system are -0.5+2.3979i, and -0.5-2.3979i. (b)

The natural frequencies of the system are 2.449, and 2.449. (b)

The damping ratios of the system are 0.2041, 0.2041. (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).

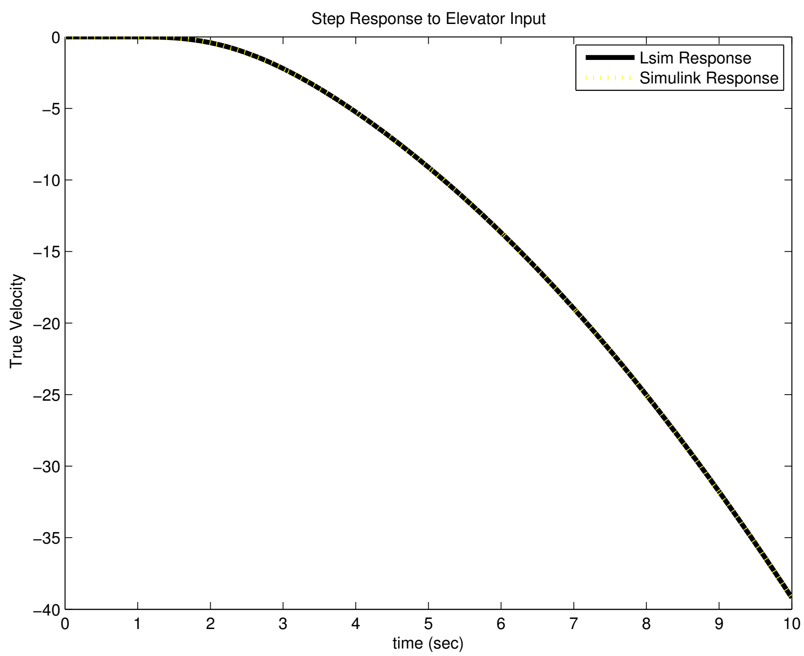
The 1st poles (-0.5+2.3979i,-0.5-2.3979i) are the same as the 3rd poles (-0.5+2.3979i,-0.5-2.3979i).

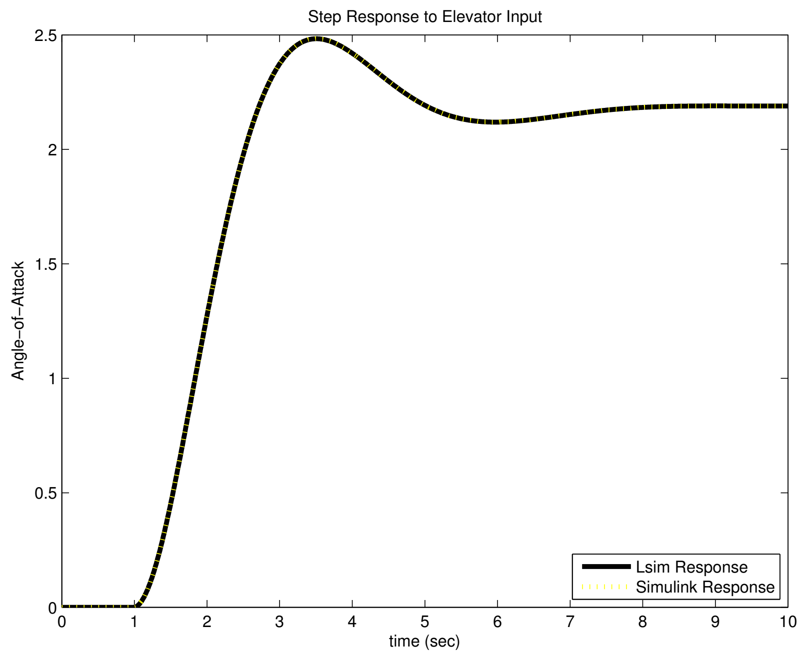
The 1st poles (-0.5+2.3979i,-0.5-2.3979i) are the same as the 4th poles (-0.5+2.3979i,-0.5-2.3979i).

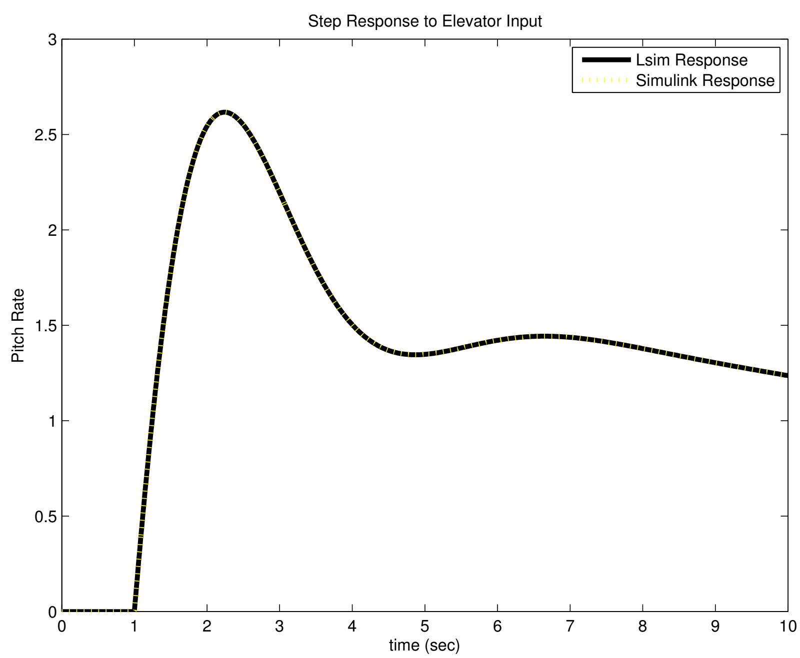
The transfer function poles (-0.5+2.3979i,-0.5-2.3979i) are the same as the eigenvalues (-0.5+2.3979i,-0.5-2.3979i). (c)

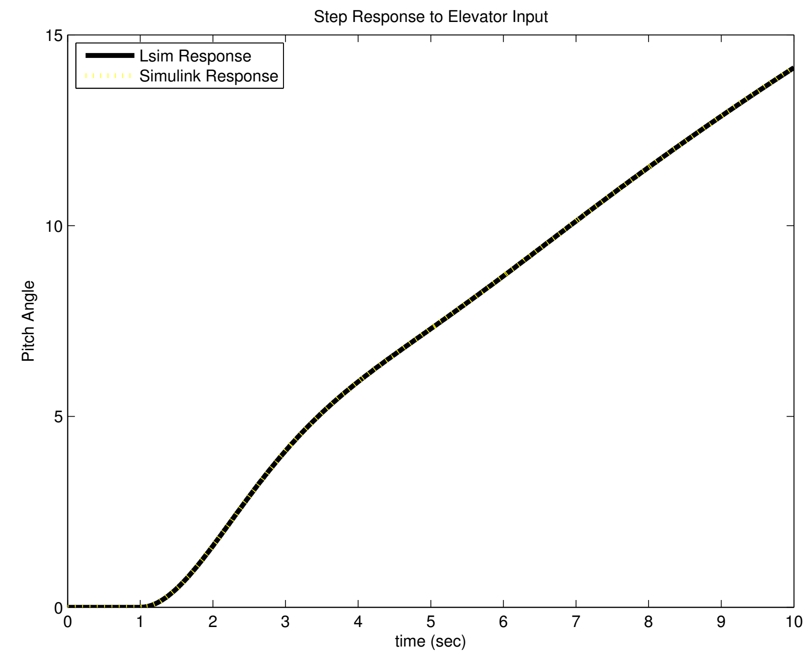
The poles of the longitudinal system are -0.0045932+0.070993i, -0.0045932-0.070993i, -0.75016+1.2508i, and -0.75016-1.2508i.
The natural frequencies of the longitudinal system are 0.07114, 0.07114, 1.458, and 1.45847.
The damping ratios of the longitudinal system are 0.06456, 0.06456.
The damping ratios 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

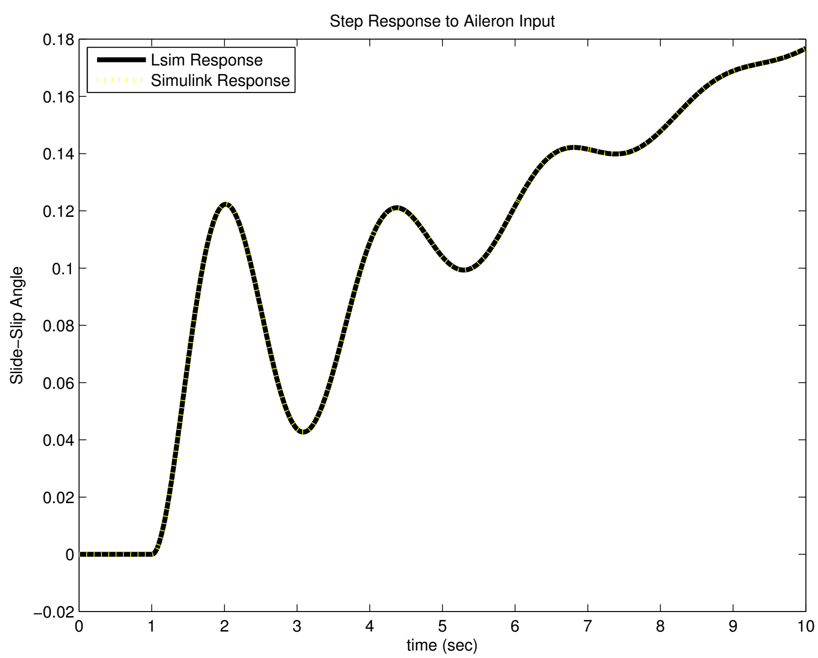
The following all show the same story, that the the lsim results and the simulink results **MATCH!**

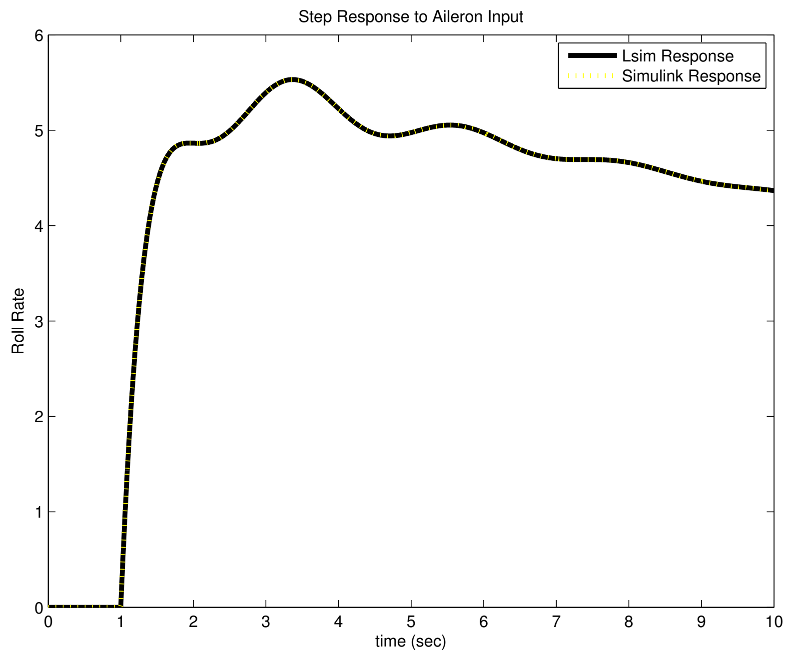


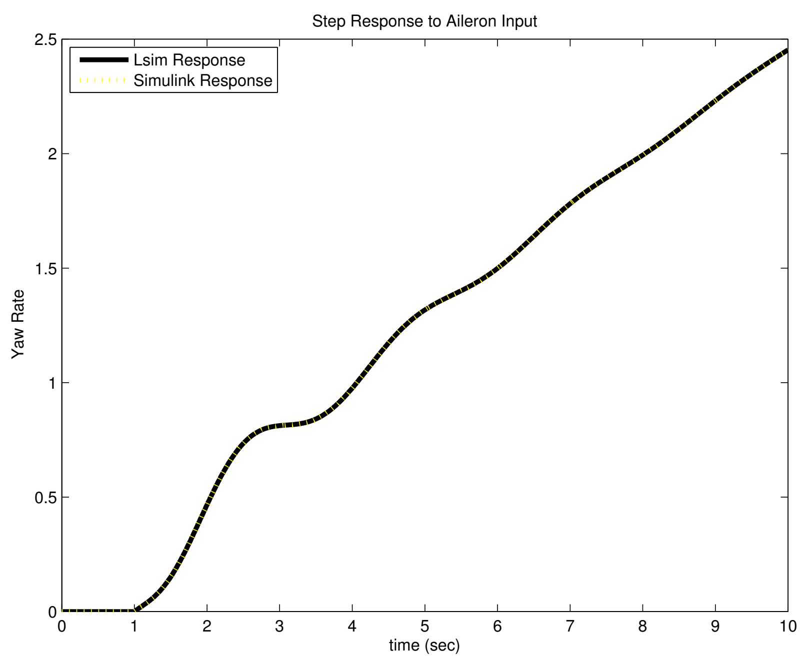

Step Response to Elevator Input
Angle–of–Attack
time (sec)
Lsim Response
Simulink Response
0
0.5
1
1.5
2
2.5
0 1 2 3 4 5 6 7 8 9 10

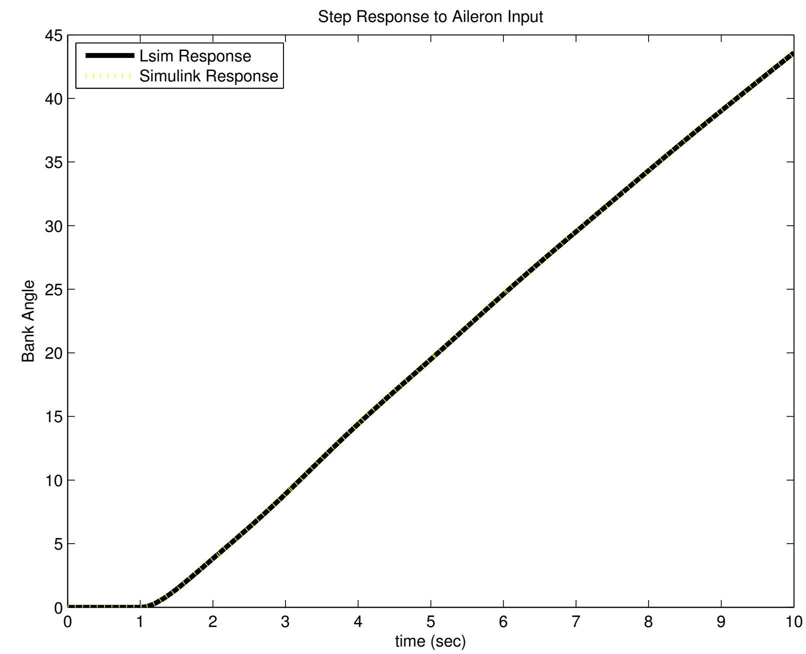

Step Response to Elevator Input
Pitch Rate
time (sec)
Lsim Response
Simulink Response
0
0.5
1
1.5
2
2.5
3
0 1 2 3 4 5 6 7 8 9 10

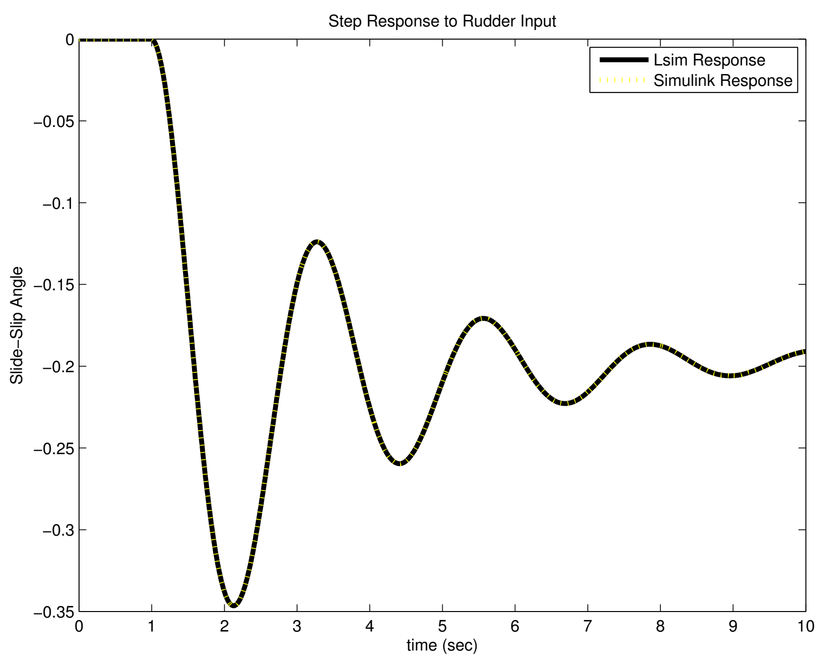

Step Response to Elevator Input
Lsim Response
Simulink Response
Pitch Angle
time (sec)
0
5
10
15
0
1
2
3
4
5
6
7
8
9
10

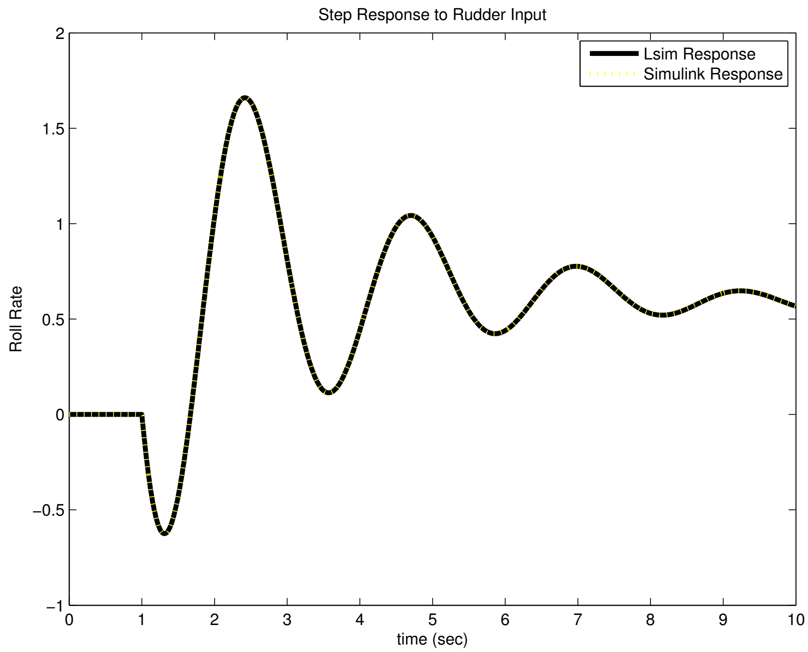

Step Response to Aileron Input
Lsim Response
Simulink Response
Slide−Slip Angle
time (sec)
−0.02
0
0.02
0.04
0.06
0.08
0.1
0.12
0.14
0.16
0.18
0
1
2
3
4
5
6
7
8
9
10

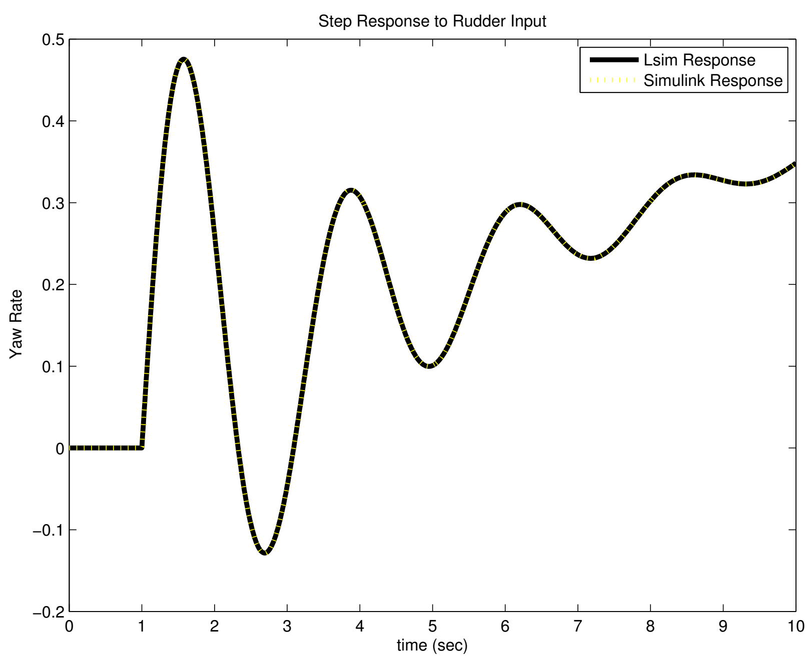

Step Response to Aileron Input
Lsim Response
Simulink Response
Roll Rate
time (sec)
0
1
2
3
4
5
6
7
8
9
10

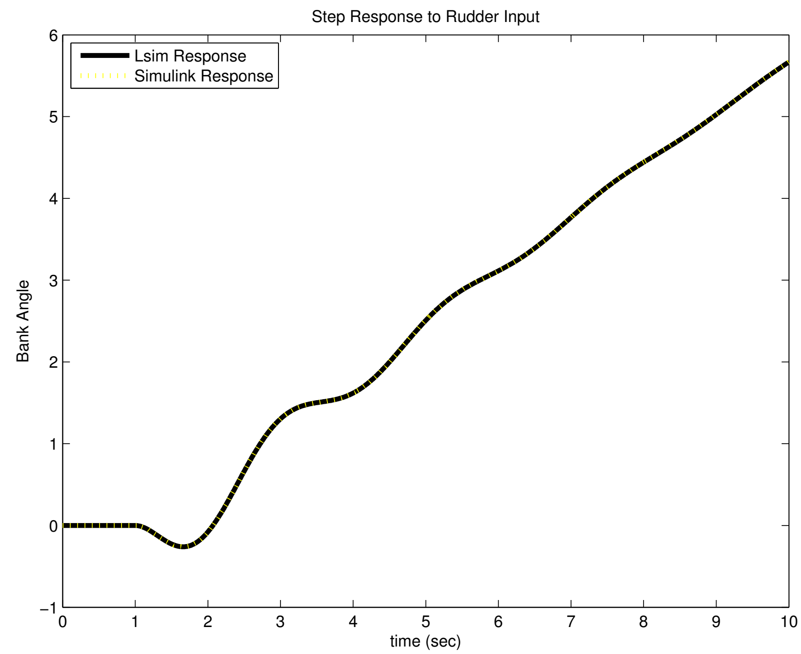

Step Response to Aileron Input
Lsim Response
Simulink Response
Yaw Rate
time (sec)
0
0.5
1
1.5
2
2.5
3
4
5
6
7
8
9
10


Step Response to Aileron Input
Lsim Response
Simulink Response
Bank Angle
time (sec)
0
5
10
15
20
25
30
35
40
45
0
1
2
3
4
5
6
7
8
9
10


Step Response to Rudder Input
Lsim Response
Simulink Response
Slide−Slip Angle
time (sec)
0
−0.05
−0.1
−0.15
−0.2
−0.25
−0.3
−0.35
0
1
2
3
4
5
6
7
8
9
10


Step Response to Rudder Input
Lsim Response
Simulink Response
Roll Rate
time (sec)
2
1.5
1
0.5
0
−0.5
−1
0
1
2
3
4
5
6
7
8
9
10


Step Response to Rudder Input
Lsim Response
Simulink Response
Yaw Rate
time (sec)
0.5
0.4
0.3
0.2
0.1
0
−0.1
−0.2
0
1
2
3
4
5
6
7
8
9
10



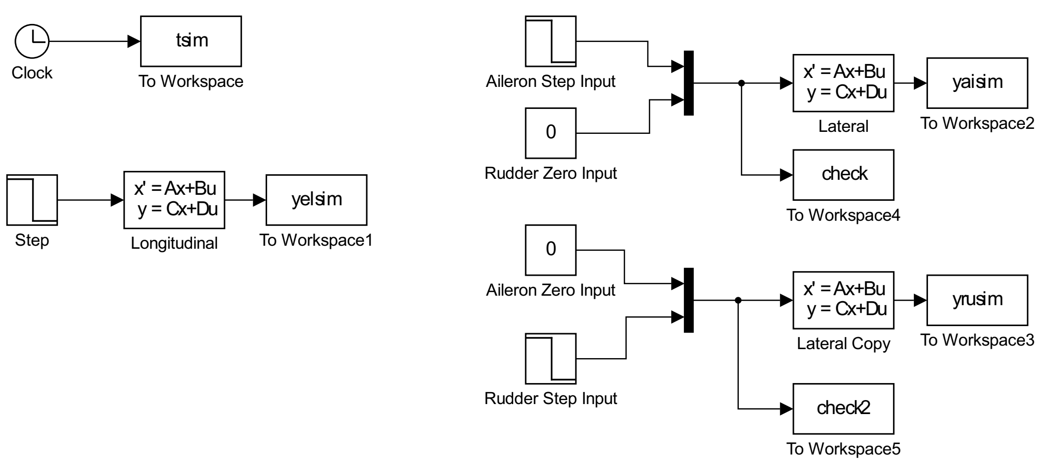


Figure 1: Simulink Diagram

```
clear all; close all; clc
%Project 1
%Problem 1
fid = fopen('Problem1.txt','w+');
A = [3 7 -1 5; 4 3 2 1; 12 -3 -8 9; 8 6 7 -4];
B = [ 2 -2 8 3; 2 4 1 2; 8 -1 2 1; 18 6 2 -9];

%Part a
AxB = A*B;
[row,col] = size(AxB);
fprintf(fid,'\\noindent The matrix product of A and
    B is: $\\begin{bmatrix}\n');
for i=1:row
    for j=1:col
        if j==col
            if i == row
                fprintf(fid,'%g', AxB(i,j));
            else
                fprintf(fid,'%g \\\\', AxB(i,j));
            end
        else
            fprintf(fid,'%g & ', AxB(i,j));
        end
    end
end
fprintf(fid,'\n\\end{bmatrix}\\hspace*{\\fill} \\
    circled{a}$\\\\\n\\\\\n');

AplusB = A+B;
fprintf(fid,'The matrix sum of A and B is: $\\begin{
    bmatrix}\n');
[row,col] = size(AplusB);
for i=1:row
    for j=1:col
        if j==col
            if i == row
                fprintf(fid,'%g', AplusB(i,j));
            else
```

```
                fprintf(fid,'%g \\\\', AplusB(i,j));
            end
        else
            fprintf(fid,'%g & ', AplusB(i,j));
        end
    end
end
fprintf(fid,'\n\\end{bmatrix}\\hspace*{\\fill} \\circled{a}$\\\\\n\\\\\n');

AminusB = A-B;
fprintf(fid,'The matrix difference of B from A is: $\\begin{bmatrix}\n');
[row,col] = size(AminusB);
for i=1:row
    for j=1:col
        if j==col
            if i == row
                fprintf(fid,'%g', AminusB(i,j));
            else
                fprintf(fid,'%g \\\\', AminusB(i,j));
            end
        else
            fprintf(fid,'%g & ', AminusB(i,j));
        end
    end
end
fprintf(fid,'\n\\end{bmatrix}\\hspace*{\\fill} \\circled{a}$\\\\\n\\\\\n');

%Part b
Atrans = transpose(A);
fprintf(fid,'The matrix transpose of A is: $\\begin{bmatrix}\n');
[row,col] = size(Atrans);
for i=1:row
    for j=1:col
```

```
            if j==col
                if i == row
                    fprintf(fid,'%g', Atrans(i,j));
                else
                    fprintf(fid,'%g \\\\', Atrans(i,j));
                end
            else
                fprintf(fid,'%g & ', Atrans(i,j));
            end
        end
end
fprintf(fid,'\n\\end{bmatrix}\\hspace*{\\fill} \\
    circled{b}$\\\\\n\\\\\n');

%Part c
C = cat(2, A, B);
fprintf(fid,'The matrix concatenation of of and A B
    $\\Rightarrow$ C is: $\\begin{bmatrix}\n');
[row,col] = size(C);
for i=1:row
    for j=1:col
        if j==col
            if i == row
                fprintf(fid,'%g', C(i,j));
            else
                fprintf(fid,'%g \\\\', C(i,j));
            end
        else
            fprintf(fid,'%g & ', C(i,j));
        end
    end
end
fprintf(fid,'\n\\end{bmatrix}\\hspace*{\\fill} \\
    circled{c}$\\\\\n\\\\\n');

%Part d
Ainv = inv(A);
fprintf(fid,'The matrix inverse of A is: $\\begin{
```

```
    bmatrix}\n');
[row,col] = size(Ainv);
for i=1:row
    for j=1:col
        if j==col
            if i == row
                fprintf(fid,'%g', Ainv(i,j));
            else
                fprintf(fid,'%g \\\\', Ainv(i,j));
            end
        else
            fprintf(fid,'%g & ', Ainv(i,j));
        end
    end
end
fprintf(fid,'\n\\end{bmatrix}\\hspace*{\\fill} \\
    circled{d}$\\\\n\\\\n');
%Part e
Adet = det(A);
fprintf(fid,'The determinant of A is %g.\\hspace*{\\
    fill} \\circled{e}\\\\n\\\\n',Adet);
%Part f
[V,D] = eig(A);

fprintf(fid,'The eigenvectors of A are :\\\\n\\\\n
    ');
[row,col] = size(V);
for j=1:col
    fprintf(fid,'$\\begin{bmatrix}');
    for i=1:row
        if i == row
        fprintf(fid,'%g', Ainv(i,j));
        else
            fprintf(fid,'%g \\\\', Ainv(i,j));
        end
    end
    if j==col
        fprintf(fid,'\\end{bmatrix}\\hspace*{\\fill}
```

```
\\circled{f}$\\\\n\\\\n ');
    else
        fprintf(fid,'\\end{bmatrix}$,\n ');
    end
end

fprintf(fid,'The eigenvalues of A are: $\\begin{bmatrix}\n');
[row,col] = size(D);
for i=1:row
    for j=1:col
        if j==col
            if i == row
                fprintf(fid,'%g', D(i,j));
            else
                fprintf(fid,'%g \\\\', D(i,j));
            end
        else
            fprintf(fid,'%g & ', D(i,j));
        end
    end
end
fprintf(fid,'\n\\end{bmatrix}\\hspace*{\\fill} \\circled{f}$\\\\n\\\\n');

%Part g
Arank = rank(A);
fprintf(fid,'The rank of matrix A is: %g\\hspace*{\\fill} \\circled{g}\\\\n\\\\n',Arank);

%Part h
Aexpm = expm(A);
fprintf(fid,'The matrix exponential of A is: $\\begin{bmatrix}\n');
[row,col] = size(Aexpm);
for i=1:row
    for j=1:col
        if j==col
```

```
            if i == row
                fprintf(fid,'%g', Aexpm(i,j));
            else
                fprintf(fid,'%g \\\\', Aexpm(i,j));
            end
        else
            fprintf(fid,'%g & ', Aexpm(i,j));
        end
    end
end
fprintf(fid,'\n\\end{bmatrix}\\hspace*{\\fill} \\circled{h}$\\\\n\\\\n');

%Part i
Clog10 = log10(C(1,2));
fprintf(fid,'The log$_{10}$ of C(1,2) is: %g\\hspace*{\\fill} \\circled{i}\\\\n\\\\n',Clog10);

%Part j
X = A\B;
fprintf(fid,'The matrix X that satisfies AX = B is: \\\\n\n$\\begin{bmatrix}\n');
[row,col] = size(X);
for i=1:row
    for j=1:col
        if j==col
            if i == row
                fprintf(fid,'%g', X(i,j));
            else
                fprintf(fid,'%g \\\\', X(i,j));
            end
        else
            fprintf(fid,'%g & ', X(i,j));
        end
    end
end
fprintf(fid,'\n\\end{bmatrix}\\hspace*{\\fill} \\circled{j}$\\\\n\\\\n');
```

```
%Part k
Ar = A;
for i = 1:4
    for j = 1:4
       if A(i,j)< 0
           Ar(i,j) = 0;
       end
    end
end
fprintf(fid,'The removal of negatives from a results
    in: $\\begin{bmatrix}\n');
[row,col] = size(Ar);
for i=1:row
    for j=1:col
        if j==col
            if i == row
                fprintf(fid,'%g', Ar(i,j));
            else
                fprintf(fid,'%g \\\\', Ar(i,j));
            end
        else
            fprintf(fid,'%g & ', Ar(i,j));
        end
    end
end
fprintf(fid,'\n\\end{bmatrix}\\hspace*{\\fill} \\
   circled{k}$\\\\\n\\\\\n');
fclose(fid);


%Problem 2
fid = fopen('Problem2.txt','w+');
A2 = [0 1; -6 -1]; B2 = [0 1; 1 1];
C2 = [1 0; 0 1]; D2 = [0.25 0; 0.1 2];
[num1,den] = ss2tf(A2,B2,C2,D2,1);
[num2,den] = ss2tf(A2,B2,C2,D2,2);
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);
E = eig(A2);
[Wn,Z] = damp(ss(A2,B2,C2,D2));

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);
p1(:,1) = pole(tf(num1(1,:),den));
p1(:,2) = pole(tf(num1(2,:),den));
p1(:,3) = pole(tf(num2(1,:),den));
p1(:,4) = pole(tf(num2(2,:),den));
nome = ['st';'nd';'rd';'th'];
n = 0;
for i = 1:4
    for j = 2:4
        if p1(:,i) == p1(:,j)
            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)));
            n = n+1;
        end
        if n == 3
            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)));
            break;
        end
    end
    if n == 3
            break;
    end
end
fclose(fid);

%Problem 3
fid = fopen('Problem3.txt','w+'); %creates/opens and overwrites text if file exists already
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];
BLo = [0.0721; -0.1178; -9.0991; 0];
CLo = eye(4); %Identity Matrix
DLo = zeros(4,1); %Zero Column Vector

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];
BLa = [0.0052 0.0310; -36.4909 8.1090; -0.4916 -2.8274; 0 0];
CLa = eye(4); %Identity Matrix
DLa = zeros(4,2);

%Part a
[WnLo,ZLo,PLo] = damp(ss(ALo,BLo,CLo,DLo));
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)));
fprintf(fid,'The natural frequencies of the longitudinal system  are %.4g, %.4g, %.4g, and %4g.\\\\n',WnLo);
fprintf(fid,'The damping ratios of the longitudinal system are %.4g, %.4g.\\\\n',ZLo);
```

```
[WnLa,ZLa,PLa] = damp(ss(ALa,BLa,CLa,DLa));
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)));
fprintf(fid,'The natural frequencies of the lateral
   system  are %.4g, %.4g, %.4g, and %4g.\\\\\n',
   WnLa);
fprintf(fid,'The damping ratios of the lateral
   system are %.4g, %.4g, %.4g, and %.4g.\\hspace
   *{\\fill} \\circled{b}\\\\\n',ZLa);

t0 = 0:0.01:10; %time sample length and step size
uel = cat(1,zeros(100,1),-0.5*ones(901,1)); %-0.5
   degree elevator step input at 1 second
uai = cat(2,uel,zeros(1001,1)); %-0.5 degree aileron
    step input at 1 second
uru = cat(2,zeros(1001,1),uel); %-0.5 degree rudder
   step input at 1 second

[yel,t] = lsim(ss(ALo,BLo,CLo,DLo),uel,t0);
%Where the First Column is True Velocity,
    %Second Column is Angle-of-Attack
        %Third Column is Pitch Rate
            %Fourth Column is Pitch Angle

[yai,t] = lsim(ss(ALa,BLa,CLa,DLa),uai,t0);
[yru,t] = lsim(ss(ALa,BLa,CLa,DLa),uru,t0);
%Where the First Column is Slide-Slip Angle,
    %Second Column is Roll Rate
        %Third Column is Yaw Rate
            %Fourth Column is Bank Angle

sim('Problem3.slx')%Run the Simulink Version
q = 1;
figure(q)
plot(t,yel(:,1),'k',tsim,yelsim(:,1),':y','LineWidth
   ',3.0)
title('Step Response to Elevator Input');
```

```
ylabel('True Velocity');
xlabel('time (sec)');
legend('Lsim Response','Simulink Response','Location','NorthEast');
q = q + 1;
figure(q)
plot(t,yel(:,2),'k',tsim,yelsim(:,2),':y','LineWidth',3.0)
title('Step Response to Elevator Input');
ylabel('Angle-of-Attack');
xlabel('time (sec)');
legend('Lsim Response','Simulink Response','Location','SouthEast');
q = q + 1;
figure(q)
plot(t,yel(:,3),'k',tsim,yelsim(:,3),':y','LineWidth',3.0)
title('Step Response to Elevator Input');
ylabel('Pitch Rate');
xlabel('time (sec)');
legend('Lsim Response','Simulink Response','Location','NorthEast');
q = q + 1;
figure(q)
plot(t,yel(:,4),'k',tsim,yelsim(:,4),':y','LineWidth',3.0)
title('Step Response to Elevator Input');
ylabel('Pitch Angle');
xlabel('time (sec)');
legend('Lsim Response','Simulink Response','Location','NorthWest');
q = q + 1;

figure(q)
plot(t,yai(:,1),'k',tsim,yaisim(:,1),':y','LineWidth',3.0)
title('Step Response to Aileron Input');
ylabel('Slide-Slip Angle');
```

```
xlabel('time (sec)');
legend('Lsim Response','Simulink Response','Location','NorthWest');
q = q + 1;
figure(q)
plot(t,yai(:,2),'k',tsim,yaisim(:,2),':y','LineWidth',3.0)
title('Step Response to Aileron Input');
ylabel('Roll Rate');
xlabel('time (sec)');
legend('Lsim Response','Simulink Response','Location','NorthEast');
q = q + 1;
figure(q)
plot(t,yai(:,3),'k',tsim,yaisim(:,3),':y','LineWidth',3.0)
title('Step Response to Aileron Input');
ylabel('Yaw Rate');
xlabel('time (sec)');
legend('Lsim Response','Simulink Response','Location','NorthWest');
q = q + 1;
figure(q)
plot(t,yai(:,4),'k',tsim,yaisim(:,4),':y','LineWidth',3.0)
title('Step Response to Aileron Input');
ylabel('Bank Angle');
xlabel('time (sec)');
legend('Lsim Response','Simulink Response','Location','NorthWest');
q = q + 1;

figure(q)
plot(t,yru(:,1),'k',tsim,yrusim(:,1),':y','LineWidth',3.0)
title('Step Response to Rudder Input');
ylabel('Slide-Slip Angle');
xlabel('time (sec)');
```

```
legend('Lsim Response','Simulink Response','Location','NorthEast');
q = q + 1;
figure(q)
plot(t,yru(:,2),'k',tsim,yrusim(:,2),':y','LineWidth',3.0)
title('Step Response to Rudder Input');
ylabel('Roll Rate');
xlabel('time (sec)');
legend('Lsim Response','Simulink Response','Location','NorthEast');
q = q + 1;
figure(q)
plot(t,yru(:,3),'k',tsim,yrusim(:,3),':y','LineWidth',3.0)
title('Step Response to Rudder Input');
ylabel('Yaw Rate');
xlabel('time (sec)');
legend('Lsim Response','Simulink Response','Location','NorthEast');
q = q + 1;
figure(12)
plot(t,yru(:,4),'k',tsim,yrusim(:,4),':y','LineWidth',3.0)
title('Step Response to Rudder Input');
ylabel('Bank Angle');
xlabel('time (sec)');
legend('Lsim Response','Simulink Response','Location','NorthWest');
fprintf(fid,'\\\\\nThe following all show the same story, that the the lsim results and the simulink results \\textbf{MATCH!}');
for i=1:q
    name = cat(2,'Figure_',num2str(i));
    print(figure(i),'-depsc',name);
    fprintf(fid,'\\begin{figure}[H]\\centering\\includegraphics[keepaspectratio=true,height=.45\\textheight,width=1\\textwidth]{%s.eps
```

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