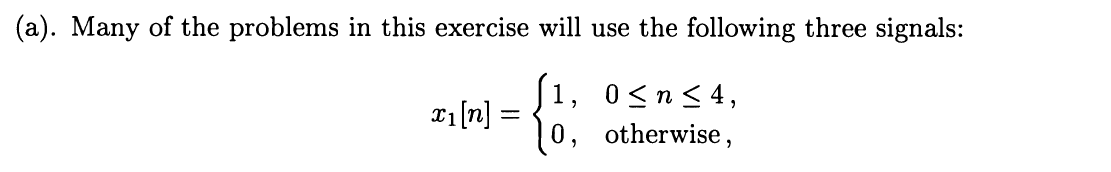
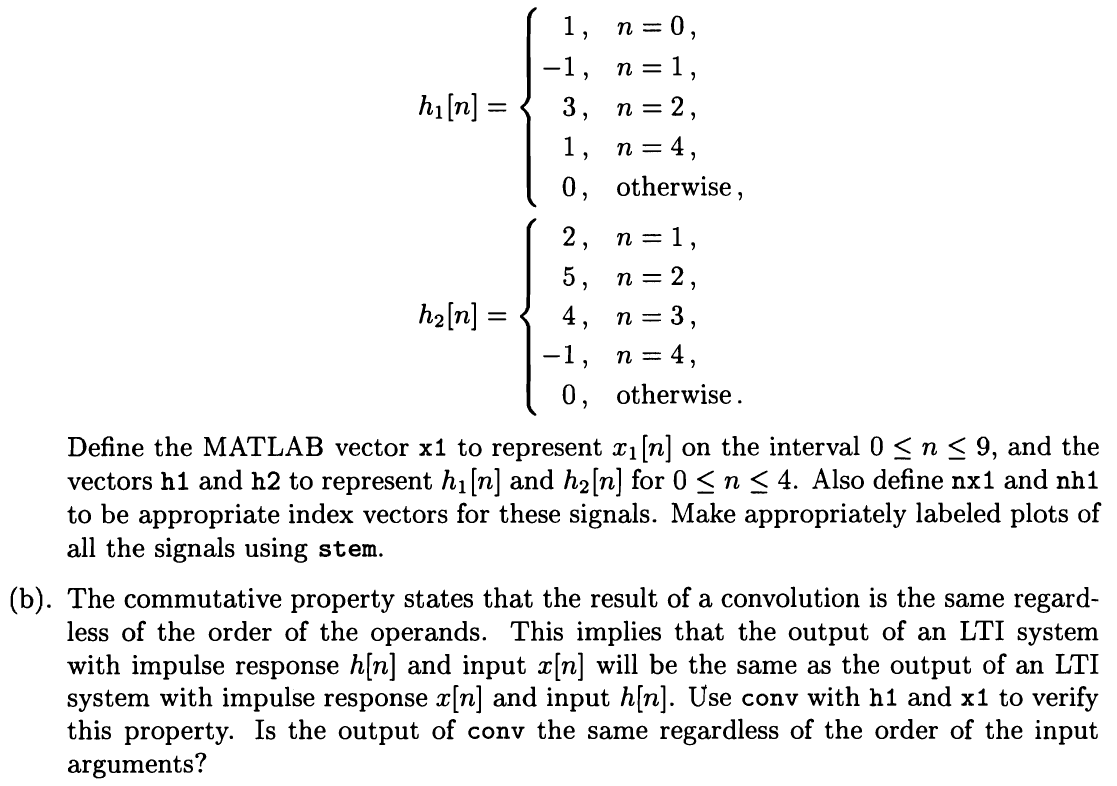
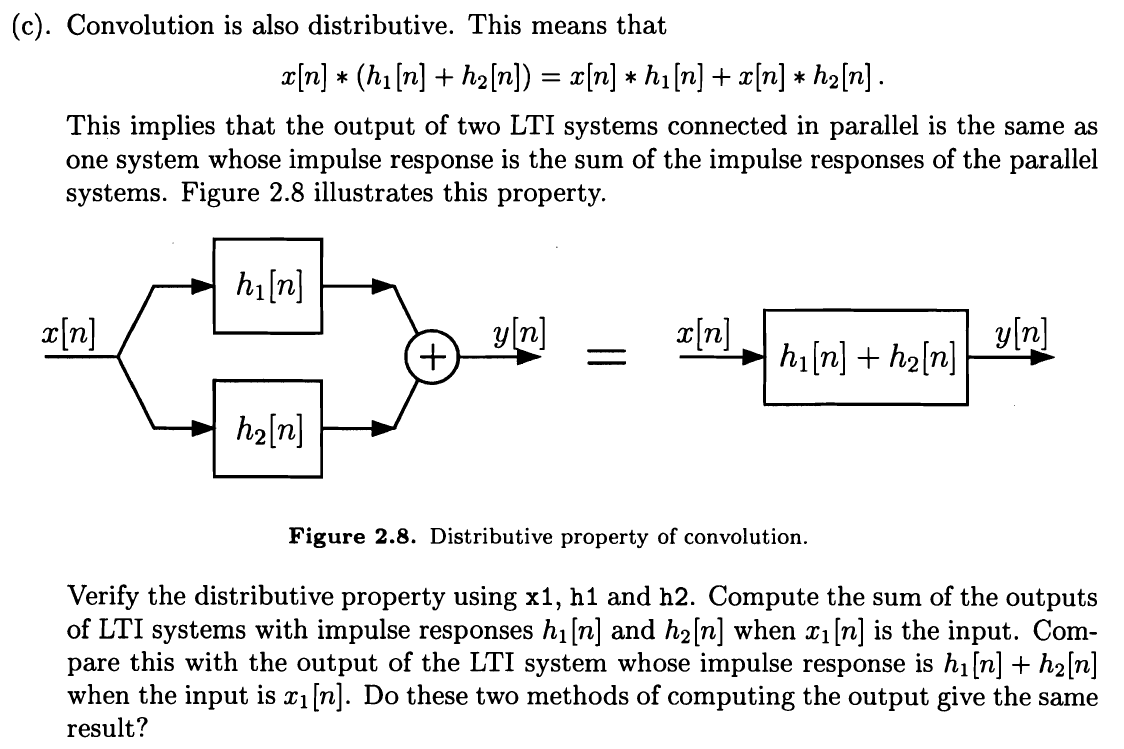
Name 1:李璇 SID 1:12010137 Name 2:张林燊 SID 2:12010424

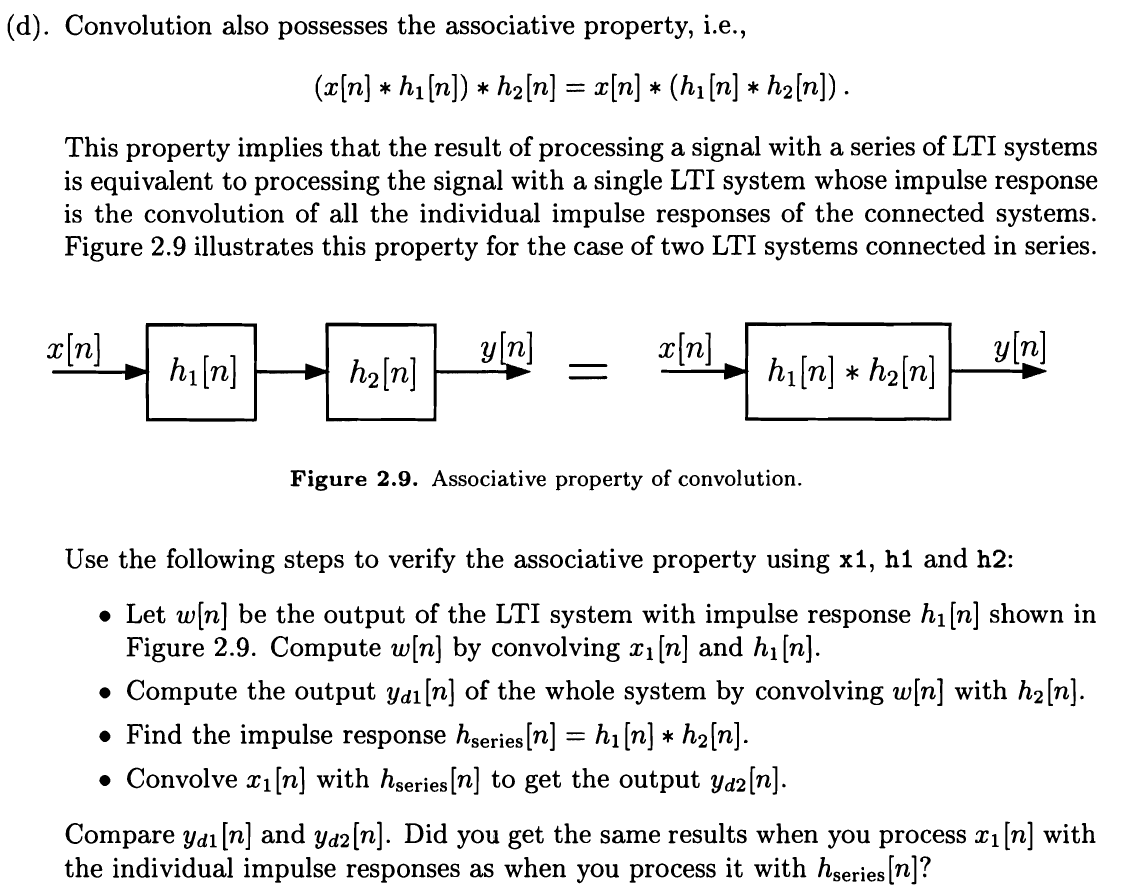
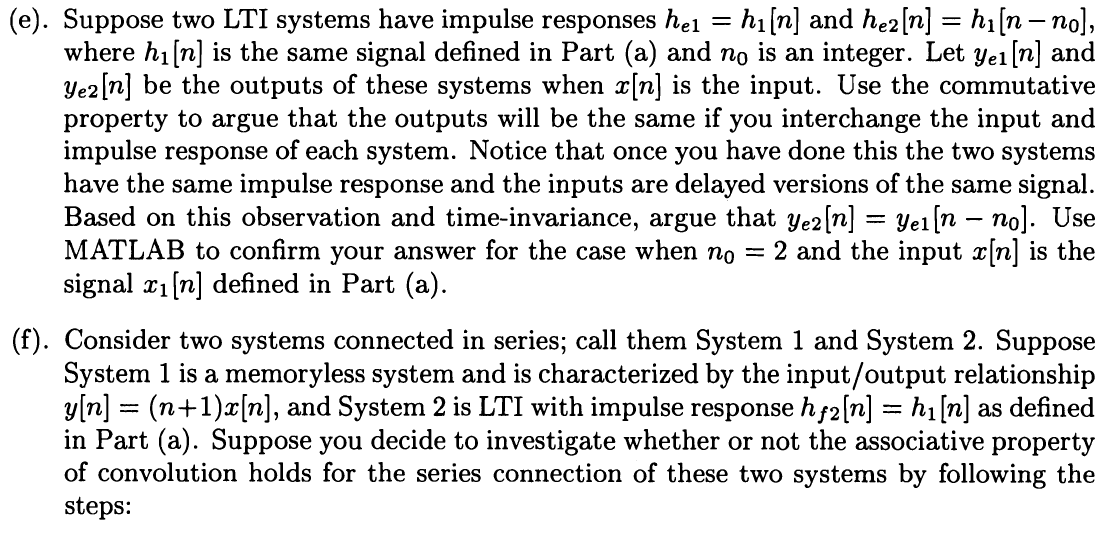
实验报告#2 (Lab#2)

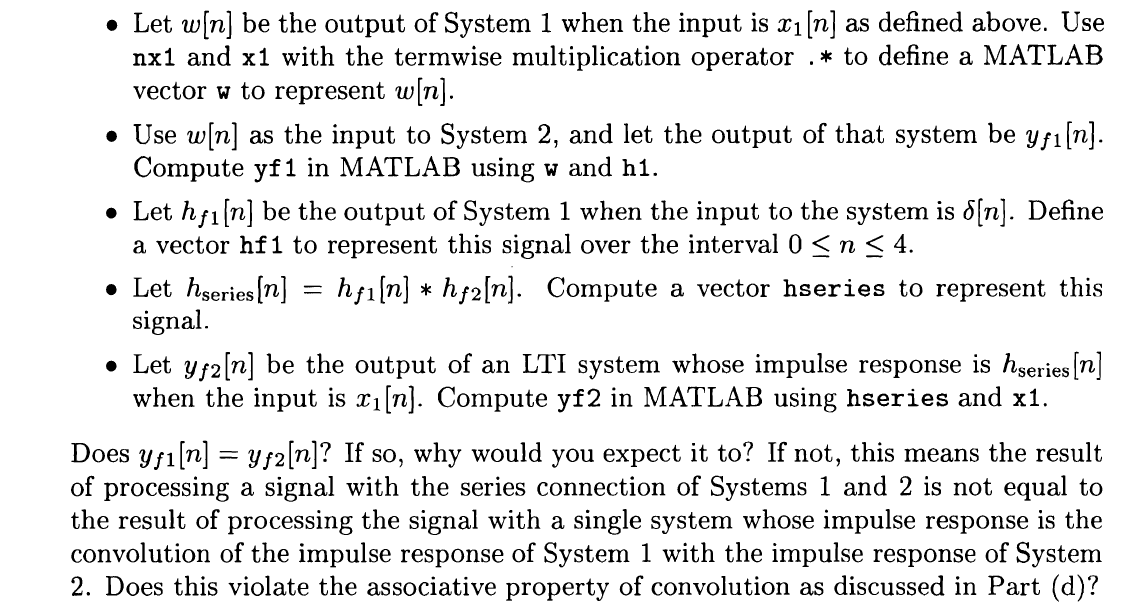
**2.4**



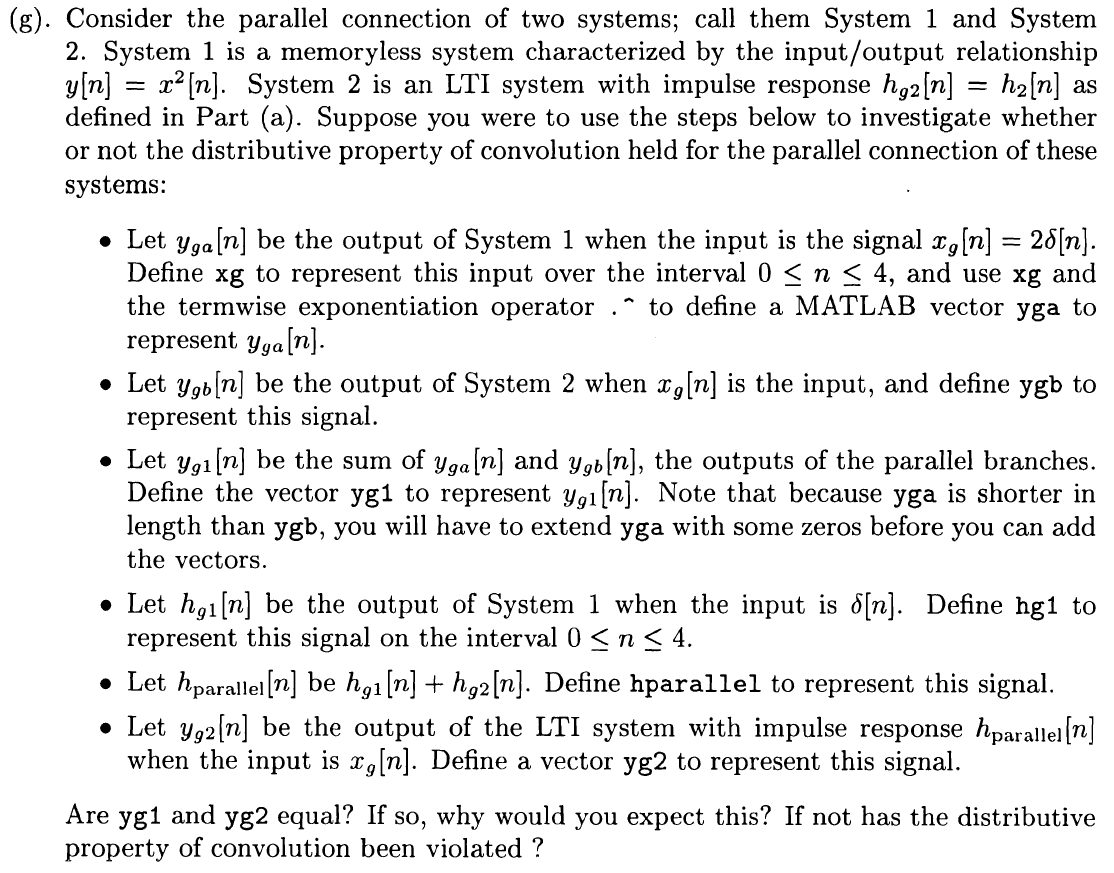




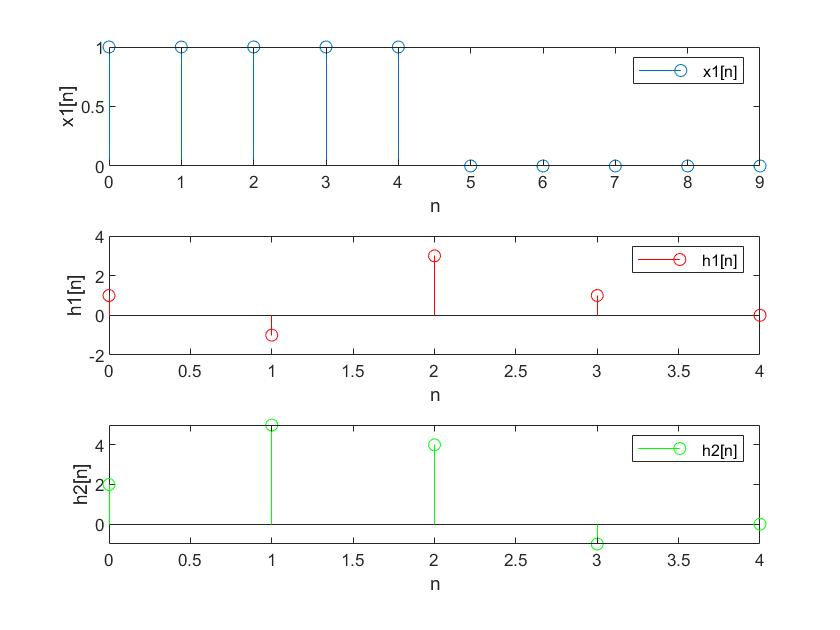




**Solution:**



**Solution:**



The plots of x1[n], h1[n], h2[n] are shown above.

**MATLAB Code:**

nx1=0:9;

nh1=0:4;

x1=[ones(1,5) zeros(1,5)];

h1=[1,-1,3,1,0];

h2=[2,5,4,-1,0];

subplot(3,1,1);

stem(nx1,x1);

xlabel('n');

ylabel('x1[n]');

legend('x1[n]');

subplot(3,1,2);

stem(nh1,h1,'red');

xlabel('n');

ylabel('h1[n]');

legend('h1[n]');

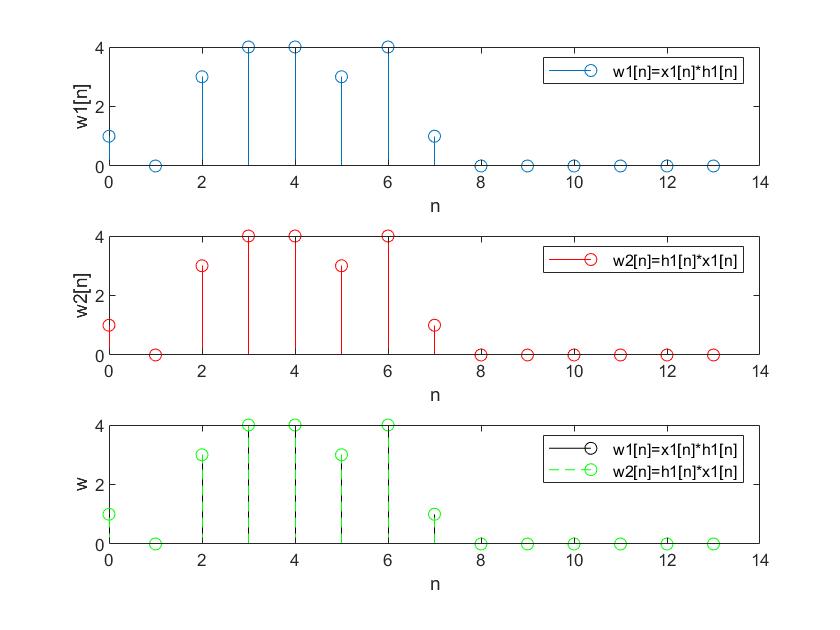
subplot(3,1,3);

stem(nh1,h2,'green');

xlabel('n');

ylabel('h2[n]');

legend('h2[n]');



As the figures shown, w1 and w2 completely overlap with each other, which means they are equal. So the output of conv is the same regardless of the order of the input arguments, which means the commutative property holds.

**MATLAB Code:**

nx1=0:9;

nh1=0:4;

x1=[ones(1,5) zeros(1,5)];

h1=[1,-1,3,1,0];

w1=conv(x1,h1);

w2=conv(h1,x1);

nw=(nx1(1)+nh1(1)):(nx1(end)+nh1(end));

subplot(3,1,1);

stem(nw,w1);

xlabel('n');

ylabel('w1[n]');

legend('w1[n]=x1[n]\*h1[n]');

subplot(3,1,2);

stem(nw,w2,'red');

xlabel('n');

ylabel('w2[n]');

legend('w2[n]=h1[n]\*x1[n]');

subplot(3,1,3);

stem(nw,w1,'black');

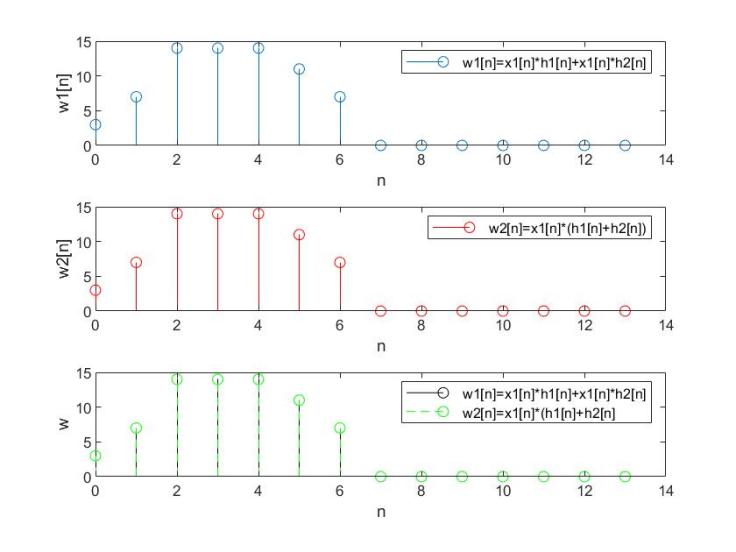
hold on;

stem(nw,w2,'green','--');

xlabel('n');

ylabel('w');

legend('w1[n]=x1[n]\*h1[n]','w2[n]=h1[n]\*x1[n]');



As the figures shown, w1 and w2 completely overlap with each other, which means they are equal. So these two methods of computing the output give the same result, which means the distributive property holds.

**MATLAB Code:**

nx1=0:9;

nh1=0:4;

x1=[ones(1,5) zeros(1,5)];

h1=[1,-1,3,1,0];

h2=[2,5,4,-1,0];

h3=h1+h2;

w1=conv(x1,h1)+conv(x1,h2);

w2=conv(x1,h3);

nw=(nx1(1)+nh1(1)):(nx1(end)+nh1(end));

subplot(3,1,1);

stem(nw,w1);

xlabel('n');

ylabel('w1[n]');

legend('w1[n]=x1[n]\*h1[n]+x1[n]\*h2[n]');

subplot(3,1,2);

stem(nw,w2,'red');

xlabel('n');

ylabel('w2[n]');

legend('w2[n]=x1[n]\*(h1[n]+h2[n])');

subplot(3,1,3);

stem(nw,w1,'black');

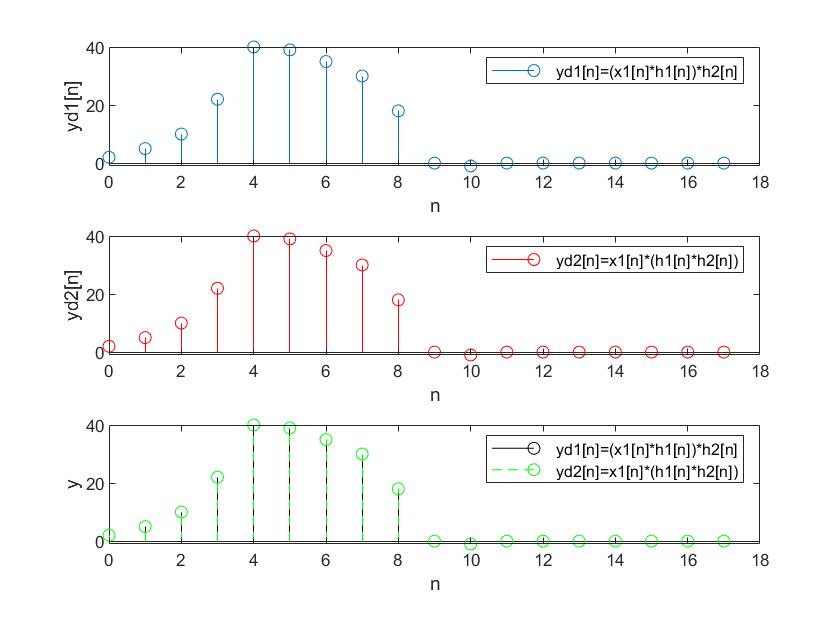
hold on;

stem(nw,w2,'green','--');

xlabel('n');

ylabel('w');

legend('w1[n]=x1[n]\*h1[n]+x1[n]\*h2[n]','w2[n]=x1[n]\*(h1[n]+h2[n]');



As the figures shown, w1 and w2 completely overlap with each other, which means they are equal. So we can get the same results when processing x1[n] with the individual impulse responses as when processing it with hseries[n], which means the associative property holds.

**MATLAB Code:**

nx1=0:9;

nh1=0:4;

x1=[ones(1,5) zeros(1,5)];

h1=[1,-1,3,1,0];

h2=[2,5,4,-1,0];

w=conv(x1,h1);

hseries=conv(h1,h2);

yd1=conv(w,h2);

yd2=conv(x1,hseries);

nw=(nx1(1)+2\*nh1(1)):(nx1(end)+2\*nh1(end));

subplot(3,1,1);

stem(nw,yd1);

xlabel('n');

ylabel('yd1[n]');

legend('yd1[n]=(x1[n]\*h1[n])\*h2[n]');

subplot(3,1,2);

stem(nw,yd2,'red');

xlabel('n');

ylabel('yd2[n]');

legend('yd2[n]=x1[n]\*(h1[n]\*h2[n])');

subplot(3,1,3);

stem(nw,yd1,'black');

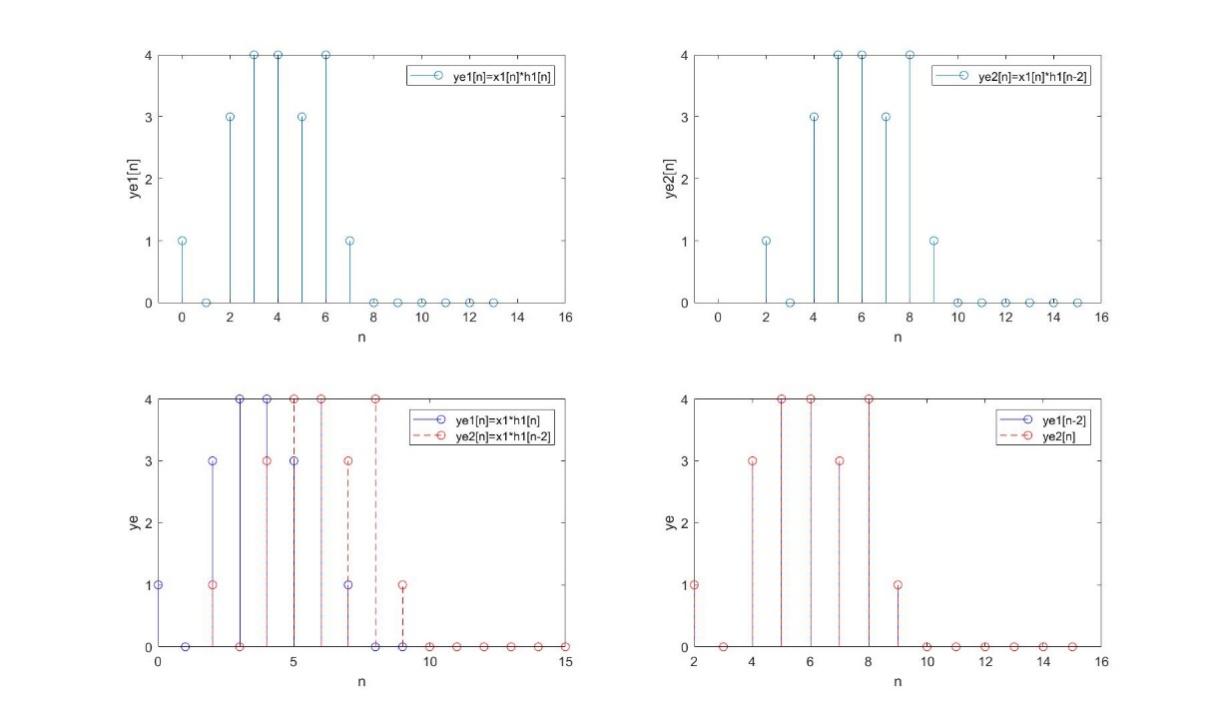
hold on;

stem(nw,yd2,'green','--');

xlabel('n');

ylabel('y');

legend('yd1[n]=(x1[n]\*h1[n])\*h2[n]','yd2[n]=x1[n]\*(h1[n]\*h2[n])');



In the figures, we get ye1[n]=x1[n]\*h1[n] and ye2[n]=x1[n]\*h2[n-2]. In the fourth figure, ye1[n-2] and ye2[n] completely overlap with each other, which means ye2[n]= ye1[n-2] is correct.

**MATLAB Code:**

nx1=0:9;

nh1=0:4;

x1=[ones(1,5) zeros(1,5)];

h1=[1,-1,3,1,0];

h2=[2,5,4,-1,0];

nhe1=nh1;

nhe2=nh1+2;

nye1=nx1(1)+nhe1(1):nx1(end)+nhe1(end);

nye2=nx1(1)+nhe2(1):nx1(end)+nhe2(end);

ye=conv(x1,h1);

subplot(2,2,1);

stem(nye1,ye);

ylabel('ye1[n]');

xlabel('n');

legend('ye1[n]=x1[n]\*h1[n]');

axis([-1 16 0 4]);

subplot(2,2,2);

stem(nye2,ye);

ylabel('ye2[n]');

xlabel('n');

legend('ye2[n]=x1[n]\*h1[n-2]');

axis([-1 16 0 4]);

subplot(2,2,3);

stem(nye1,ye,'blue');

hold on;

stem(nye2,ye,'red','--');

ylabel('ye');

legend('ye1[n]=x1\*h1[n]','ye2[n]=x1\*h1[n-2]');

xlabel('n');

subplot(2,2,4);

stem(nye2,ye,'blue');

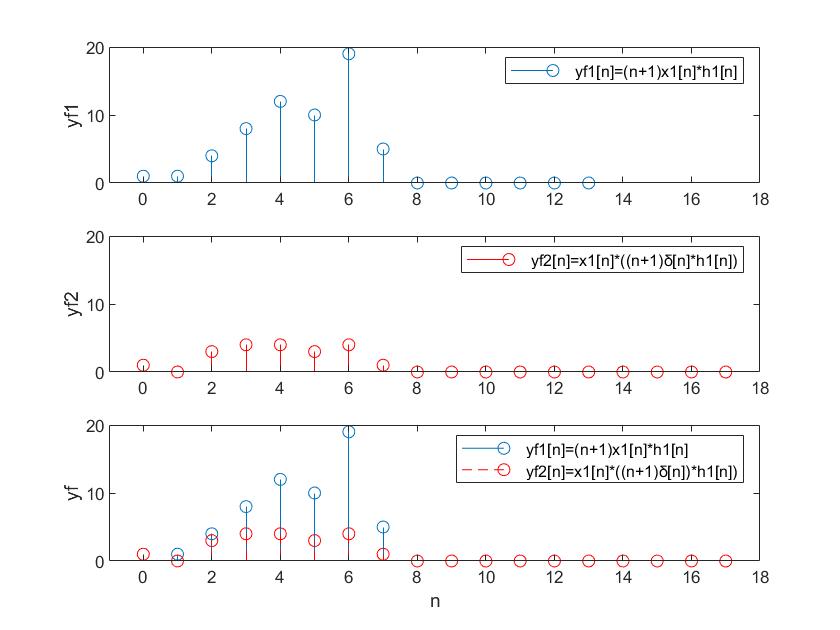
hold on;

stem(nye2,ye,'red','--');

ylabel('ye');

legend('ye1[n-2]','ye2[n]');

xlabel('n');



As the figures show, yf1[n] and yf2[n] do not overlap with each other completely, which means yf1[n]≠yf2[n]. That is because y[n]=(n+1)x[n] is not LTI.

**MATLAB Code:**

nx1=0:9;

nh1=0:4;

x1=[ones(1,5) zeros(1,5)];

h1=[1,-1,3,1,0];

h2=[2,5,4,-1,0];

h3=[1,0,0,0,0];

hf2=h1;

w=(nx1+1).\*x1;

nw=nx1;

yf1=conv(w,h1);

nyf1=nw(1)+nh1(1):nw(end)+nh1(end);

hf1=(nh1+1).\*h3;

hseries=conv(hf1,hf2);

nhseries=nh1(1)\*2:nh1(end)\*2;

yf2=conv(x1,hseries);

nyf2=nx1(1)+nhseries(1):nx1(end)+nhseries(end);

subplot(3,1,1)

stem(nyf1,yf1);

ylabel('yf1');

legend('yf1[n]=(n+1)x1[n]\*h1[n]');

axis([-1 18 0 20]);

subplot(3,1,2)

stem(nyf2,yf2,'red');

ylabel('yf2');

legend('yf2[n]=x1[n]\*((n+1)¦Ä[n]\*h1[n])');

axis([-1 18 0 20]);

subplot(3,1,3)

stem(nyf1,yf1);

hold on;

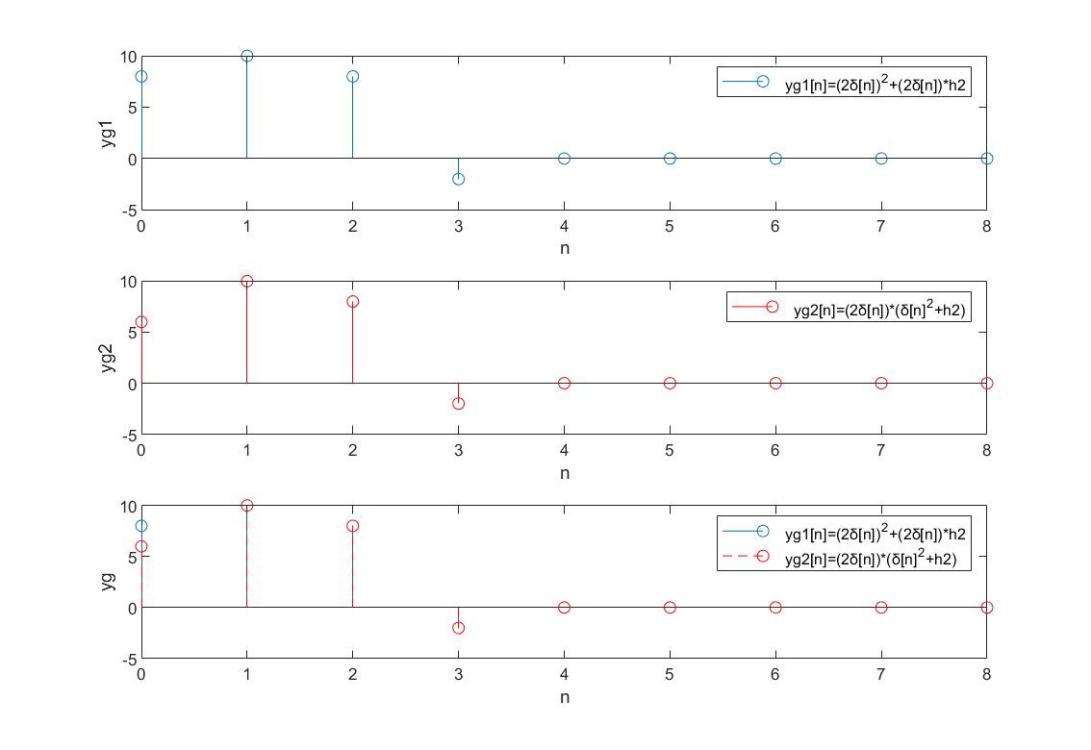
stem(nyf2,yf2,'red','--');

ylabel('yf');

xlabel('n');

legend('yf1[n]=(n+1)x1[n]\*h1[n]','yf2[n]=x1[n]\*((n+1)¦Ä[n])\*h1[n])');

axis([-1 18 0 20]);



As the figures show, yg1[n] and yg2[n] does not overlap with each other completely, which means yg1[n]≠yg2[n]. However, this does not violate the distributive property because the system is not LTI.

**MATLAB Code:**

nx1=0:9;

nh1=0:4;

x1=[ones(1,5) zeros(1,5)];

h1=[1,-1,3,1,0];

h2=[2,5,4,-1,0];

xg=[2,0,0,0,0];

nxg=nh1;

yga=xg.^2;

nyga=nxg;

ygb=conv(xg,h2);

nygb=nxg(1)+nh1(1):nxg(end)+nh1(end);

yg1=[yga 0 0 0 0]+ygb;

nyg1=nygb;

hg1=[1,0,0,0,0].^2;

hg2=h2;

hparallel=hg1+hg2;

nhparallel=nh1;

yg2=conv(xg,hparallel);

nyg2=nxg(1)+nhparallel(1):nxg(end)+nhparallel(end);

subplot(3,1,1);

stem(nyg1,yg1);

xlabel('n');

ylabel('yg1');

legend('yg1[n]=(2¦Ä[n])^2+(2¦Ä[n])\*h2');

subplot(3,1,2);

stem(nyg2,yg2,'red');

xlabel('n');

ylabel('yg2');

legend('yg2[n]=(2¦Ä[n])\*(¦Ä[n]^2+h2)');

subplot(3,1,3);

stem(nyg1,yg1);

hold on;

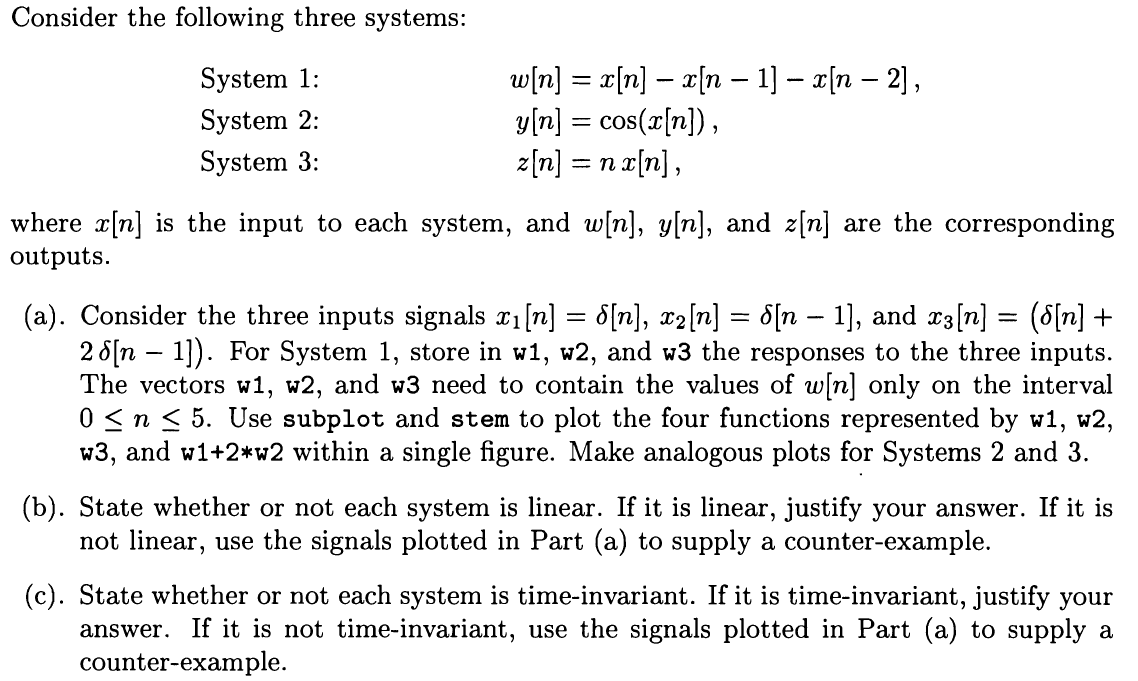
stem(nyg1,yg2,'red','--');

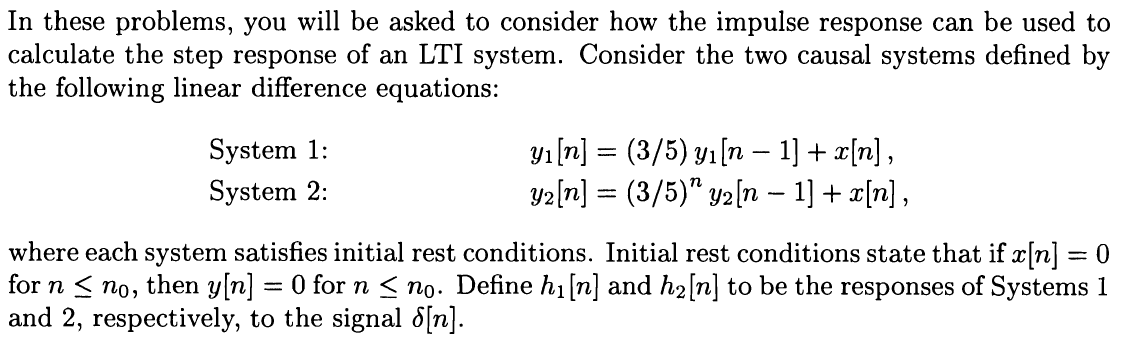
ylabel('yg');

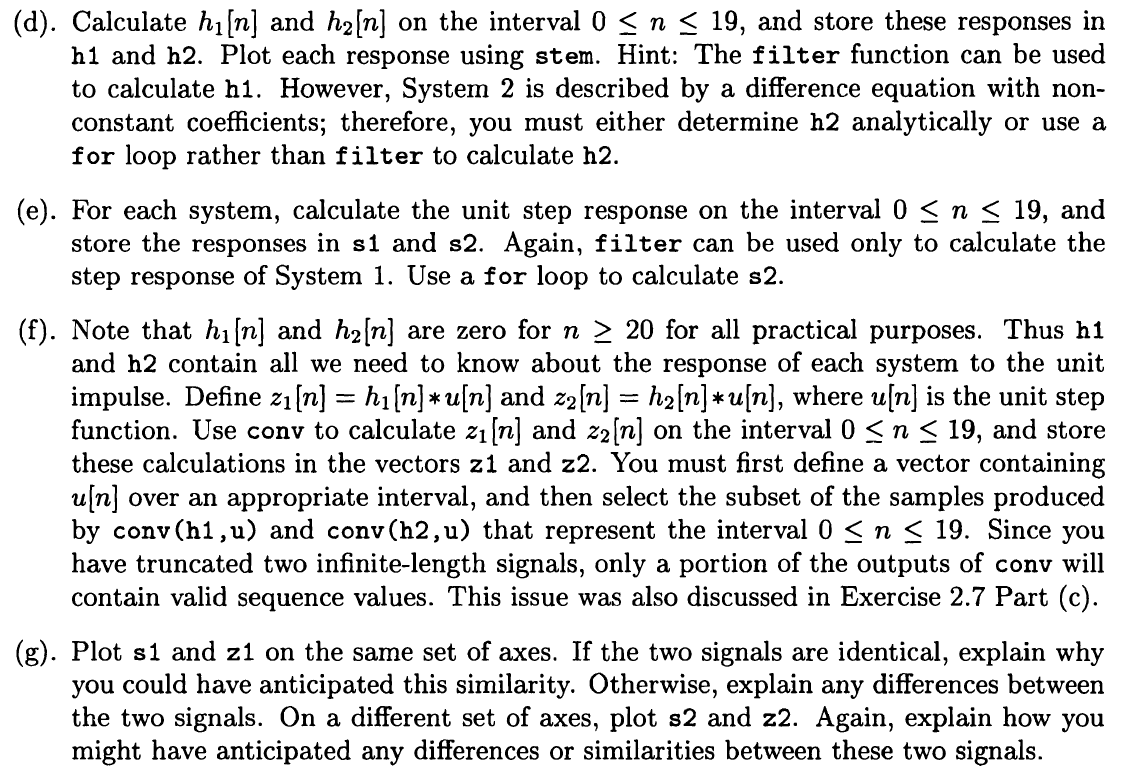
legend('yg1[n]=(2¦Ä[n])^2+(2¦Ä[n])\*h2','yg2[n]=(2¦Ä[n])\*(¦Ä[n]^2+h2)');

xlabel('n');

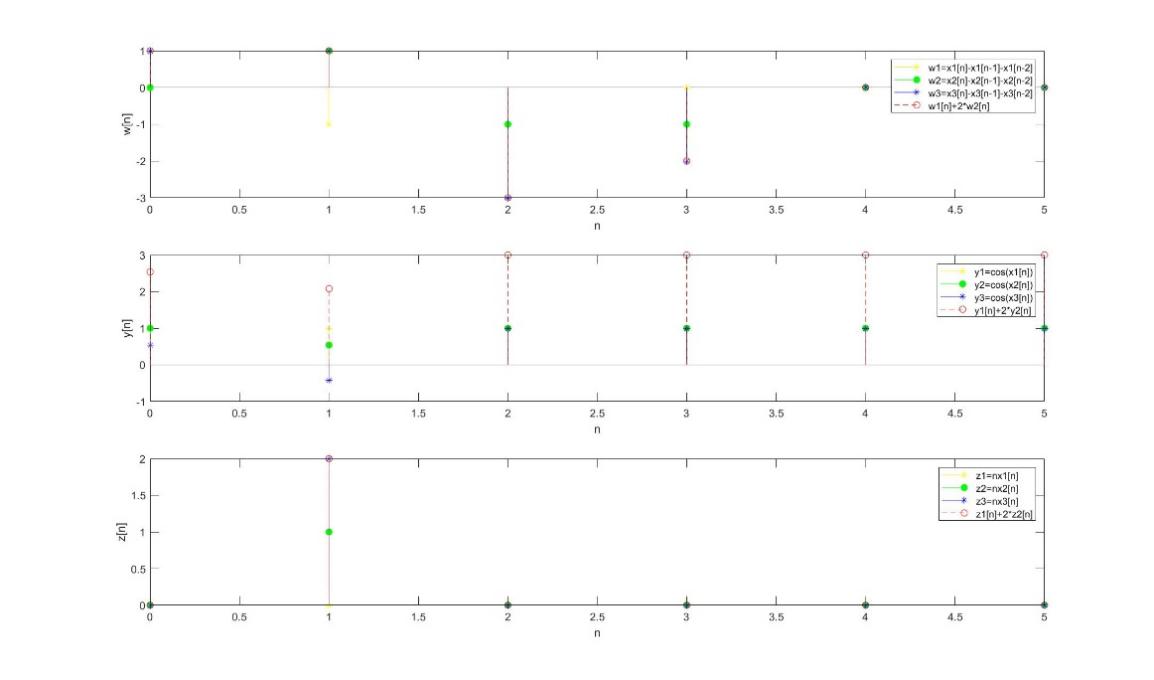
**2.5**







**Solution:**



Figures of the systems are shown above.

**MATLAB Code:**

nx=0:5;

x1=[1,0,0,0,0,0];

x2=[0,1,0,0,0,0];

x3=[1,2,0,0,0,0];

a1=1;

b1=[1,-1,-1];

w1=filter(b1,a1,x1);

w2=filter(b1,a1,x2);

w3=filter(b1,a1,x3);

y1=cos(x1);

y2=cos(x2);

y3=cos(x3);

z1=nx.\*x1;

z2=nx.\*x2;

z3=nx.\*x3;

subplot(3,1,1);

stem(nx,w1,'yellow','\*');

hold on;

stem(nx,w2,'green','filled');

hold on;

stem(nx,w3,'blue','\*');

hold on;

stem(nx,w1+2.\*w2,'red','--');

xlabel('n');

ylabel('w[n]');

legend('w1=x1[n]-x1[n-1]-x1[n-2]','w2=x2[n]-x2[n-1]-x2[n-2]','w3=x3[n]-x3[n-1]-x3[n-2]','w1[n]+2\*w2[n]');

subplot(3,1,2);

stem(nx,y1,'yellow','\*');

hold on;

stem(nx,y2,'green','filled');

hold on;

stem(nx,y3,'blue','\*');

hold on;

stem(nx,y1+2.\*y2,'red','--');

xlabel('n');

ylabel('y[n]');

legend('y1=cos(x1[n])','y2=cos(x2[n])','y3=cos(x3[n])','y1[n]+2\*y2[n]');

subplot(3,1,3);

stem(nx,z1,'yellow','\*');

hold on;

stem(nx,z2,'green','filled');

hold on;

stem(nx,z3,'blue','\*');

hold on;

stem(nx,z1+2.\*z2,'red','--');

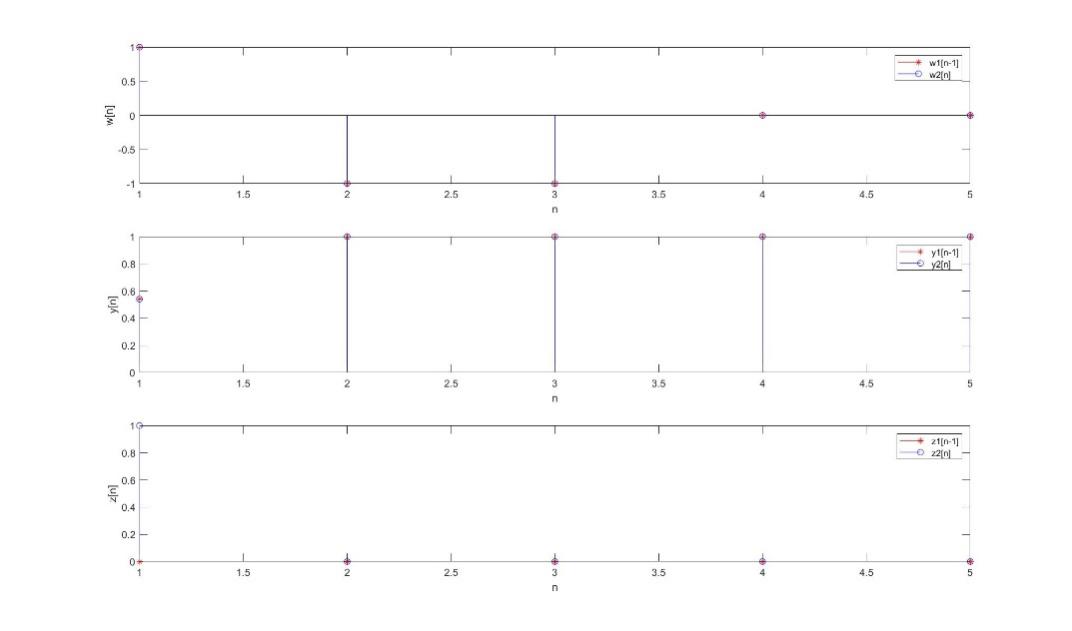
xlabel('n');

ylabel('z[n]');

legend('z1=nx1[n]','z2=nx2[n]','z3=nx3[n]','z1[n]+2\*z2[n]');

1. System 1 and System 3 are linear while System 2 is not linear.

If the system is linear, the response to x1[n]+2x2[n] should be y1[n]+2y2[n]. As the figures show, w3[n] completely overlap with w1[n]+2w2[n] and z3[n] completely overlap with z1[n]+z2[n], so System 1 and System 3 are linear. The figure of y3[n] does not overlap with y1[n]+2y2[n] completely, so System 2 is not linear.



If the system is time-invariant, there should be y1[n-1]=y2[n] as x2[n]=x1[n-1]. As the figures show, w1[n-1] completely overlap with w2[n] and y1[n-1] completely overlap with y2[n], so System 1 and System 2 are time-invariant. But z1[n-1] does not overlap with z2[n] completely, so System 3 is not time-invariant.

**MATLAB Code:**

nx=0:5;

x1=[1,0,0,0,0,0];

x2=[0,1,0,0,0,0];

a1=1;

b1=[1,-1,-1];

w1=filter(b1,a1,x1);

w2=filter(b1,a1,x2);

y1=cos(x1);

y2=cos(x2);

y3=cos(x3);

z1=nx.\*x1;

z2=nx.\*x2;

z3=nx.\*x3;

subplot(3,1,1);

stem(nx+1,w1,'red','\*');

hold on;

stem(nx,w2,'blue');

xlabel('n');

ylabel('w[n]');

legend('w1[n-1]','w2[n]');

xlim([1 5]);

subplot(3,1,2);

stem(nx+1,y1,'red','\*');

hold on;

stem(nx,y2,'blue');

xlabel('n');

ylabel('y[n]');

legend('y1[n-1]','y2[n]');

xlim([1 5]);

subplot(3,1,3);

stem(nx+1,z1,'red','\*');

hold on;

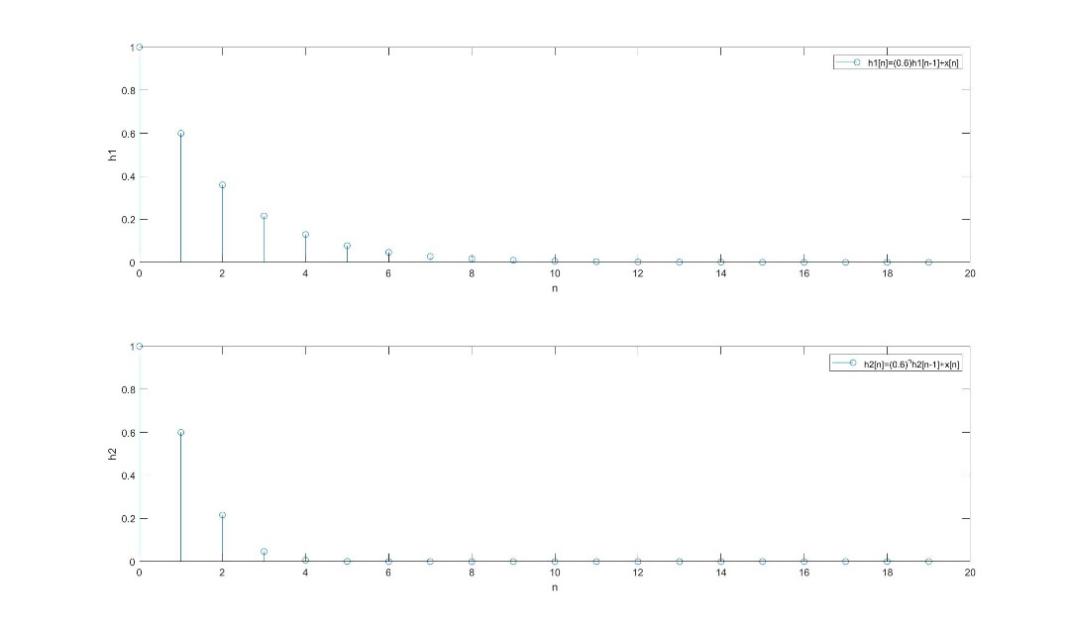
stem(nx,z2,'blue');

xlabel('n');

ylabel('z[n]');

legend('z1[n-1]','z2[n]');

xlim([1 5]);



The figures of h1[n] and h2[n] are shown above.

**MATLAB Code:**

nx=0:19;

x=[1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0];

a1=[1 -0.6];

b1=1;

h1=filter(b1,a1,x);

m=0;

s=0;

for n=1:20

s=(0.6^(n-1))\*m+x(n);

h2(n)=s;

m=s;

end

subplot(2,1,1);

stem(nx,h1);

xlabel('n');

ylabel('h1');

legend('h1[n]=(0.6)h1[n-1]+x[n]');

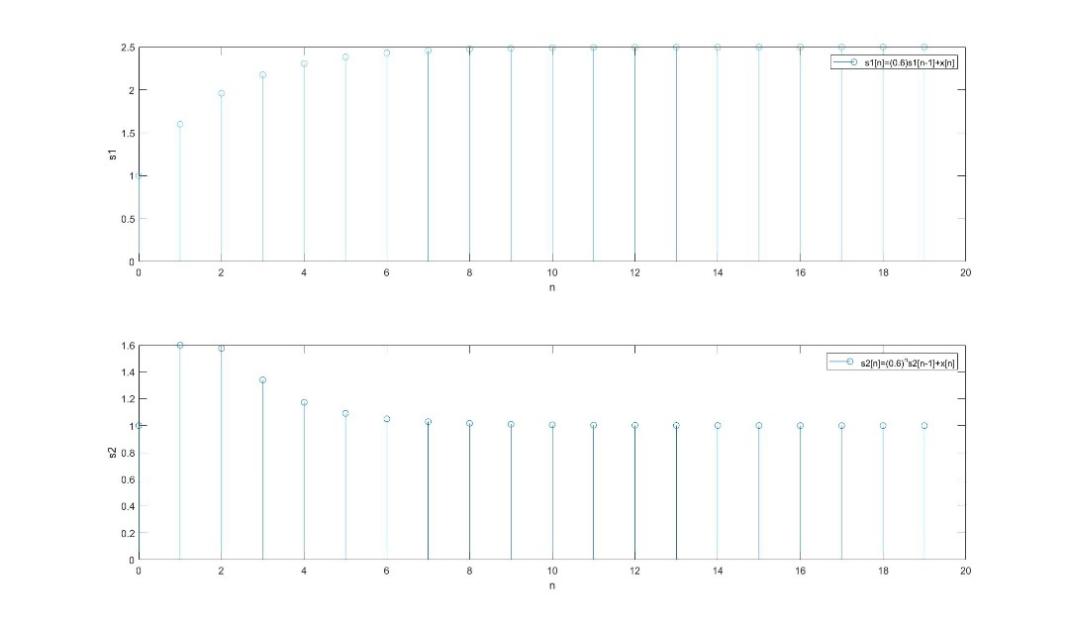
subplot(2,1,2);

stem(nx,h2);

xlabel('n');

ylabel('h2');

legend('h2[n]=(0.6)^nh2[n-1]+x[n]');



The figures of s1[n] and s2[n] are shown above.

**MATLAB Code:**

nx=0:19;

x=ones(20);

a1=[1 -0.6];

b1=1;

s1=filter(b1,a1,x);

m=0;

p=0;

for n=1:20

p=(0.6^(n-1))\*m+x(n);

s2(n)=p;

m=p;

end

subplot(2,1,1);

stem(nx,s1);

xlabel('n');

ylabel('s1');

legend('s1[n]=(0.6)s1[n-1]+x[n]');

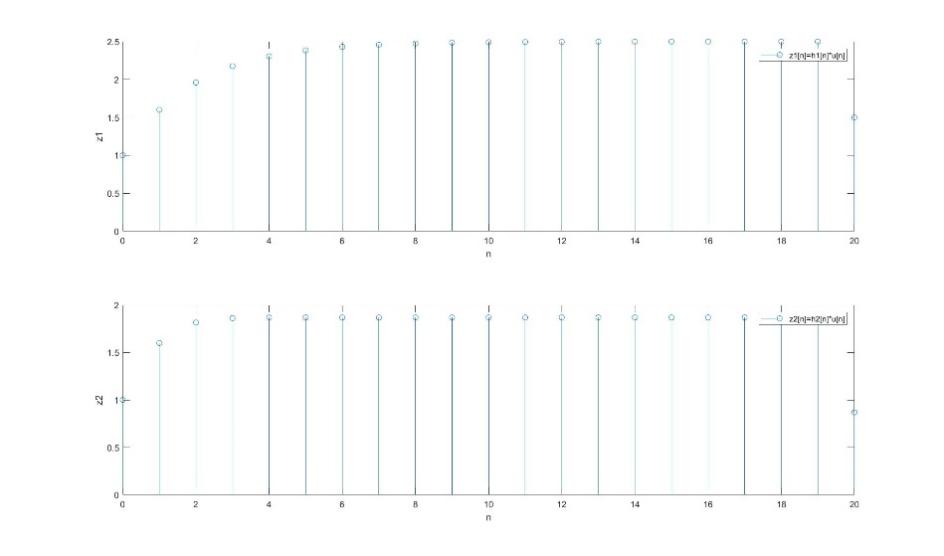
subplot(2,1,2);

stem(nx,s2);

xlabel('n');

ylabel('s2');

legend('s2[n]=(0.6)^ns2[n-1]+x[n]');



The figures of z1[n] and z2[n] are shown above.

Since h1 and h2 are zero when n<0 or n≥20, the appropriate interval of u[n] should be [0, 20), and only 0≤n<20 is the valid interval of z1 and z2.

**MATLAB Code:**

nx=0:19;

x=[1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0];

u=[1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1];

a1=[1 -0.6];

b1=1;

h1=filter(b1,a1,x);

m=0;

s=0;

for n=1:20

s=(0.6^(n-1))\*m+x(n);

h2(n)=s;

m=s;

end

z1=conv(h1,u);

z2=conv(h2,u);

nz1=2\*nx(1):2\*nx(end);

nz2=2\*nx(1):2\*nx(end);

subplot(2,1,1);

stem(nz1,z1);

xlabel('n');

ylabel('z1');

legend('z1[n]=h1[n]\*u[n]');

xlim([0 20]);

subplot(2,1,2);

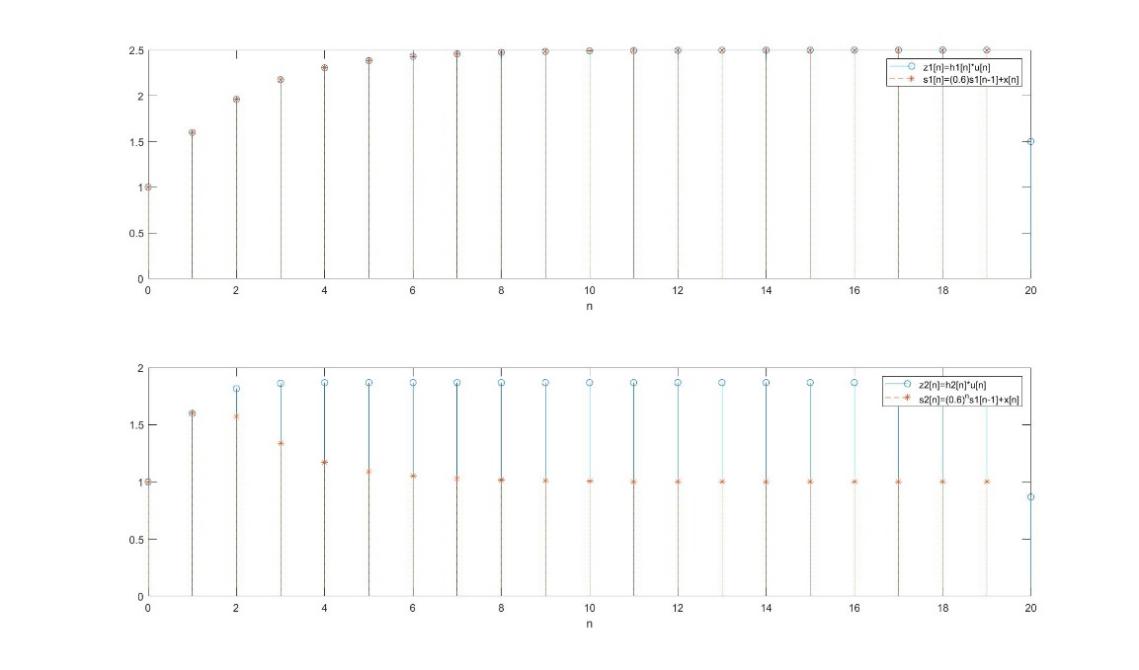
stem(nz2,z2);

xlabel('n');

ylabel('z2');

legend('z2[n]=h2[n]\*u[n]');

xlim([0 20]);



As the figures shown, z1[n] and s1[n] completely overlap with each other, which means z1[n]=s1[n]. However, z2[n] and s2[n] does not overlap with each other completely, which means z2[n]≠s2[n].

Because h1[n] has a fixed coefficient (3/5) so the effect of the previous value h1[n-1] is always the same as convolution.

Because h2[n] is different from h1[n], it has an unfixed coefficient (3/5^n) of h2[n-1], so the effect of h2[n-1] (the previous value) is different from the convolution and changes with n.

**MATLAB Code:**

nx=0:19;

x=[1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0];

u=[1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1];

a1=[1 -0.6];

b1=1;

h1=filter(b1,a1,x);

m=0;

s=0;

for n=1:20

s=(0.6^(n-1))\*m+x(n);

h2(n)=s;

m=s;

end

m=0;

s=0;

z1=conv(h1,u);

z2=conv(h2,u);

nz1=2\*nx(1):2\*nx(end);

nz2=2\*nx(1):2\*nx(end);

a1=[1 -0.6];

b1=1;

s1=filter(b1,a1,u);

for n=1:20

p=(0.6^(n-1))\*m+u(n);

s2(n)=p;

m=p;

end

subplot(2,1,1);

stem(nz1,z1);

hold on;

stem(nx,s1,'--\*');

xlabel('n');

legend('z1[n]=h1[n]\*u[n]','s1[n]=(0.6)s1[n-1]+x[n]');

xlim([0 20]);

subplot(2,1,2);

stem(nz2,z2);

hold on;

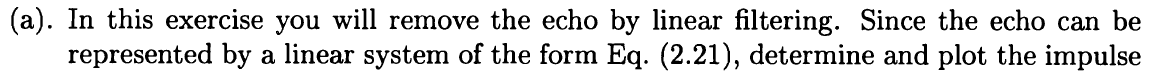
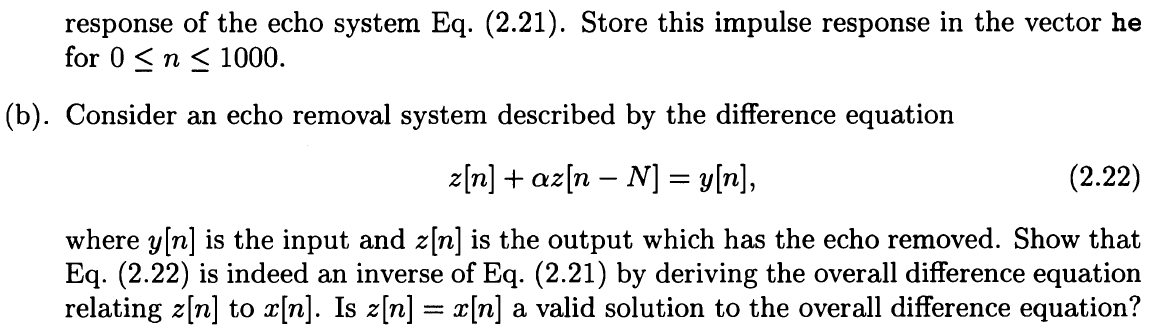
stem(nx,s2,'--\*');

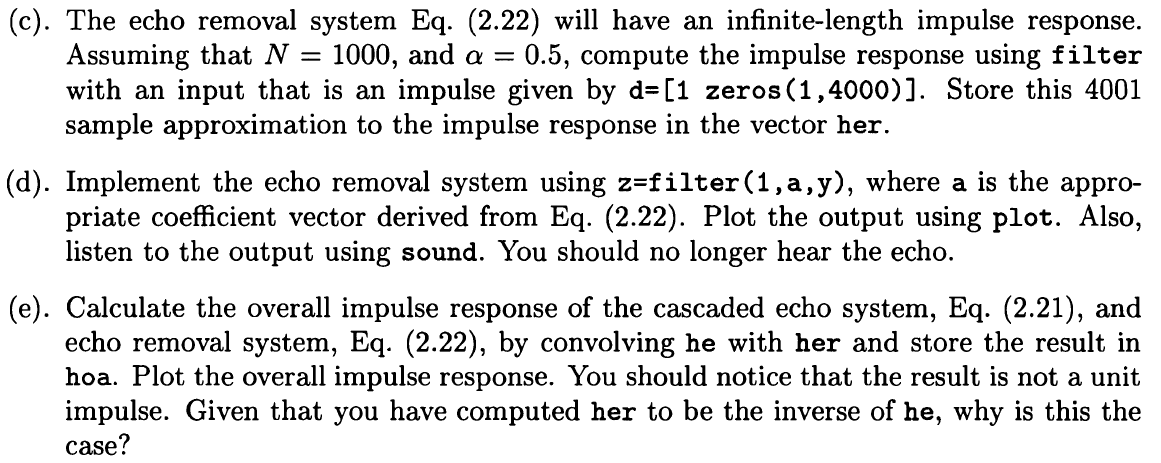
xlabel('n');

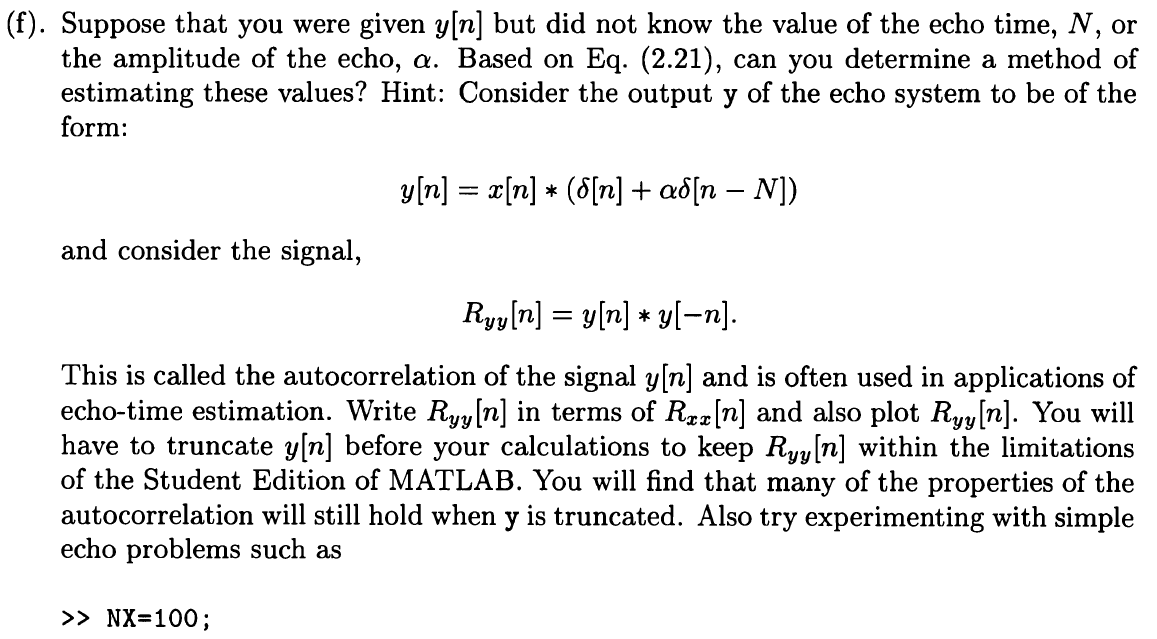
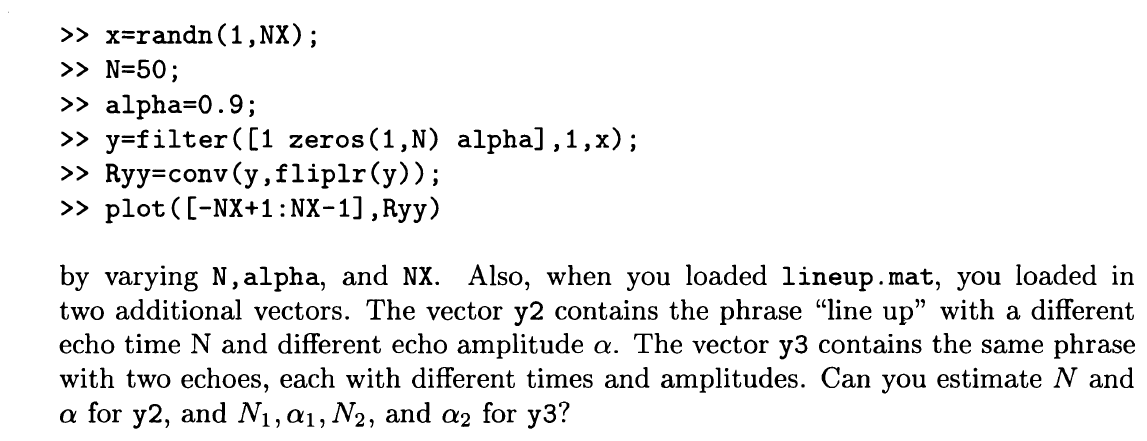
legend('z2[n]=h2[n]\*u[n]','s2[n]=(0.6)^ns1[n-1]+x[n]');

xlim([0 20]);

**2.10**

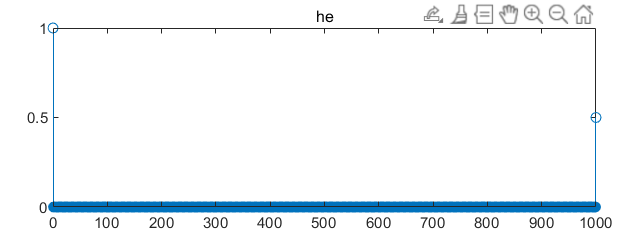






**Solution:**

（a）



The unit impulse response of m(y[n] = x[n]+0.5x[n-1000]) is calculated.

**MATLAB Code:**

%2.10（a）

A=1;

B=[1, zeros(1,999) ,0.5];

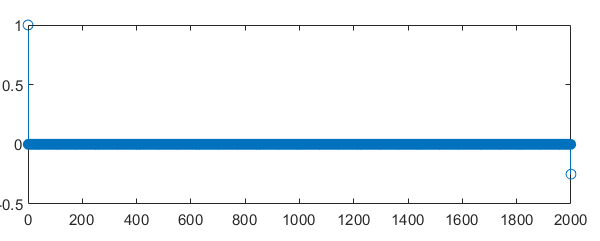
he=impz(B,A,1001);

nhe=0:1000;

stem(nhe,he);

title('he')

（b）



Because x\*h1=y can get x\*h1\*h2=y\*h2=x and h1\*h2 is equal to the unit impulse function, it is only necessary to prove that the convolution of the unit impulse responses of the two systems is a unit impulse function.

**MATLAB Code:**

%(b)

B=1;

A=[1, zeros(1,999) ,0.5];

nz=0:1000;

z=impz(B,A,1001);

A1=1;

B1=[1, zeros(1,999) ,0.5];

he=impz(B1,A1,1001);

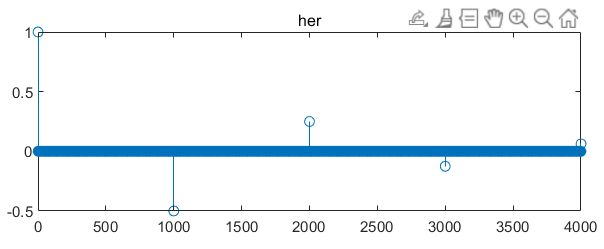
nhe=0:1000;

n=0:2000;

fi=conv(z,he);

stem(n,fi);

（c）



The unit impulse response of z[n]+0.5z[n-1000]=y[n] is her[n], and d = [1 zeroes (1,4000)].

**MATLAB Code:**

B=1;

A=[1, zeros(1,999) ,0.5];

d=[1,zeros(1,4000)]

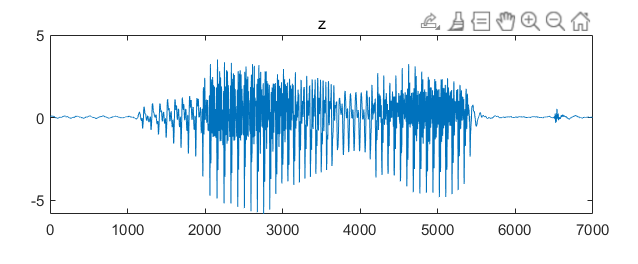
her=filter(B,A,d);

ny=0:4000;

stem(ny,her);

title('her');

（d）



Pass y through the echo cancellation system.

**MATLAB Code:**

%(d)

Bd=[1, zeros(1,999) ,0.5];

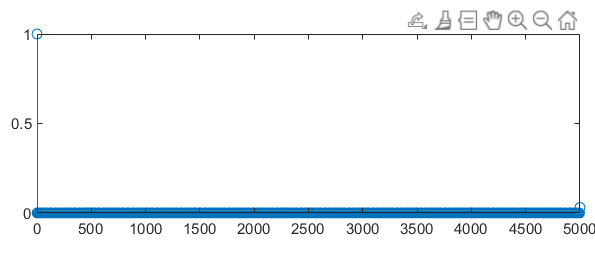
z2=filter(1,Bd,y);

plot(z2);

title('z')

sound(z2,8192);

（e）



Hoa[n] is not a unit impulse response, hoa(5001)! =0, because the unit impulse response (her) of the echo cancellation system is not infinite in this topic, and only the value in the range of 0: 4000 is taken in this topic, resulting in the convolution of he and her having a value at 5001.

**MATLAB Code:**

%(e)

B=1;

A=[1, zeros(1,999) ,0.5];

d=[1,zeros(1,4000)]

her=filter(B,A,d);

A1=1;

B1=[1, zeros(1,999) ,0.5];

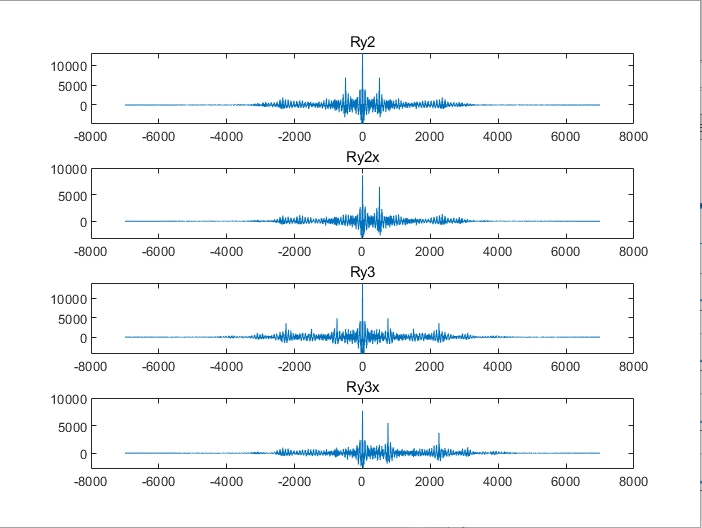
he=impz(B1,A1,1001);

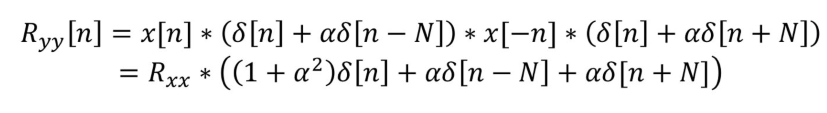
hoa=conv(he,her);

nhoa=0:5000;

stem(nhoa,hoa);

（f）



1. x is obtained through the echo cancellation system by calculating y;
2. 2. Make y2 auto-correlated and cross-correlated with X.
3. 3. By traversing Ry2x, it can be found that the maximum time delay is N=501, and N1=751 and N2=2252 can be obtained in the same way.
4. 4. Because
5. Therefore, ryy [0] = (1+a 2) rxx [0]+arxx [n]+arxx [-n] can be obtained. A=0.7 can be calculated from the data in the figure. In the same way, a1=0.76 and a2=0.61 can be obtained.

**MATLAB Code:**

%(f)

%对于y2而言:

NY=length(y2);

x=filter(1,[1,zeros(1,999),0.5],y);

Rxx=conv(x,flipud(x));

Ry2=conv(y2,flipud(y2));

Ry2x=conv(y2,flipud(x));

figure

subplot(4,1,1);

plot([-NY+1:NY-1],Ry2);

title('Ry2');

subplot(4,1,2);

plot([-NY+1:NY-1],Ry2x);

title('Ry2x');

p=0;

N=0;

for i=10:6999

if abs(Ry2x(7000+i))>p

p=Ry2x(7000+i);

N=i;

end

end

a=Ry2x(NY);

%对于y3而言

Ry3=conv(y3,flipud(y3));

Ry3x=conv(y3,flipud(x));

subplot(4,1,3);

plot([-NY+1:NY-1],Ry3);

title('Ry3');

subplot(4,1,4);

plot([-NY+1:NY-1],Ry3x);

title('Ry3x');

p1=0;

p2=0;

N2=0;

N1=0;

for i=10:6999

if abs(Ry3x(7000+i))>p1

p1=Ry3x(7000+i);

N1=i;

end

end

for i=1000:6999

if abs(Ry3x(7000+i))>p2

p2=Ry3x(7000+i);

N2=i;

end

end