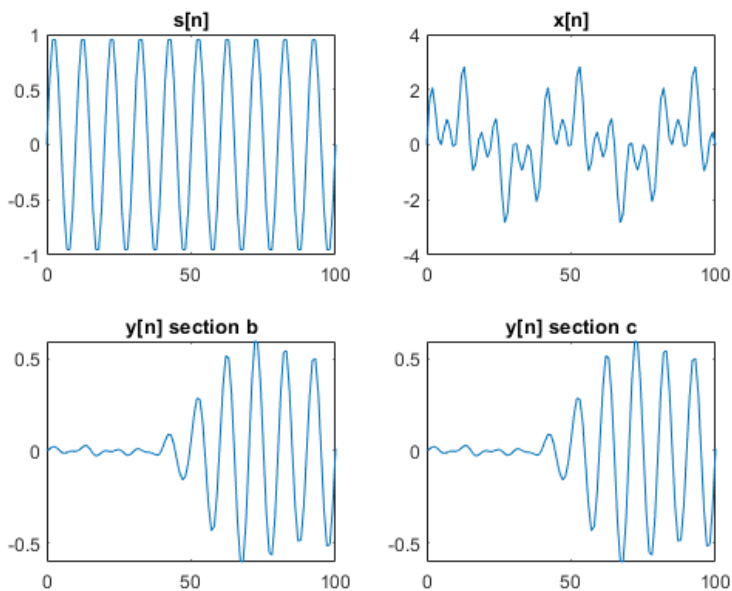


```

%Experiment 2-2
%section a
M = 100;
n = 0:M;
w1 = 0.05*pi;
w2 = 0.2*pi;
w3 = 0.35*pi;
s = sin(w2*n);%s[n]
v = sin(w1*n) + sin(w3*n);%v[n]
x = s + v;%x[n]
subplot(2,2,1)
plot(n,s)
title('s[n]')
subplot(2,2,2)
plot(n,x)
title('x[n]')
%section b
wa = 0.15*pi;
wb = 0.25*pi;
wn1 = 0.54 - 0.46 * sin(2*pi*(n)/M);
hn = wn1 .* ((wb/pi)*sinc((wb/pi)*(n-M/2)) - (wa/pi)*sinc((wa/pi)*(n-M/2)));
yn = filter(hn,1,x);
subplot(2,2,3)
plot(n,yn)
title('y[n] section b')
%section c
load("Experiment2.mat")

yn1 = filter(Num,1,x);
subplot(2,2,4)
plot(n,yn)
title('y[n] section c')

```



```

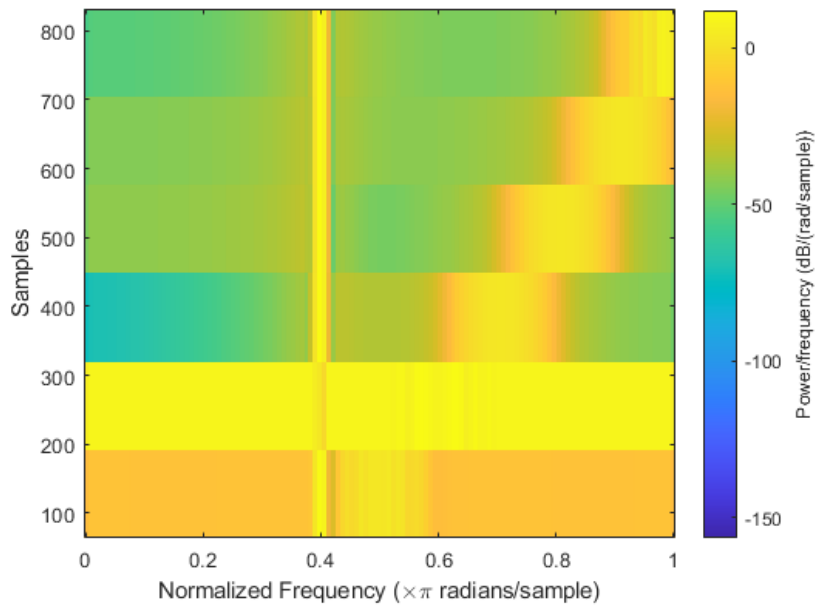
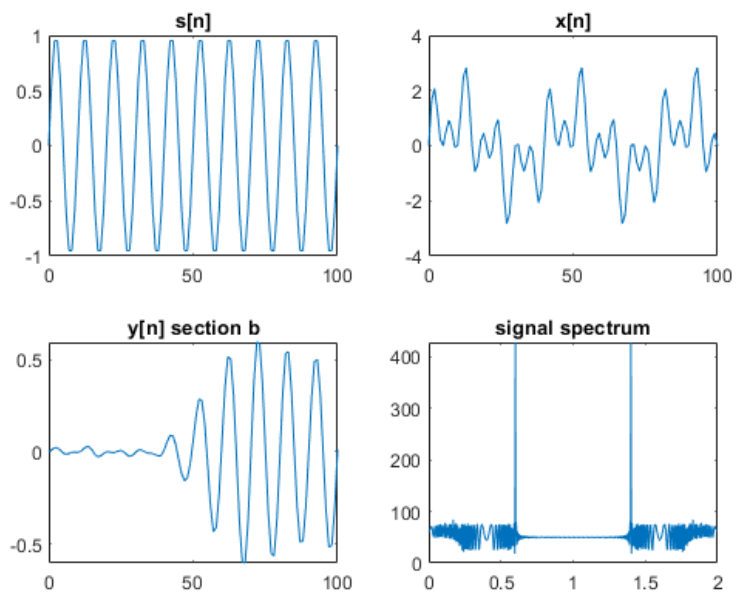
%Experiment 2-3
%section a
[xn,fs] = audioread("Audio01.wav");
f0 = 10000;%cutoff frequency
%section c
load("filter.mat")
y0 = filter(Num1,1,xn);
n = 1:length(xn);
sn = 2*cos(2*pi*f0*n/fs);
y1 = transpose(sn).*y0;
y2 = filter(Num1,1,y1);
%sound(y2,fs)
%section d
y00 = filter(Num1,1,y2);
n = 1:length(y2);
y11 = transpose(sn).*y00;
xn_recovered = filter(Num1,1,y11);
sound(xn_recovered,fs)

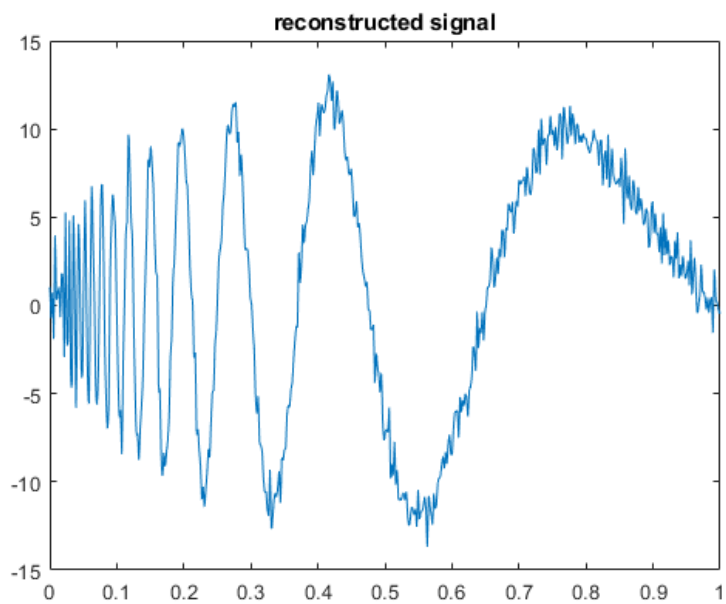
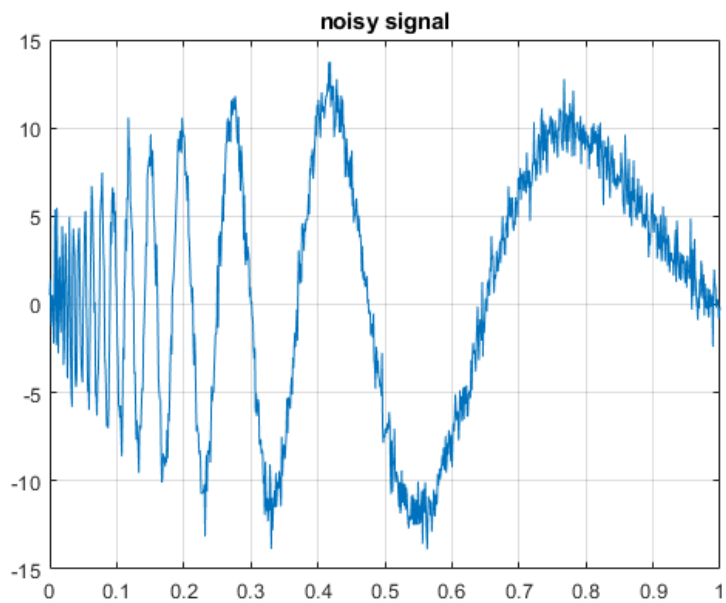
```

```

%Experiment 2-4
%section a
fs = 500;%sampling frequency
n = 0:1/fs:2;
x1 = sin(2*pi*100*n);
%creating delta function
x2 = zeros(1,length(x1));
x2(250) = 50;
x3 = chirp(n,400,2,200);
x = x1 + x2 + x3;
Xf = abs(fftshift(fft(x)));
plot(n,Xf)
title('signal spectrum')
%section b
L = 256;
w1 = hamming(L);
figure;
spectrogram(x,w1);
%%Experiment 2-5
loc = linspace(0,1,2^10);
[xx,noisyx] = wnoise('doppler',10,7);
[cA,cD] = dwt(noisyx,'sym4');
xrec = idwt(cA,zeros(size(cA)),'sym4');
figure;
plot(loc,noisyx)
title('noisy signal')
hold on
grid on
figure;
plot(loc,xrec)
title('reconstructed signal')

```





```
%Experiment 2-1
%section b
h = ones(1, 10) * 0.1;%creating impulse response function
x = zeros(1, 200);%initializing input signal
%creating the input signal
x(1:50)=ones(1,50);
x(101:150)=ones(1,50);
%convolving
y = myconv(h,x);
%input plot
subplot(1,2,1)
stem(0:199,x)
xlabel('n')
ylabel('x[n]')
title('input signal')
%output plot
subplot(1,2,2)
stem(0:length(y)-1,y)
xlabel('n')
ylabel('y[n]')
title('output signal')
%section c
%creating the impulse response
hh = zeros(1,15);
for i=1:15
    hh(i) = 0.25*(0.75)^i;
end
%convolving
yy = myconv(hh,x);
figure;
%input plot
```

```

subplot(1,2,1)
stem(0:199,x)
xlabel('n')
ylabel('x[n]')
title('input signal')
%output plot
subplot(1,2,2)
stem(0:length(yy)-1,yy)
xlabel('n')
ylabel('y[n]')
title('output signal')
%section d
syms n z;
%recreating the input signal in a different form in order to use 'ztrans'
xx = heaviside(n) - heaviside(n-50) + heaviside(n-100) - heaviside(n-150);
%z-transform formula
X = ztrans(xx);
hz = 0.2*(1-1/z)^5;%impulse response in z domain
yz = hz*X;
%Inverse Z-transform to return to time domain
yyy = iztrans(yz);
figure;
%input plot
subplot(1,2,1)
stem(0:199,x)
xlabel('n')
ylabel('x[n]')
title('input signal')
%output plot
subplot(1,2,2)
b=double(subs(yyy, n, 0:199));%n existing in yyy will be substituted by an array and then will be converted to a double from syms
stem(b);
xlabel('n')
ylabel('y[n]')
title('output signal')
%section a
function y = myconv(h,x)
M = length(h)-1;
L = length(x)-1;
y=zeros(1,M+L);%initializing y with zeros
%coding sigma part
for n = 1: M+L
    for m = 1:M+1
        if n-m > 0 && n-m <= length(x)
            y(n) = y(n) + h(m)*x(n-m);
        else
            break;
        end
    end
end
end
end

```

