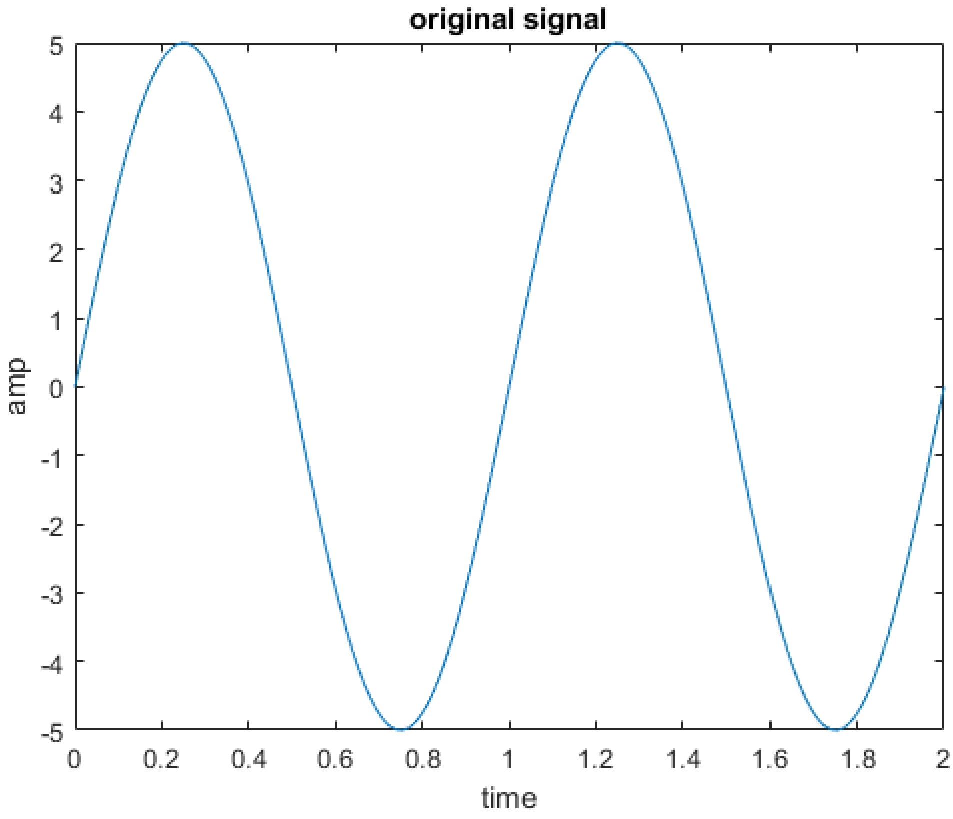
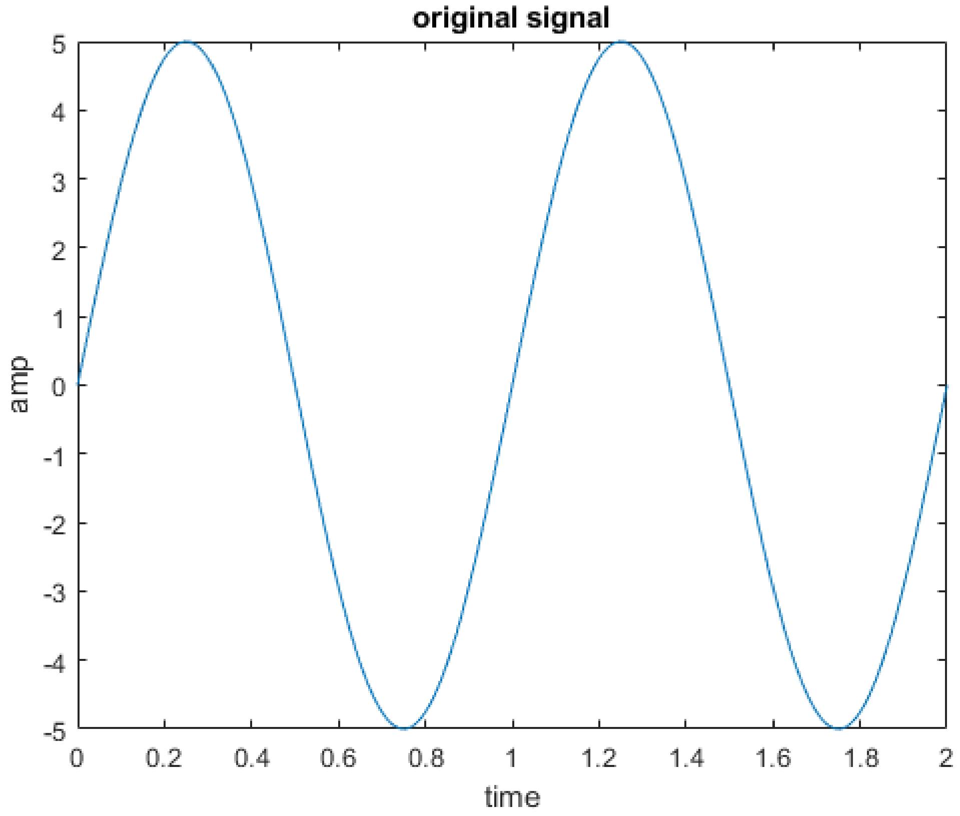
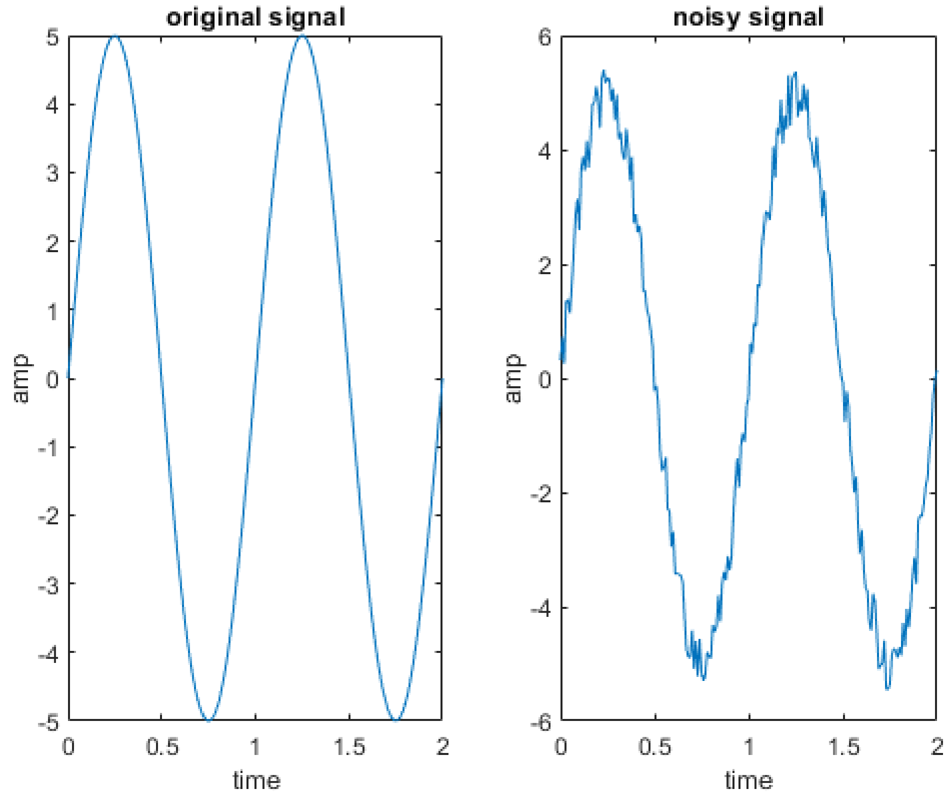
```
clc, clear, close all;
%1-1
t=0:0.01:2;
x=5*sin(2*pi*t);%original signal
figure;
plot(t,x)
title('original signal')
xlabel('time')
ylabel('amp')
```

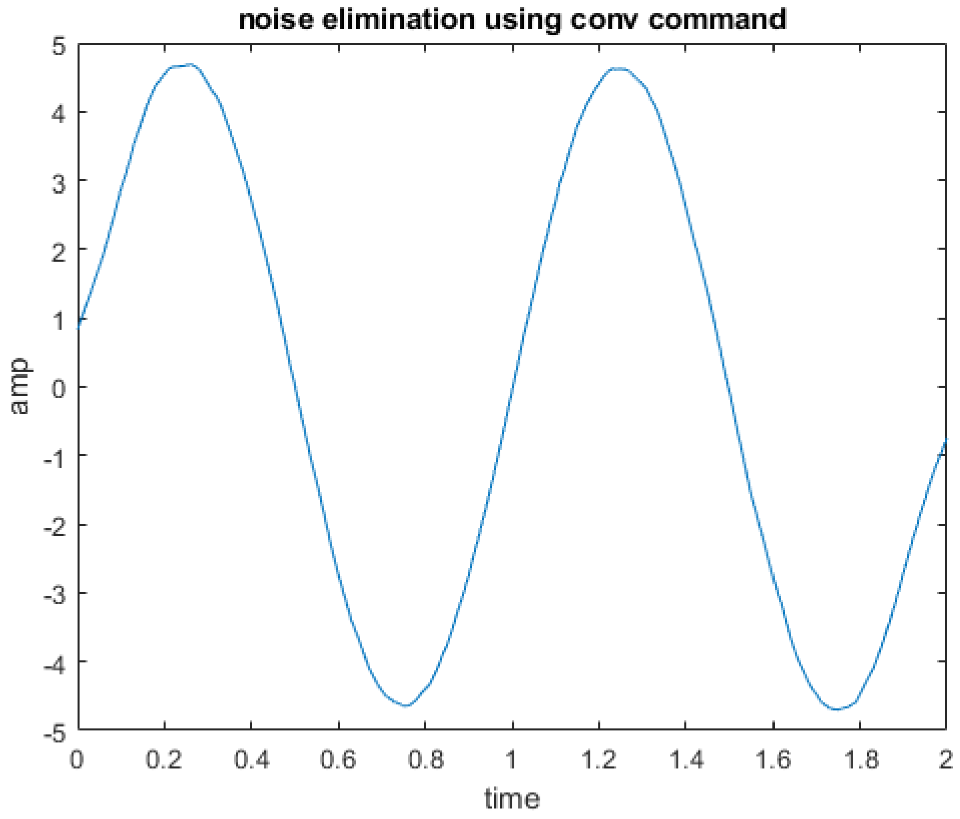


```
%1-2
x_noise=zeros(size(t));
for i=1:length(t)
    randd = rand - 0.5;
    x_noise(i) = x(i) + randd;
end
figure;
subplot(1,2,1)
plot(t,x)
xlabel('time')
ylabel('amp')
title('original signal')
subplot(1,2,2)
plot(t,x_noise)
xlabel('time')
ylabel('amp')
title('noisy signal')
```

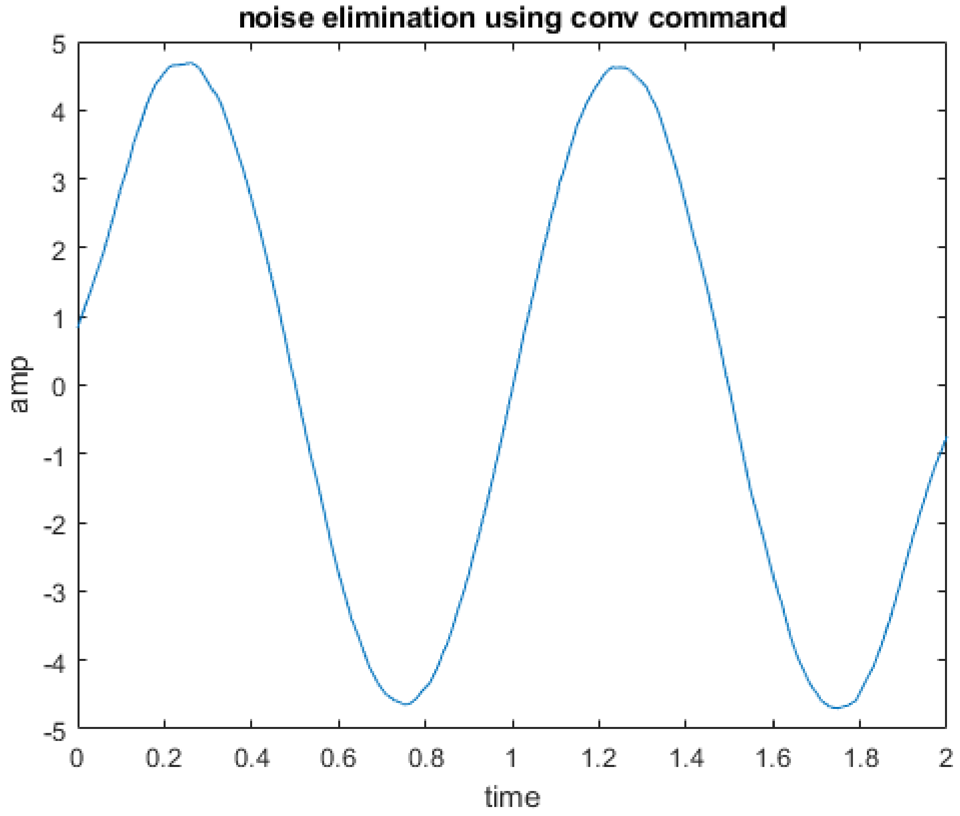


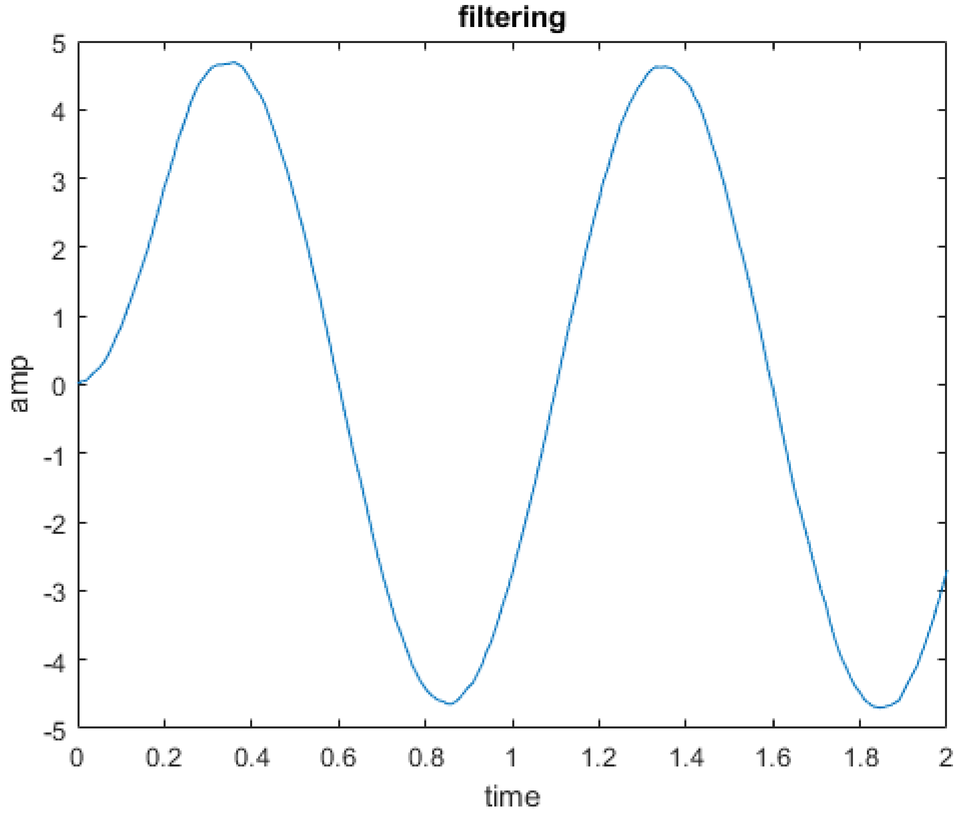


```
%1-3
m1=0;
m2=20;
M=m2-m1+1;
filterCoeff = ones(1, M) / M;
result = conv(x_noise, filterCoeff, 'same');
figure;
plot(t,result)
xlabel('time')
ylabel('amp')
title('noise elimination using conv command')
```



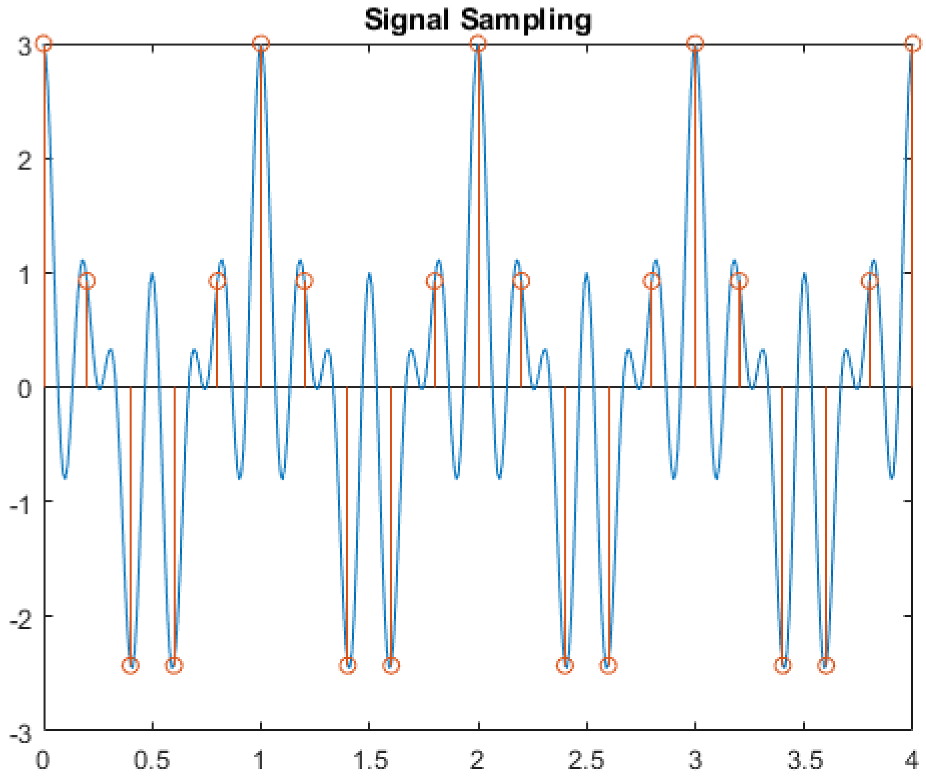
```
%1-4
b = (1/M)*ones(1,M);
a = 1;
y = filter(b,a,x_noise);
figure;
plot(t,y)
title('filtering')
xlabel('time')
ylabel('amp')
```

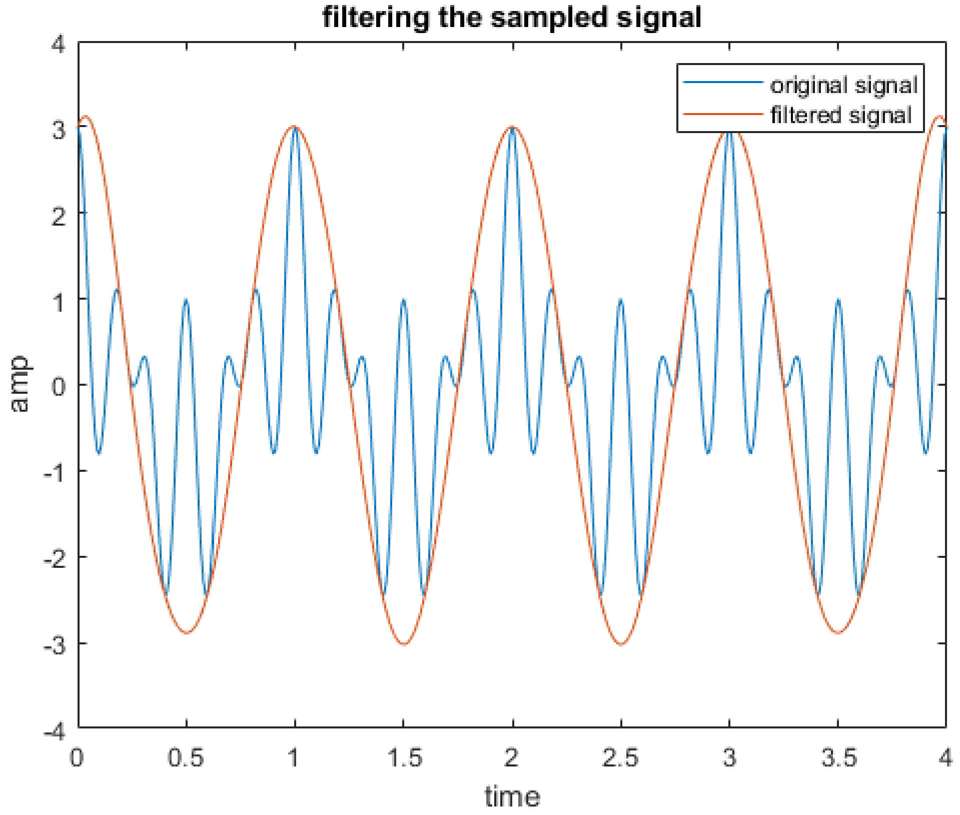




```
%1-6
fs =5 ;% 5KHz Sampling Frequency
T_s = 1/fs;
t_s = 0:T_s:4;
t=0:0.01:4;
t1=0: T_s : 4;
x = cos(2*pi*t) + cos (8*pi*t) + cos(12*pi*t);%original signal
x1 = cos(2*pi*t1) + cos (8*pi*t1) + cos(12*pi*t1);%sampled signal
plot(t,x)
hold on
stem(t1,x1)
```

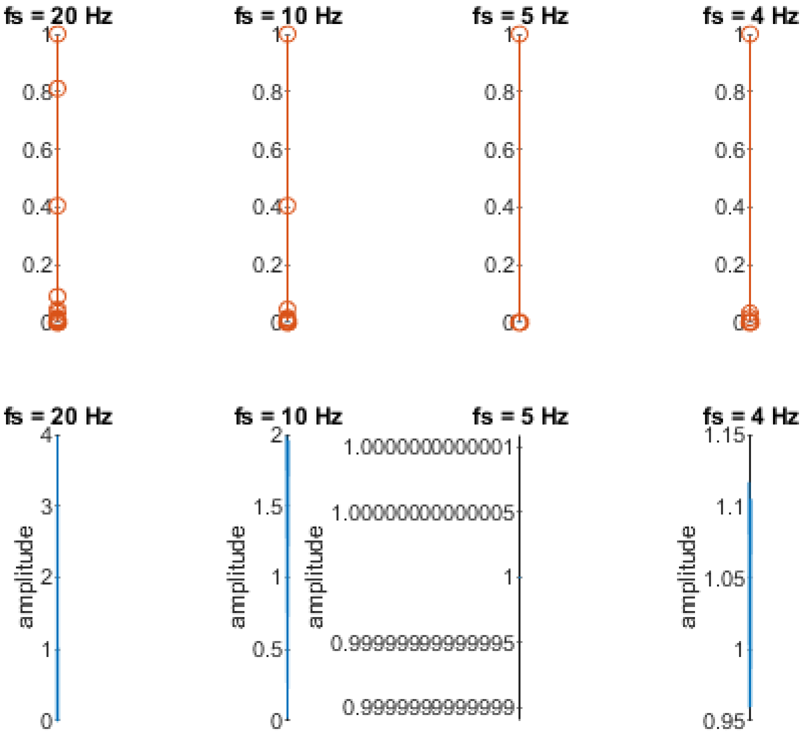
```
title('Signal Sampling')
%recreating the original signal
t_r=zeros(length(t_s),length(t));
for i=1:length(t_s)
    t_r(i,:) = t/T_s - t_s(i)/T_s;
end
h_r = sinc(t_r);
x_r = x1 * h_r;
figure;
plot(t,x,t,x_r)
title('filtering the sampled signal')
legend('original signal','filtered signal')
xlabel('time')
ylabel('amp')
```



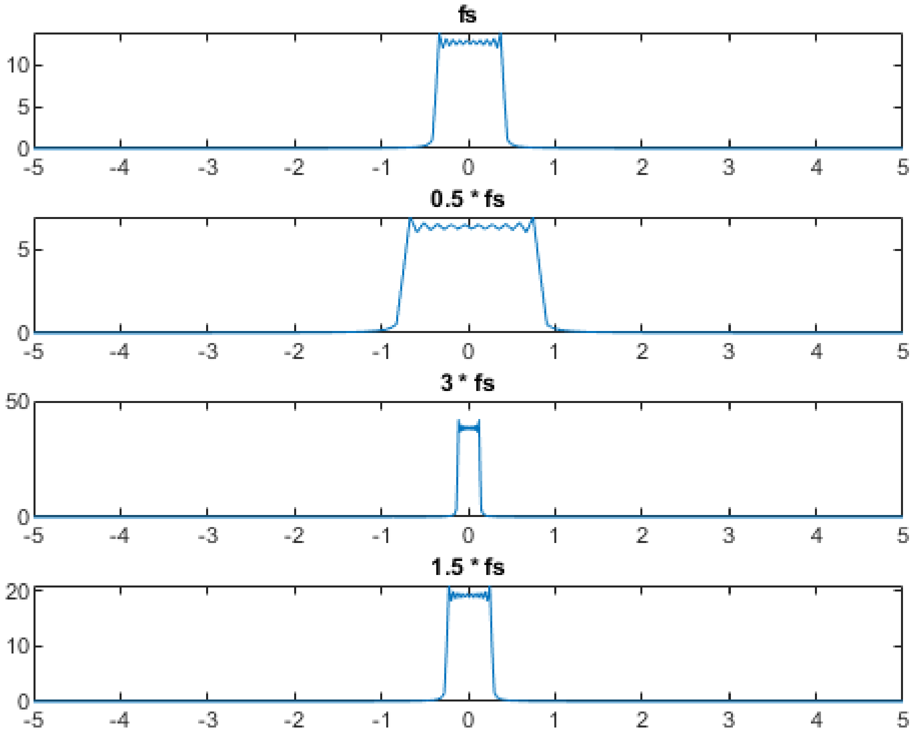


```
%1-7
fs1=20;
fs2=10;
fs3=5;
fs4=4;
t=-5:0.01:5;
t1=-5: 1/fs1 : 5;
t2=-5: 1/fs2 : 5;
t3=-5: 1/fs3 : 5;
t4=-5: 1/fs4 : 5;
x = sinc(5*t).^2;%original signal
```

```
x1 = sinc(5*t1).^2;
x2 = sinc(5*t2).^2;
x3 = sinc(5*t3).^2;
x4 = sinc(5*t4).^2;
%time domain
subplot(2,4,1)
plot(t,x)
hold on
stem(t1,x1)
title('fs = 20 Hz')
subplot(2,4,2)
plot(t,x)
hold on
stem(t2,x2)
title('fs = 10 Hz')
subplot(2,4,3)
plot(t,x)
hold on
stem(t3,x3)
title('fs = 5 Hz')
subplot(2,4,4)
plot(t,x)
hold on
stem(t4,x4)
title('fs = 4 Hz')
%frequency domain
subplot(2,4,5)
plot(t1,abs(fftshift((fft(x1)))))
xlabel('frequency')
ylabel('amplitude')
title('fs = 20 Hz')
subplot(2,4,6)
plot(t2,abs(fft(x2)))
xlabel('frequency')
ylabel('amplitude')
title('fs = 10 Hz')
subplot(2,4,7)
plot(t3,abs(fft(x3)))
xlabel('frequency')
ylabel('amplitude')
title('fs = 5 Hz')
subplot(2,4,8)
plot(t4,abs(fft(x4)))
xlabel('frequency')
ylabel('amplitude')
title('fs = 4 Hz')
```

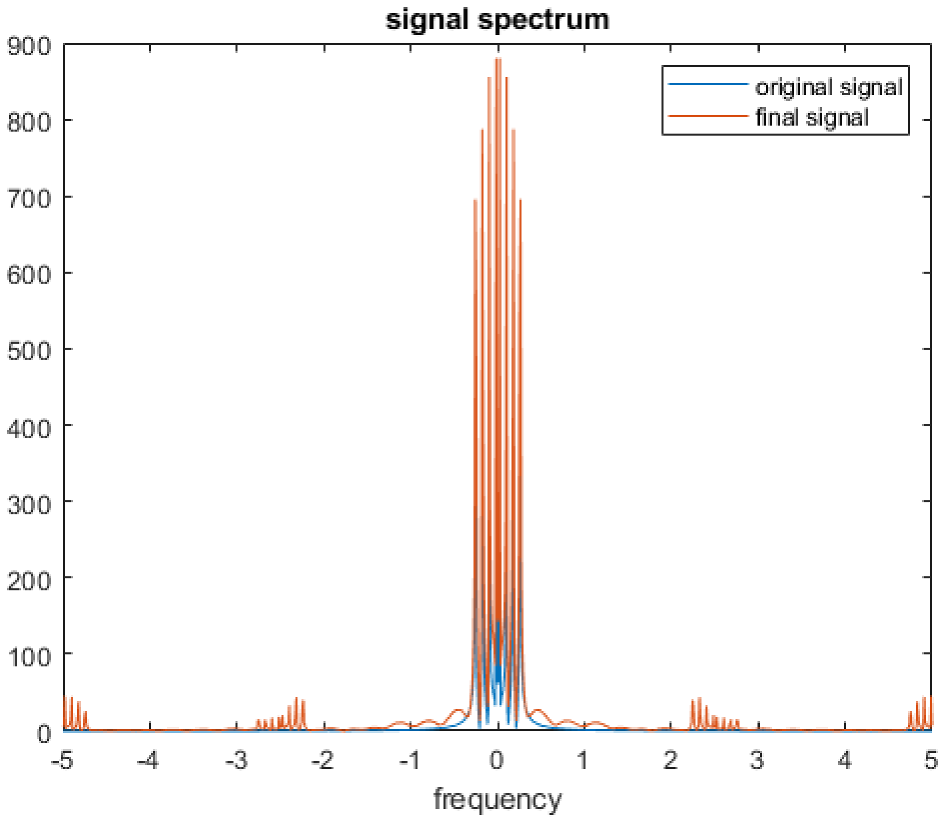


```
%1-8
figure;
n = 256;
n1 = n*0.5;
n2 = n*3;
n3 = n*1.5;
tt=linspace(-5,5,n);
tt1=linspace(-5,5,n1);
tt2=linspace(-5,5,n2);
tt3=linspace(-5,5,n3);
y = sinc(2*tt);%original signal
y1 = sinc(2*tt1);
y2 = sinc(2*tt2);
y3 = sinc(2*tt3);
subplot(4,1,1)
yy = abs(fftshift(fft(y)));
plot(tt,yy)
title('fs')
subplot(4,1,2)
yy1 = abs(fftshift(fft(y1)));
plot(tt1,yy1)
title('0.5 * fs')
subplot(4,1,3)
yy2 = abs(fftshift(fft(y2)));
plot(tt2,yy2)
title('3 * fs')
subplot(4,1,4)
yy3 = abs(fftshift(fft(y3)));
plot(tt3,yy3)
title('1.5 * fs')
```

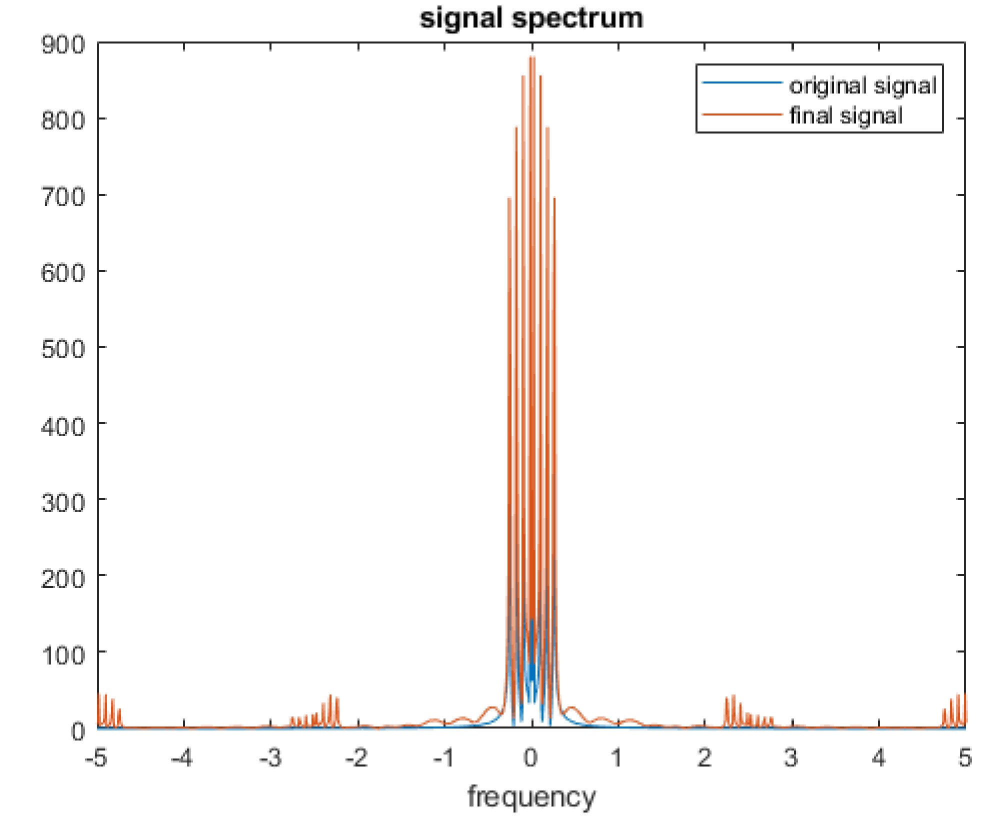


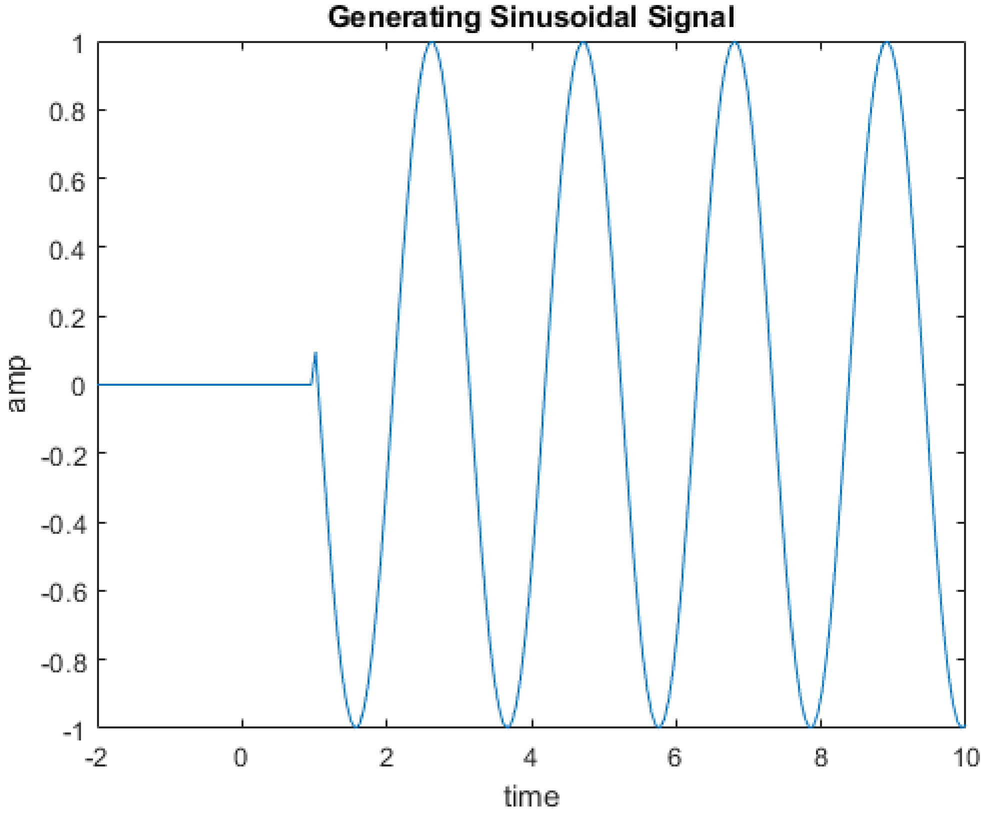
```
%1-9a
clc;
close all;
t = linspace(-5,5,1000);
f1 = pi/16;
f2 = 5*pi/16;
f3 = 9*pi/16;
f4 = 13*pi/16;
x_t1 = cos(2*pi*f1*t);
x_t2 = cos(2*pi*f2*t);
x_t3 = cos(2*pi*f3*t);
x_t4 = cos(2*pi*f4*t);
x_t = x_t1 + x_t2 + x_t3 + x_t4;
x_f = fftshift(fft(x_t));
plot(t,abs(x_f))
hold on
title('signal spectrum')
xlabel('frequency')
%1-9b
filter1 = xlsread('filters',1);%analysis filter
filter2 = xlsread('filters',2);%synthesise filter
%1-9c
a=1;
b1=filter1(1,:);
b2=filter1(2,:);
b3=filter1(3,:);
b4=filter1(4,:);
%filtering
x_filter1 = filter(b1,a,x_t);
x_filter2 = filter(b2,a,x_t);
x_filter3 = filter(b3,a,x_t);
x_filter4 = filter(b4,a,x_t);
%downsampling
x_down1 = downsample(x_filter1,4);
x_down2 = downsample(x_filter2,4);
x_down3 = downsample(x_filter3,4);
x_down4 = downsample(x_filter4,4);
%processing
```

```
x_pro1 = x_down1*2;
x_pro2 = x_down2*0;
x_pro3 = x_down3*1;
x_pro4 = x_down4*0.5;
%upsampling
x_up1 = upsample(x_pro1,4);
x_up2 = upsample(x_pro2,4);
x_up3 = upsample(x_pro3,4);
x_up4 = upsample(x_pro4,4);
%filtering again
c1=filter2(1,:);
c2=filter2(2,:);
c3=filter2(3,:);
c4=filter2(4,:);
x_final1 = filter(c1,a,x_up1);
x_final2 = filter(c2,a,x_up2);
x_final3 = filter(c3,a,x_up3);
x_final4 = filter(c4,a,x_up4);
x_final = x_final1 + x_final2 + x_final3 + x_final4;
plot(t,abs(fftshift(fft(x_final))))
legend('original signal','final signal')
```



```
%1-5
singen(3,200)
function [] = singen(w,n)
%1-5
%creating unit function
n1 = linspace(-2,10,n);
u = zeros(size(n1));
u(n1>=1) = 1;
y= u.*sin(w*n1);
figure;
plot(n1,y)
title('Generating Sinusoidal Signal')
xlabel('time')
ylabel('amp')
end
```





Published with MATLAB® R2020b