MATLAB Programs

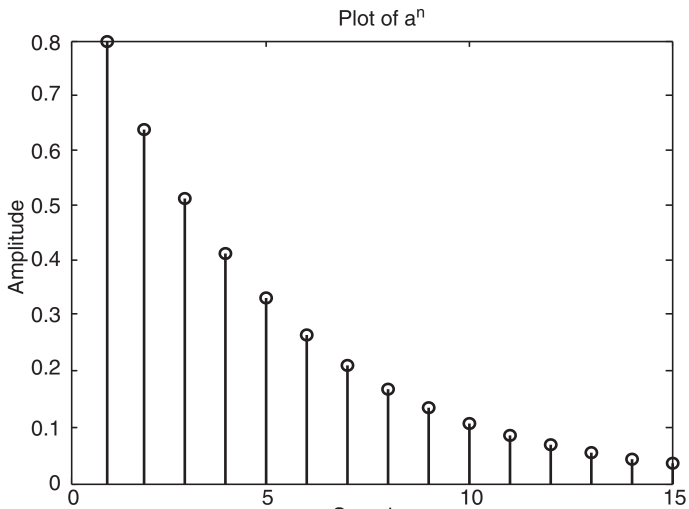
Program 1

Write a program to show aliasing in time domain using sampling theorem in frequency domain and plot the signal.

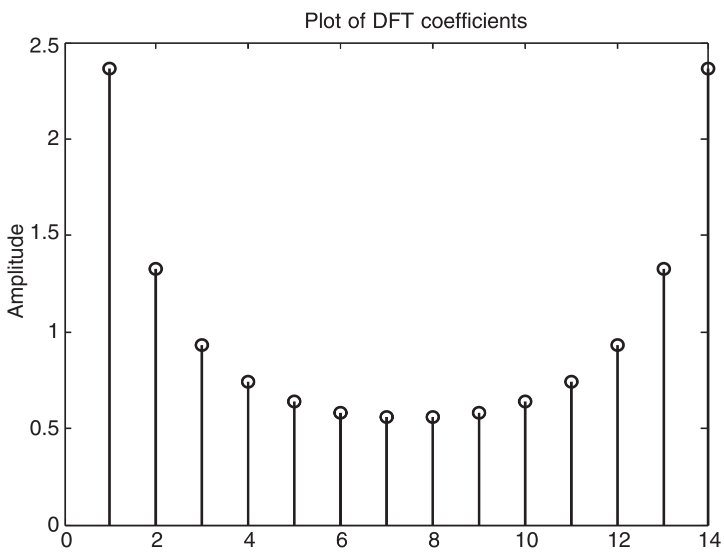
```
%alasing in time domain-sampling theorem in frequency domain
clear all;
N=15;
for i=1:15,
    x(i)=(0.8)^i;
end
stem(x);
xlabel('sample no.');
ylabel('amplitude');
title('plot of a^n');

figure;
for i=1:14,
    y(i)=abs(1/(1-(0.8)*(exp(-j*2*pi*i/N))));
end
stem(y);
xlabel('freq. sample no.');
ylabel('amplitude');
title('plot of DFT coefficients');
for i=1:14,
x1(i)=(0.8)^i/(1-(0.8)^15);
end
figure;
stem(x1);
xlabel('sample no.');
ylabel('amplitude');
title('plot of reconstructed x(n)');
for i=1:3,
disp(x(i));
disp(x1(i));
end
x2=x(1)+(0.8)^14;
disp(x2);
sum=0.0;
for w=0.1:0.1:2*pi,
    for l=1:15,
        sum(w)=1/(1-((0.8)*exp(-j*w)));
        end
```

```
end
figure;
plot(sum);
```

Output



(a)



(b)

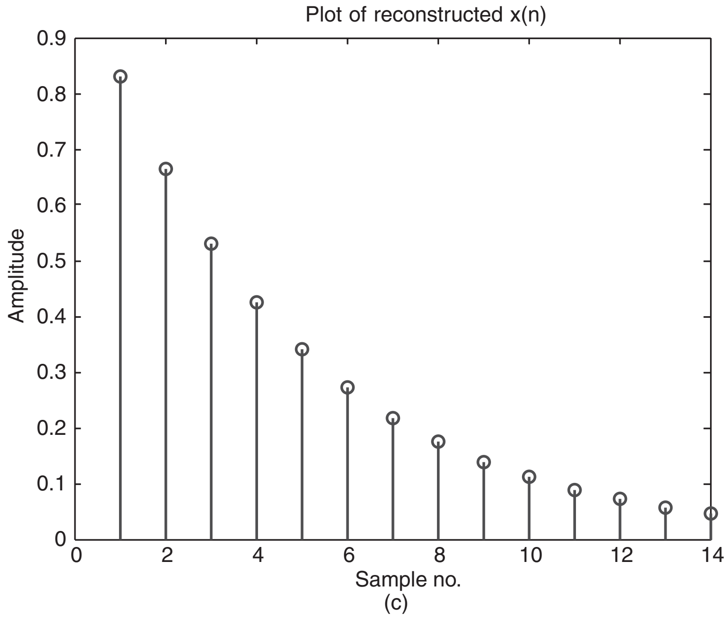


Figure 1 (a) Plot of the signal $x(n)$ using a MATLAB program; (b) plot of the DFT output $X(k)$ using a MATLAB program; (c) plot of the recovered aliased signal $x(n)$ using a MATLAB program.

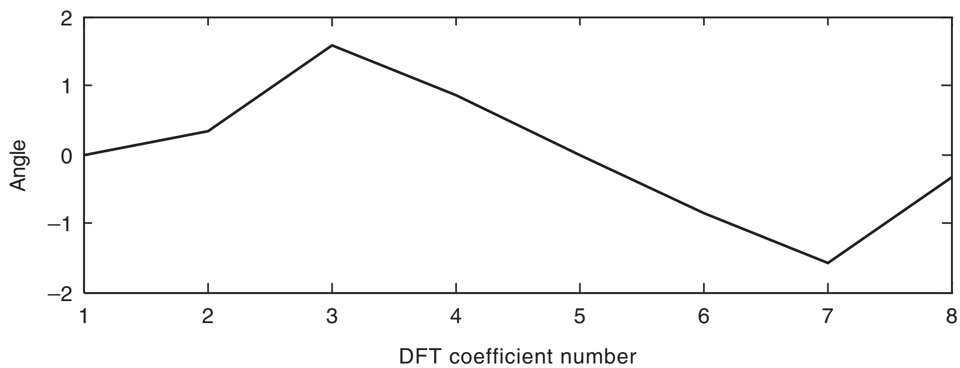
Program 2

Write a program to show the time-shifting property of DFT and plot the magnitude response of original and time-shifted sequence.

```
%Time shifting property
clear all;
x=[1 2 3 4 5 2 3 1];
y=abs(fft(x));
y3=fft(x);
subplot(2,1,1);
plot(y);
title('DFT of x(n)');
xlabel('DFT coefficient number');
ylabel('Amplitude');
x1=[3 4 5 2 3 1 1 2]
y2=fft(x1);
y1=abs(fft(x1));
%disp(y1);
subplot(2,1,2);
```

```
plot(y1);
disp(y2)
disp(y3);
figure;
for i=1:8,
z(i)=atan(imag(y3(i))/real(y3(i)));
z1(i)=atan(imag(y2(i))/real(y2(i)));
end
%disp(z);
%disp(z1);
subplot(2,1,1);
plot(z);
title('DFT of x(n)');
xlabel('DFT coefficient number');
ylabel('angle');
subplot(2,1,2);
plot(z1)
title('DFT of x(n)');
xlabel('DFT coefficient number');
ylabel('Angle');
```

Output



(a)

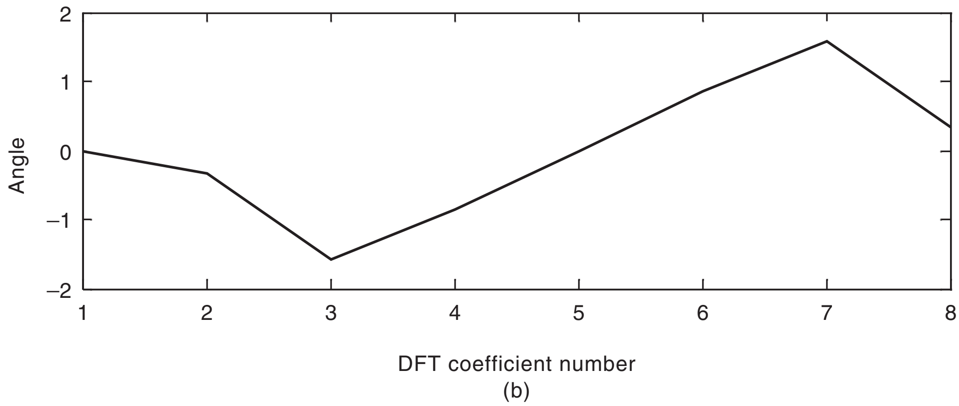


Figure 2 Phase plots of (a) original and (b) time-reversed sequence.

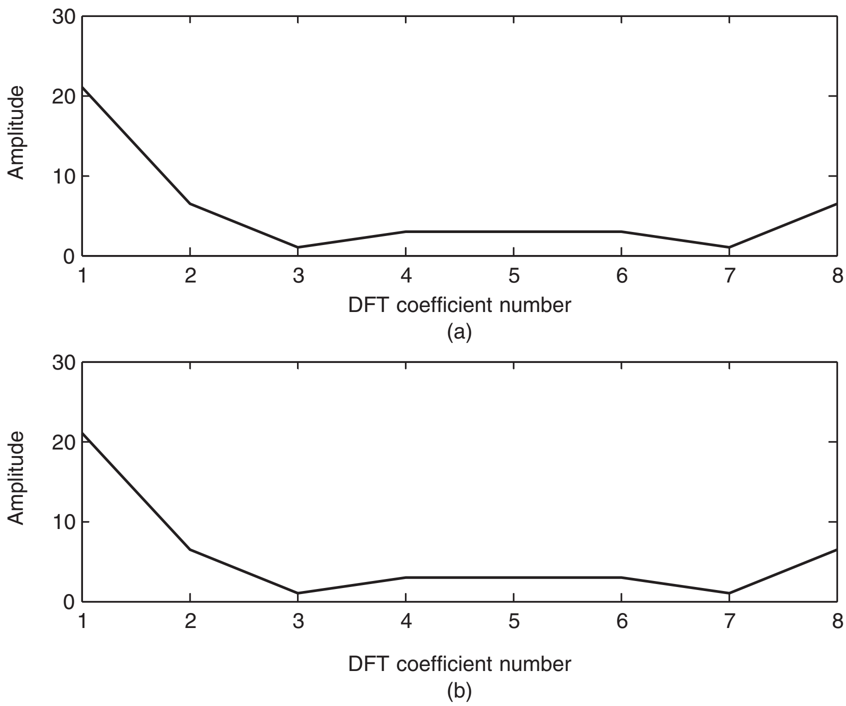


Figure 3 Magnitude plots for DFT of (a) original and (b) time-shifted sequence.

Program 3

Write program to show the frequency-shifting property of DFT.

```
%frequency shifting property
clear all;
x=[1 2 3 4 5 2 3 1];
y=abs(fft(x));
y3=fft(x);
subplot(2,1,1);
plot(y);
title('DFT of x(n)');
xlabel('DFT coefficient number');
ylabel('Amplitude');
for i=1:8,
    x1(i)=x(i)*exp(-j*2*pi*i*2/8);
end
y1=abs(fft(x1));
y2=fft(x1);
subplot(2,1,2);
plot(y1);
title('DFT of x1(n)');
xlabel('DFT coefficient number');
ylabel('Amplitude');
a=[1 2 3 4 5  2 3 1];
for i= 1:8,
    a=a*exp((j*2*pi*2*i)/8);
end
figure;
subplot(2,1,1);
stem(abs(a));
title('magnitude plot of the signal');
xlabel('sample number');
ylabel('amplitude');
for i=1:8,
p(i)=2*pi*2*i/8;
p(i)=p(i)*180/pi;
end
subplot(2,1,2);
stem(p);
title('phase plot of the signal');
xlabel('sample number');
ylabel('degrees');
```

Output

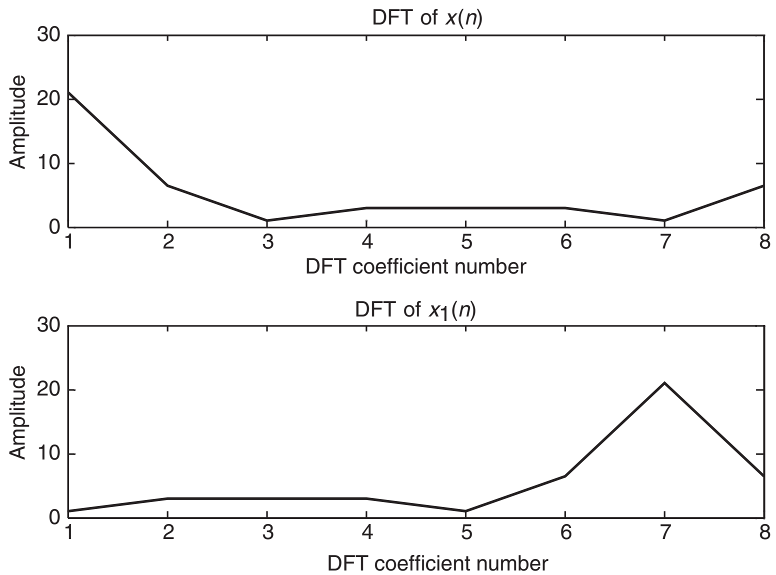


Figure 4 The magnitude plots for the original and the modified sequence.

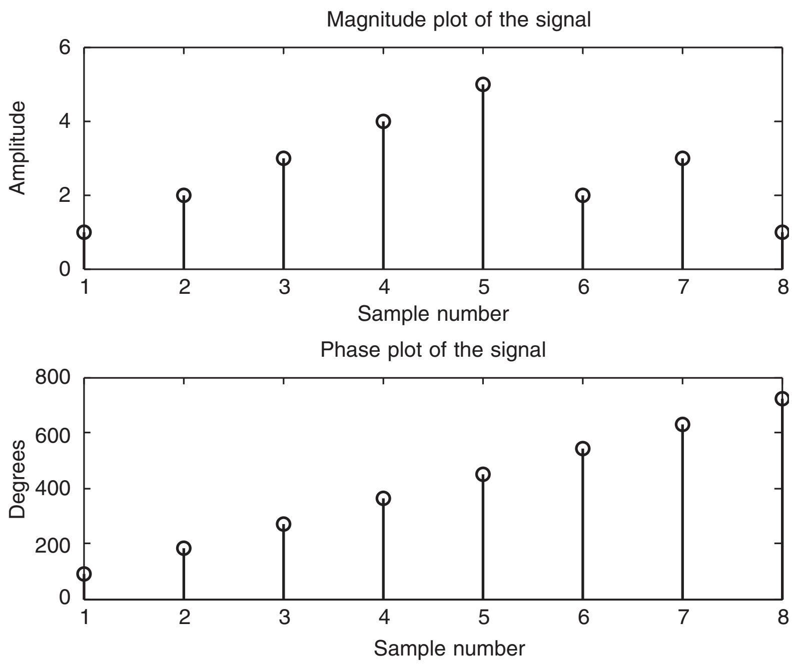


Figure 5 Magnitude and phase plots of the modified sequence in time domain.

Program 4

Write a program to explain the sampling theorem in frequency domain.

```
%sampling theorem in Frequency domain
clear all;
x=[1 2 2 1 ];
h=[ 1 2 1 0 ]
x1=fft(x);
h1=fft(h);
for i=1:4,
z(i)=x1(i)*h1(i);
end
disp(z);
w=ifft(z);
disp(abs(w));
x=[1 2 2 1 0 0 0 0 ];
h=[ 1 2 1 0 0 0 0 0];
x1=fft(x,8);
h1=fft(h,8);
for i=1:8,
z(i)=x1(i)*h1(i);
end
disp(z);
w=ifft(z);
disp(abs(w));
```

Output

The result of circular convolution using 4 point FFT is

5 5 7 7

The result of circular convolution using 8 point FFT is

1 4 7 7 4 1

We see that when we take a 4-point FFT, the first and the fifth points in 8-point FFT output are added to get the first sample in 4-point FFT (1 + 4 = 5). The second and the sixth points in 8-point FFT output are added to get the second sample in 4-point FFT (4 + 1 = 5).

Program 5

Write a program to explain the use of overlap and add algorithm.

```
%overlap &add algorithm
clear all;
a=[1 2 3 4 5 1 2 3 4 5 1 2 3 4 5];
b=[3 2 1 1 0 0 0 0];
c=conv(a,b);
```

```
disp(c);
b1=fft(b,8);
a1=[1 2 3 4 5 0 0 0];
a11=fft(a1,8);
for i=1:8,
   a111(i)=a11(i)*b1(i);
end
c1=ifft(a111,8);
disp(c1);
a2=[1 2 3 4 5 0 0 0];
a22=fft(a2,8);
for i=1:8,
   a222(i)=a22(i)*b1(i);
end
c2=ifft(a222,8);
disp(c2);
a3=[1 2 3 4 5 0 0 0];
a33=fft(a3,8);
for i=1:8,
   a333(i)=a33(i)*b1(i);
end
   c3=ifft(a333,8);
   disp(c3);
   %disp(c4);
for i=1:5,
   d(i)=c1(i);
end
for i=6:8,
   d(i)=c2(i-5)+c1(i);
end
for i=9:10,
   d(i)=c2(i-5);
end
for i=11:13,
   d(i)=c3(i-10)+c2(i-5);
end
for i=14:18,
   d(i)=c3(i-10);
end
disp(d);
```

Output

The resultant output using MATLAB program is

3 8 14 21 28 20 17 19 21 28 20 17 19 21 28 17 9 5 0 0

Program 6

Write a program to explain the use of overlap and save algorithm.

```
%overlap & save algorithm
clear all;
%a=[1 2 3 4 5 1 2 3 4 5 1 2 3 4 5];
a=[1 2 3 2 3 4 3 2 1 4 5 6 7 8 9];
b=[1 2 1 0 0 0 0 0];
c=conv(a,b);
disp(c);
b1=fft(b,8);
%a1=[0 0 0 1 2 3 4 5];
a1=[0 0 0 1 2 3 2 3];
a11=fft(a1,8);
for i=1:8,
   a111(i)=a11(i)*b1(i);
end
c1=ifft(a111,8);
disp(c1);
%a2=[3 4 5 1 2 3 4 5];
a2=[3 2 3 4 3 2 1 4];
a22=fft(a2,8);
for i=1:8,
   a222(i)=a22(i)*b1(i);
end
c2=ifft(a222,8);
disp(c2);
%a3=[3 4 5 1 2 3 4 5];
a3=[2 1 4 5 6 7 8 9];
a33=fft(a3,8);
for i=1:8,
   a333(i)=a33(i)*b1(i);
end
   c3=ifft(a333,8);
   disp(c3);
   %a4=[3 4 5 0 0 0 0 0];
   a4=[7 8 9 0 0 0 0 0 ];
   a44=fft(a4,8);
   for i=1:8,
```

```
        a444(i)=a44(i)*b1(i);
     end
     c4=ifft(a444,8);
     disp(c4);
     for i=1:5,
     d(i)=c1(3+i);
  end
  for i=6:10,
     d(i)=c2(i-2);
  end
  for i=11:15,
     d(i)=c3(i-7);
  end
  for i=16:20,
     d(i)=c4(i-12);
  end
  disp(d);
```

Output

The result of MATLAB program is

[3 8 14 21 28 20 17 19 21 28 20 17 19 21 28 17 9 5]

Program 7

Write a program to explain the use of Goertzel algorithm.

```
% Goertzel algorithm
clear all;
x=[1 2 3 4 1 2 3 4];N=8;
y=fft(x);
disp(y);
b=[1 -exp(-j*2*pi*2/8)];
a=[1 -2*cos(2*pi*2/8) 1];
y=filter(b,a,x);
disp(y(8));
```

Output

```
20.0000          0          -4.0000 + 4.0000i
 Columns 4 through 6
     0       -4.0000              0
 Columns 7 through 8
 -4.0000 - 4.0000i     0
  4.0000 + 4.0000i   →    X(2)
```

Here, DFT coefficient $X(2)$ is obtained as the Nth output, that is, eighth point output is same as $X(2)$.

Program 8

Write a program to plot frequency contents of a signal using different window sizes with different resolutions.

```
%program to plot frequency contents of signal
clear all;
N1=50; N2=100; N3=200;
fs=800; f1=200; f2=210; f3=300;
for n=1:100,
x(n)=cos(2*pi*f1*n/fs)+cos(2*pi*f2*n/fs)+cos(2*pi*f3*n/fs);
end
plot(x);title('Signal magnitude'); xlabel('sample number');
ylabel('magnitude');
figure;
y1=fft(x,N1);
y2=fft(x,N2);
y3=fft(x,N3);
for n=1:50,
f11(n)=n*800/50;
end
for n=1:100,
f22(n)=n*800/100;
end
for n=1:200,
f33(n)=n*800/200;
end
plot(f11,abs(y1));title('Magnitude spectrum of the signal
with window length=0.0625'); xlabel('frequency');ylabel('ma
gnitude');
figure;
plot(f22,abs(y2));title('Magnitude spectrum of the signal
with window length=0.125'); xlabel('frequency');ylabel('mag
nitude');
figure;
plot(f33,abs(y3));title('Magnitude spectrum of the signal
with window length=0.25'); xlabel('frequency');ylabel('magn
itude');
```

Output

We see that when the window size is 50 we cannot resolve 200 Hz and 210 Hz. But, using window size of 100 and 200 we can resolve 200 Hz and 210 Hz.

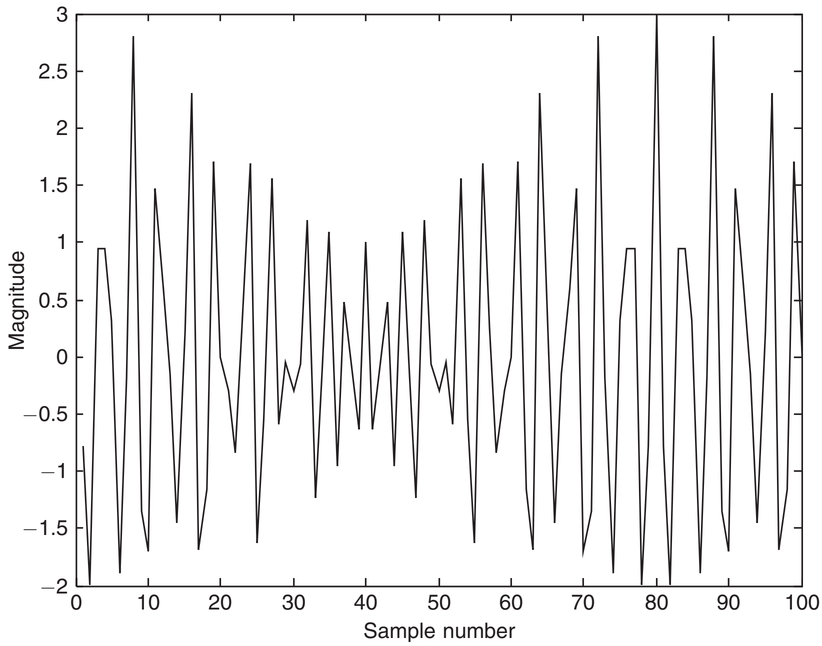


Figure 6 Plot of signal with 200, 210 and 300 Hz.

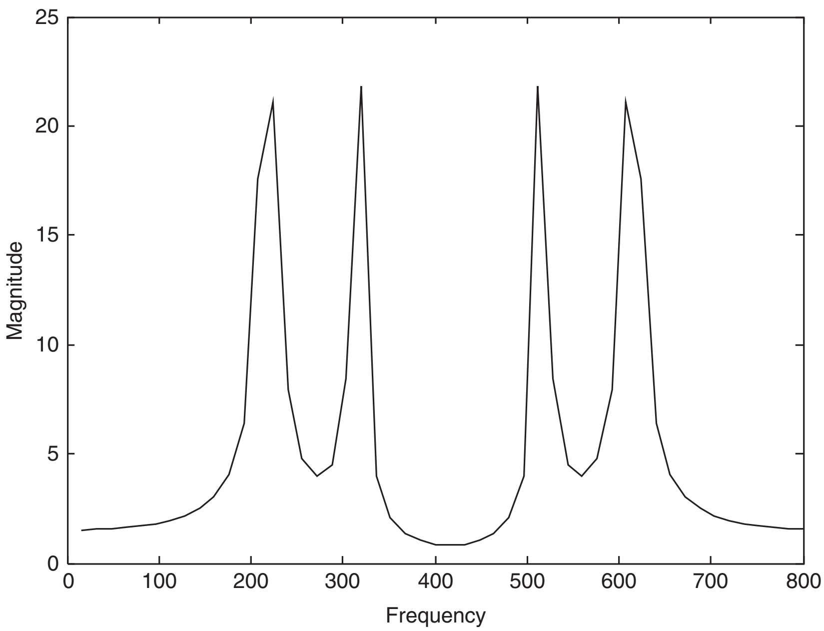


Figure 7 Plot of signal spectrum with window length of 0.0625 and resolution of 16 Hz.

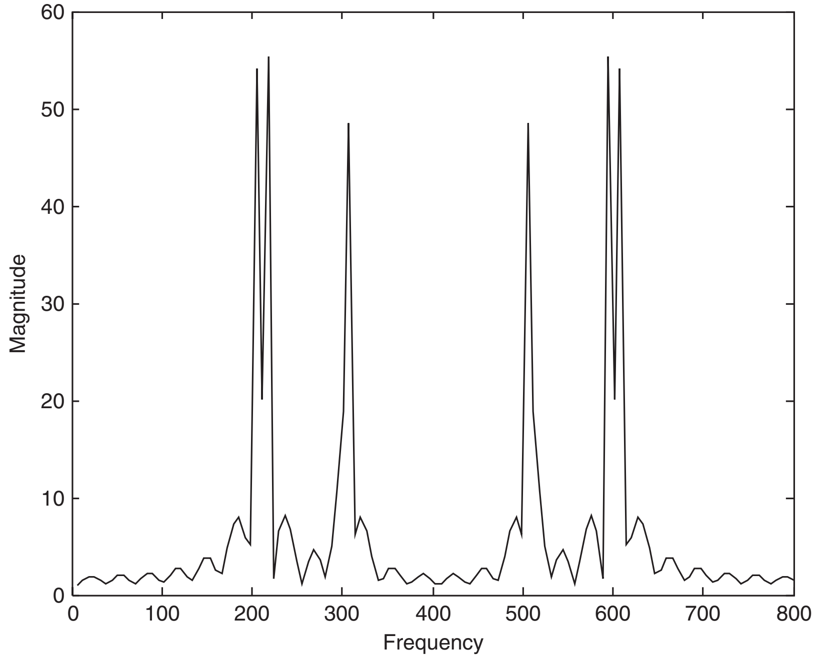


Figure 8 Plot of signal spectrum with window length of 0.15625 and resolution of 6.4 Hz.

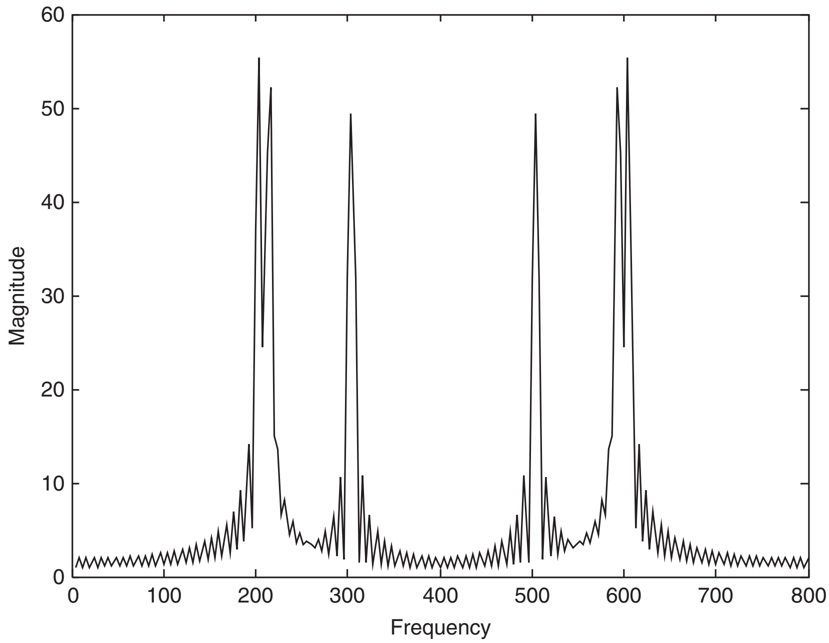


Figure 9 Plot of signal spectrum with window length of 0.25 and resolution of 4 Hz.

Program 9

Write a program to do the frequency analysis of signal with window length selected to resolve all frequencies.

```
%Frequency analysis of signal with window length selected to
resolve all frequencies.
 clear all;
N1=20;
fs=1000; f1=200; f2=300; f3=500;
for n=1:100,
x(n)=cos(2*pi*f1*n/fs)+cos(2*pi*f2*n/fs)+cos(2*pi*f3*n/fs);
end
plot(x);title('Signal magnitude'); xlabel('sample number');
ylabel('magnitude');
figure;
y1=fft(x,N1);
for n=1:20,
f11(n)=n*1000/50;
end
plot(f11,abs(y1));title('Magnitude spectrum of the signal
with window length=0.005-frequency resolution of 20 Hz');
xlabel('frequency');ylabel('magnitude');
```

Output

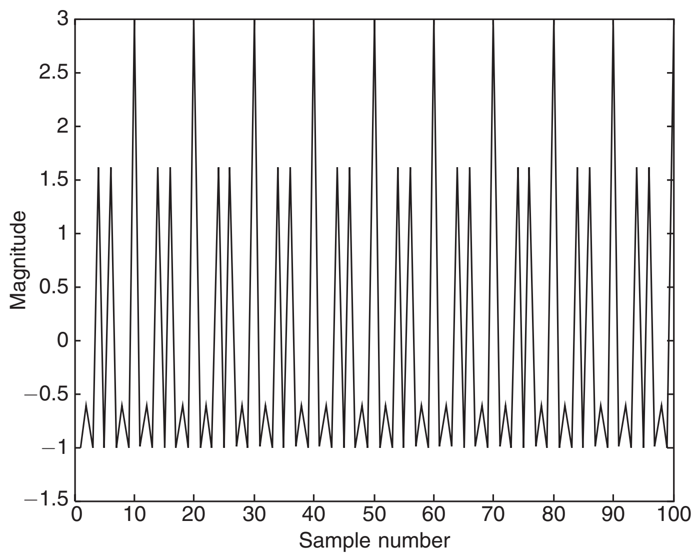


Figure 10 Plot of signal magnitude vs. sample number.

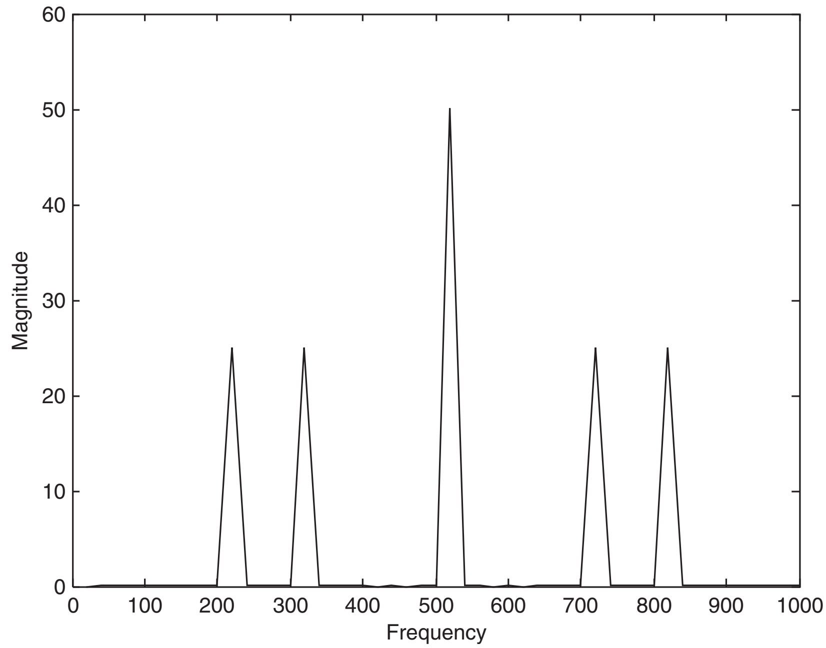


Figure 11 Plot of signal spectrum with window length of 0.005 and resolution of 20 Hz.

Program 10

Write a program to find the frequency contents of speech signal.

```
%to find frequency contents of speech signal
clear all;
fp=fopen('watermark.wav','r');
fseek(fp,44,-1);
a=fread(fp,1024);
plot(a);title('plot  of  speech  signal');xlabel('sample
number'); ylabel('Amplitude');
b=fft(a);
figure;
c=abs(b);
for i=1:1023,
    d(i)=c(i+1);
f(i)=(i/512)*4000;
end
plot(f,d);title('plot of frequency contents of speech
signal');xlabel('frequency in Hz'); ylabel('Amplitude');
```

Output

Here, we see that we have used a wav file watermark.wav. This file is provided in the CD. The student must move this file to the work directory of MATLAB to execute the program.

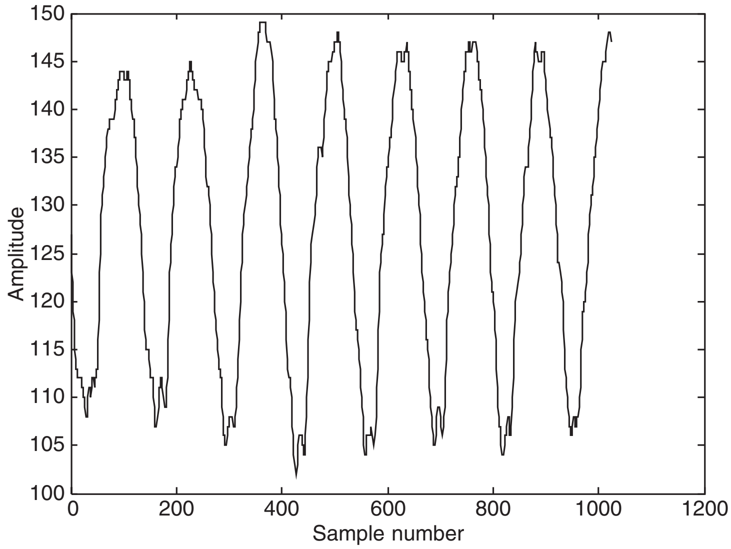


Figure 12 Plot of a speech signal.

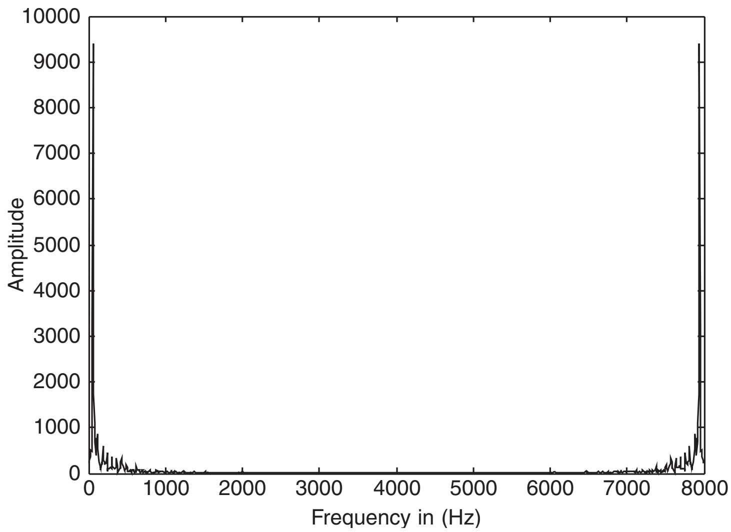


Figure 13 Plot of speech signal spectrum with resolution of 7.8125 Hz.