Hacettepe University

Department of Electrical and Electronics Engineering

ELE 409 Digital Signal Proccessing Laboratory

EXPERIMENT 4 – DISCRETE-TIME FILTERING

PRELIMINARY WORK

Ziya Tuna Bölükbaşı

21628166

03/01/2023

50th order Hamming Window Matlab Code :

```
h1=fir1(50,0.2,hamming(51));
```

1)Matlab Code

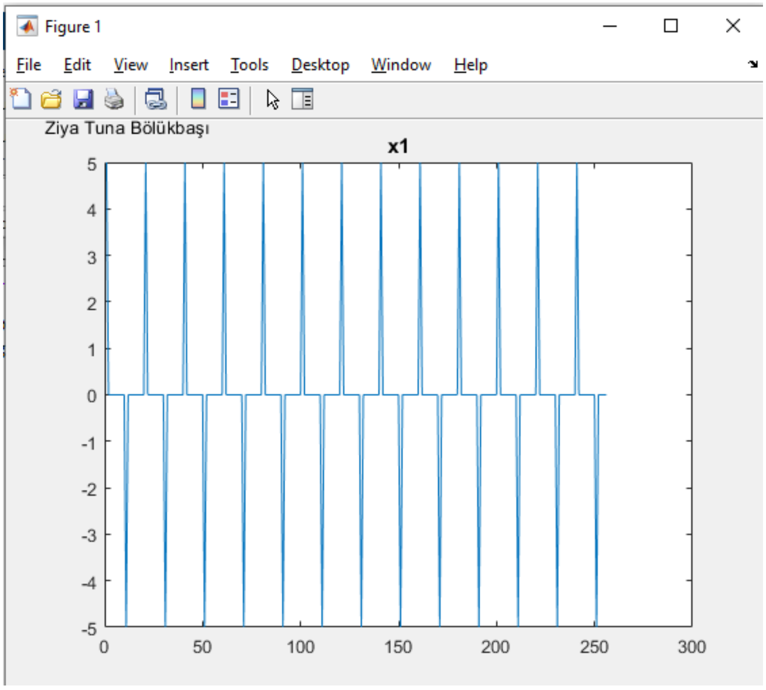
```
function y=dftfilt(x,h,N)
X=fft(x,N);
H=fft(h,N);
Y=X.*H;
y=ifft(Y,N);
end
```

2)Matlab Code

```
function  y  = convfilt( x, h, N )
x = [x , zeros(1,N-length(x))];
r = [h , zeros(1,N-length(h))];
c = [h(1) , fliplr(r(2:end))];
H = toeplitz(c, r);
y = x*H;
end
```

3)Matlab Code

```
clc,clear all,close all
n=0:1:255
x1=cos(0.1*pi*n)+cos(0.3*pi*n)+cos(0.5*pi*n)+cos(0.7*pi*n)+cos(0.9*pi*n)
plot(x1), title('x1'),gtext('Ziya Tuna Bölükbaşı')
```

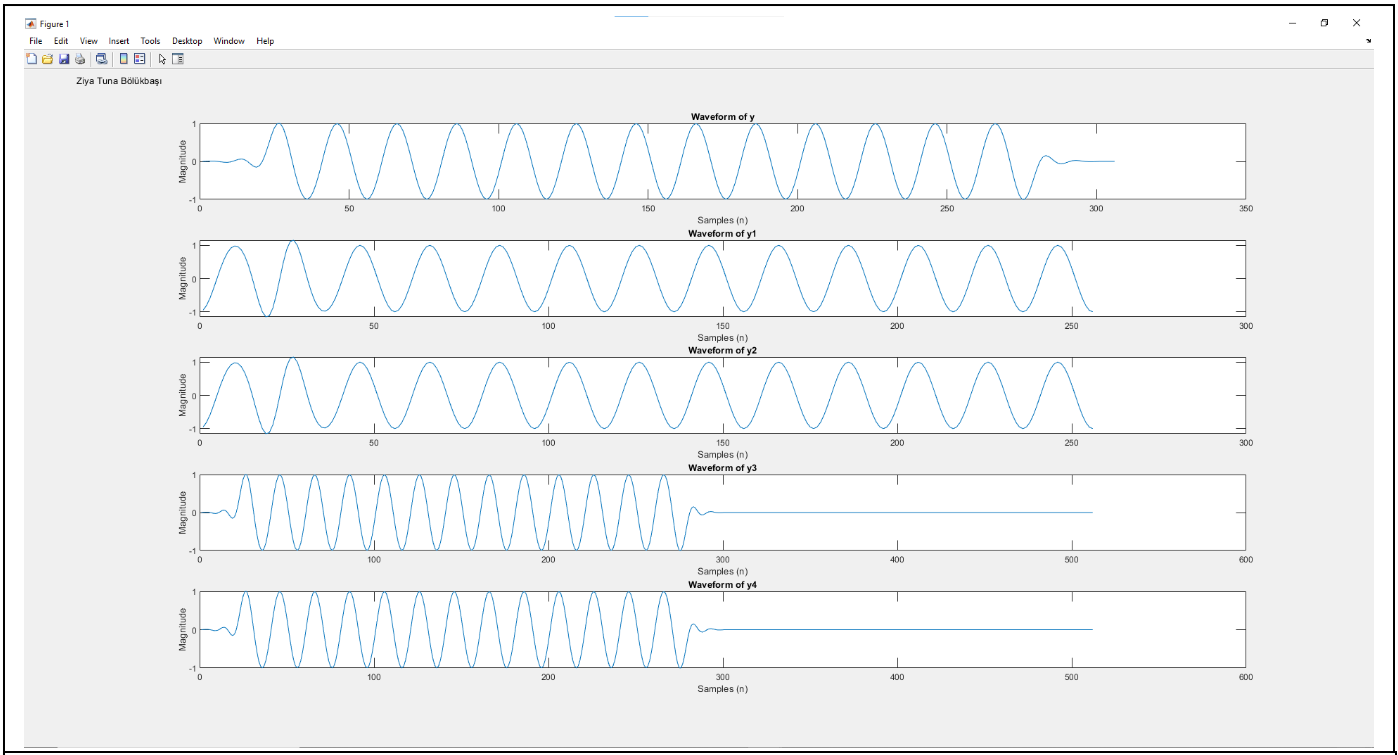


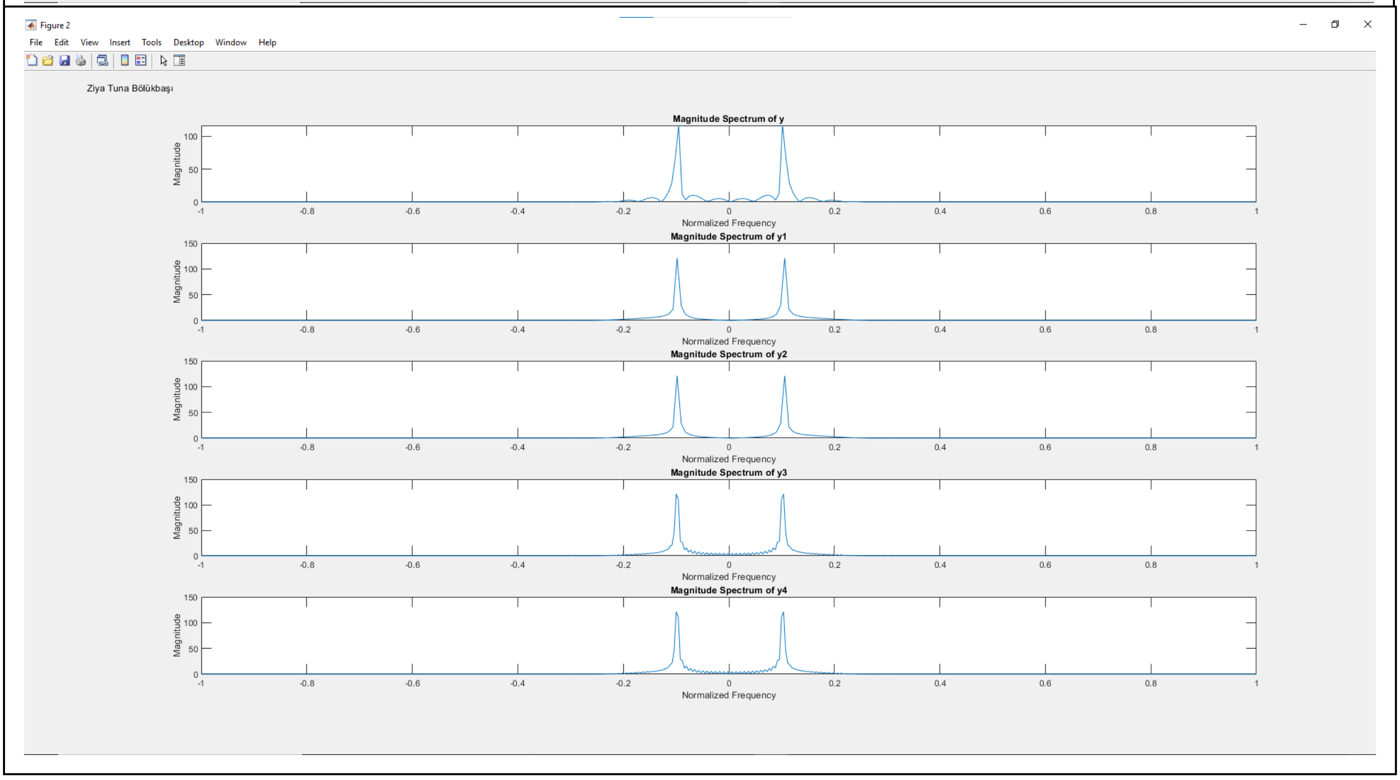
4)Matlab Code

```
clc,clear all,close all
h1=fir1(50,0.2,hamming(51));
n=0:1:255;
x=cos(0.1*pi*n)+cos(0.3*pi*n)+cos(0.5*pi*n)+cos(0.7*pi*n)+cos(0.9*pi*n);
 y = conv(x, h1);
 y1 = dftfilt(x, h1, 256);
 y2 = convfilt(x, h1, 256);
 y3 = dftfilt(x, h1, 512);
 y4 = convfilt(x, h1, 512);
```

5)Matlab Code

```
clc,clear all,close all
h1=fir1(50,0.2,hamming(51));
n=0:1:255;
x1=cos(0.1*pi*n)+cos(0.3*pi*n)+cos(0.5*pi*n)+cos(0.7*pi*n)+cos(0.9*pi*n);
y=conv(x1,h1);
y1=dftfilt(x1,h1,256);
y2=convfilt(x1,h1,256);
y3=dftfilt(x1,h1,512);
y4=convfilt(x1,h1,512);
[freq_y,fftshift_y]=mresponse(y)
[freq_y1,fftshift_y1]=mresponse(y1)
[freq_y2,fftshift_y2]=mresponse(y2)
[freq_y3,fftshift_y3]=mresponse(y3)
[freq_y4,fftshift_y4]=mresponse(y4)
figure
subplot(5,1,1),plot(y);title('Waveform of y'),xlabel('Samples
(n)'),ylabel('Magnitude');
subplot(5,1,2),plot(y1);title('Waveform of y1'),xlabel('Samples
(n)'),ylabel('Magnitude');
subplot(5,1,3),plot(y2);title('Waveform of y2'),xlabel('Samples
(n)'),ylabel('Magnitude');
subplot(5,1,4),plot(y3);title('Waveform of y3'),xlabel('Samples
(n)'),ylabel('Magnitude');
subplot(5,1,5),plot(y4);title('Waveform of y4'),xlabel('Samples
(n)'),ylabel('Magnitude');
figure
subplot(5,1,1),plot(freq_y,fftshift_y)
title('Magnitude Spectrum of y'),xlabel('Normalized Frequency'),ylabel('Magnitude');
subplot(5,1,2),plot(freq_y1,fftshift_y1)
title('Magnitude Spectrum of y1'),xlabel('Normalized Frequency'),ylabel('Magnitude');
subplot(5,1,3),plot(freq_y2,fftshift_y2)
title('Magnitude Spectrum of y2'),xlabel('Normalized Frequency'),ylabel('Magnitude');
subplot(5,1,4),plot(freq_y3,fftshift_y3)
```





FIR filters filter with linear convolution.

Therefore, linear convolution becomes more complex as the signal length increases.

So it is better to use circular convolution.

In circular convolution, if the sum of the length of the input signal and the length of the system minus 1 is greater than the sampling number, it is the same as for linear convolution.

The system length in this question is 51.

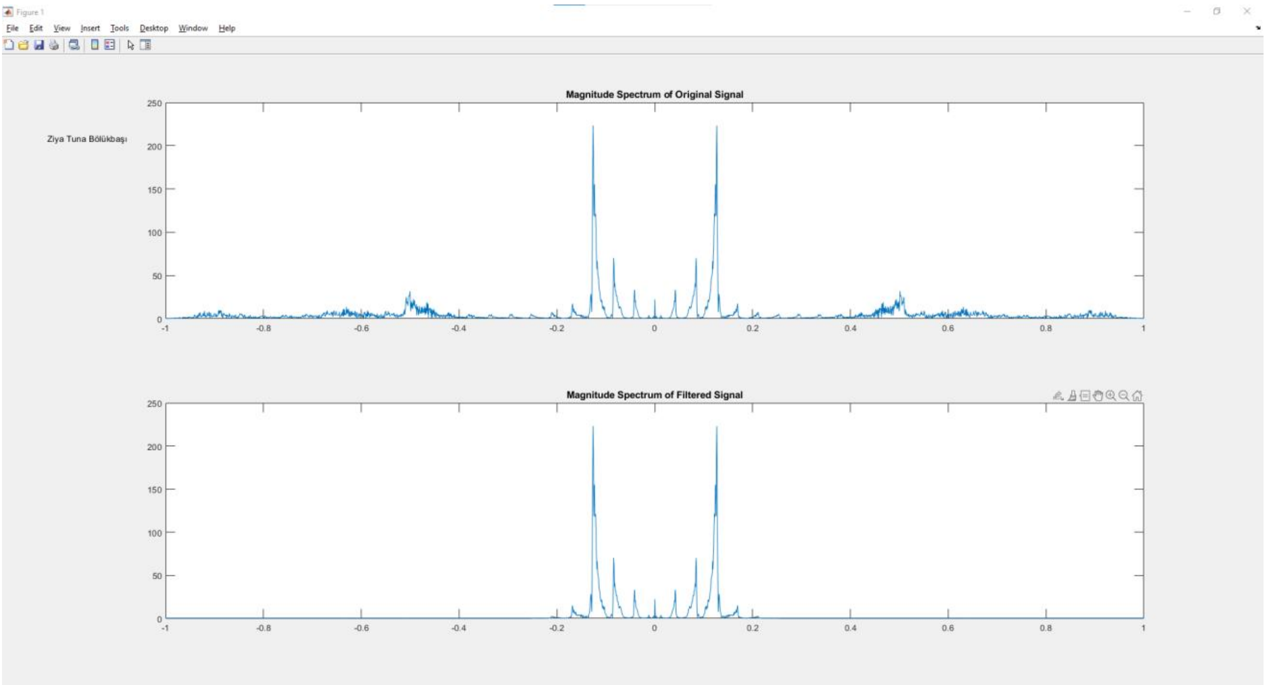
The lengths of the signals in this question are 256, so the sampling number must be greater than 306.

Since the sampling of y1 and y2 signals is 256, it is not equal to linear convolution.

Since the sampling of y3 and y4 signals is 512, circular convolution is equal to linear convolution. So we can say that the y3 and y4 signal is the correct filtered signal.

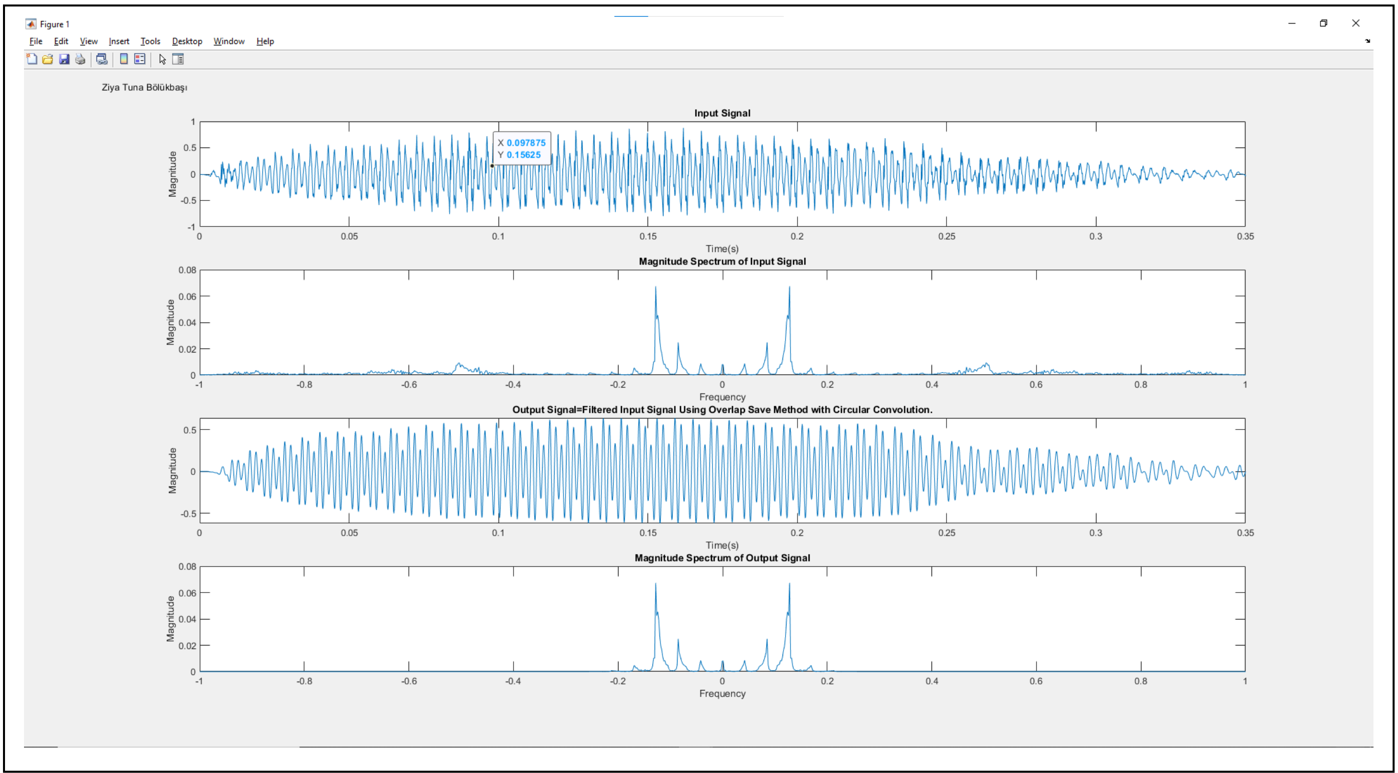
6)Matlab Code

```
clc,clear all,close all
[x,fs]=audioread('sound.wav')
h1=fir1(50,0.2,hamming(51));
x=x'
l_x=length(x);
dft_x=dftfilt(x,h1,l_x)
[freq1,ffts_x]=mresponse(x);
[freq2,ffts_dft_x]=mresponse(dft_x);
figure;
subplot(211),plot(freq1,ffts_x),title('Original signal magnitude spectra')
subplot(212),plot(freq2,ffts_dft_x),title('Filtered signal magnitude spectra')
function y=dftfilt(x,h,N)
X=fft(x,N);
H=fft(h,N);
Y=X.*H;
y=ifft(Y,N);
end
function [range_of_frequency,fftshift_of_x] = mresponse(in_x)
fft_of_x=fft(in_x,length(in_x))
fftshift_of_x=fftshift(abs(fft_of_x));
range_of_frequency=linspace(-1,1,length(fftshift_of_x))
end
```



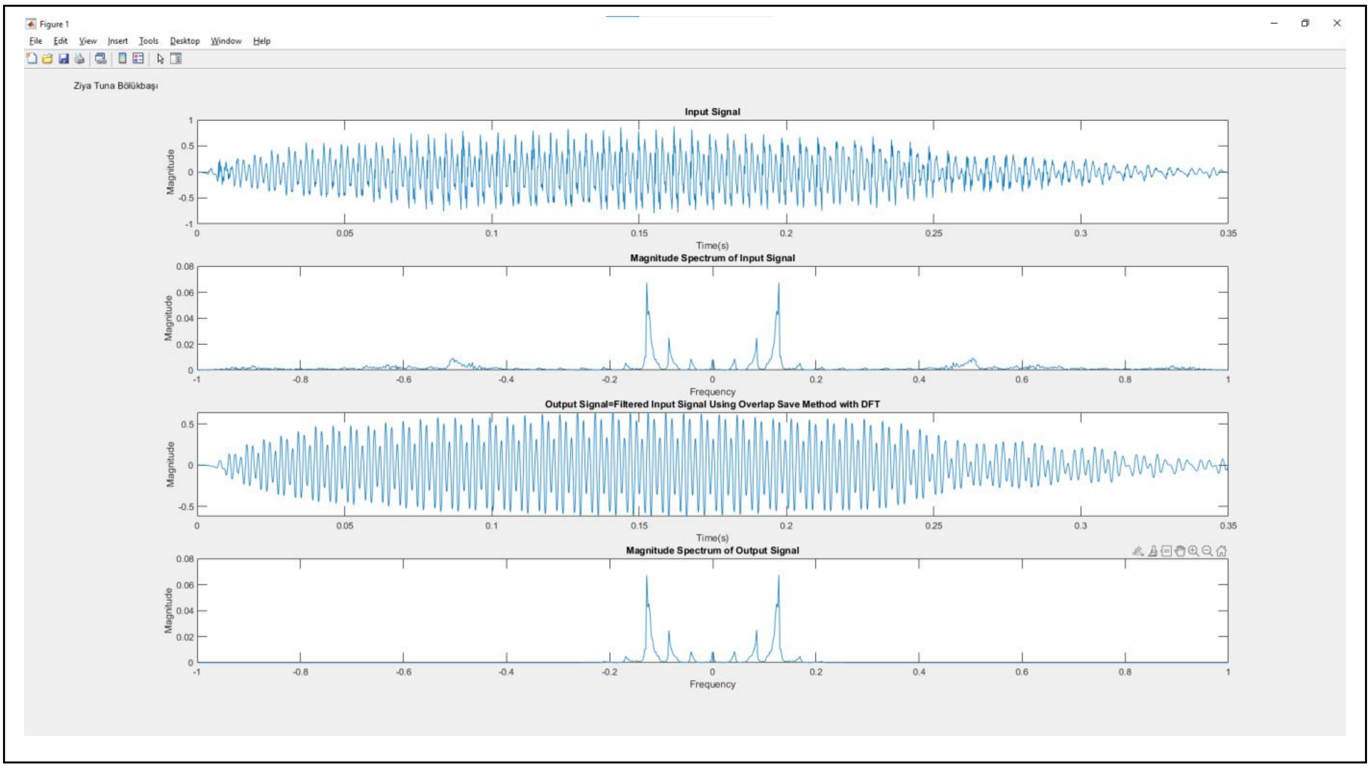
7)Matlab Code

```
clc,clear all,close all
[x,fs]=audioread('sound.wav');
x=x'
h1=fir1(50,0.2,hamming(51));
l_x=length(x);
conv_x=convsave(x,h1,l_x);
t=[1:l_x]/fs;
[fr_x,w_x]=freqz(x);
[fr_convx,w_o]=freqz(conv_x);
fr_x=[flipud(fr_x);fr_x];
fr_convx=[flipud(fr_convx);fr_convx];
freq=linspace(-1,1,length(fr_x));
subplot(4,1,1),plot(t,x);
title('Input Signal'),xlabel('Time(s)'),ylabel('Magnitude');
subplot(4,1,2),plot(freq,abs(fr_x)/l_x);
title('Magnitude Spectrum of Input Signal'),xlabel('Frequency'),ylabel('Magnitude');
subplot(4,1,3),plot(t,conv_x);
title('Output Signal=Filtered Input Signal Using Overlap Save Method with Circular Convolution.');
xlabel('Time(s)'),ylabel('Magnitude');
subplot(4,1,4),plot(freq,abs(fr_convx)/l_x)
title('Magnitude Spectrum of Output Signal'),xlabel('Frequency'),ylabel('Magnitude');
function  y  = convsave( x,h,L )
l_x = length(x);
l_h = length(h);
N = L + l_h - 1;
Num_Of_Segment = ceil(l_x/(L));
x = [zeros(1,l_h-1) x zeros(1,Num_Of_Segment*L-l_x)];
y = zeros(Num_Of_Segment,N);
for n=1:Num_Of_Segment
    StartAdd = (n-1)*L +1;
    y(n,:) = convfilt(x(StartAdd:StartAdd+N-1), h , N);
end
y = y(:,l_h:end);
y = reshape(y',[1,L*Num_Of_Segment]);
end
```



8)Matlab Code

```
clc,clear all,close all
[x,fs]=audioread('sound.wav');
x=x'
h1=fir1(50,0.2,hamming(51));
l_x=length(x);
dft_x=dftsave(x,h1,l_x);
l_x=length(x);
t=[1:l_x]/fs;
[fr_x,w_x]=freqz(x);
[fr_dftx,w_o]=freqz(dft_x);
fr_x=[flipud(fr_x);fr_x];
fr_dftx=[flipud(fr_dftx);fr_dftx];
freq=linspace(-1,1,length(fr_x));
subplot(4,1,1),plot(t,x);
title('Input Signal'),xlabel('Time(s)'),ylabel('Magnitude');
subplot(4,1,2),plot(freq,abs(fr_x)/l_x);
title('Magnitude Spectrum of Input Signal'),xlabel('Frequency'),ylabel('Magnitude');
subplot(4,1,3),plot(t,dft_x);
title('Output Signal=Filtered Input Signal Using Overlap Save Method with DFT')
xlabel('Time(s)'),ylabel('Magnitude');
subplot(4,1,4),plot(freq,abs(fr_dftx)/l_x)
title('Magnitude Spectrum of Output Signal'),xlabel('Frequency'),ylabel('Magnitude');
gtext('Ziya Tuna Bölükbaşı')
function y = dftsave( x,h,L )
L_x = length(x);
L_h = length(h);
N = L + L_h - 1;
NumOfSegment = ceil(L_x/(L));
x = [zeros(1,L_h-1) x zeros(1,NumOfSegment*L-L_x)];
temp_y = zeros(NumOfSegment,N);
for n=1:NumOfSegment
StartAdd = (n-1)*L +1;
temp_y(n,:) = dftfilt(x(StartAdd:StartAdd+N-1), h , N);
end
temp_y = temp_y(:,L_h:end);
y = reshape(temp_y',[1,L*NumOfSegment]);
end
```



9) These 3 signals are identical to one another. The overlap save technique produces the same outcome as linear convolution. Utilizing the overlap save method is advantageous when the signal is lengthy since we can only process the completed portion of the signal.

Although the operational load is smaller, circular convolution is more difficult than linear convolution. To calculate linear convolution, more calculations are performed.

10)Matlab Code

```
clc,clear all,close all
[x,fs]=audioread('sound.wav');
x=x'
h1=fir1(50,0.2,hamming(51));
l_x=length(x);
dft_x=dftadd(x,h1,l_x);
t=[1:l_x]/fs;
[fr_x,w_x]=freqz(x);
[fr_dftx,w_o]=freqz(dft_x);
fr_x=[flipud(fr_x);fr_x];
fr_dftx=[flipud(fr_dftx);fr_dftx];
freq=linspace(-1,1,length(fr_x));
figure,subplot(3,2,1),plot(t,x);
title('Input Signal'),xlabel('Time(s)'),ylabel('Magnitude');
subplot(3,2,2),plot(freq,abs(fr_x)/l_x);
title('Magnitude Spectrum of Input Signal'),xlabel('Frequency'),ylabel('Magnitude');
subplot(3,2,3),plot(t,dft_x);
title('Output Signal=Filtered Input Signal Using Overlap Add Method with DFT')
xlabel('Time(s)'),ylabel('Magnitude');
subplot(3,2,4),plot(freq,abs(fr_dftx)/l_x)
title('Magnitude Spectrum of Output Signal'),xlabel('Frequency'),ylabel('Magnitude');
conv_x=convadd(x,h1,l_x);
t=[1:l_x]/fs;
[fr_x,w_x]=freqz(x);
[fr_convx,w_o]=freqz(conv_x);
fr_x=[flipud(fr_x);fr_x];
fr_convx=[flipud(fr_convx);fr_convx];
freq=linspace(-1,1,length(fr_x));
subplot(3,2,5),plot(t,x);
title('Output Signal=Filtered Input Signal Using Overlap Add Method with Circular
Convolution.');
xlabel('Time(s)'),ylabel('Magnitude');
subplot(3,2,6),plot(freq,abs(fr_convx)/l_x)
title('Magnitude Spectrum of Output Signal'),xlabel('Frequency'),ylabel('Magnitude');
gtext('Ziya Tuna Bölükbaşı')
 function y=convadd(x,h,L)
  P=length(h);
  y=convfilt(x(1:L),h,L);
    for i=1:length(x)/L-1
        k=convfilt(x(i*L+1:(i+1)*L),h,L);
        y(end-P+1:end)=y(end-P+1:end)+k(1:P);
        y=cat(2,y,k(P+1:end));
    end
  end
 function y=dftadd(x,h,L)
  P=length(h);
  y=dftfilt(x(1:L),h,L);
    for i=1:length(x)/L-1
        k=dftfilt(x(i*L+1:(i+1)*L),h,L+P-1);
        y(end-P+1:end)=y(end-P+1:end)+k(1:P);
        y=cat(2,y,k(P+1:end));
    end
  end
```

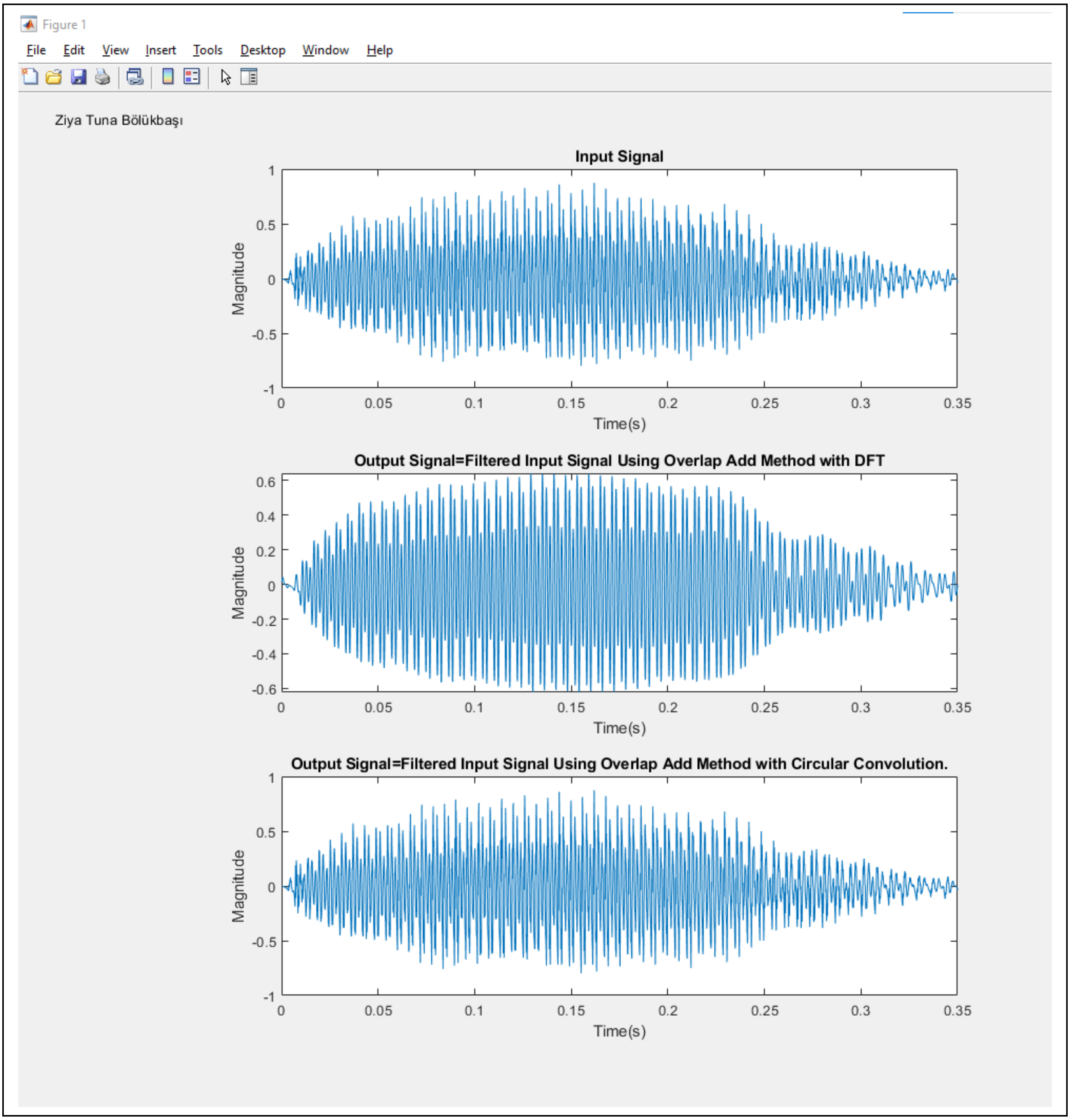
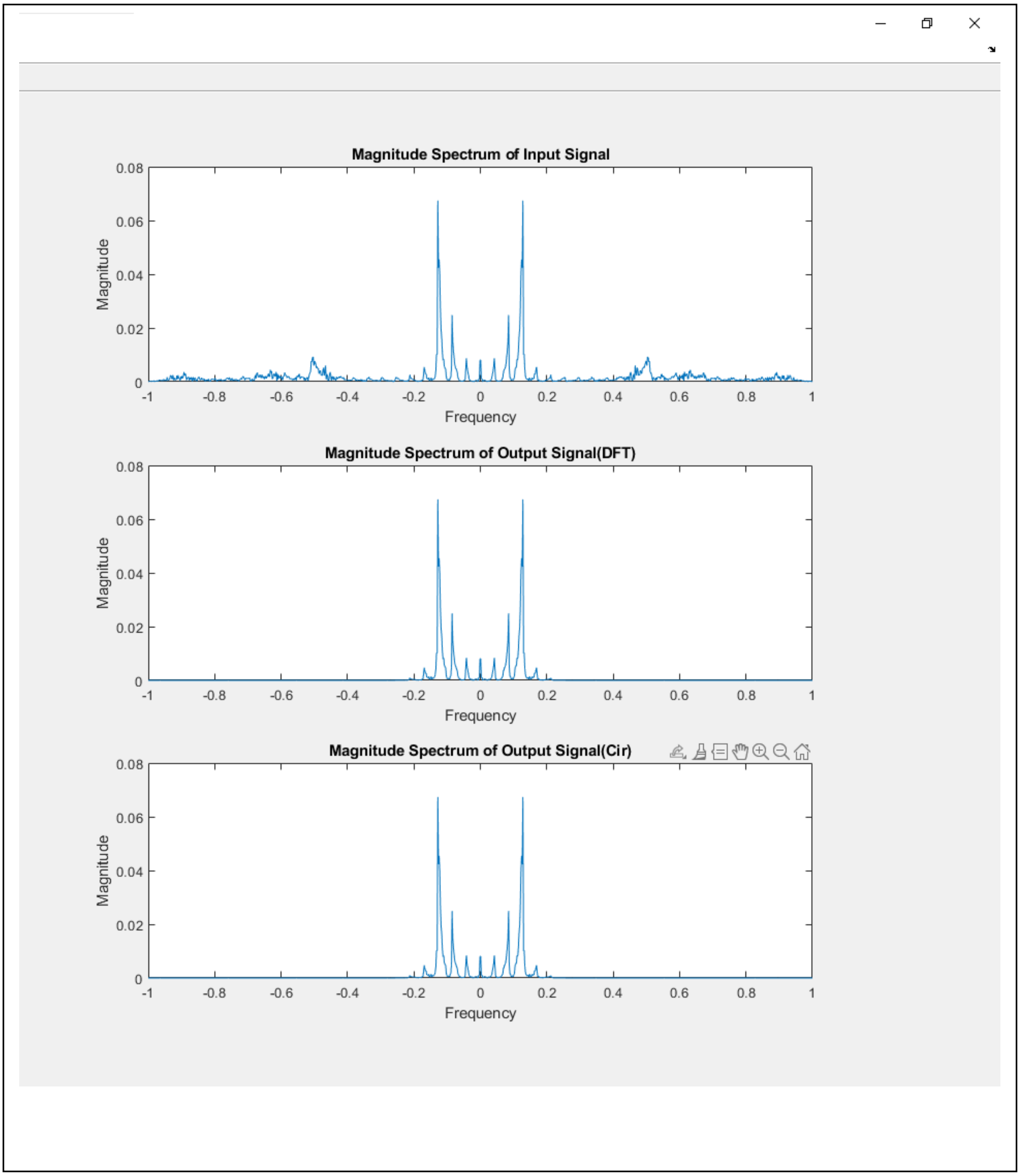


Figure 1
File
Edit
View
Insert
Tools
Desktop
Window
Help
Ziya Tuna Bölükbaşı
Input Signal
Magnitude
Time(s)
1
0.5
0
-0.5
-1
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
Output Signal=Filtered Input Signal Using Overlap Add Method with DFT
Magnitude
Time(s)
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
Output Signal=Filtered Input Signal Using Overlap Add Method with Circular Convolution.
Magnitude
Time(s)



Magnitude Spectrum of Input Signal
Magnitude
Frequency
Magnitude Spectrum of Output Signal(DFT)
Magnitude
Frequency
Magnitude Spectrum of Output Signal(Cir)
Magnitude
Frequency