Department of Electrical, Computer, & Biomedical Engineering
Faculty of Engineering & Architectural Science

Course Title:	Signals and Systems II
Course Number:	ELE632
Semester/Year (e.g.F2016)	W2022

Instructor:	Dimitri Androutsos

Assignment/Lab Number:	5
Assignment/Lab Title:	Sampling and the Discrete Fourier Transform

Submission Date:	Sunday, April 10th, 2022
Due Date:	Sunday, April 10th, 2022

Student LAST Name	Student FIRST Name	Student Number	Section	Signature*
Fahmy	Ahmad	500913092	9	A.F.

**By signing above, you attest that you have contributed to this submission and confirm that all work you have contributed to this submission is your own work. Any suspicion of copying or plagiarism in this work <u>will result in an investigation of Academic Misconduct and may result in a "0" on the work,</u> an "F" in the course, or possibly more severe penalties, as well as a Disciplinary Notice on your academic record under the Student Code of Academic Conduct, which can be found online at:*

<u>https://www.ryerson.ca/senate/policies/pol60.pdf</u>

Appendix

A) Discrete Fourier Transform and Zero Padding

B) Sampling

C) Filter Design

```
%%Student #:500913092
%         ABCDEFGHI
% H = 9, I = 2
```

A) Discrete Fourier Transform and Zero Padding

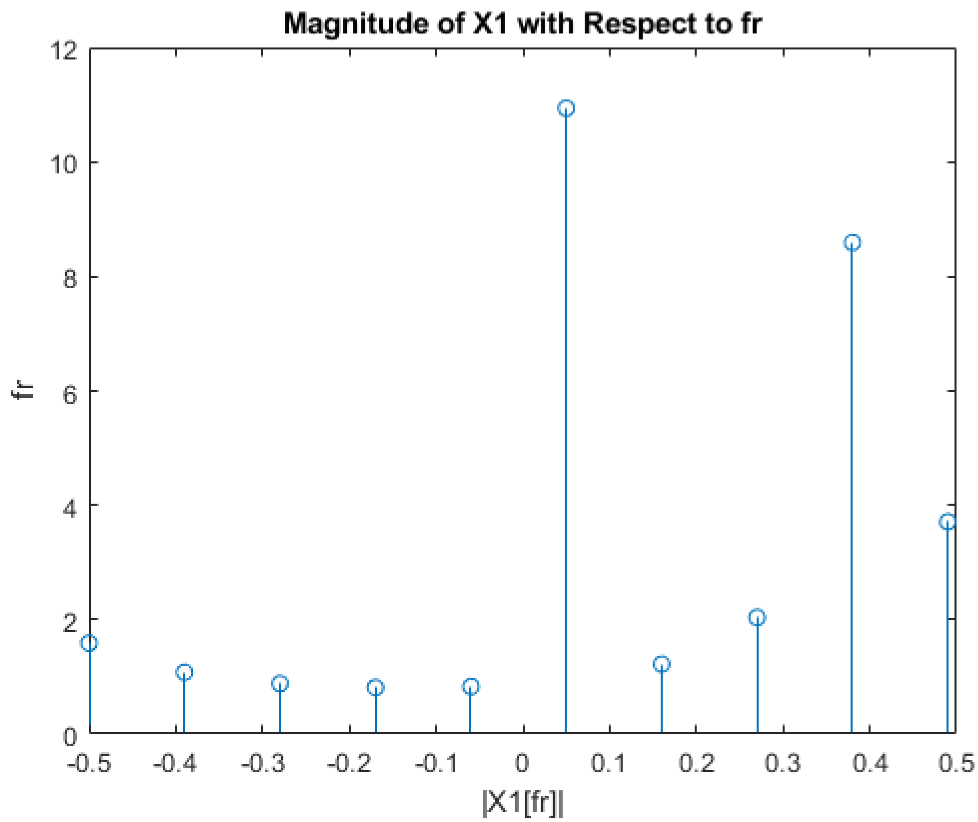
Part 1 - Discrete Fourier Transform

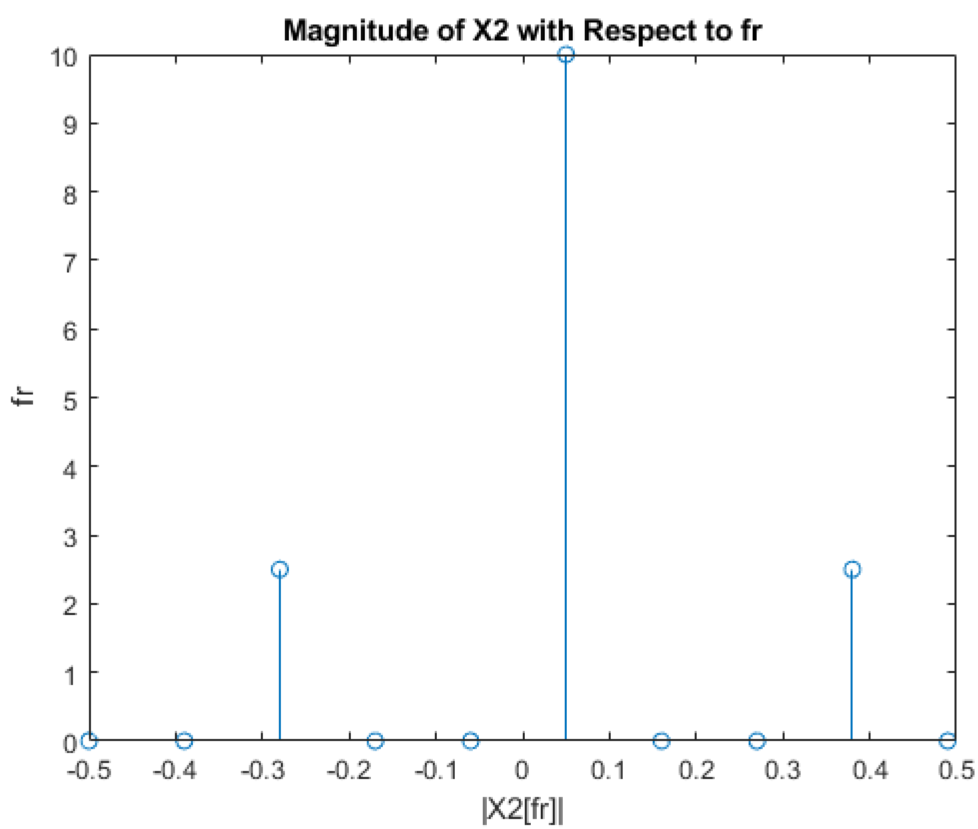
```
n = 0:9;
N = length(n);
fr = linspace(-0.5, 0.49, 10);
x1 = exp(1i*2*pi*(1)*n)+exp(1i*2*pi*(33/100)*n);
x2 = cos(2*pi*(1)*n)+0.5*cos(2*pi*(0.3*n));

X1 = fftshift(fft(x1));
X2 = fftshift(fft(x2));

figure();
stem(fr, abs(X1));
title("Magnitude of X1 with Respect to fr");
xlabel('|X1[fr]|')
ylabel('fr')

figure();
stem(fr, abs(X2));
title("Magnitude of X2 with Respect to fr");
xlabel('|X2[fr]|')
ylabel('fr')
```


Magnitude of X1 with Respect to fr
fr
|X1[fr]|
12
10
8
6
4
2
0
-0.5
-0.4
-0.3
-0.2
-0.1
0
0.1
0.2
0.3
0.4
0.5


Magnitude of X2 with Respect to fr
fr
|X2[fr]|
10
9
8
7
6
5
4
3
2
1
0
-0.5
-0.4
-0.3
-0.2
-0.1
0
0.1
0.2
0.3
0.4
0.5

1. X2 has a symmetrical signal because the sum of the function is even, whereas the sum of X1 is not even.
2. Yes it is possible to distinguish the two signals because each plot has different frequency components that belong to the two signals
3. Other frequency components can be observed in the DFT plot of x1[n] because it is the summation of two signals with different frequencies.

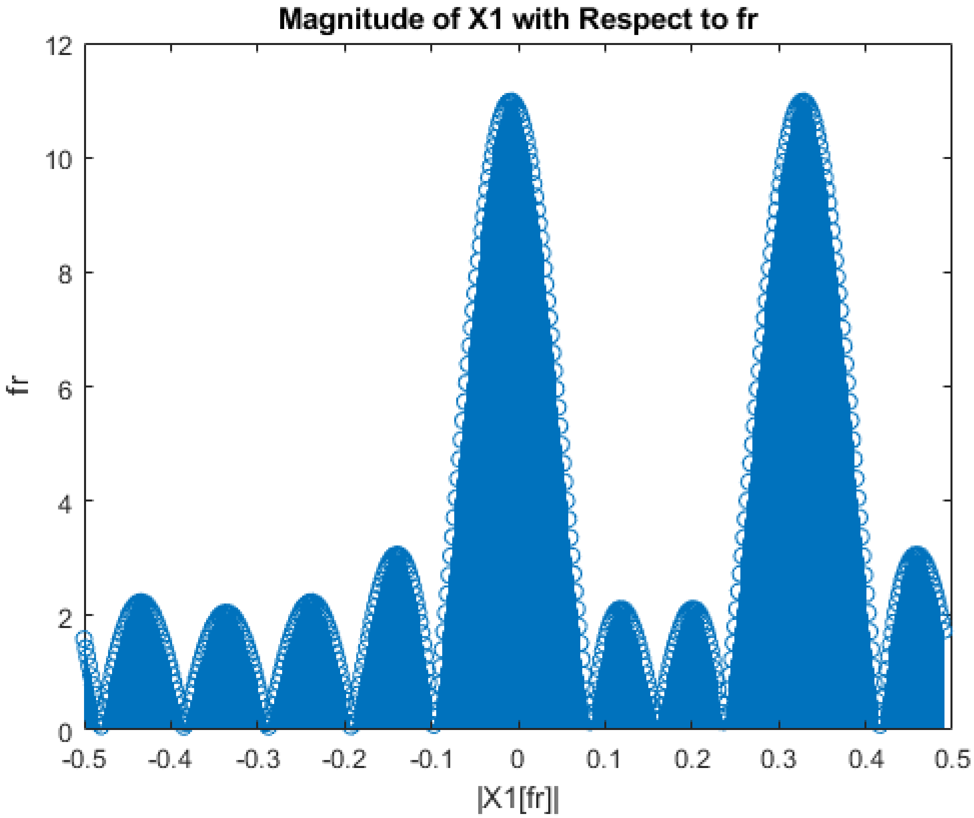
Part 2 - 490-zero padded DFT with 500 samples

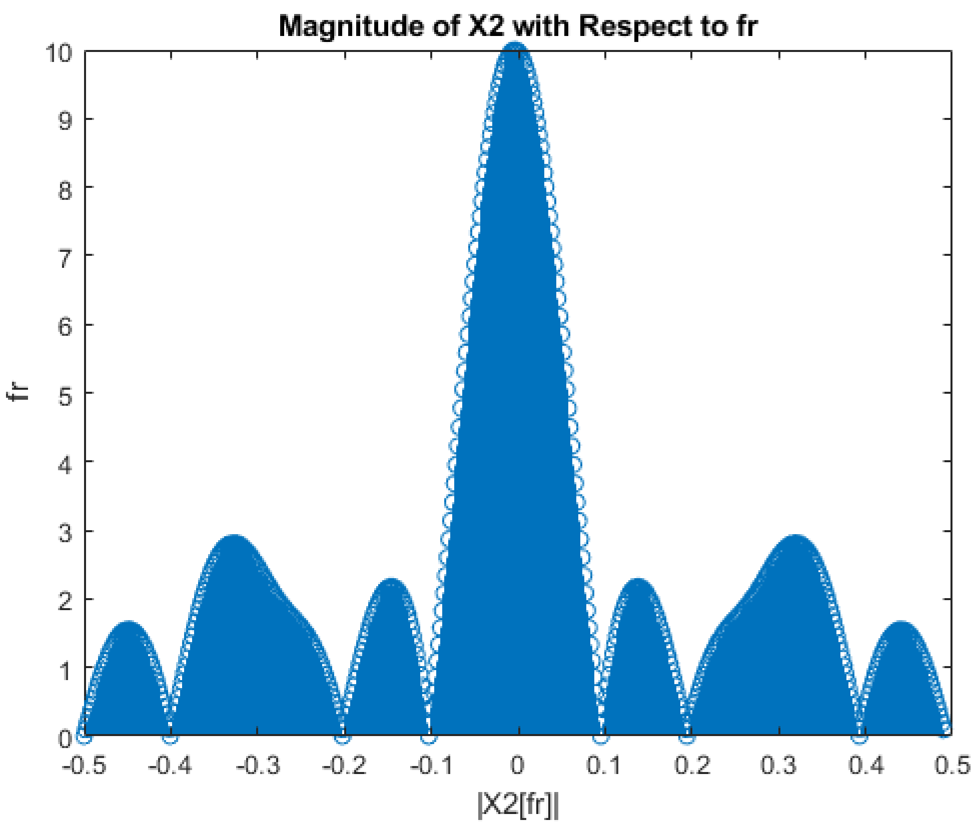
```
n = 0:9;
N = 500;
fr = linspace(-0.5, 0.49, N);
x1 = exp(1i*2*pi*(1)*n)+exp(1i*2*pi*(33/100)*n);
x3 = [zeros(1, 245), x1, zeros(1, 245)];
x2 = cos(2*pi*(1)*n)+0.5*cos(2*pi*(0.3*n));
x4 = [zeros(1, 245), x2, zeros(1, 245)];

X3 = fftshift(fft(x3));
X4 = fftshift(fft(x4));

figure();
stem(fr, abs(X3));
title("Magnitude of X1 with Respect to fr");
xlabel('|X1[fr]|')
ylabel('fr')

figure();
stem(fr, abs(X4));
title("Magnitude of X2 with Respect to fr");
xlabel('|X2[fr]|')
ylabel('fr')
```


Magnitude of X1 with Respect to fr
fr
|X1[fr]|
0
2
4
6
8
10
12
-0.5
-0.4
-0.3
-0.2
-0.1
0
0.1
0.2
0.3
0.4
0.5


Magnitude of X2 with Respect to fr
fr
|X2[fr]|
0
1
2
3
4
5
6
7
8
9
10
-0.5
-0.4
-0.3
-0.2
-0.1
0
0.1
0.2
0.3
0.4
0.5

Yes we can see an improvement due to the zero-padding compared to the plots in Part 1 since the plots are now perfectly centered about the y-axis.

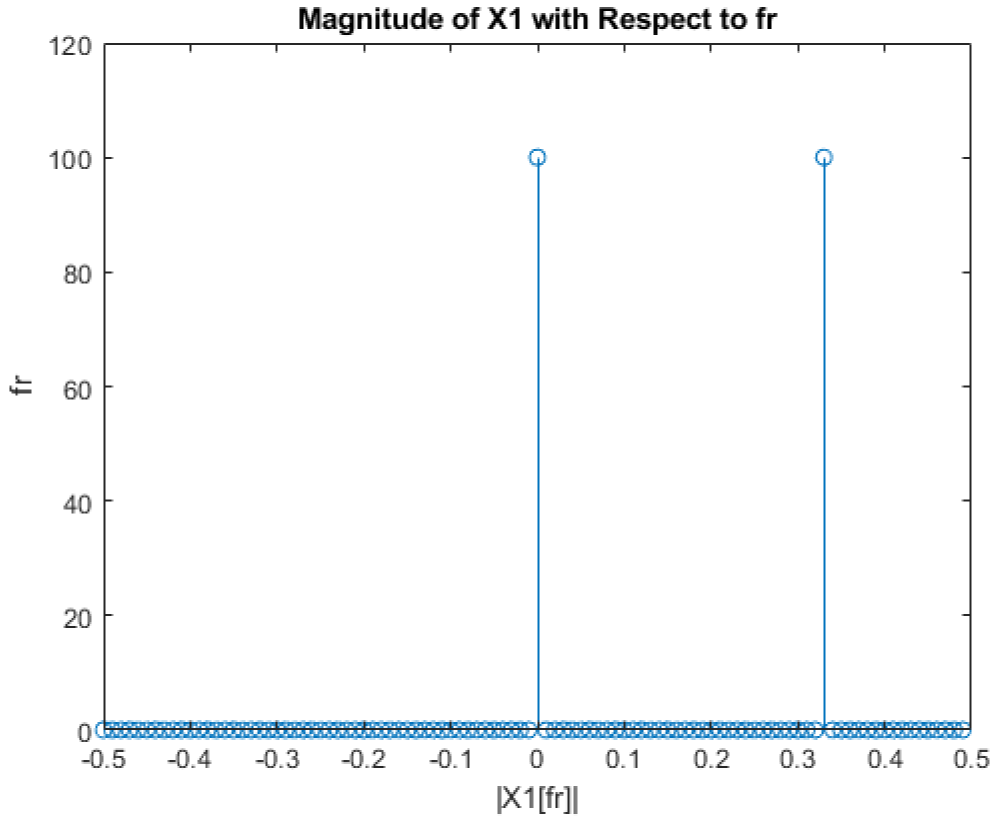
Part 3 - Part 1 DFT with 100 samples

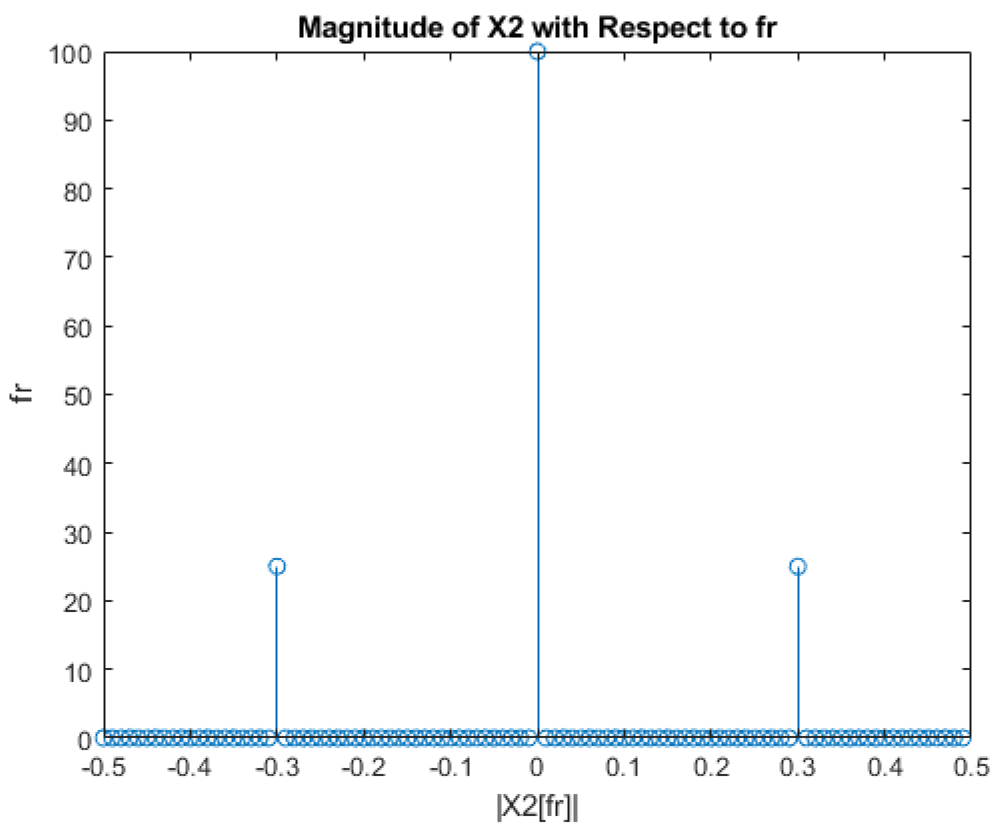
```
n = 0:99;
N = 100;
fr = linspace(-0.5, 0.49, N);
x1 = exp(1i*2*pi*(1)*n)+exp(1i*2*pi*(33/100)*n);
x2 = cos(2*pi*(1)*n)+0.5*cos(2*pi*(0.3*n));

X3 = fftshift(fft(x1));
X4 = fftshift(fft(x2));

figure();
stem(fr, abs(X3));
title("Magnitude of X1 with Respect to fr");
xlabel('|X1[fr]|')
ylabel('fr')

figure();
stem(fr, abs(X4));
title("Magnitude of X2 with Respect to fr");
xlabel('|X2[fr]|')
ylabel('fr')
```


Magnitude of X1 with Respect to fr
120
100
80
60
40
20
0
fr
-0.5
-0.4
-0.3
-0.2
-0.1
0
0.1
0.2
0.3
0.4
0.5
|X1[fr]|


Magnitude of X2 with Respect to fr
100
90
80
70
60
50
40
30
20
10
0
fr
-0.5
-0.4
-0.3
-0.2
-0.1
0
0.1
0.2
0.3
0.4
0.5
|X2[fr]|

X2 has a symmetric spectrum because the signal is even.

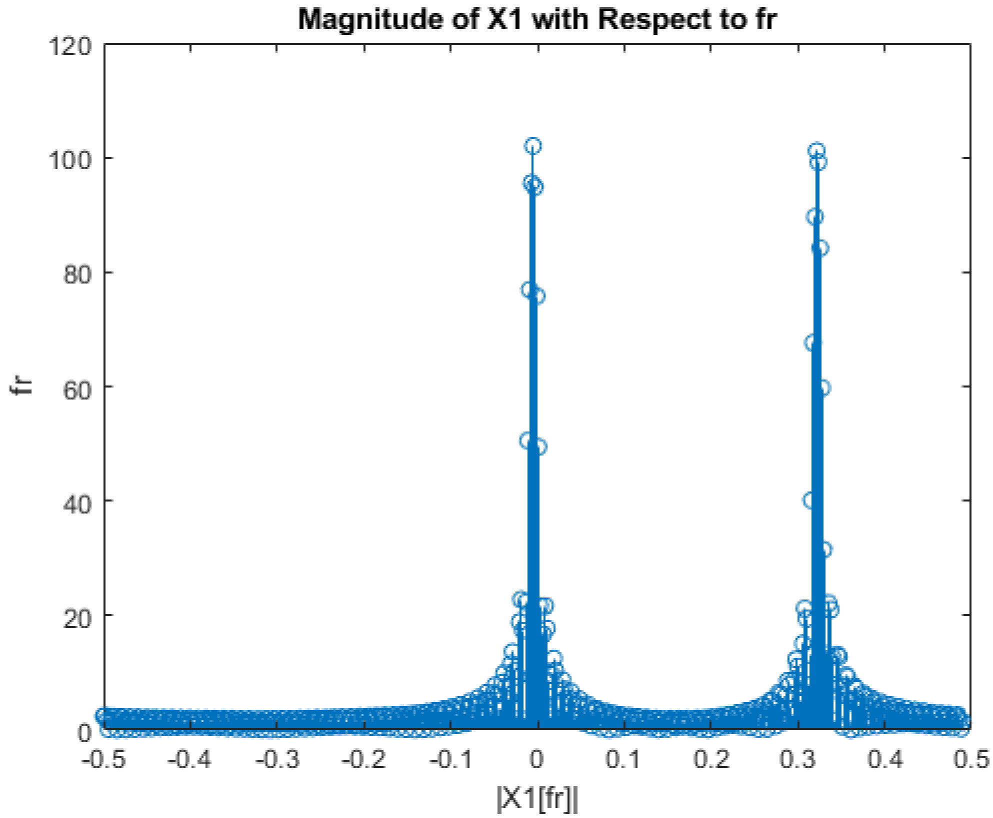
Part 4 - 400-zero padded DFT with 500 samples

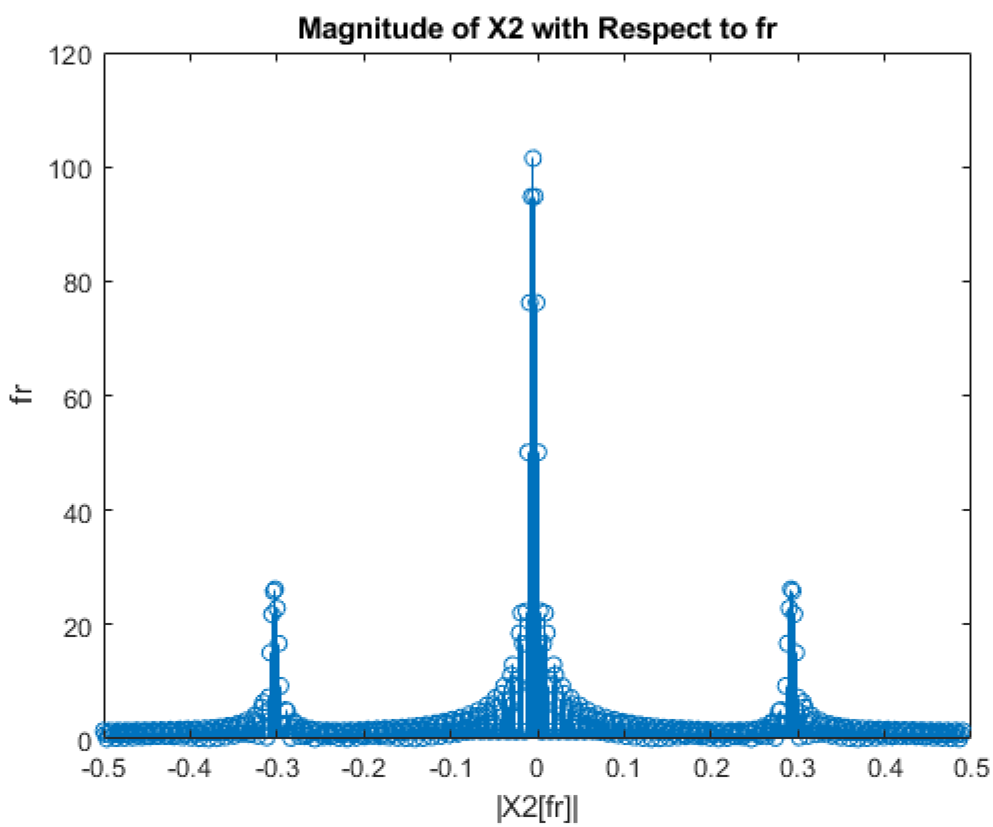
```
n = 0:100;
N = 501;
fr = linspace(-0.5, 0.49, N);
x1 = exp(1i*2*pi*(1)*n)+exp(1i*2*pi*(33/100)*n);
x3 = [zeros(1, 200), x1, zeros(1, 200)];
x2 = cos(2*pi*(1)*n)+0.5*cos(2*pi*(0.3*n));
x4 = [zeros(1, 200), x2, zeros(1, 200)];

X3 = fftshift(fft(x3));
X4 = fftshift(fft(x4));

figure();
stem(fr, abs(X3));
title("Magnitude of X1 with Respect to fr");
xlabel('|X1[fr]|')
ylabel('fr')

figure();
stem(fr, abs(X4));
title("Magnitude of X2 with Respect to fr");
xlabel('|X2[fr]|')
ylabel('fr')
```


Magnitude of X1 with Respect to fr
fr
|X1[fr]|
0
20
40
60
80
100
120
-0.5
-0.4
-0.3
-0.2
-0.1
0
0.1
0.2
0.3
0.4
0.5


Magnitude of X2 with Respect to fr
fr
|X2[fr]|
0
20
40
60
80
100
120
-0.5
-0.4
-0.3
-0.2
-0.1
0
0.1
0.2
0.3
0.4
0.5

Yes there is an improvement due to a higher degree of accuracy.

B) Sampling

Part 1 - Signal Properties

```
clc
clear
load laughter.mat
filename = 'laughter.wav';
audiowrite(filename, y, Fs);
clear y Fs
[y, Fs] = audioread("laughter.wav");

No = length(y);
To = length(y)/Fs;
Ti = 1/Fs;

%Number of Samples
No

%Duration of Singal
To

%Sampling Interval
Ti
```

Part 2 - Plotted Signal

```
t = 0:(No-1);
figure();
plot(t, y);
title("Signal y vs Time");
```

Part 3 - Plot of Discrete Fourier Transformed Signal

```
Y = fftshift(fft(y));
fr = (-No/2) : ((No/2)-1);

figure();
plot(fr, abs(Y));
title("DFT of audio signal");
```

Part 4 - Rate 2 Subsampled Signal Properties

```
rate = 2;
y1 = y(1:rate:end);
No2 = length(y1);
To2 = 2*length(y1)/Fs;
Ti2 = 2/Fs;

%Number of Samples
No2

%Duration of Singal
To2

%Sampling Interval
Ti2
```

Part 5 - Plot of Rate 2 Subsampled Signal

```
figure();
plot(y1);
title("Signal y vs Time");
```

Part 6 - Plot of Discrete Fourier Transformed Subsampled Signal

```
Y1 = fftshift(fft(y1));
fr = (-No2/2) : ((No2/2)-1);

figure();
plot(fr, abs(Y1));
title("DFT of audio signal");
```

The signal y1 hasn’t changed very much, however its frequency spectrum now contains more components, this is due to the sampling with a rate of two and the loss of data to form an accurate representation of the signal.

Part 7 - Listening to the two signals

```
sound(y, Fs);
sound(y1, Fs);
```

The audio signals contain the same sound, however the subsample audio sounds as if it’s been sped up.

Part 8 - Subsample of signal with rate 5

```
rate = 5;
y2 = y(1:rate:end);
No3 = length(y2);
To3 = 2*length(y2)/Fs;
Ti3 = 2/Fs;

No3

To3

Ti3
```

```
figure();
plot(y2);
title("Signal y vs Time");

Y2 = fftshift(fft(y2));
fr = (-No3/2) : ((No3/2)-1);

figure();
plot(fr, abs(Y2));
title("DFT of audio signal");
```

The effect on the spectrum seen in Part 5 is similar to the effect seen in Part 4, in which more frequency components have been introduced.

```
No =

       52634


To =

    6.4250


Ti =

   1.2207e-04
```

No2 =

26317

To2 =

6.4250

Ti2 =

2.4414e-04

No3 =

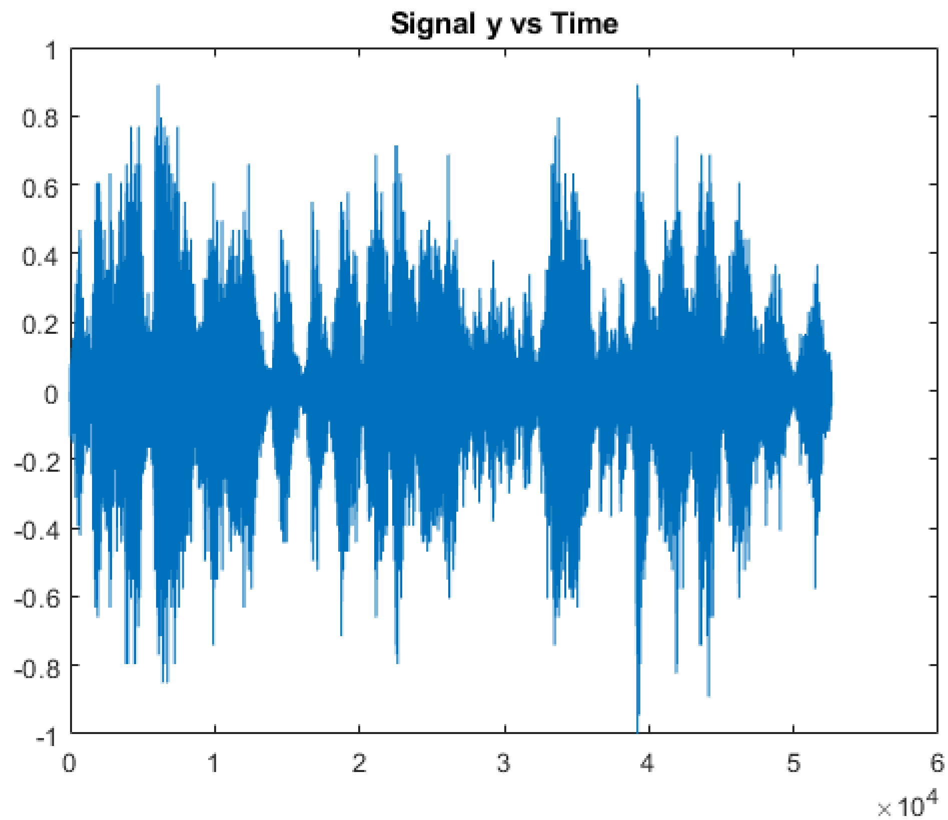
10527

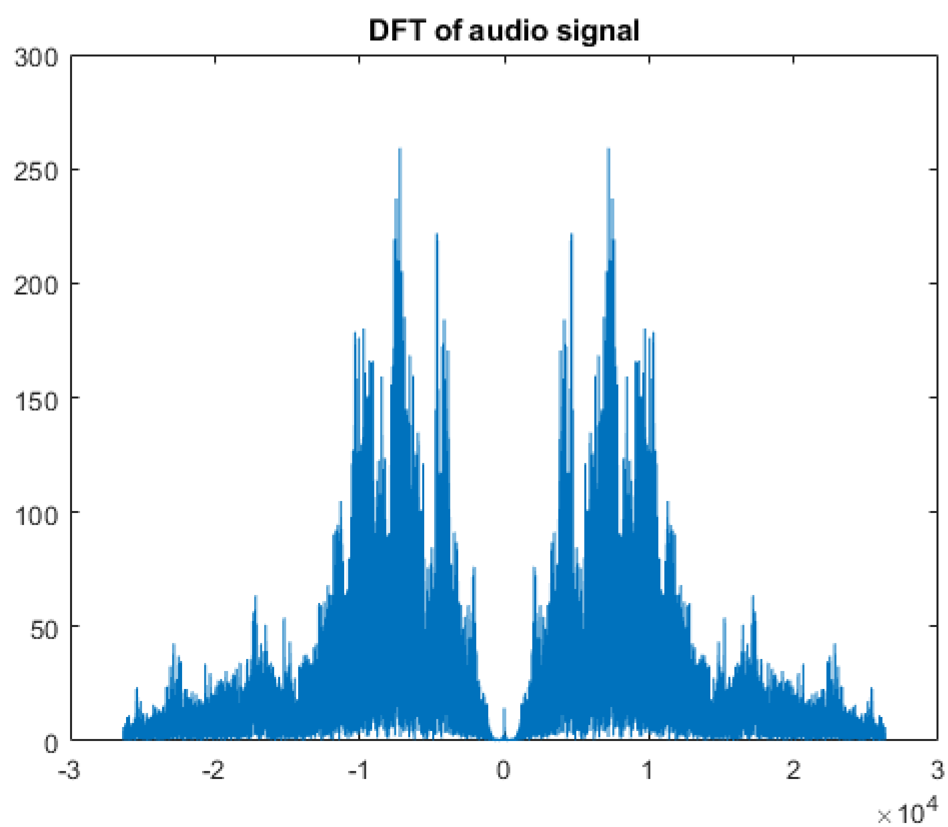
To3 =

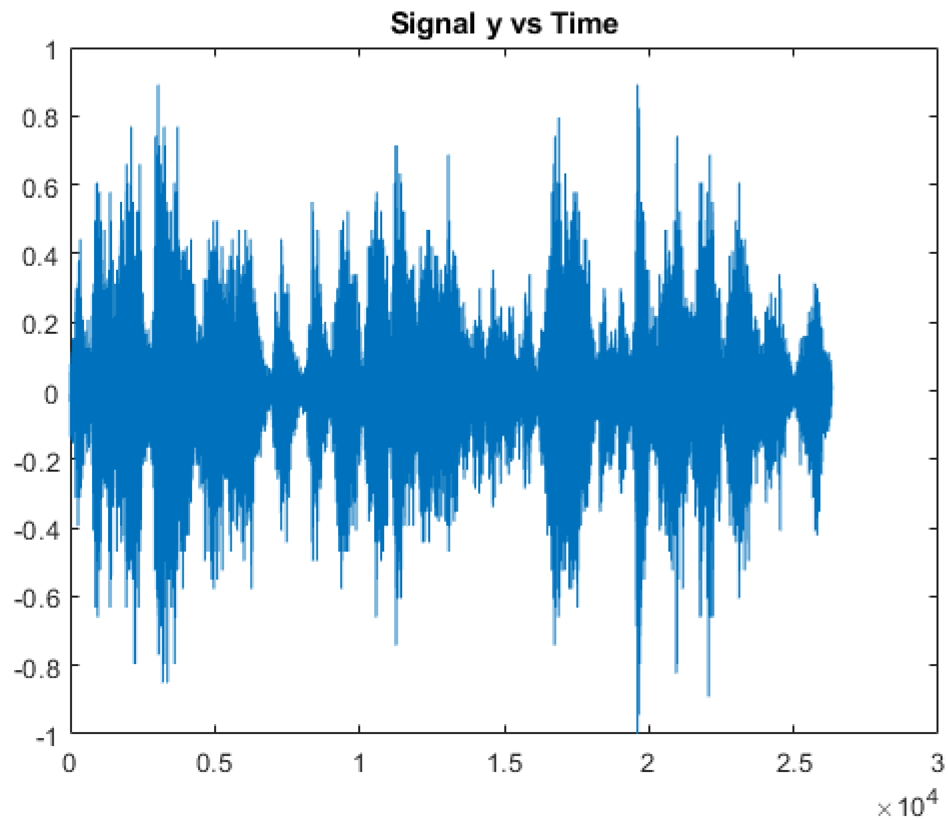
2.5701

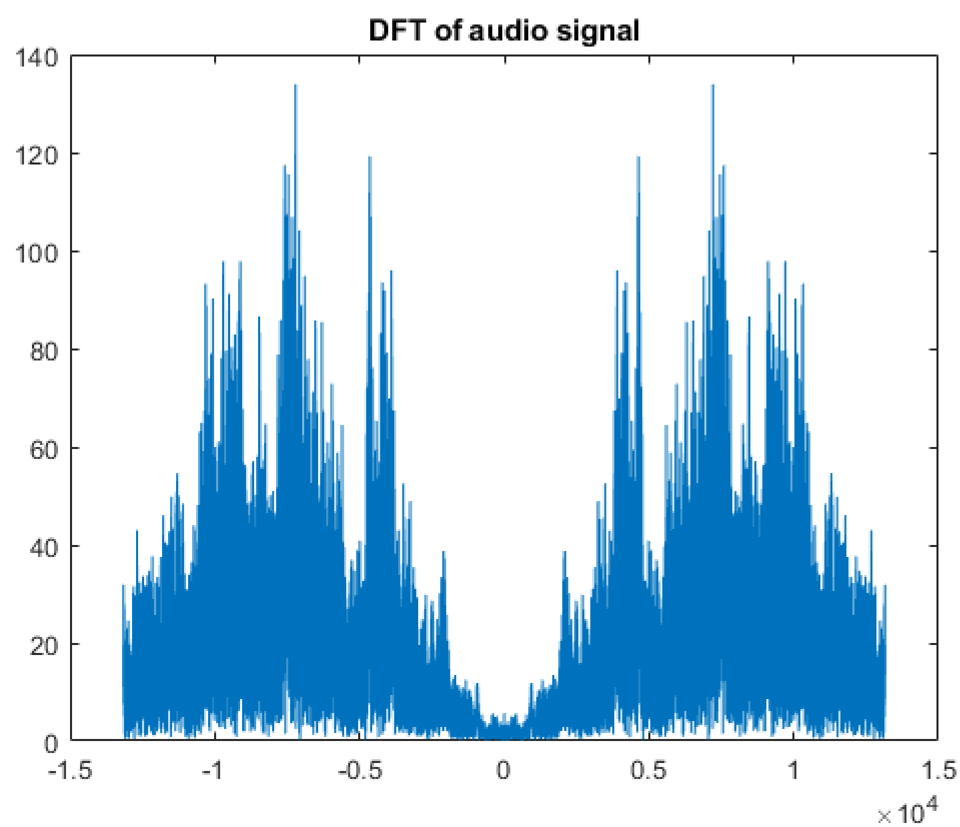
Ti3 =

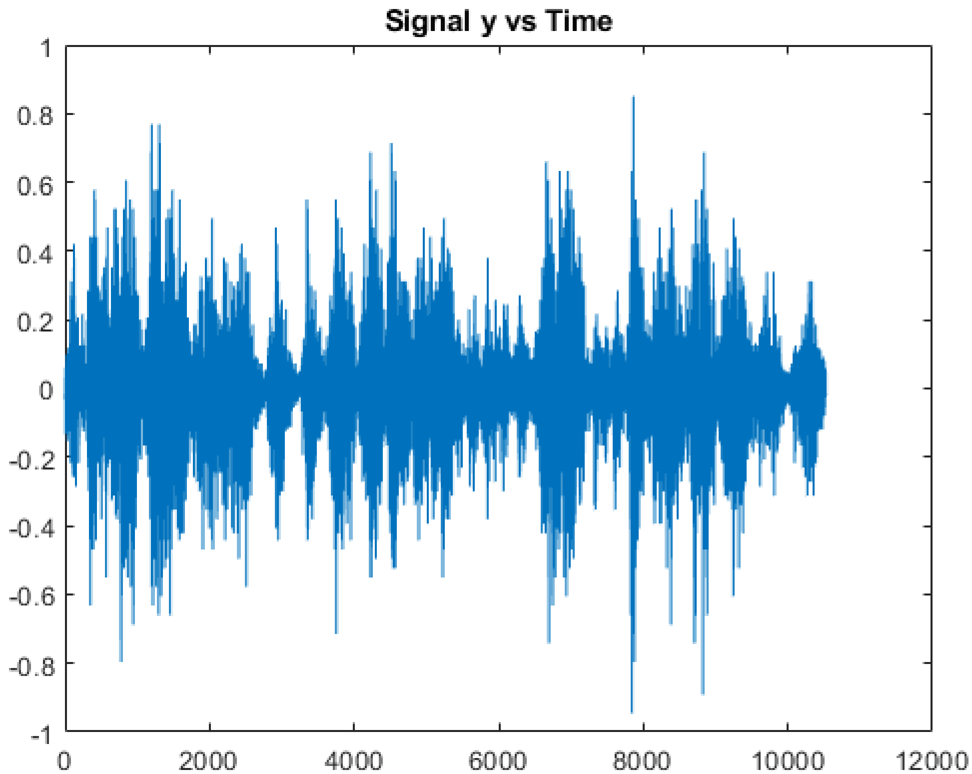
2.4414e-04

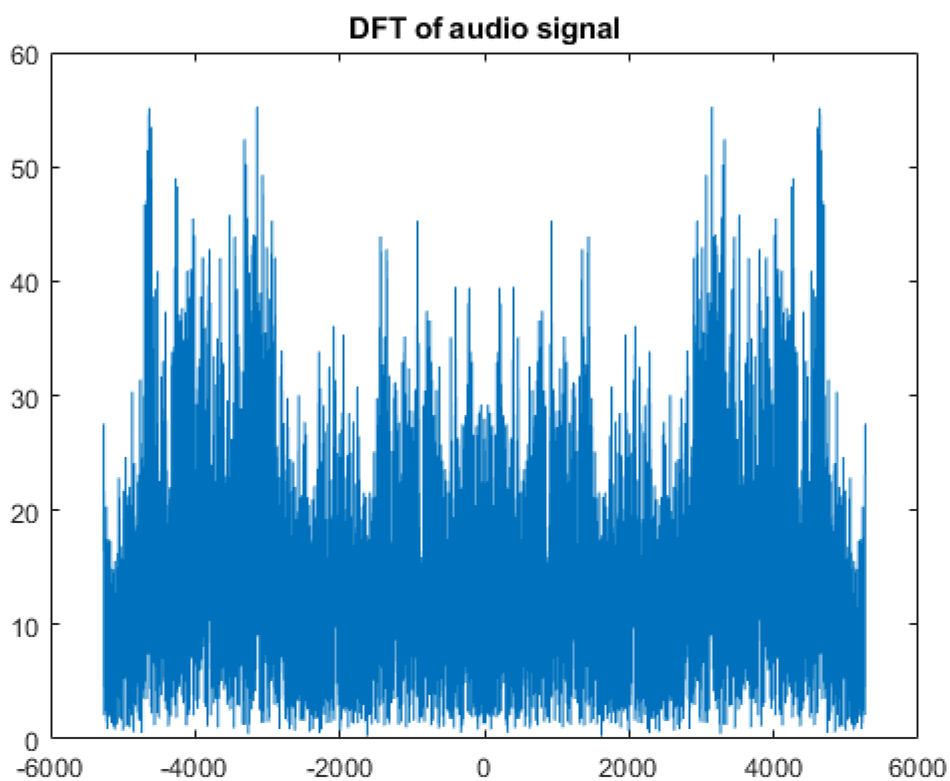

Signal y vs Time
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
1
2
3
4
5
6
×10^4


DFT of audio signal
300
250
200
150
100
50
0
-3
-2
-1
0
1
2
3
×10^4


Signal y vs Time
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
0.5
1
1.5
2
2.5
3
$\times 10^4$


DFT of audio signal
140
120
100
80
60
40
20
0
-1.5
-1
-0.5
0
0.5
1
1.5
$\times 10^4$


Signal y vs Time
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
2000
4000
6000
8000
10000
12000


DFT of audio signal
60
50
40
30
20
10
0
-6000
-4000
-2000
0
2000
4000
6000

C) Filter Design

Part 1 - Rect Filter

```
clc
clear
load laughter.mat;
filename = "laughter.wav";
audiowrite (filename, y, Fs);
[y, Fs] = audioread ("laughter.wav")
; n = length(y);
range = -n/2:((n/2)-1);
period = 1/Fs; p = n*period;
t = (0 : (n-1)); j = 1/p;
f = range*j;
filter_2000 = abs(f) < 2000; Y = fftshift(fft(y.'));
Yfiltered = Y.*filter_2000;
ytime = ifft(fftshift(Yfiltered));
figure(); plot(t,real(ytime));
title ("Plot of filtered sound (2000kHz) in time domain");
xlabel('t'); ylabel('amplitude');
figure(); plot(f, abs(Yfiltered));
title ("Plot of filtered sound (2000kHz) in frequency domain");
xlabel('frequency'); ylabel('|Y(Ω)|');
```

Part 2 - Playing the Rect Filtered Signal

```
sound(real(ytime),Fs);
```

Part 3 - Bass Frequency (16Hz - 256Hz) Filter

```
[y, Fs] = audioread ("laughter.wav");
n = length(y);
range = -n/2:((n/2)-1);
```

```
period = 1/Fs; p = n*period;
t = (0 : (n-1)); j = 1/p;
f = range*j;
filter_16_256 = abs(f) < 16 | abs(f) > 256;
Y = fftshift(fft(y.'));
Yfiltered = Y.*filter_16_256;
ytime = ifft(fftshift(Yfiltered));
figure(); plot(t,real(ytime));
title ("Plot of filtered sound (16-256kHz) in time domain");
xlabel('t'); ylabel('amplitude');
figure(); plot(f, abs(Yfiltered));
title ("Plot of filtered sound (16-256kHz) in frequency domain");
xlabel('frequency');
ylabel('|Y(Ω)|');
sound(real(ytime),Fs);
```

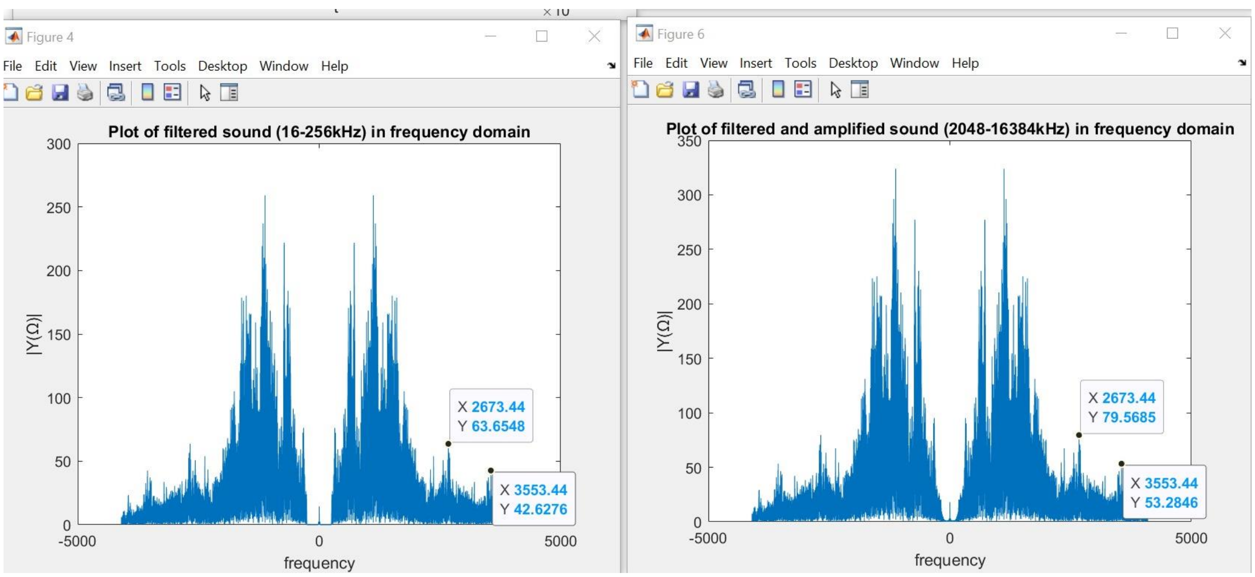
The frequency spectrum no longer contains frequencies between 16 Hz and 256 Hz.

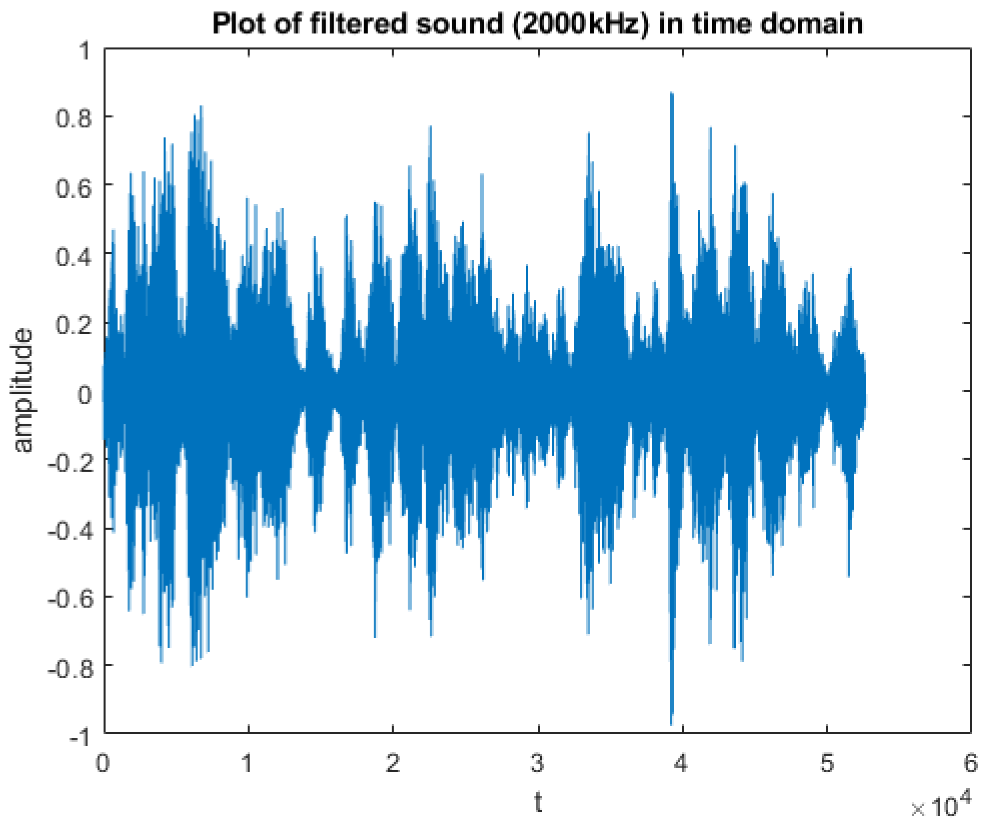
Part 4 - Treble Frequency (2048Hz - 16384Hz) Amplifying Filter

```
[y, Fs] = audioread ("laughter.wav");
n = length(y); range = -n/2:((n/2)-1);
period = 1/Fs; p = n*period;
t = (0 : (n-1)); j = 1/p;
f = range*j;
filter_2048_16384 = abs(f) > 2048 | abs(f) < 16384;
Y = fftshift(fft(y.'));
Yfiltered = Y.*filter_2048_16384.*1.25; ytime = ifft(fftshift(Yfiltered));
figure(); plot(t,real(ytime));
title ("Plot of filtered and amplified sound (2048-16384kHz) in time domain");
xlabel('t');
ylabel('amplitude');
```

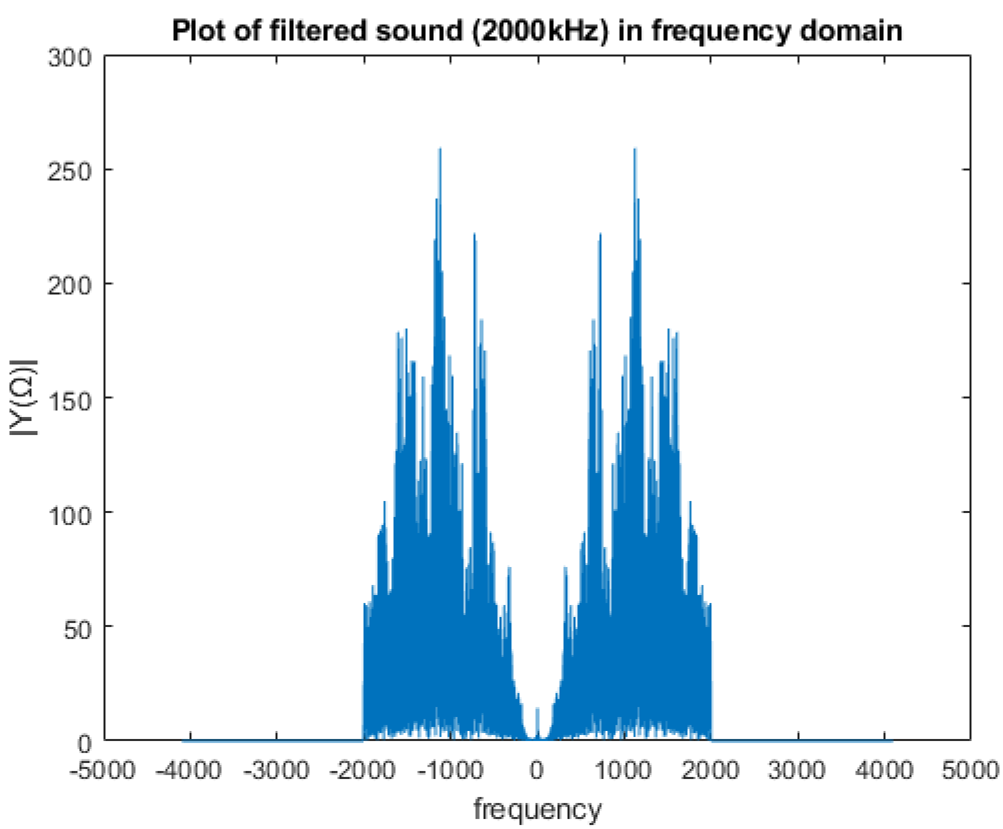
```
figure();
plot(f, abs(Yfiltered));
title ("Plot of filtered and amplified sound (2048-16384kHz) in frequency domain");
xlabel('frequency');
ylabel('|Y(Ω)|');
sound(real(ytime),Fs);
```

The frequency has been amplified by 25% for frequencies between 2048 Hz and 16384 Hz.

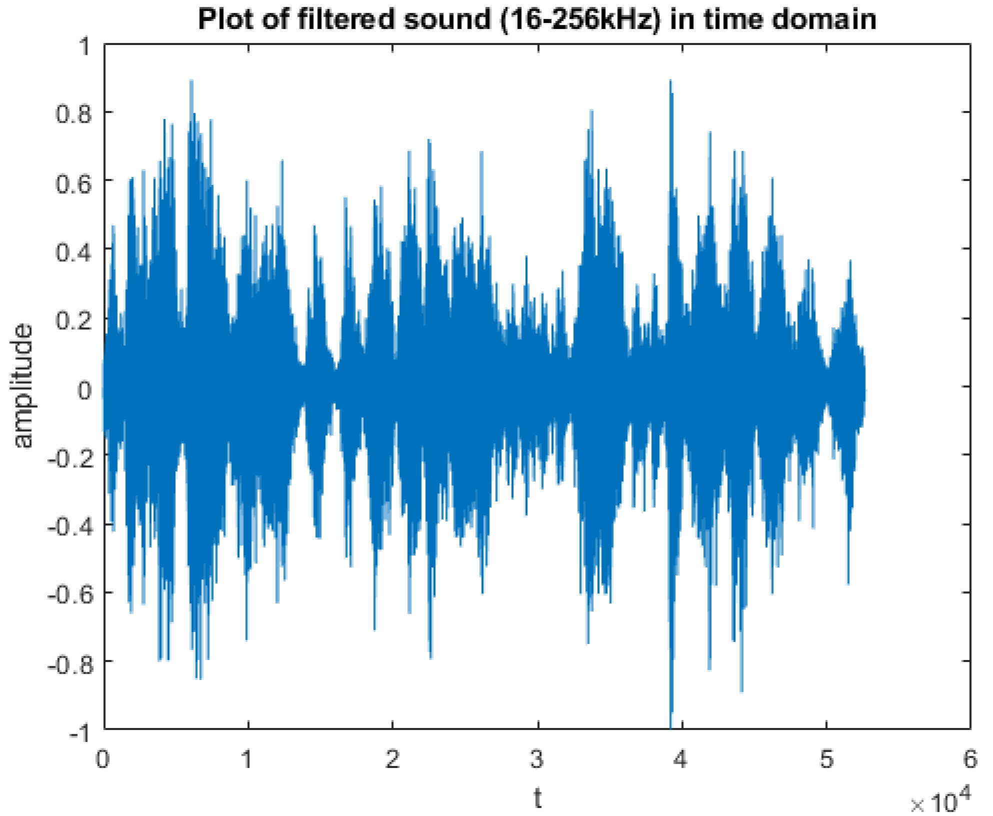


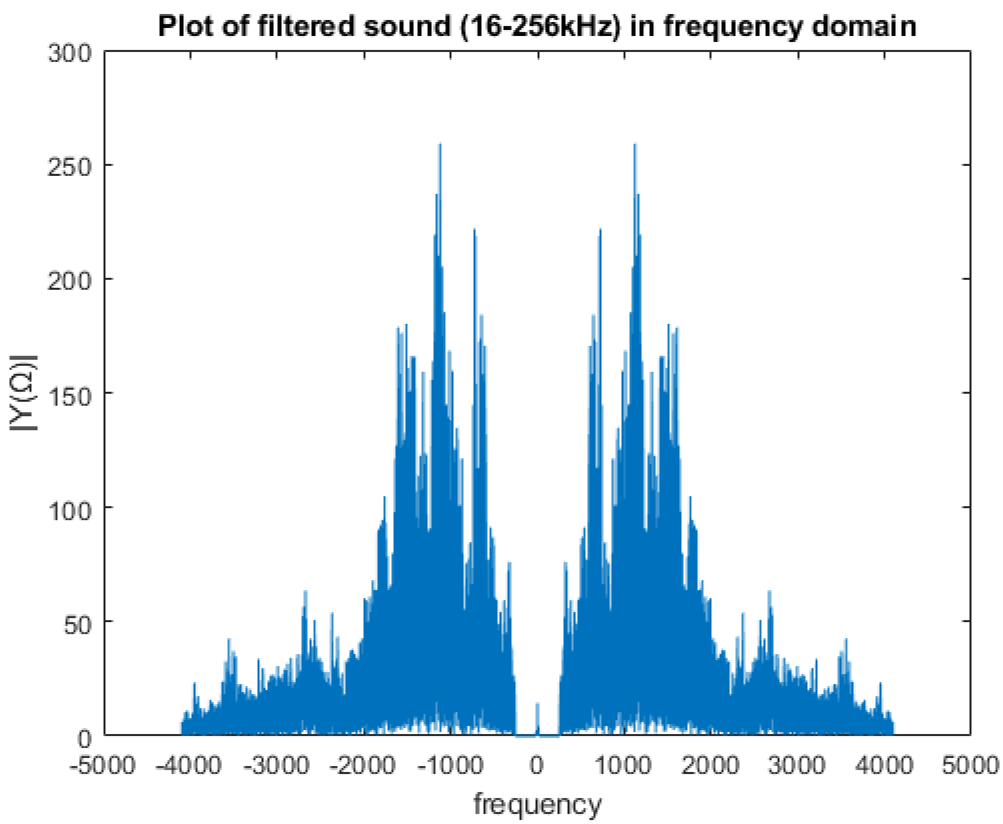


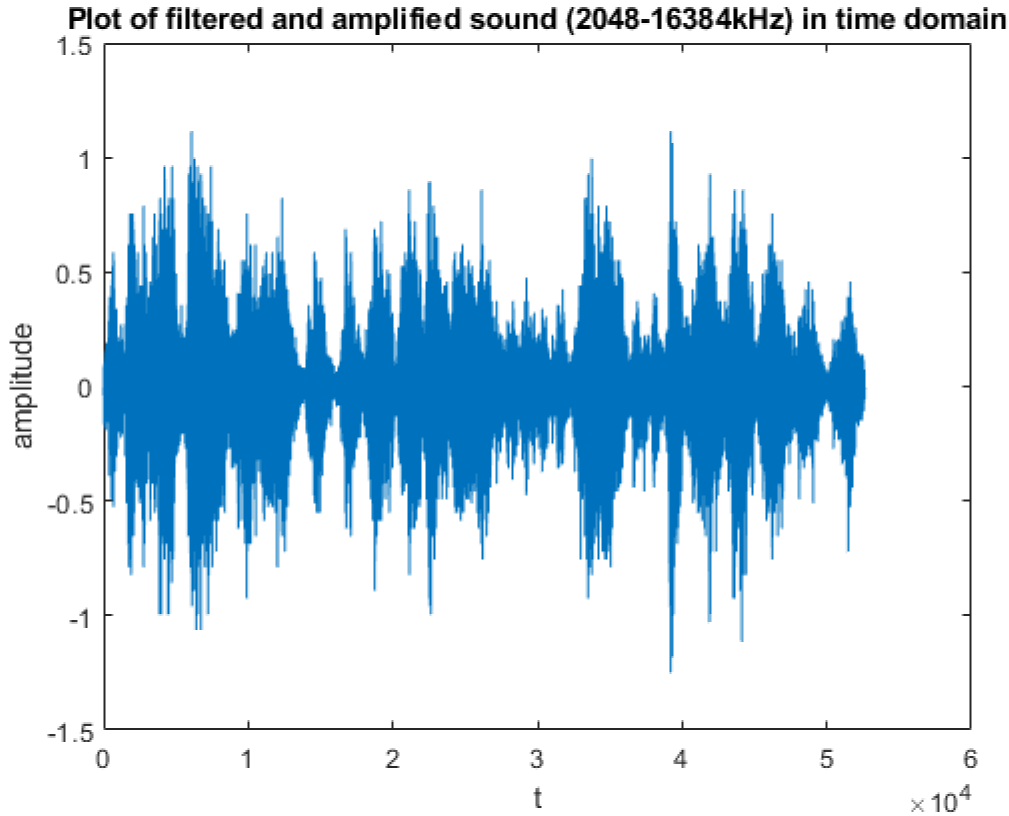
Plot of filtered sound (2000kHz) in time domain
amplitude
t
×10^4

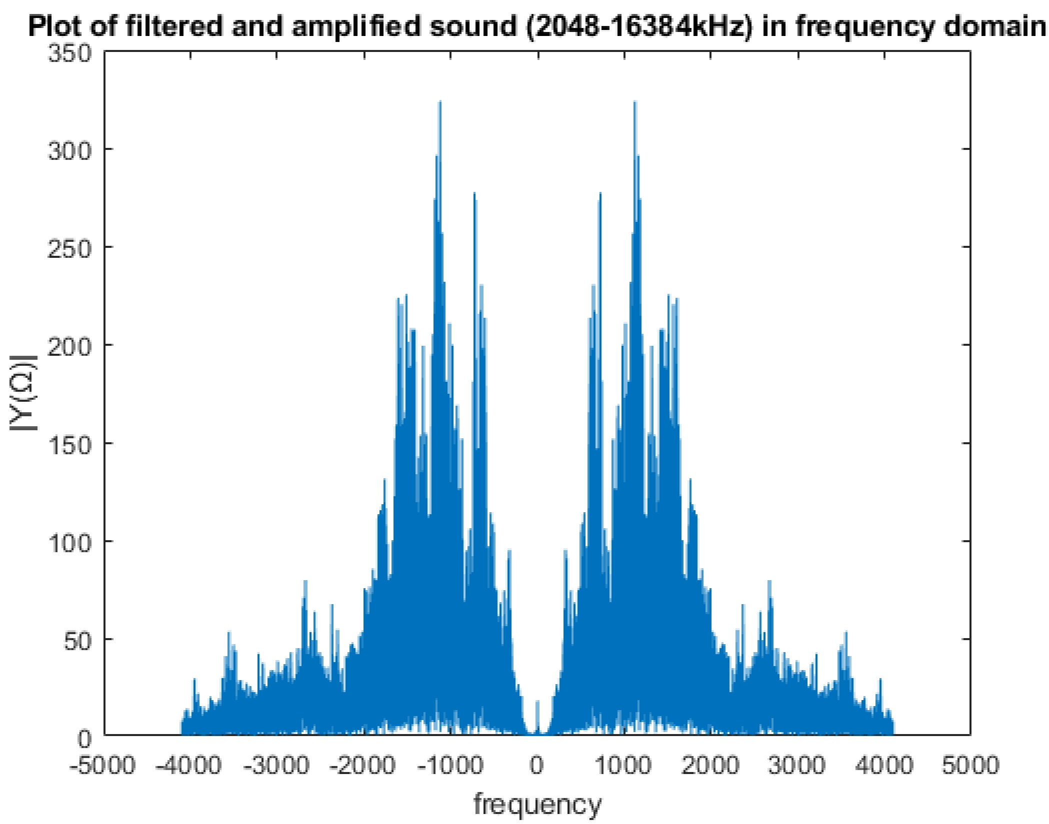


Plot of filtered sound (2000kHz) in frequency domain
|Y(Ω)|
frequency


Plot of filtered sound (16-256kHz) in time domain
amplitude
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
1
2
3
4
5
6
t
$\times 10^4$


Plot of filtered sound (16-256kHz) in frequency domain
$|Y(\Omega)|$
300
250
200
150
100
50
0
-5000
-4000
-3000
-2000
-1000
0
1000
2000
3000
4000
5000
frequency


Plot of filtered and amplified sound (2048-16384kHz) in time domain
amplitude
1.5
1
0.5
0
-0.5
-1
-1.5
0
1
2
3
4
5
6
t
$\times 10^4$


Plot of filtered and amplified sound (2048-16384kHz) in frequency domain
|Y(Ω)|
350
300
250
200
150
100
50
0
-5000
-4000
-3000
-2000
-1000
0
1000
2000
3000
4000
5000
frequency

Part 5 - Question Response

The property of DFT used to perform the task in Part 4 is the Convolution theorem, in which convolution in the time domain is equivalent to multiplication in the frequency domain.