American International University- Bangladesh (AIUB)
Faculty of Engineering
Principles of Communications Lab
OBE Assessment Lab Project

Course Name:	**Principles of Communications**		
Course Code:	EEE 3215	**Section:**	B
Semester:	Spring 2021-22	**Group No:**	**03**

Assignment Name:	**Final Term Lab Project (OBE)**		
Assessed CO2:	Investigate a Pulse coded modulation (PCM) transmitter system for an audio signal to verify sampling and quantization through appropriate research.		
Assessed POI:	**P.d.1.C5**		
Student Name:	**Shahrukh Islam**	**Student ID:**	**18-36099-1**
Student Name:	**Momtahin Ahammed**	**Student ID:**	**19-40698-1**
Student Name:	**Moon Md Umar Faruk**	**Student ID:**	**18-37425-1**
Student Name:	**Md. Shafiqul Islam**	**Student ID:**	**18-37155-1**
Student Name:	**Tonmoy Hassan**	**Student ID:**	**18-37777-2**
Student Name:	**Ashik Ashraf Chowdhury**	**Student ID:**	**18-38025-2**
Student Name:	**Md. Muhit Alam Khan**	**Student ID:**	**18-38831-3**

Mark distribution (to be filled by Faculty):

Objectives	Proficient [10-8]	Good [7-4]	Needs Improvement [3-1]	Secured Marks
Depth of knowledge displayed through appropriate research (P1)	Student was able to apply in-depth engineering knowledge achieved by appropriate research about digital/analog communication to design the communication model correctly and **fulfilled all design criteria**.	Design process is not completely supported by in-depth engineering knowledge achieved by appropriate research about digital/analog communication, **some but not all of the design criteria are fulfilled**.	Design process contains mistakes and does not display enough in-depth engineering knowledge achieved by appropriate research about digital/analog communication. **Most of the design criteria are not fulfilled**.	
Depth of analysis (P3)	Student defended the diversified approach taken to solve the problem with **well-justified in-depth analysis that demonstrated abstract thinking**.	Student's attempts to analyze the diversified approach taken to solve the problem **is not enough in-depth, some of design choices do not demonstrate adequate abstract thinking** and are not properly justified.	Student **did not attempt any in-depth analysis** of the designed system and **displayed no abstract thinking**.	
Level of integration of multiple sections of design for solution of high-level problem (P7)	Student correctly identified all problems and successfully integrated the interdependent parts into a high-level design using a block diagram. Block diagram was at best match with the given problem.	Student was able to identify some of the problems correctly and integrated the interdependent parts into a high-level design using a block diagram. Some parts of the block diagram were not a good match for the given problem.	Student was able to identify only one/two of the problems correctly and could not properly integrate the interdependent parts into a high-level design using a block diagram. Only one/two blocks were correct and/or block diagram was incomplete.	
Comments:			**Total Marks (Out of 30):**	

Title: Design a PCM system having input audio signal, sampler, and quantization modules

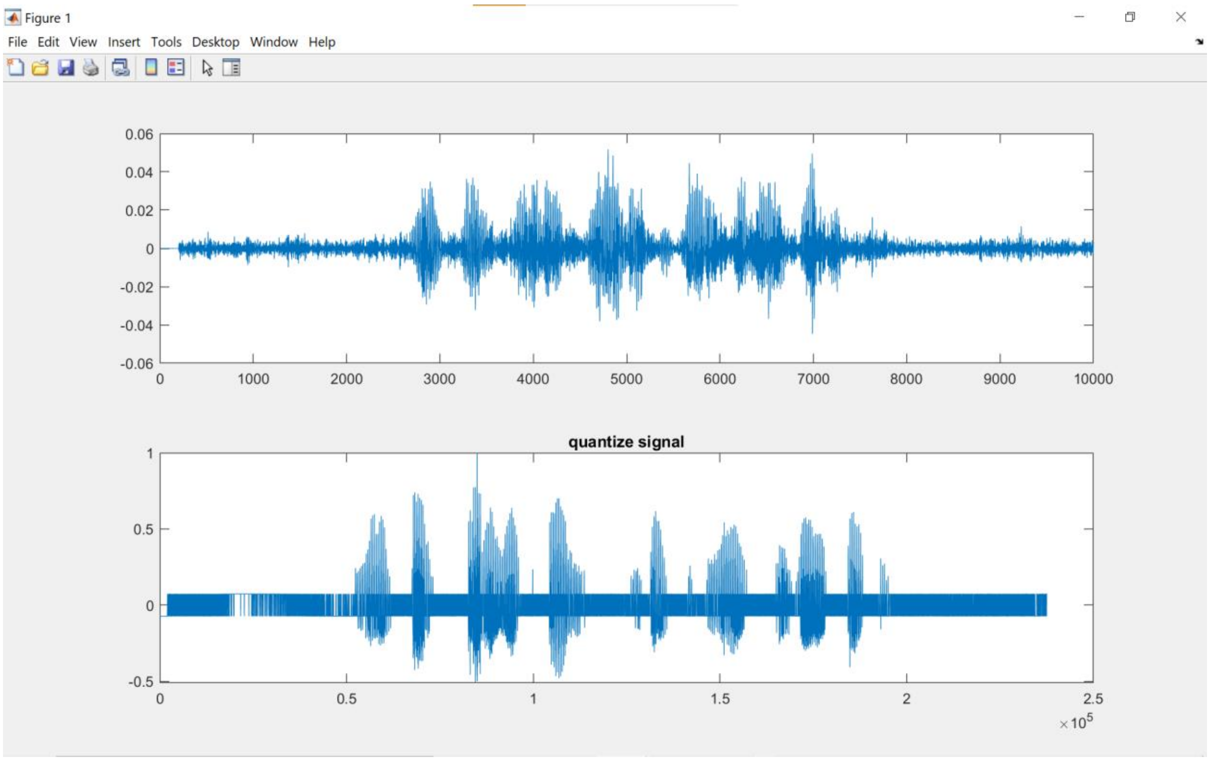
MATLAB CODE

2000 hz 2 bit

```
clc;
clear all;
close all;

rectime = 5;

fs= 2000; %use  2k, 8k, 44.1k, 48k

record = audiorecorder (fs, 16, 1) ;

disp ('Start Speaking');

recordblocking (record, rectime);

disp ('End Recording');

y= getaudiodata(record) ;

sound (y, fs) ;%Play the sound

subplot(2,1,1)

plot(y)

pause (9)

filename_ori = 'OBE_PROJECT.aac';

[y,fso] = audioread(filename_ori);

fs= 2000; %change the sampling rate, use 1000, 2k, 8k, 44.1k, 48k

audiowrite('sampled.wav',y,fs)

[mySpeech,fs] = audioread('sampled.wav');%Read the data back into

sound(mySpeech,fs);

n=2
y_max = max(y)
y_min = min(y)
qu = y/y_max(1)
d = (y_max - y_min)/n
d = d(1)
q = d.*[0:n-1]
q = q-((n-1)/2)*d
for i = 1:n
qu (find((q(i)-d/2<=qu )&(qu <= q(i)+d/2)))= q(i).*ones
(1,length(find((q(i)- d/2<= qu )&(qu <=q(i) +d/2))));
end
sound (qu )
subplot (2,1,2)
plot (qu )
```

```
title ('quantize signal');
%saving file
filename='quants_2000_2.wav';
audiowrite (filename,qu ,fs) ;
```

Output

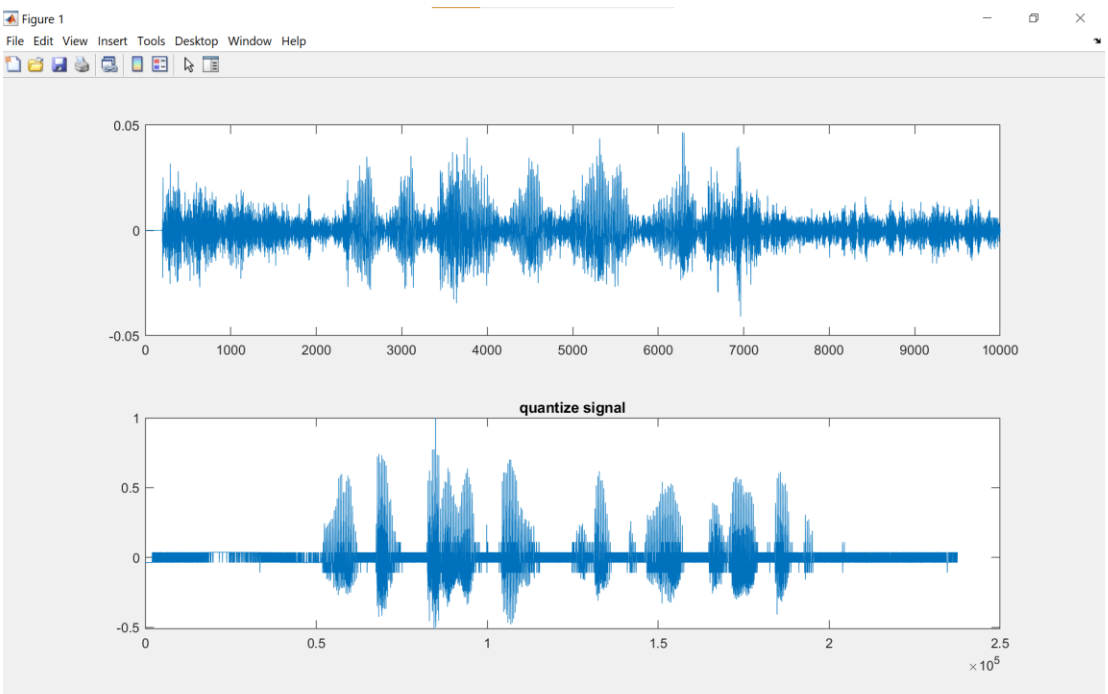


2000 hz 4 bit

```
clc;
clear all;
close all;

rectime = 5;

fs= 2000; %use  2k, 8k, 44.1k, 48k

record = audiorecorder (fs, 16, 1) ;

disp ('Start Speaking');

recordblocking (record, rectime);

disp ('End Recording');

y= getaudiodata(record) ;

sound (y, fs) ;%Play the sound

subplot(2,1,1)
```

```
plot(y)

pause (9)

filename_ori = 'OBE_PROJECT.aac';

[y,fso] = audioread(filename_ori);

fs= 2000; %change the sampling rate, use 1000, 2k, 8k, 44.1k, 48k

audiowrite('sampled.wav',y,fs)

[mySpeech,fs] = audioread('sampled.wav');%Read the data back into
sound(mySpeech,fs);

n=4
y_max = max(y)
y_min = min(y)
qu = y/y_max(1)
d = (y_max - y_min)/n
d = d(1)
q = d.*[0:n-1]
q = q-((n-1)/2)*d
for i = 1:n
qu (find((q(i)-d/2<=qu )&(qu <= q(i)+d/2)))= q(i).*ones
(1,length(find((q(i)- d/2<= qu )&(qu <=q(i) +d/2))));

end
sound (qu )
subplot (2,1,2)
plot (qu )
title ('quantize signal');
%saving file
filename='quants_2000_4.wav';
audiowrite (filename,qu ,fs) ;
```

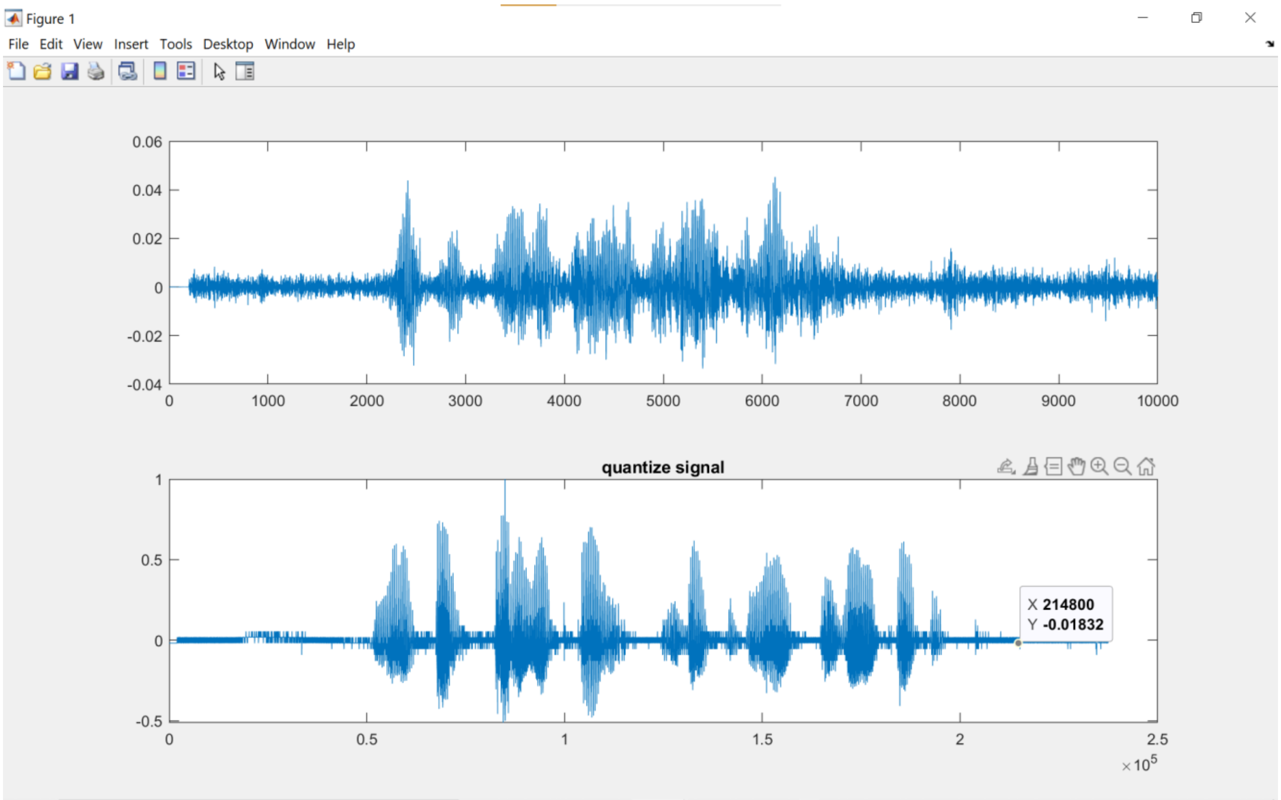
Output



2000 hz 8 bit

```
clc;
clear all;
close all;

rectime = 5;

fs= 2000; %use  2k, 8k, 44.1k, 48k

record = audiorecorder (fs, 16, 1) ;

disp ('Start Speaking');

recordblocking (record, rectime);

disp ('End Recording');

y= getaudiodata(record) ;

sound (y, fs) ;%Play the sound

subplot(2,1,1)

plot(y)

pause (9)

filename_ori = 'OBE_PROJECT.aac';

[y,fso] = audioread(filename_ori);

fs= 2000; %change the sampling rate, use 1000, 2k, 8k, 44.1k, 48k

audiowrite('sampled.wav',y,fs)

[mySpeech,fs] = audioread('sampled.wav');%Read the data back into

sound(mySpeech,fs);

n=8
y_max = max(y)
y_min = min(y)
qu = y/y_max(1)
d = (y_max - y_min)/n
d = d(1)
q = d.*[0:n-1]
q = q-((n-1)/2)*d
for i = 1:n
qu (find((q(i)-d/2<=qu )&(qu <= q(i)+d/2)))= q(i).*ones
(1,length(find((q(i)- d/2<= qu )&(qu <=q(i) +d/2))));

end
sound (qu )
subplot (2,1,2)
plot (qu )
title ('quantize signal');
%saving file
filename='quants_2000_8.wav';
audiowrite (filename,qu ,fs) ;
```

Output

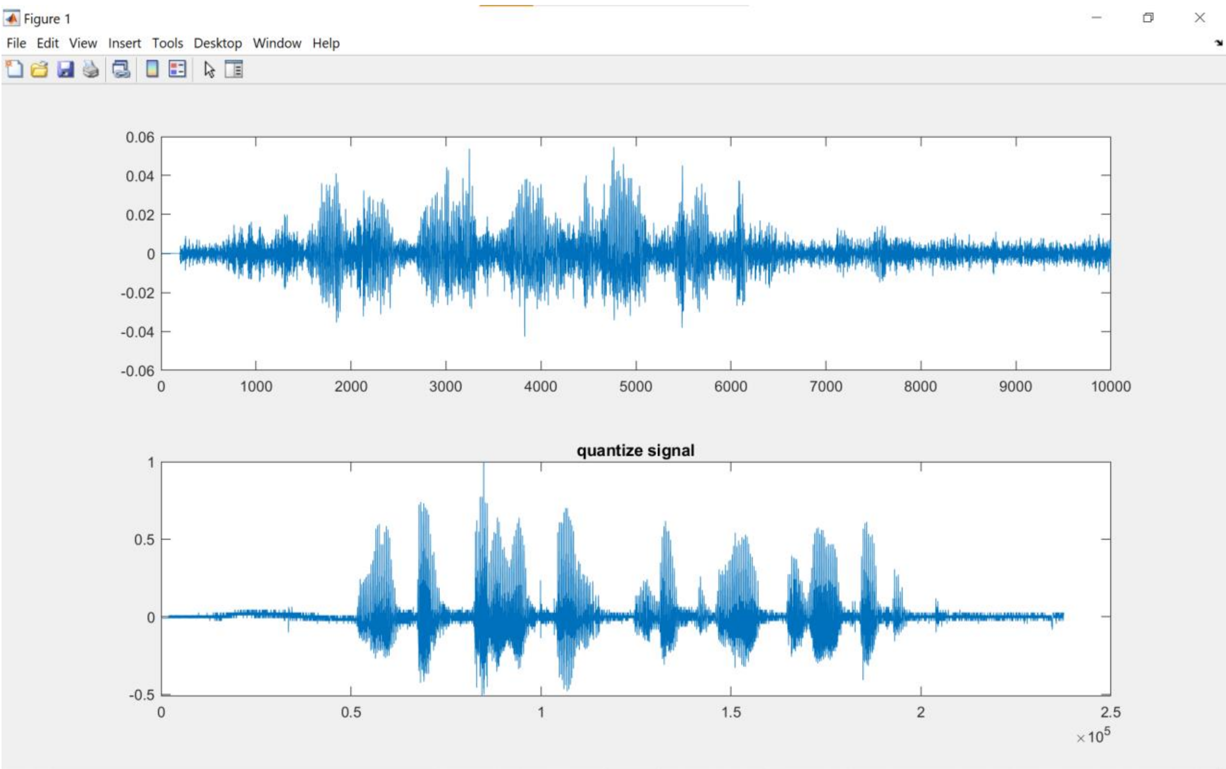


2000 hz 16 bit

```
clc;
clear all;
close all;

rectime = 5;

fs= 2000; %use  2k, 8k, 44.1k, 48k

record = audiorecorder (fs, 16, 1) ;

disp ('Start Speaking');

recordblocking (record, rectime);

disp ('End Recording');

y= getaudiodata(record) ;

sound (y, fs) ;%Play the sound

subplot(2,1,1)

plot(y)

pause (9)

filename_ori = 'OBE_PROJECT.aac';
```

```
[y,fso] = audioread(filename_ori);

fs= 2000; %change the sampling rate, use 1000, 2k, 8k, 44.1k, 48k

audiowrite('sampled.wav',y,fs)

[mySpeech,fs] = audioread('sampled.wav');%Read the data back into

 sound(mySpeech,fs);

n=16
y_max = max(y)
y_min = min(y)
qu = y/y_max(1)
d = (y_max - y_min)/n
d = d(1)
q = d.*[0:n-1]
q = q-((n-1)/2)*d
for i = 1:n
qu (find((q(i)-d/2<=qu )&(qu <= q(i)+d/2)))= q(i).*ones
(1,length(find((q(i)- d/2<= qu )&(qu <=q(i) +d/2))));

end
sound (qu )
subplot (2,1,2)
plot (qu )
title ('quantize signal');
%saving file
filename='quants_2000_16.wav';
audiowrite (filename,qu ,fs) ;
```

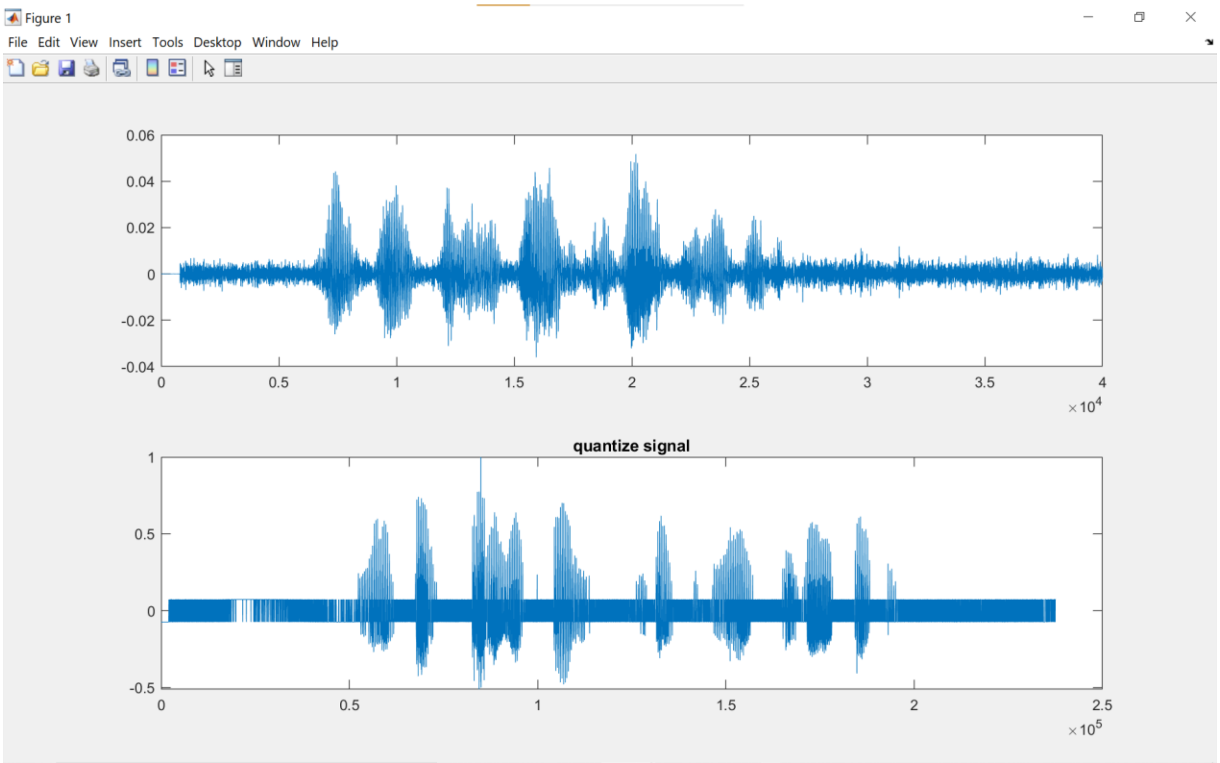
Output



8000 hz 2 bit

```
clc;
clear all;
close all;

rectime = 5;

fs= 8000; %use  2k, 8k, 44.1k, 48k

record = audiorecorder (fs, 16, 1) ;

disp ('Start Speaking');

recordblocking (record, rectime);

disp ('End Recording');

y= getaudiodata(record) ;

sound (y, fs) ;%Play the sound

subplot(2,1,1)

plot(y)

pause (9)

filename_ori = 'OBE_PROJECT.aac';

[y,fso] = audioread(filename_ori);

fs= 8000; %change the sampling rate, use 1000, 2k, 8k, 44.1k, 48k

audiowrite('sampled.wav',y,fs)

[mySpeech,fs] = audioread('sampled.wav');%Read the data back into

 sound(mySpeech,fs);

n=2
y_max = max(y)
y_min = min(y)
qu = y/y_max(1)
d = (y_max - y_min)/n
d = d(1)
q = d.*[0:n-1]
q = q-((n-1)/2)*d
for i = 1:n
qu (find((q(i)-d/2<=qu )&(qu <= q(i)+d/2)))= q(i).*ones
(1,length(find((q(i)- d/2<= qu )&(qu <=q(i) +d/2))));

end
sound (qu )
subplot (2,1,2)
plot (qu )
title ('quantize signal');
%saving file
filename='quants_8000_2.wav';
audiowrite (filename,qu ,fs) ;
```

<u>Output</u>

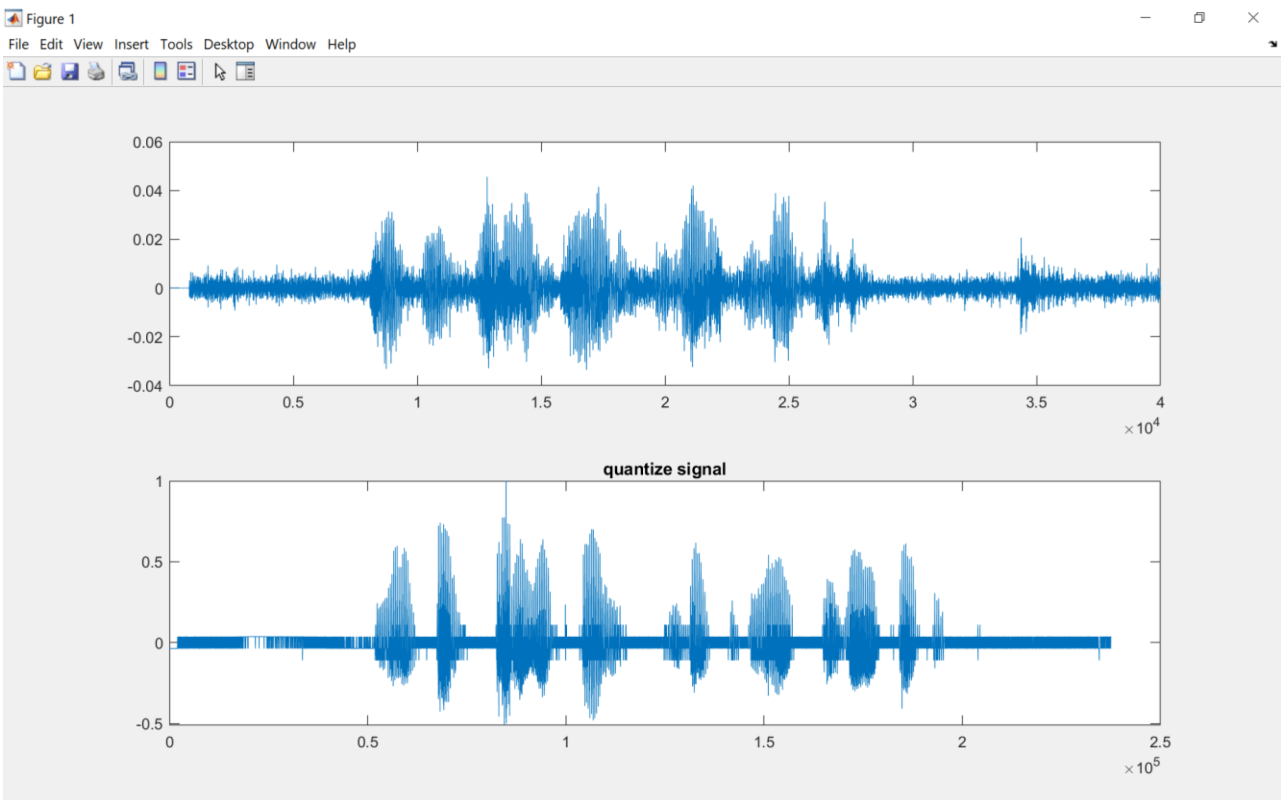


8000 hz 4 bit

```
clc;
clear all;
close all;

rectime = 5;

fs= 8000; %use  2k, 8k, 44.1k, 48k

record = audiorecorder (fs, 16, 1) ;

disp ('Start Speaking');

recordblocking (record, rectime);

disp ('End Recording');

y= getaudiodata(record) ;

sound (y, fs) ;%Play the sound

subplot(2,1,1)

plot(y)

pause (9)

filename_ori = 'OBE_PROJECT.aac';

[y,fso] = audioread(filename_ori);
```

```
fs= 8000; %change the sampling rate, use 1000, 2k, 8k, 44.1k, 48k

audiowrite('sampled.wav',y,fs)

[mySpeech,fs] = audioread('sampled.wav');%Read the data back into

 sound(mySpeech,fs);

n=4
y_max = max(y)
y_min = min(y)
qu = y/y_max(1)
d = (y_max - y_min)/n
d = d(1)
q = d.*[0:n-1]
q = q-((n-1)/2)*d
for i = 1:n
qu (find((q(i)-d/2<=qu )&(qu <= q(i)+d/2)))= q(i).*ones
(1,length(find((q(i)- d/2<= qu )&(qu <=q(i) +d/2))));

end
sound (qu )
subplot (2,1,2)
plot (qu )
title ('quantize signal');
%saving file
filename='quants_8000_4.wav';
audiowrite (filename,qu ,fs) ;
```

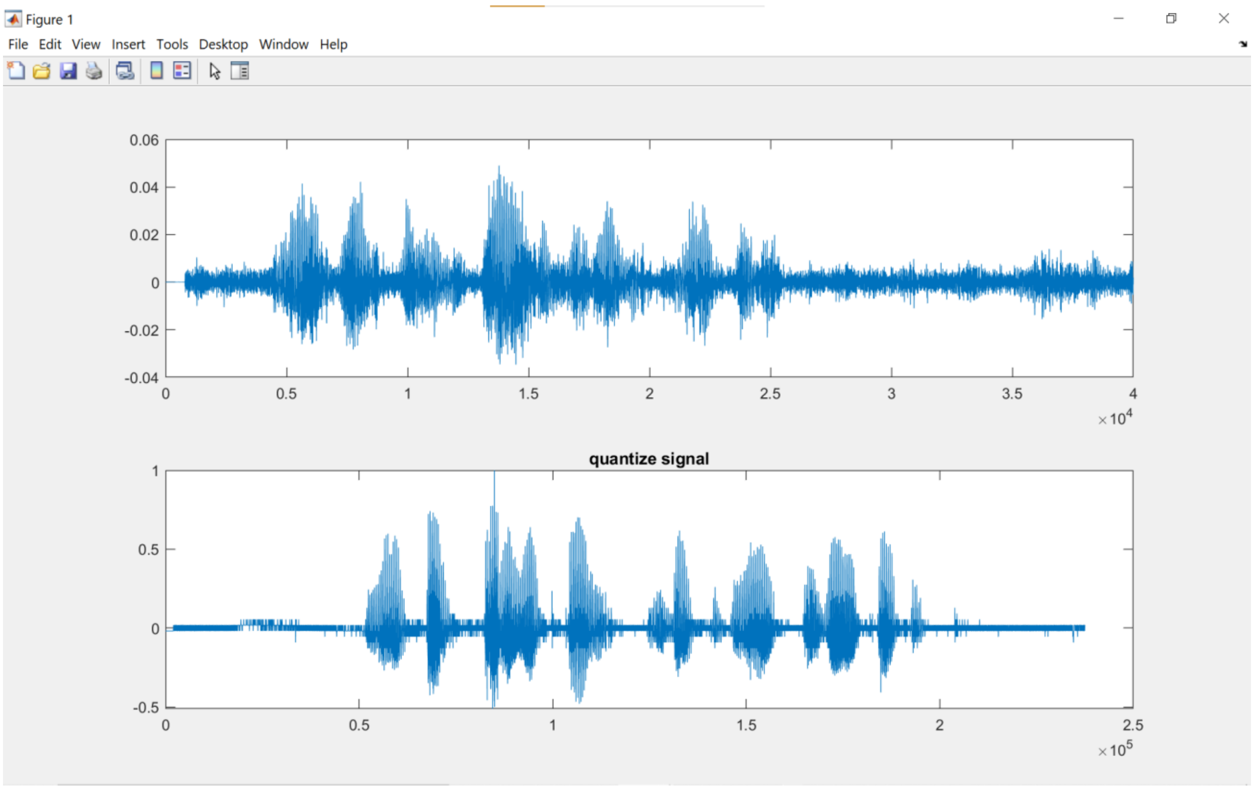
Output



8000 hz 8 bit

```
clc;
clear all;
close all;

rectime = 5;

fs= 8000; %use  2k, 8k, 44.1k, 48k

record = audiorecorder (fs, 16, 1) ;

disp ('Start Speaking');

recordblocking (record, rectime);

disp ('End Recording');

y= getaudiodata(record) ;

sound (y, fs) ;%Play the sound

subplot(2,1,1)

plot(y)

pause (9)

filename_ori = 'OBE_PROJECT.aac';

[y,fso] = audioread(filename_ori);

fs= 8000; %change the sampling rate, use 1000, 2k, 8k, 44.1k, 48k

audiowrite('sampled.wav',y,fs)

[mySpeech,fs] = audioread('sampled.wav');%Read the data back into

 sound(mySpeech,fs);

n=8
y_max = max(y)
y_min = min(y)
qu = y/y_max(1)
d = (y_max - y_min)/n
d = d(1)
q = d.*[0:n-1]
q = q-((n-1)/2)*d
for i = 1:n
qu (find((q(i)-d/2<=qu )&(qu <= q(i)+d/2)))= q(i).*ones
(1,length(find((q(i)- d/2<= qu )&(qu <=q(i) +d/2))));

end
sound (qu )
subplot (2,1,2)
plot (qu )
title ('quantize signal');
%saving file
filename='quants_8000_8.wav';
audiowrite (filename,qu ,fs) ;
```

<u>Output</u>

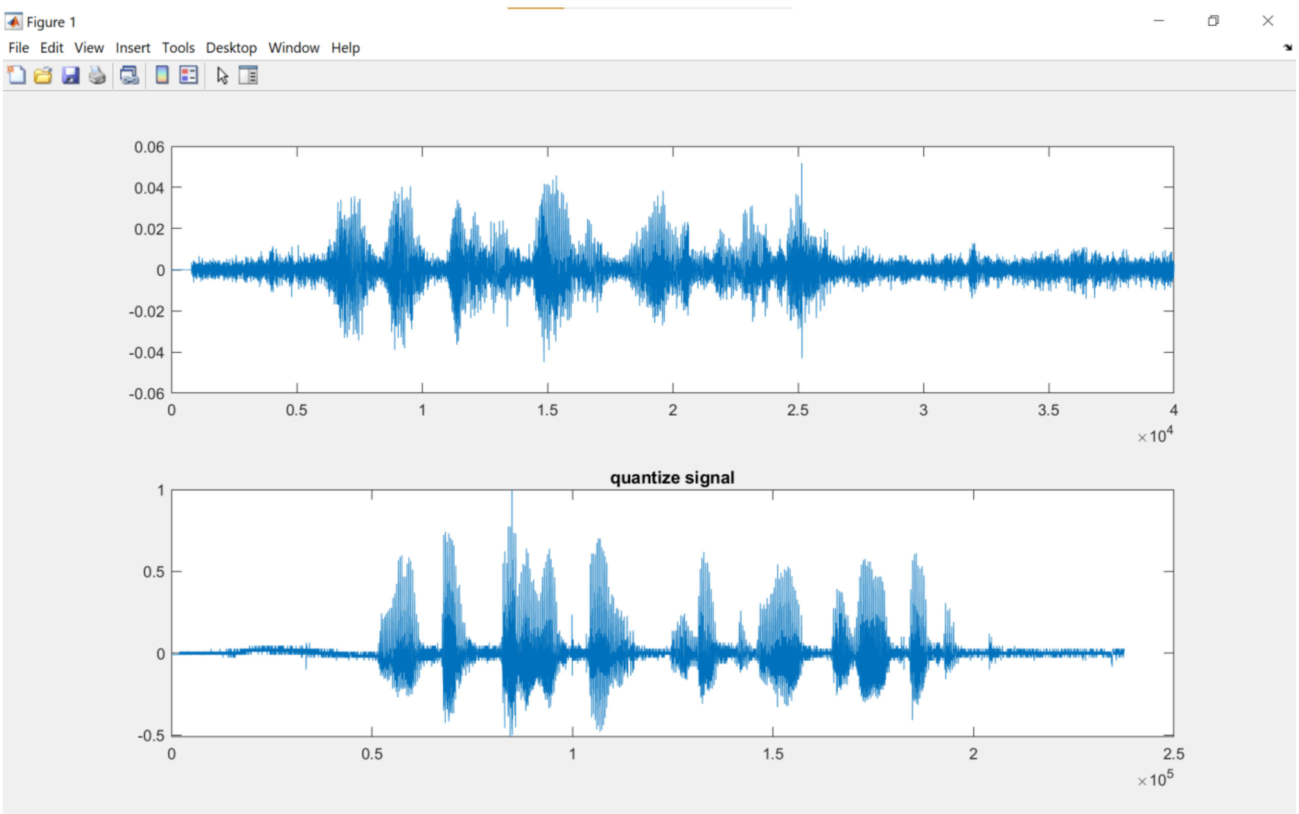


8000 hz 16 bit

```
clc;
clear all;
close all;

rectime = 5;

fs= 8000; %use  2k, 8k, 44.1k, 48k

record = audiorecorder (fs, 16, 1) ;

disp ('Start Speaking');

recordblocking (record, rectime);

disp ('End Recording');

y= getaudiodata(record) ;

sound (y, fs) ;%Play the sound

subplot(2,1,1)

plot(y)

pause (9)

filename_ori = 'OBE_PROJECT.aac';
```

```
[y,fso] = audioread(filename_ori);

fs= 8000; %change the sampling rate, use 1000, 2k, 8k, 44.1k, 48k

audiowrite('sampled.wav',y,fs)

[mySpeech,fs] = audioread('sampled.wav');%Read the data back into

 sound(mySpeech,fs);

n=16
y_max = max(y)
y_min = min(y)
qu = y/y_max(1)
d = (y_max - y_min)/n
d = d(1)
q = d.*[0:n-1]
q = q-((n-1)/2)*d
for i = 1:n
qu (find((q(i)-d/2<=qu )&(qu <= q(i)+d/2)))= q(i).*ones
(1,length(find((q(i)- d/2<= qu )&(qu <=q(i) +d/2))));

end
sound (qu )
subplot (2,1,2)
plot (qu )
title ('quantize signal');
%saving file
filename='quants_8000_16.wav';
audiowrite (filename,qu ,fs) ;
```

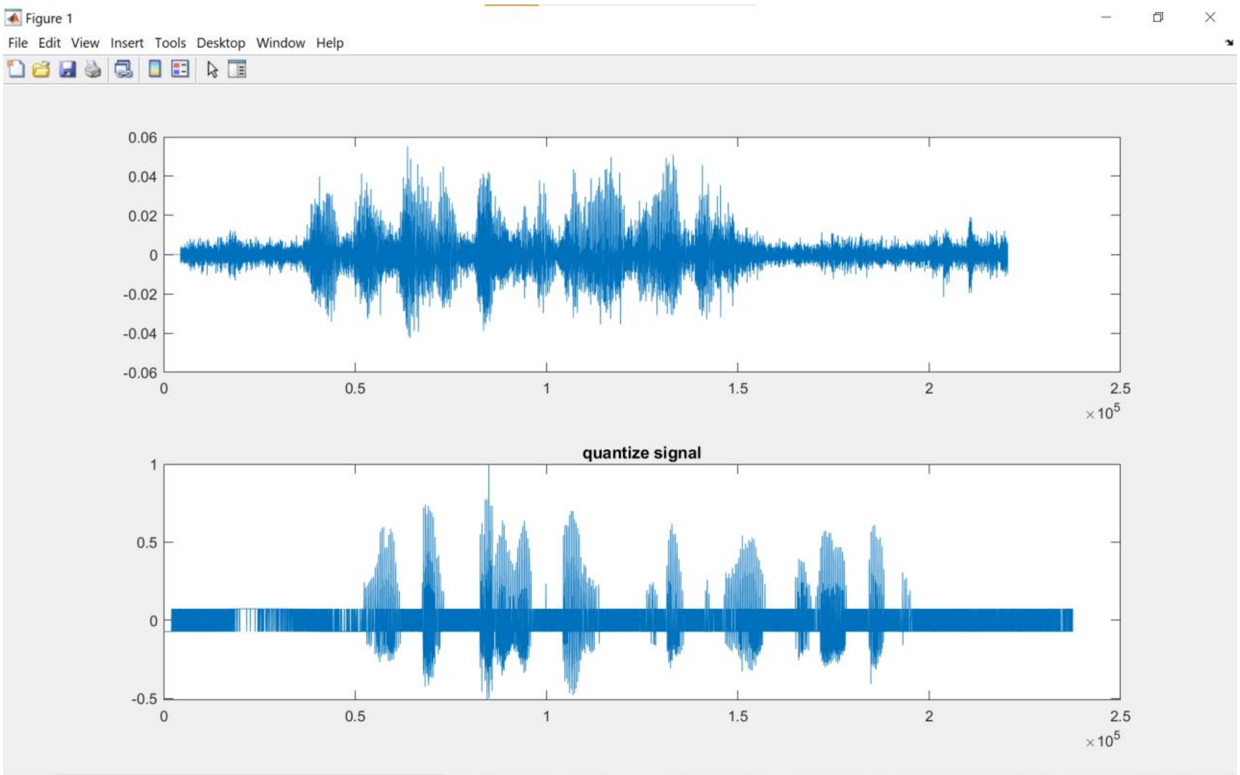
<u>Output</u>



44100 hz 2 bit

```
clc;
clear all;
close all;

rectime = 5;

fs= 44100; %use  2k, 8k, 44.1k, 48k

record = audiorecorder (fs, 16, 1) ;

disp ('Start Speaking');

recordblocking (record, rectime);

disp ('End Recording');

y= getaudiodata(record) ;

sound (y, fs) ;%Play the sound

subplot(2,1,1)

plot(y)

pause (9)

filename_ori = 'OBE_PROJECT.aac';

[y,fso] = audioread(filename_ori);

fs= 44100; %change the sampling rate, use 1000, 2k, 8k, 44.1k, 48k

audiowrite('sampled.wav',y,fs)

[mySpeech,fs] = audioread('sampled.wav');%Read the data back into

 sound(mySpeech,fs);

n=2
y_max = max(y)
y_min = min(y)
qu = y/y_max(1)
d = (y_max - y_min)/n
d = d(1)
q = d.*[0:n-1]
q = q-((n-1)/2)*d
for i = 1:n
qu (find((q(i)-d/2<=qu )&(qu <= q(i)+d/2)))= q(i).*ones
(1,length(find((q(i)- d/2<= qu )&(qu <=q(i) +d/2))));

end
sound (qu )
subplot (2,1,2)
plot (qu )
title ('quantize signal');
%saving file
filename='quants_44100_2.wav';
audiowrite (filename,qu ,fs) ;
```

Output

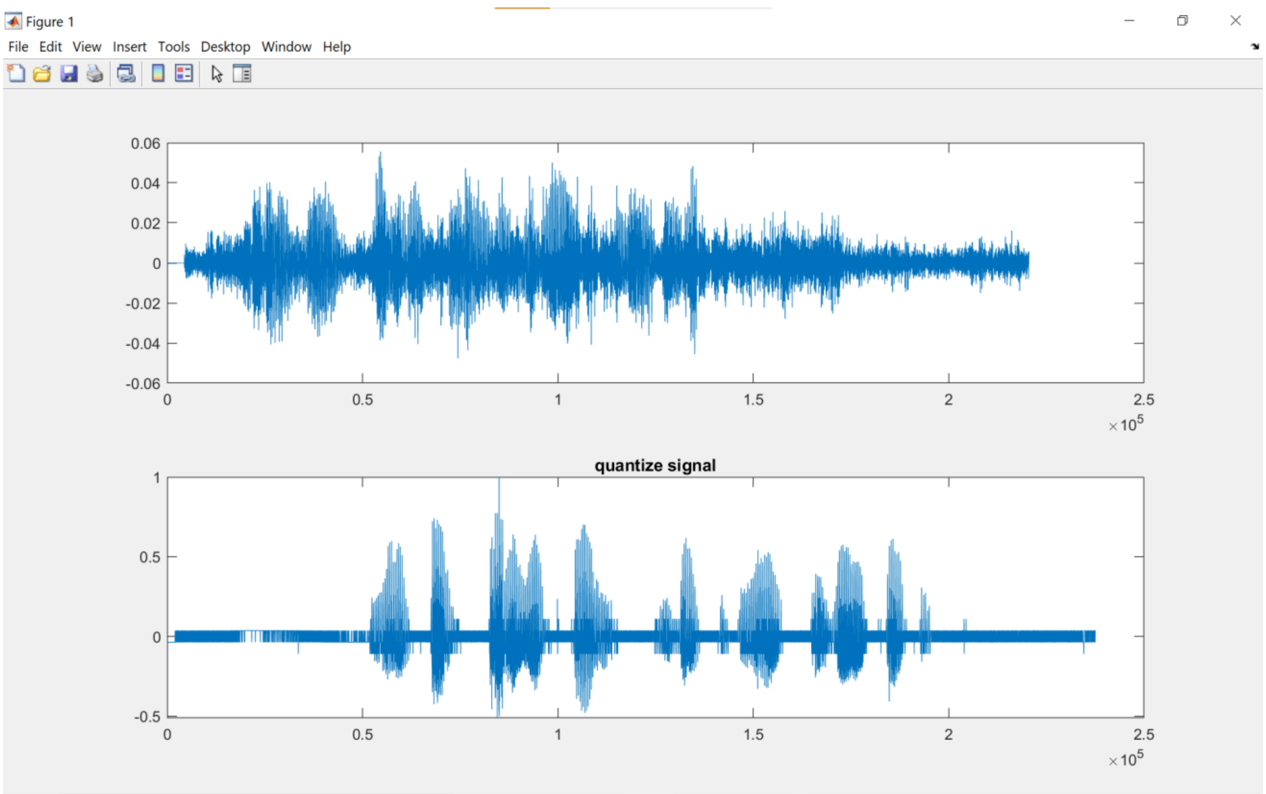


44100 hz 4 bit

```
clc;
clear all;
close all;

rectime = 5;

fs= 44100; %use  2k, 8k, 44.1k, 48k

record = audiorecorder (fs, 16, 1) ;

disp ('Start Speaking');

recordblocking (record, rectime);

disp ('End Recording');

y= getaudiodata(record) ;

sound (y, fs) ;%Play the sound

subplot(2,1,1)

plot(y)

pause (9)

filename_ori = 'OBE_PROJECT.aac';
```

```
[y,fso] = audioread(filename_ori);

fs= 44100; %change the sampling rate, use 1000, 2k, 8k, 44.1k, 48k

audiowrite('sampled.wav',y,fs)

[mySpeech,fs] = audioread('sampled.wav');%Read the data back into

 sound(mySpeech,fs);

n=4
y_max = max(y)
y_min = min(y)
qu = y/y_max(1)
d = (y_max - y_min)/n
d = d(1)
q = d.*[0:n-1]
q = q-((n-1)/2)*d
for i = 1:n
qu (find((q(i)-d/2<=qu )&(qu <= q(i)+d/2)))= q(i).*ones
(1,length(find((q(i)- d/2<= qu )&(qu <=q(i) +d/2))));

end
sound (qu )
subplot (2,1,2)
plot (qu )
title ('quantize signal');
%saving file
filename='quants_44100_4.wav';
audiowrite (filename,qu ,fs) ;
```

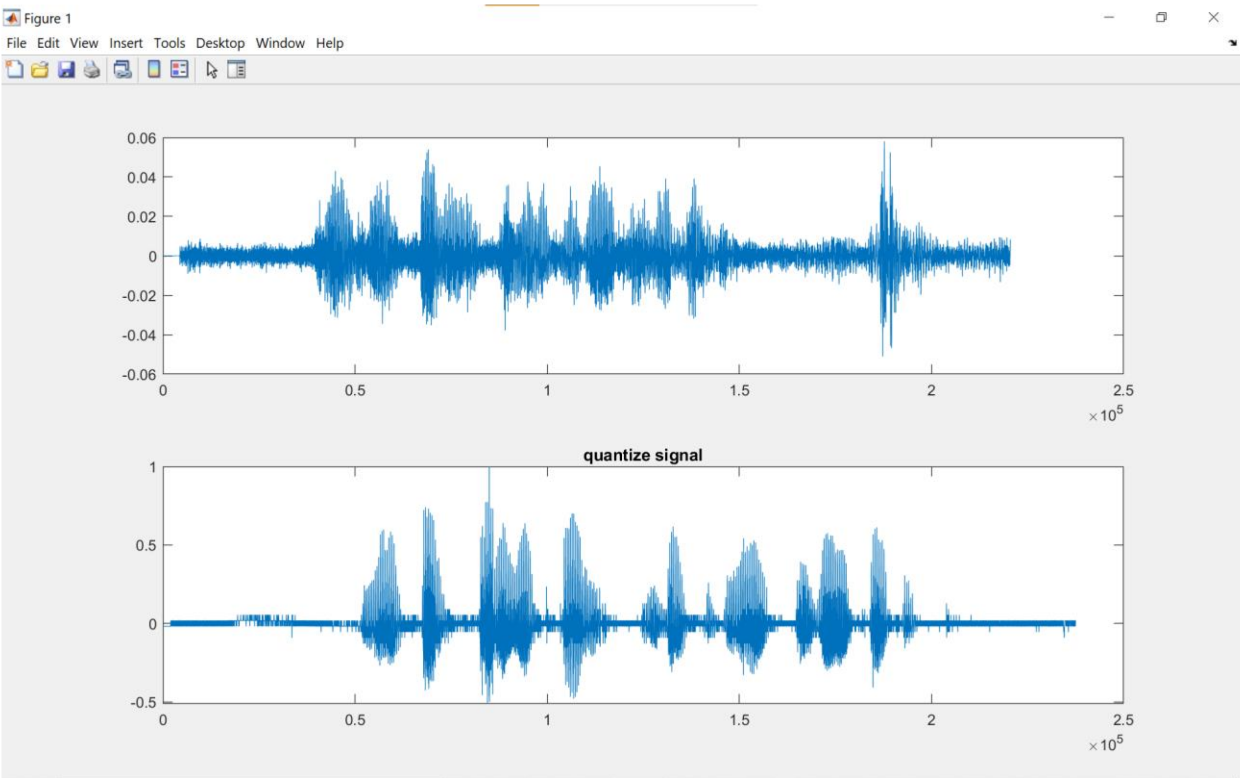
Output



44100 hz 8 bit

```
clc;
clear all;
close all;

rectime = 5;

fs= 44100; %use  2k, 8k, 44.1k, 48k

record = audiorecorder (fs, 16, 1) ;

disp ('Start Speaking');

recordblocking (record, rectime);

disp ('End Recording');

y= getaudiodata(record) ;

sound (y, fs) ;%Play the sound

subplot(2,1,1)

plot(y)

pause (9)

filename_ori = 'OBE_PROJECT.aac';

[y,fso] = audioread(filename_ori);

fs= 44100; %change the sampling rate, use 1000, 2k, 8k, 44.1k, 48k

audiowrite('sampled.wav',y,fs)

[mySpeech,fs] = audioread('sampled.wav');%Read the data back into

 sound(mySpeech,fs);

n=8
y_max = max(y)
y_min = min(y)
qu = y/y_max(1)
d = (y_max - y_min)/n
d = d(1)
q = d.*[0:n-1]
q = q-((n-1)/2)*d
for i = 1:n
qu (find((q(i)-d/2<=qu )&(qu <= q(i)+d/2)))= q(i).*ones
(1,length(find((q(i)- d/2<= qu )&(qu <=q(i) +d/2))));

end
sound (qu )
subplot (2,1,2)
plot (qu )
title ('quantize signal');
%saving file
filename='quants_44100_8.wav';
audiowrite (filename,qu ,fs) ;
```

Output

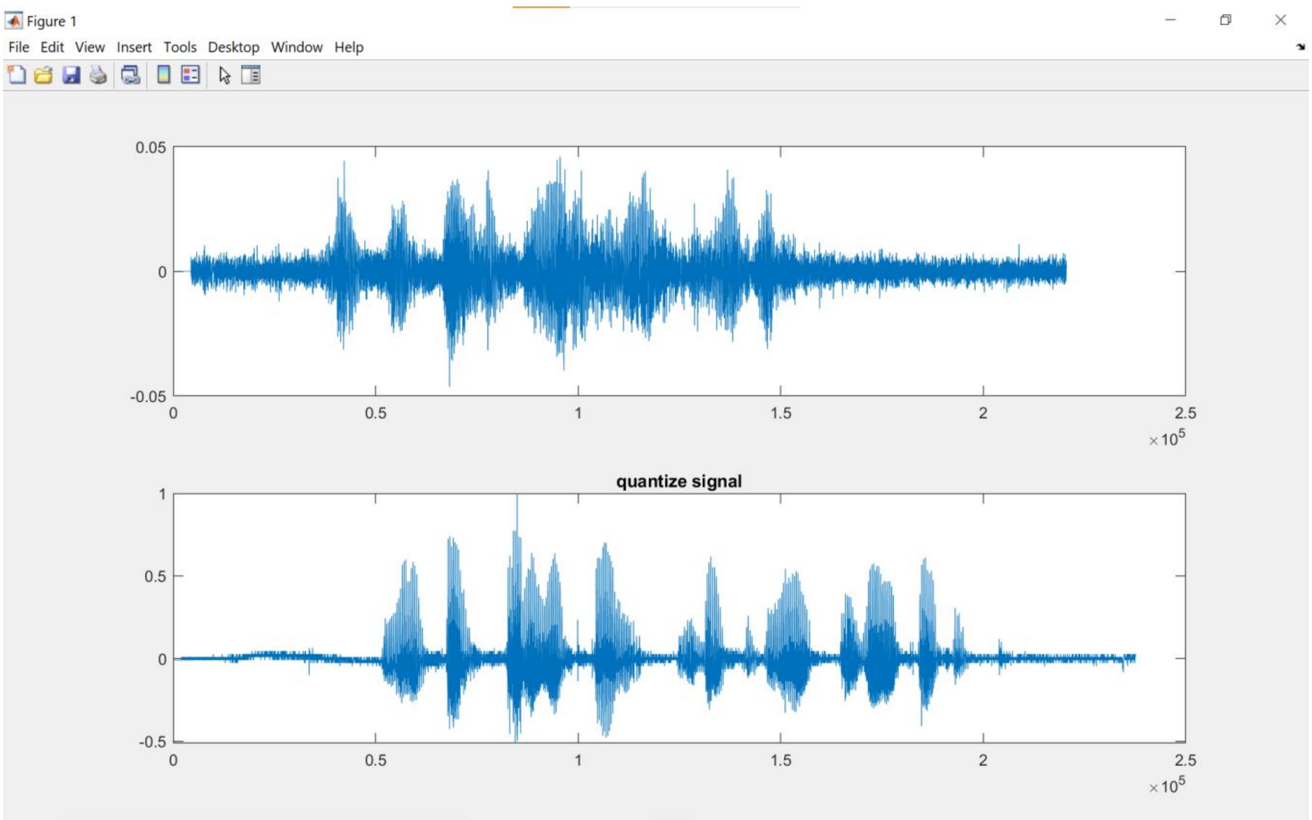


44100 hz 16 bit

```
clc;
clear all;
close all;

rectime = 5;

fs= 44100; %use  2k, 8k, 44.1k, 48k

record = audiorecorder (fs, 16, 1) ;

disp ('Start Speaking');

recordblocking (record, rectime);

disp ('End Recording');

y= getaudiodata(record) ;

sound (y, fs) ;%Play the sound

subplot(2,1,1)

plot(y)

pause (9)

filename_ori = 'OBE_PROJECT.aac';

[y,fso] = audioread(filename_ori);
```

```
fs= 44100; %change the sampling rate, use 1000, 2k, 8k, 44.1k, 48k

audiowrite('sampled.wav',y,fs)

[mySpeech,fs] = audioread('sampled.wav');%Read the data back into

 sound(mySpeech,fs);

n=16
y_max = max(y)
y_min = min(y)
qu = y/y_max(1)
d = (y_max - y_min)/n
d = d(1)
q = d.*[0:n-1]
q = q-((n-1)/2)*d
for i = 1:n
qu (find((q(i)-d/2<=qu )&(qu <= q(i)+d/2)))= q(i).*ones
(1,length(find((q(i)- d/2<= qu )&(qu <=q(i) +d/2))));

end
sound (qu )
subplot (2,1,2)
plot (qu )
title ('quantize signal');
%saving file
filename='quants_44100_16.wav';
audiowrite (filename,qu ,fs) ;
```

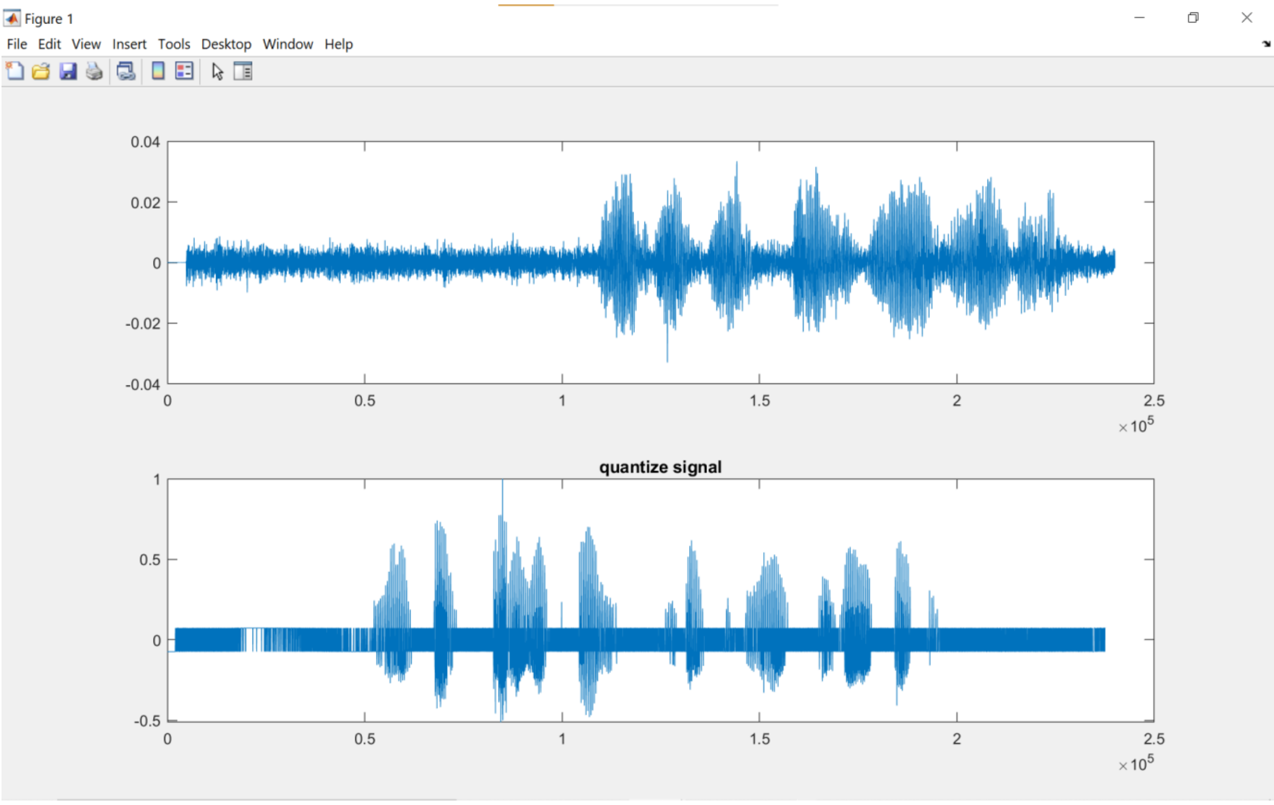
<u>Output</u>



48000 hz 2 bit

```
clc;
clear all;
close all;

rectime = 5;

fs= 48000; %use  2k, 8k, 44.1k, 48k

record = audiorecorder (fs, 16, 1) ;

disp ('Start Speaking');

recordblocking (record, rectime);

disp ('End Recording');

y= getaudiodata(record) ;

sound (y, fs) ;%Play the sound

subplot(2,1,1)

plot(y)

pause (9)

filename_ori = 'OBE_PROJECT.aac';

[y,fso] = audioread(filename_ori);

fs= 48000; %change the sampling rate, use 1000, 2k, 8k, 44.1k, 48k

audiowrite('sampled.wav',y,fs)

[mySpeech,fs] = audioread('sampled.wav');%Read the data back into

sound(mySpeech,fs);

n=2
y_max = max(y)
y_min = min(y)
qu = y/y_max(1)
d = (y_max - y_min)/n
d = d(1)
q = d.*[0:n-1]
q = q-((n-1)/2)*d
for i = 1:n
qu (find((q(i)-d/2<=qu )&(qu <= q(i)+d/2)))= q(i).*ones
(1,length(find((q(i)- d/2<= qu )&(qu <=q(i) +d/2))));

end
sound (qu )
subplot (2,1,2)
plot (qu )
title ('quantize signal');
%saving file
filename='quants_48000_2.wav';
audiowrite (filename,qu ,fs) ;
```

<u>Output</u>

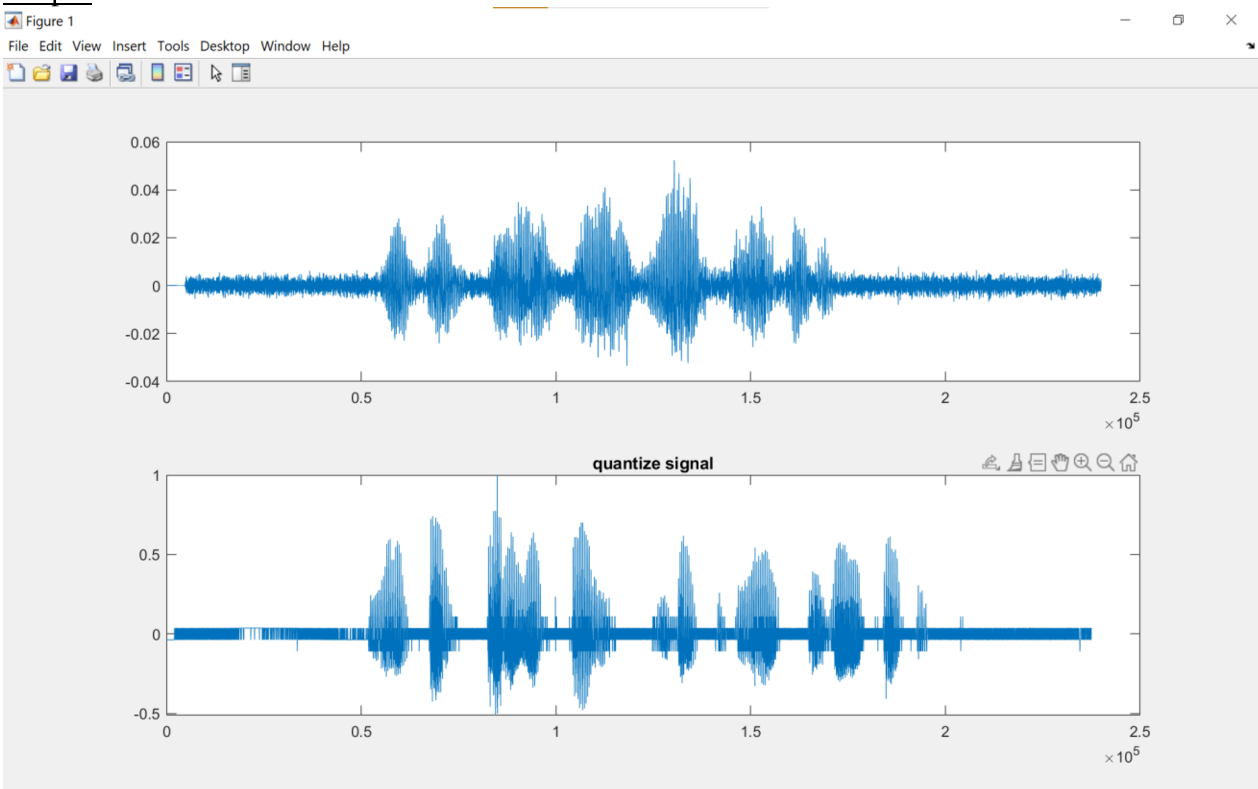


48000 hz 4 bit

```
clc;
clear all;
close all;

rectime = 5;

fs= 48000; %use  2k, 8k, 44.1k, 48k

record = audiorecorder (fs, 16, 1) ;

disp ('Start Speaking');

recordblocking (record, rectime);

disp ('End Recording');

y= getaudiodata(record) ;

sound (y, fs) ;%Play the sound

subplot(2,1,1)

plot(y)

pause (9)

filename_ori = 'OBE_PROJECT.aac';
```

```
[y,fso] = audioread(filename_ori);

fs= 48000; %change the sampling rate, use 1000, 2k, 8k, 44.1k, 48k

audiowrite('sampled.wav',y,fs)

[mySpeech,fs] = audioread('sampled.wav');%Read the data back into

sound(mySpeech,fs);

n=4
y_max = max(y)
y_min = min(y)
qu = y/y_max(1)
d = (y_max - y_min)/n
d = d(1)
q = d.*[0:n-1]
q = q-((n-1)/2)*d
for i = 1:n
qu (find((q(i)-d/2<=qu )&(qu <= q(i)+d/2)))= q(i).*ones
(1,length(find((q(i)- d/2<= qu )&(qu <=q(i) +d/2))));

end
sound (qu )
subplot (2,1,2)
plot (qu )
title ('quantize signal');
%saving file
filename='quants_48000_4.wav';
audiowrite (filename,qu ,fs) ;
```

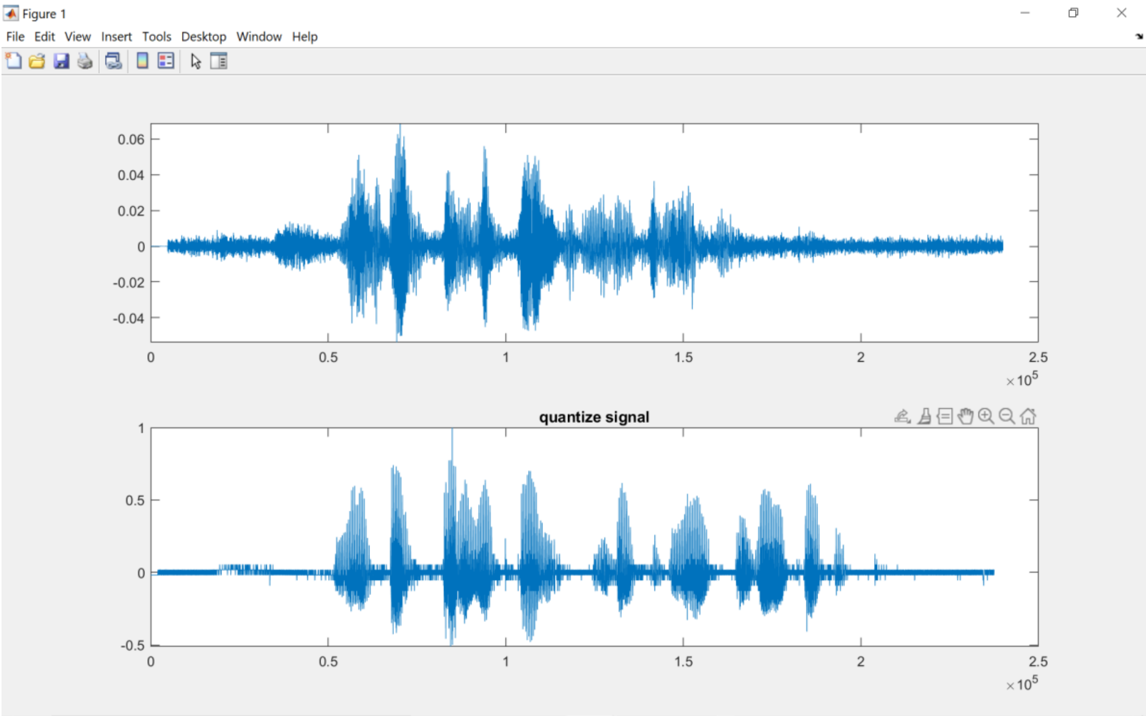
Output



48000 hz 8 bit

```
clc;
clear all;
close all;

rectime = 5;

fs= 48000; %use  2k, 8k, 44.1k, 48k

record = audiorecorder (fs, 16, 1) ;

disp ('Start Speaking');

recordblocking (record, rectime);

disp ('End Recording');

y= getaudiodata(record) ;

sound (y, fs) ;%Play the sound

subplot(2,1,1)

plot(y)

pause (9)

filename_ori = 'OBE_PROJECT.aac';

[y,fso] = audioread(filename_ori);

fs= 48000; %change the sampling rate, use 1000, 2k, 8k, 44.1k, 48k

audiowrite('sampled.wav',y,fs)

[mySpeech,fs] = audioread('sampled.wav');%Read the data back into

sound(mySpeech,fs);

n=8
y_max = max(y)
y_min = min(y)
qu = y/y_max(1)
d = (y_max - y_min)/n
d = d(1)
q = d.*[0:n-1]
q = q-((n-1)/2)*d
for i = 1:n
qu (find((q(i)-d/2<=qu )&(qu <= q(i)+d/2)))= q(i).*ones
(1,length(find((q(i)- d/2<= qu )&(qu <=q(i) +d/2))));

end
sound (qu )
subplot (2,1,2)
plot (qu )
title ('quantize signal');
%saving file
filename='quants_48000_8.wav';
audiowrite (filename,qu ,fs) ;
```

<u>Output</u>

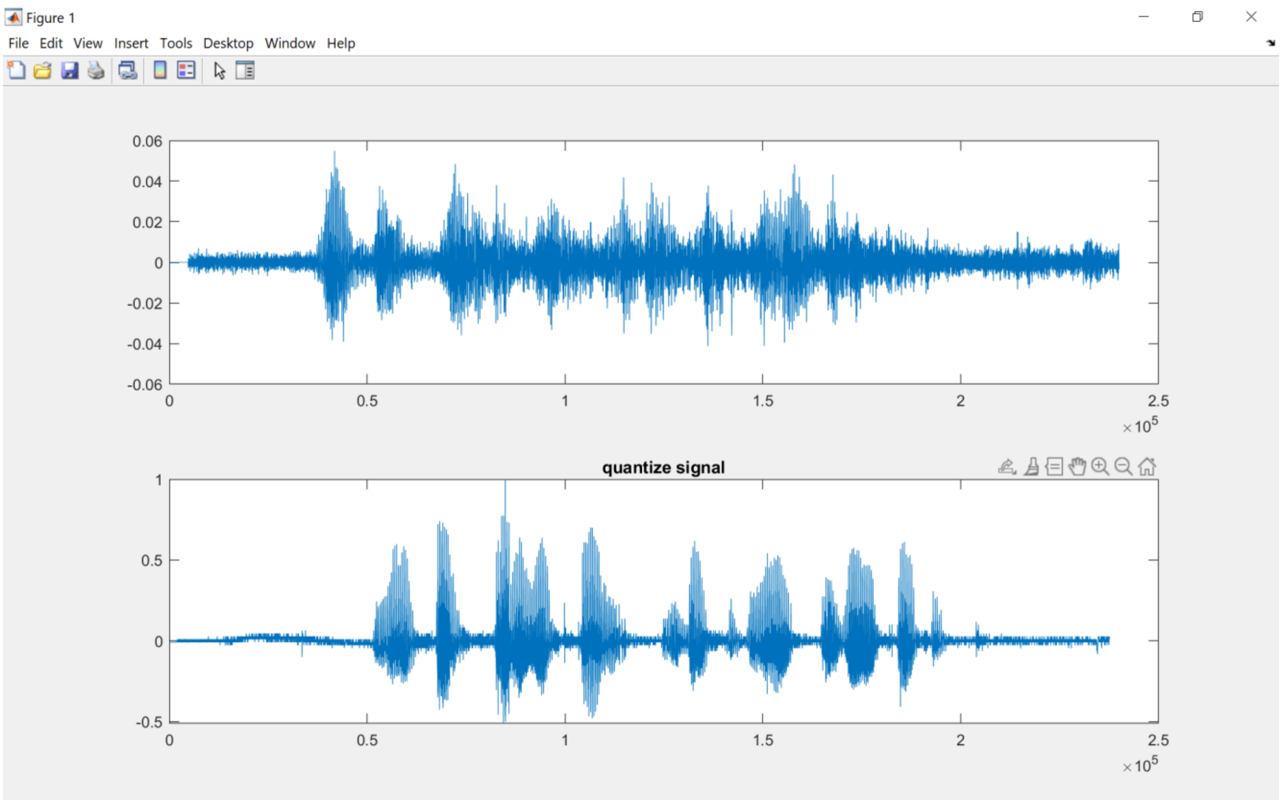


4800 Hz_16 bit

```
clc;
clear all;
close all;

rectime = 5;

fs= 48000; %use  2k, 8k, 44.1k, 48k

record = audiorecorder (fs, 16, 1) ;

disp ('Start Speaking');

recordblocking (record, rectime);

disp ('End Recording');

y= getaudiodata(record) ;

sound (y, fs) ;%Play the sound

subplot(2,1,1)

plot(y)

pause (9)

filename_ori = 'OBE_PROJECT.aac';

[y,fso] = audioread(filename_ori);

fs= 48000; %change the sampling rate, use 1000, 2k, 8k, 44.1k, 48k
```

```
audiowrite('sampled.wav',y,fs)

[mySpeech,fs] = audioread('sampled.wav');%Read the data back into

sound(mySpeech,fs);

n=16
y_max = max(y)
y_min = min(y)
qu = y/y_max(1)
d = (y_max - y_min)/n
d = d(1)
q = d.*[0:n-1]
q = q-((n-1)/2)*d
for i = 1:n
qu (find((q(i)-d/2<=qu )&(qu <= q(i)+d/2)))= q(i).*ones
(1,length(find((q(i)- d/2<= qu )&(qu <=q(i) +d/2))));

end
sound (qu )
subplot (2,1,2)
plot (qu )
title ('quantize signal');
%saving file
filename='quants_48000_16.wav';
audiowrite (filename,qu ,fs) ;
```

Output



Table

Sampling Freq / No. of bits	2000 Hz	8000 Hz	44100 Hz	48000 Hz
2	Very Bad Sound	Not Hearing (Error)	Noisy Sound	Extra Noise But Good
4	Very Bad Sound	Not Hearing (Error)	Noisy Sound	Some Noise But Good
8	Very Bad Sound	Not Hearing (Error)	Noisy Sound	Sound seems good
16	Very Bad Sound	Not Hearing (Error)	Good	Very good

Conclusion

After conducting the experiment, some important fact was founded:

1. Audio at 2000 Hz is slightly distorted in every possible bit.
2. On the other hand, audio at 8000 Hz is slightly clear than 2000 Hz.
3. 44100 Hz in every possible bit, audio frequency is clearer than others.
4. But audio frequency at 48000 Hz is very much distorted.

Thus, As experimented for difference of Frequency and different Bit rate, we get different results