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:**	C
Semester:	Summer 2021-22	**Group No:**	8

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:	**Saruwarjahan Sojib**	**Student ID:**	**18-37811-2**
Student Name:	**MD. Yasin Mahmud**	**Student ID:**	**20-42402-1**
Student Name:	**MD Nazmul Sadik**	**Student ID:**	**18-37933-2**
Student Name:	**Asif AL Abrar Niloy**	**Student ID:**	**20-42823-1**
Student Name:	**Al Tausif Mamun**	**Student ID:**	**18-38044-2**
Student Name:		**Student ID:**	

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 indepth 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 indepth 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 **welljustified 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 indepth 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

Introduction:

In this project, we recorded a small sample of our own voice before creating a PCM system that included an input audio stream, a sampler, and quantization modules. PCM (pulse-code modulation) is a way of digitally representing sampled analog signals. In a PCM stream, the amplitude of an analog signal is sampled at regular, consistent intervals, and each sample is quantized to the nearest value within a range of digital steps. [1]

Objective:

The primary goal of the chosen research is to investigate the influence of quantization on the sampled audio stream in terms of audio enhancement.

Methodology:

The following data was recorded from an audio input device, such as a microphone attached to the system:

1. First, an audio recorder object was selected.

2. The record or record blocking method was invoked with sequential parameters. Even as recording is in progress, record restores control to the calling function or command prompt.

3. Specify the duration of the recording in seconds was specified by using the stop function to stop it. The recording was done in an asynchronous manner.

4. ‘Recordblocking’ keeps control of the recording until it's finished. The recording's duration was set in seconds.

5. The recording was done in real time.

6. In the next examples, a numeric array corresponding to the signal data was formed using the "getaudiodata" method, and record techniques were demonstrated.

7. After that, the recorded sound was sampled and quantized using the sampling frequency of 3000hz, 7000hz, 35000hz, and 50000hz for a total of 2 bits.

8. The same sampling method is used in sequence, followed by quantization of the recorded sound in those sample frequencies for 4,8, and 16 bits. 9. After in sequence the same procedure of sampling method and then the quantization of the recorded sound in that sampling frequency for corresponding 4,8 & 16 bits.

10. It was discovered that shifting frequencies a have effect on voice quality.

11. It was noted that altering bits had an effect on voice quality.

12. The effects of changing frequencies and bits on voice quality were discussed together and integrated into a table. [2]

Frequency 3000, Bilt 2

```
clc;
close all;

rectime = 10;
fs= 3000;
n=2;


filename = 'Voice 001.m4a';
[y,Fs] = audioread(filename);

%sound (y, fs) ;%Play the sound
subplot(2,1,1)
plot(y)

%pause (9)

filename_ori = 'mySpeech.wav';

[y,fso] = audioread(filename_ori);


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

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

 %sound(mySpeech,fs);

y_max = max(y)
```

```
y_max = 1×2
    0.5575    0.5575
```

```
y_min = min(y)
```

```
y_min = 1×2
   -0.4309   -0.4309
```

```
qu = y/y_max(1)
```

```
qu = 457488×2
     0     0
     0     0
     0     0
     0     0
     0     0
     0     0
     0     0
     0     0
     0     0
     0     0
```

```
d = (y_max - y_min)/n
```

```
d = 1×2
    0.4942    0.4942
```

```
d = d(1)
```

```
d = 0.4942
```

```
q = d.*[0:n-1]
```

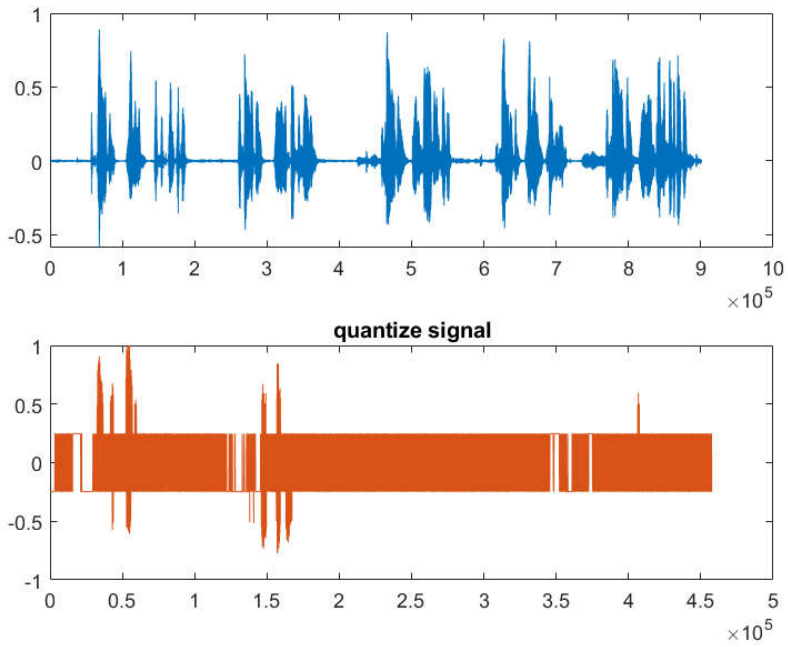
```
q = 1×2
         0    0.4942
```

```
q = q-((n-1)/2)*d
```

```
q = 1×2
   -0.2471    0.2471
```

```
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_3000_2.wav';
audiowrite (filename,qu ,fs) ;
```

Frequency 3000, Bilt 4

```
clc;
close all;

rectime = 10;
fs= 3000;
n=4;


filename = 'Voice 001.m4a';
[y,Fs] = audioread(filename);

%sound (y, fs) ;%Play the sound
subplot(2,1,1)
plot(y)

%pause (9)

filename_ori = 'mySpeech.wav';
```

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


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

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

 %sound(mySpeech,fs);

y_max = max(y)
```

```
y_max = 1×2
    0.5575    0.5575
```

```
y_min = min(y)
```

```
y_min = 1×2
   -0.4309   -0.4309
```

```
qu = y/y_max(1)
```

```
qu = 457488×2
     0     0
     0     0
     0     0
     0     0
     0     0
     0     0
     0     0
     0     0
     0     0
     0     0
```

```
d = (y_max - y_min)/n
```

```
d = 1×2
    0.2471    0.2471
```

```
d = d(1)
```

```
d = 0.2471
```

```
q = d.*[0:n-1]
```

```
q = 1×4
         0    0.2471    0.4942    0.7413
```

```
q = q-((n-1)/2)*d
```

```
q = 1×4
   -0.3706   -0.1235    0.1235    0.3706
```

```
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_3000_4.wav';
audiowrite (filename,qu ,fs) ;
```

Frequency 3000, Bilt 8

```
clc;
close all;

rectime = 10;
fs= 3000;
n=8;


filename = 'Voice 001.m4a';
[y,Fs] = audioread(filename);

%sound (y, fs) ;%Play the sound
subplot(2,1,1)
plot(y)

%pause (9)

filename_ori = 'mySpeech.wav';

[y,fso] = audioread(filename_ori);


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

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

 %sound(mySpeech,fs);

y_max = max(y)
```

```
y_max = 1×2
    0.5575    0.5575
```

```
y_min = min(y)
```

```
y_min = 1×2
   -0.4309   -0.4309
```

```
qu = y/y_max(1)
```

```
qu = 457488×2
     0     0
     0     0
     0     0
     0     0
     0     0
     0     0
     0     0
     0     0
     0     0
     0     0
```

```
d = (y_max - y_min)/n
```

```
d = 1×2
    0.1235    0.1235
```

```
d = d(1)
```

```
d = 0.1235
```

```
q = d.*[0:n-1]
```

```
q = 1×8
         0    0.1235    0.2471    0.3706    0.4942    0.6177    0.7413    0.8648
```

```
q = q-((n-1)/2)*d
```

```
q = 1×8
   -0.4324   -0.3089   -0.1853   -0.0618    0.0618    0.1853    0.3089    0.4324
```

```
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_3000_8.wav';
audiowrite (filename,qu ,fs) ;
```

Frequency 3000, Bilt 16

```
clc;
close all;

rectime = 10;
fs= 3000;
n=16;


filename = 'Voice 001.m4a';
[y,Fs] = audioread(filename);

%sound (y, fs) ;%Play the sound
subplot(2,1,1)
plot(y)

%pause (9)

filename_ori = 'mySpeech.wav';

[y,fso] = audioread(filename_ori);


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

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

 %sound(mySpeech,fs);

y_max = max(y)
```

```
y_max = 1×2
    0.5575    0.5575
```

```
y_min = min(y)
```

```
y_min = 1×2
   -0.4309   -0.4309
```

```
qu = y/y_max(1)
```

```
qu = 457488×2
```

```
     0     0
     0     0
     0     0
     0     0
     0     0
     0     0
     0     0
     0     0
     0     0
     0     0
```

```
d = (y_max - y_min)/n
```

```
d = 1⍰2
    0.0618    0.0618
```

```
d = d(1)
```

```
d = 0.0618
```

```
q = d.*[0:n-1]
```

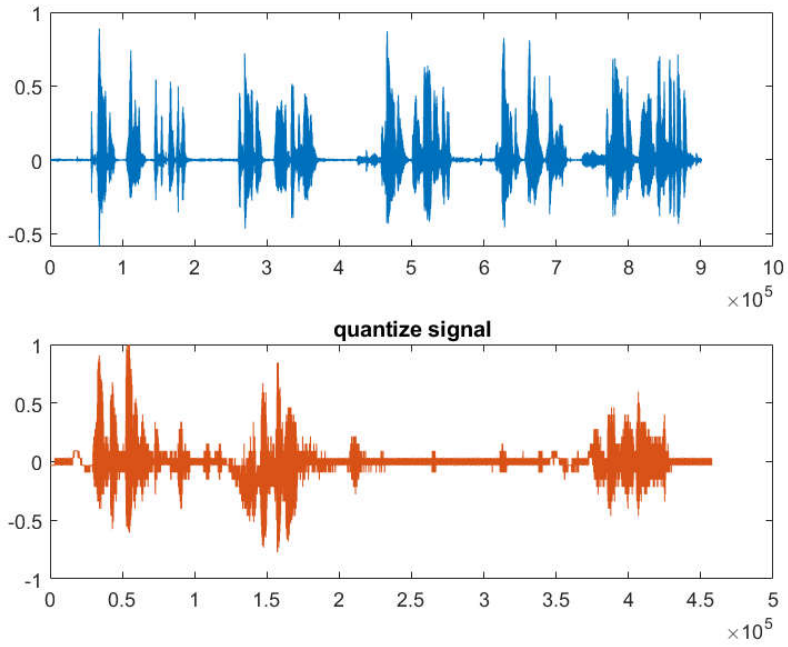
```
q = 1⍰16
         0    0.0618    0.1235    0.1853    0.2471    0.3089    0.3706    0.4324    0.4942    0.5560    0.6177    0.6795
```

```
q = q-((n-1)/2)*d
```

```
q = 1⍰16
   -0.4633   -0.4015   -0.3398   -0.2780   -0.2162   -0.1544   -0.0927   -0.0309    0.0309    0.0927    0.1544    0.2162
```

```
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_3000_16.wav';
audiowrite (filename,qu ,fs) ;
```

```
clc;
close all;

rectime = 10;
fs= 7000;
n=4;


filename = 'Voice 001.m4a';
[y,Fs] = audioread(filename);

%sound (y, fs) ;%Play the sound
subplot(2,1,1)
plot(y)

%pause (9)

filename_ori = 'mySpeech.wav';

[y,fso] = audioread(filename_ori);


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

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

 %sound(mySpeech,fs);

y_max = max(y)
```

```
y_max = 1×2
    0.5575    0.5575
```

```
y_min = min(y)
```

```
y_min = 1×2
   -0.4309   -0.4309
```

```
qu = y/y_max(1)
```

```
qu = 457488×2
     0     0
     0     0
     0     0
     0     0
     0     0
     0     0
     0     0
     0     0
     0     0
     0     0
```

```
d = (y_max - y_min)/n
```

```
d = 1×2
    0.2471    0.2471
```

```
d = d(1)
```

```
d = 0.2471
```

```
q = d.*[0:n-1]
```

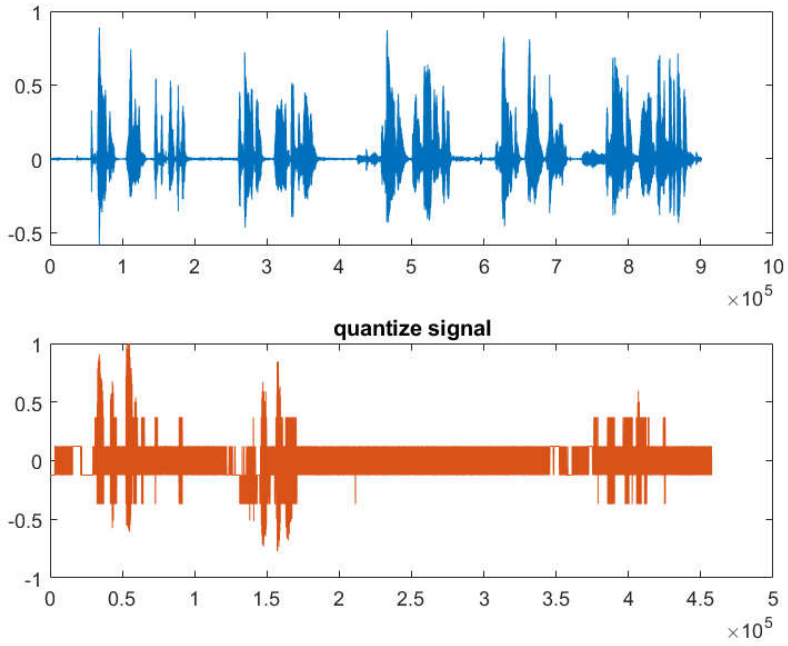
```
q = 1×4
         0    0.2471    0.4942    0.7413
```

```
q = q-((n-1)/2)*d
```

```
q = 1×4
   -0.3706   -0.1235    0.1235    0.3706
```

```
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_7000_8.wav';
audiowrite (filename,qu ,fs) ;
```

Frequency 7000, Bit 2

```
clc;
close all;

rectime = 10;
fs= 7000;
n=2;


filename = 'Voice 001.m4a';
[y,Fs] = audioread(filename);

%sound (y, fs) ;%Play the sound
subplot(2,1,1)
plot(y)

%pause (9)

filename_ori = 'mySpeech.wav';
```

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


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

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

 %sound(mySpeech,fs);

y_max = max(y)
```

```
y_max = 1×2
    0.5575    0.5575
```

```
y_min = min(y)
```

```
y_min = 1×2
   -0.4309   -0.4309
```

```
qu = y/y_max(1)
```

```
qu = 457488×2
     0     0
     0     0
     0     0
     0     0
     0     0
     0     0
     0     0
     0     0
     0     0
     0     0
```

```
d = (y_max - y_min)/n
```

```
d = 1×2
    0.4942    0.4942
```

```
d = d(1)
```

```
d = 0.4942
```

```
q = d.*[0:n-1]
```

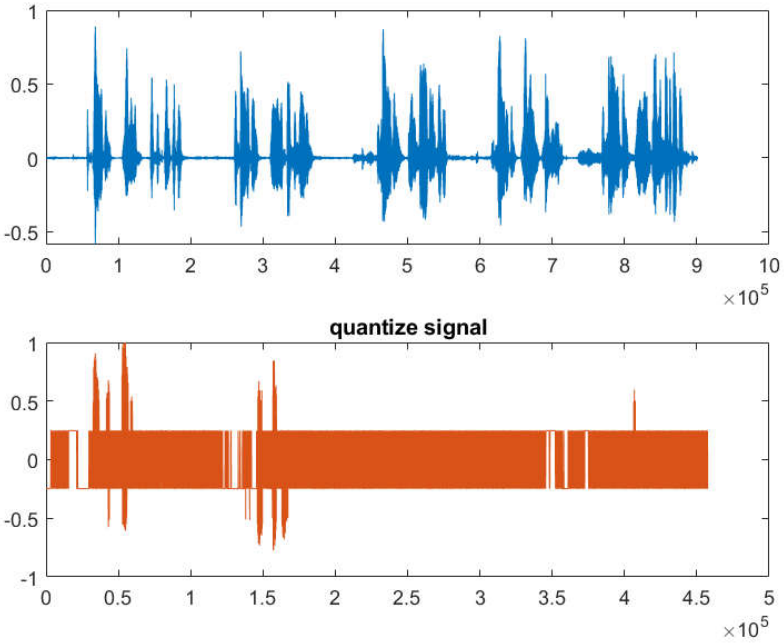
```
q = 1×2
         0    0.4942
```

```
q = q-((n-1)/2)*d
```

```
q = 1×2
   -0.2471    0.2471
```

```
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_7000_2.wav';
audiowrite (filename,qu ,fs) ;
```

```
clc;
close all;

rectime = 10;
fs= 7000;
n=8;


filename = 'Voice 001.m4a';
[y,Fs] = audioread(filename);

%sound (y, fs) ;%Play the sound
subplot(2,1,1)
plot(y)

%pause (9)

filename_ori = 'mySpeech.wav';

[y,fso] = audioread(filename_ori);


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

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

 %sound(mySpeech,fs);

y_max = max(y)
```

```
y_max = 1×2
    0.5575    0.5575
```

```
y_min = min(y)
```

```
y_min = 1×2
   -0.4309   -0.4309
```

```
qu = y/y_max(1)
```

```
qu = 457488×2
     0     0
     0     0
     0     0
     0     0
     0     0
     0     0
     0     0
     0     0
     0     0
     0     0
```

```
d = (y_max - y_min)/n
```

```
d = 1×2
    0.1235    0.1235
```

```
d = d(1)
```

```
d = 0.1235
```

```
q = d.*[0:n-1]
```

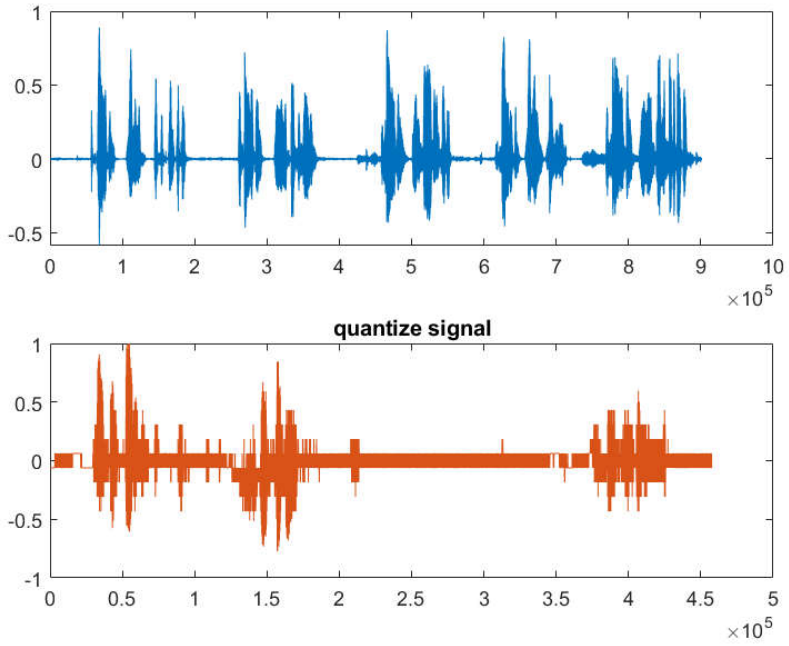
```
q = 1×8
         0    0.1235    0.2471    0.3706    0.4942    0.6177    0.7413    0.8648
```

```
q = q-((n-1)/2)*d
```

```
q = 1×8
   -0.4324   -0.3089   -0.1853   -0.0618    0.0618    0.1853    0.3089    0.4324
```

```
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_7000_8.wav';
audiowrite (filename,qu ,fs) ;
```

```
clc;
close all;

rectime = 10;
fs= 7000;
n=16;


filename = 'Voice 001.m4a';
[y,Fs] = audioread(filename);

%sound (y, fs) ;%Play the sound
subplot(2,1,1)
plot(y)

%pause (9)

filename_ori = 'mySpeech.wav';

[y,fso] = audioread(filename_ori);


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

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

 %sound(mySpeech,fs);

y_max = max(y)
```

```
y_max = 1×2
    0.5575    0.5575
```

```
y_min = min(y)
```

```
y_min = 1×2
   -0.4309   -0.4309
```

```
qu = y/y_max(1)
```

```
qu = 457488×2
     0     0
     0     0
     0     0
     0     0
     0     0
     0     0
     0     0
     0     0
     0     0
     0     0
```

```
d = (y_max - y_min)/n
```

```
d = 1×2
    0.0618    0.0618
```

```
d = d(1)
```

```
d = 0.0618
```

```
q = d.*[0:n-1]
```

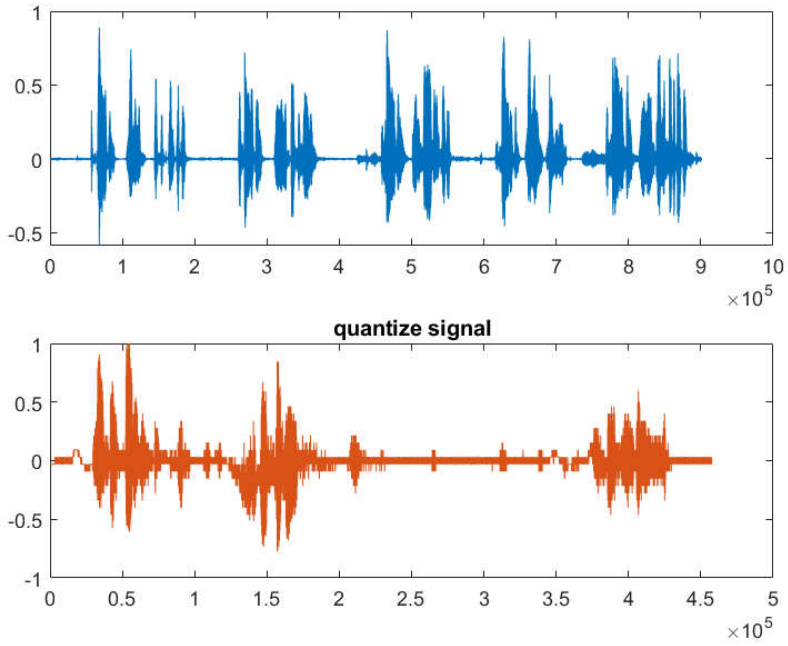
```
q = 1×16
         0    0.0618    0.1235    0.1853    0.2471    0.3089    0.3706    0.4324    0.4942    0.5560    0.6177    0.6795
```

```
q = q-((n-1)/2)*d
```

```
q = 1×16
   -0.4633   -0.4015   -0.3398   -0.2780   -0.2162   -0.1544   -0.0927   -0.0309    0.0309    0.0927    0.1544    0.2162
```

```
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_7000_16.wav';
audiowrite (filename,qu ,fs) ;
```

```
clc;
close all;

rectime = 10;
fs= 35000;
n=2;


filename = 'Voice 001.m4a';
[y,Fs] = audioread(filename);

%sound (y, fs) ;%Play the sound
subplot(2,1,1)
plot(y)

%pause (9)

filename_ori = 'mySpeech.wav';

[y,fso] = audioread(filename_ori);


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

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

 %sound(mySpeech,fs);

y_max = max(y)
```

```
y_max = 1×2
    0.5575    0.5575
```

```
y_min = min(y)
```

```
y_min = 1×2
   -0.4309   -0.4309
```

```
qu = y/y_max(1)
```

```
qu = 457488×2
     0     0
     0     0
     0     0
     0     0
     0     0
     0     0
     0     0
     0     0
     0     0
     0     0
```

```
d = (y_max - y_min)/n
```

```
d = 1×2
    0.4942    0.4942
```

```
d = d(1)
```

```
d = 0.4942
```

```
q = d.*[0:n-1]
```

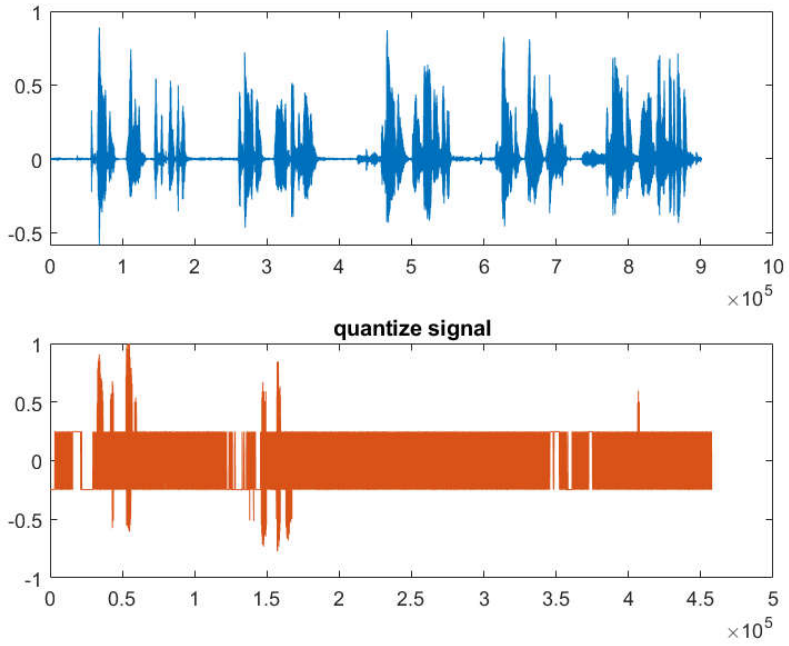
```
q = 1×2
         0    0.4942
```

```
q = q-((n-1)/2)*d
```

```
q = 1×2
   -0.2471    0.2471
```

```
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_35000_2.wav';
audiowrite (filename,qu ,fs) ;
```

```
clc;
close all;

rectime = 10;
fs= 35000;
n=4;


filename = 'Voice 001.m4a';
[y,Fs] = audioread(filename);

%sound (y, fs) ;%Play the sound
subplot(2,1,1)
plot(y)

%pause (9)

filename_ori = 'mySpeech.wav';

[y,fso] = audioread(filename_ori);


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

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

 %sound(mySpeech,fs);

y_max = max(y)
```

```
y_max = 1×2
    0.5575    0.5575
```

```
y_min = min(y)
```

```
y_min = 1×2
   -0.4309   -0.4309
```

```
qu = y/y_max(1)
```

```
qu = 457488×2
     0     0
     0     0
     0     0
     0     0
     0     0
     0     0
     0     0
     0     0
     0     0
     0     0
```

```
d = (y_max - y_min)/n
```

```
d = 1×2
    0.2471    0.2471
```

```
d = d(1)
```

```
d = 0.2471
```

```
q = d.*[0:n-1]
```

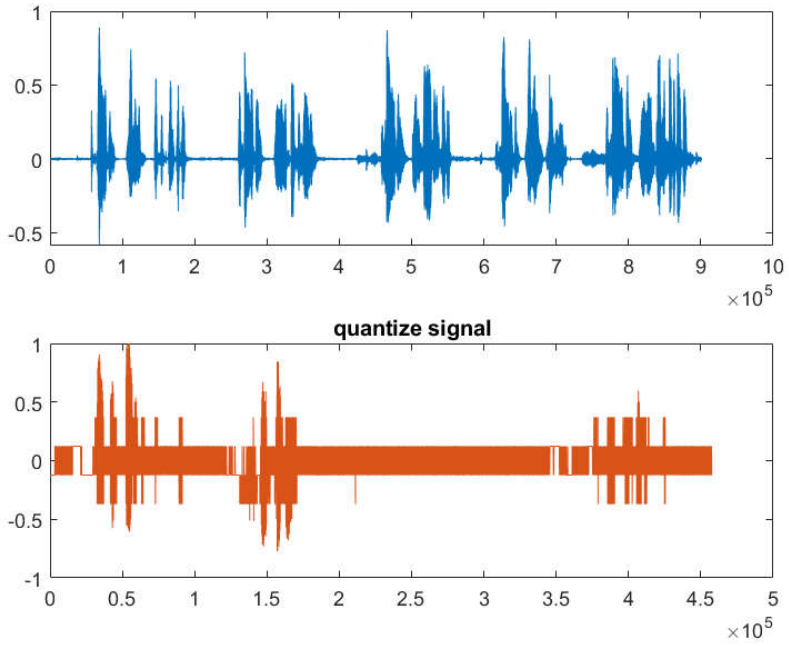
```
q = 1×4
         0    0.2471    0.4942    0.7413
```

```
q = q-((n-1)/2)*d
```

```
q = 1×4
   -0.3706   -0.1235    0.1235    0.3706
```

```
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_35000_4.wav';
audiowrite (filename,qu ,fs) ;
```

```
clc;
close all;

rectime = 10;
fs= 35000;
n=8;


filename = 'Voice 001.m4a';
[y,Fs] = audioread(filename);

%sound (y, fs) ;%Play the sound
subplot(2,1,1)
plot(y)

%pause (9)

filename_ori = 'mySpeech.wav';

[y,fso] = audioread(filename_ori);


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

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

 %sound(mySpeech,fs);

y_max = max(y)
```

```
y_max = 1×2
    0.5575    0.5575
```

```
y_min = min(y)
```

```
y_min = 1×2
   -0.4309   -0.4309
```

```
qu = y/y_max(1)
```

```
qu = 457488×2
     0     0
     0     0
     0     0
     0     0
     0     0
     0     0
     0     0
     0     0
     0     0
     0     0
```

```
d = (y_max - y_min)/n
```

```
d = 1×2
    0.1235    0.1235
```

```
d = d(1)
```

```
d = 0.1235
```

```
q = d.*[0:n-1]
```

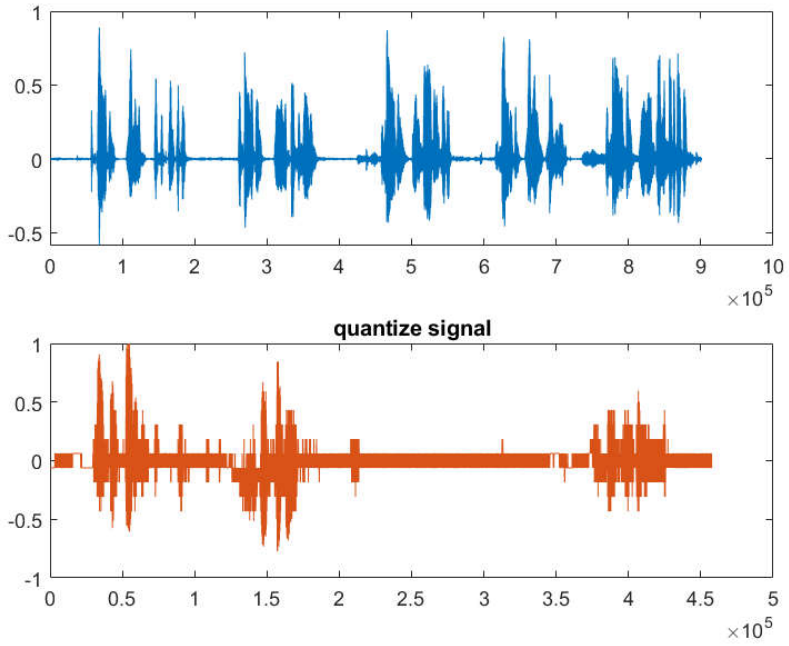
```
q = 1×8
         0    0.1235    0.2471    0.3706    0.4942    0.6177    0.7413    0.8648
```

```
q = q-((n-1)/2)*d
```

```
q = 1×8
   -0.4324   -0.3089   -0.1853   -0.0618    0.0618    0.1853    0.3089    0.4324
```

```
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_35000_8.wav';
audiowrite (filename,qu ,fs) ;
```

```
clc;
close all;

rectime = 10;
fs= 35000;
n=16;


filename = 'Voice 001.m4a';
[y,Fs] = audioread(filename);

%sound (y, fs) ;%Play the sound
subplot(2,1,1)
plot(y)

%pause (9)

filename_ori = 'mySpeech.wav';

[y,fso] = audioread(filename_ori);


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

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

 %sound(mySpeech,fs);

y_max = max(y)
```

```
y_max = 1×2
    0.5575    0.5575
```

```
y_min = min(y)
```

```
y_min = 1×2
   -0.4309   -0.4309
```

```
qu = y/y_max(1)
```

```
qu = 457488×2
     0     0
     0     0
     0     0
     0     0
     0     0
     0     0
     0     0
     0     0
     0     0
     0     0
```

```
d = (y_max - y_min)/n
```

```
d = 1×2
    0.0618    0.0618
```

```
d = d(1)
```

```
d = 0.0618
```

```
q = d.*[0:n-1]
```

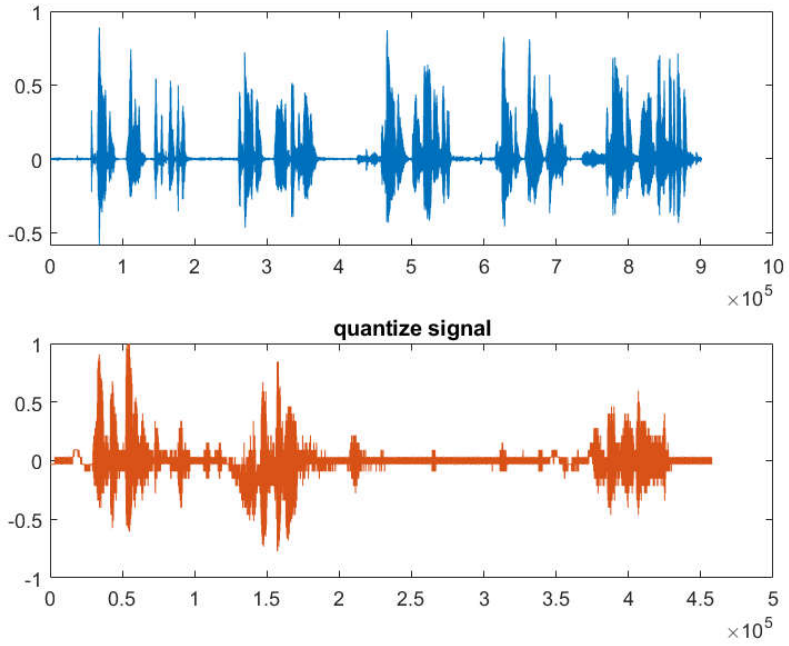
```
q = 1×16
         0    0.0618    0.1235    0.1853    0.2471    0.3089    0.3706    0.4324    0.4942    0.5560    0.6177    0.6795
```

```
q = q-((n-1)/2)*d
```

```
q = 1×16
   -0.4633   -0.4015   -0.3398   -0.2780   -0.2162   -0.1544   -0.0927   -0.0309    0.0309    0.0927    0.1544    0.2162
```

```
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_35000_16.wav';
audiowrite (filename,qu ,fs) ;
```

```
clc;
close all;

rectime = 10;
fs= 50000;
n=2;


filename = 'Voice 001.m4a';
[y,Fs] = audioread(filename);

%sound (y, fs) ;%Play the sound
subplot(2,1,1)
plot(y)

%pause (9)

filename_ori = 'mySpeech.wav';

[y,fso] = audioread(filename_ori);


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

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

 %sound(mySpeech,fs);

y_max = max(y)
```

```
y_max = 1×2
    0.5575    0.5575
```

```
y_min = min(y)
```

```
y_min = 1×2
   -0.4309   -0.4309
```

```
qu = y/y_max(1)
```

```
qu = 457488×2
     0     0
     0     0
     0     0
     0     0
     0     0
     0     0
     0     0
     0     0
     0     0
     0     0
```

```
d = (y_max - y_min)/n
```

```
d = 1×2
    0.4942    0.4942
```

```
d = d(1)
```

```
d = 0.4942
```

```
q = d.*[0:n-1]
```

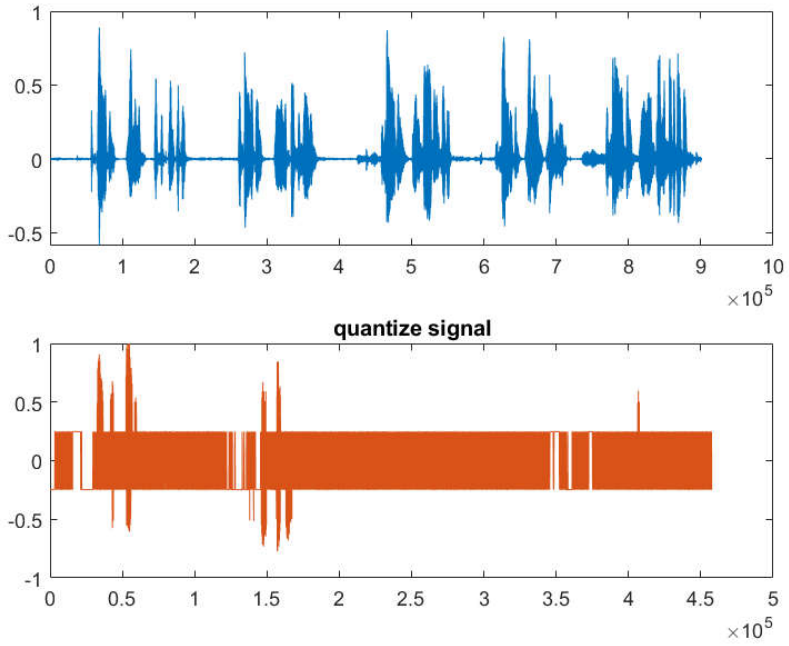
```
q = 1×2
         0    0.4942
```

```
q = q-((n-1)/2)*d
```

```
q = 1×2
   -0.2471    0.2471
```

```
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_50000_2.wav';
audiowrite (filename,qu ,fs) ;
```

```
clc;
close all;

rectime = 10;
fs= 50000;
n=4;


filename = 'Voice 001.m4a';
[y,Fs] = audioread(filename);

%sound (y, fs) ;%Play the sound
subplot(2,1,1)
plot(y)

%pause (9)

filename_ori = 'mySpeech.wav';

[y,fso] = audioread(filename_ori);


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

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

 %sound(mySpeech,fs);

y_max = max(y)
```

```
y_max = 1×2
    0.5575    0.5575
```

```
y_min = min(y)
```

```
y_min = 1×2
   -0.4309   -0.4309
```

```
qu = y/y_max(1)
```

```
qu = 457488×2
     0     0
     0     0
     0     0
     0     0
     0     0
     0     0
     0     0
     0     0
     0     0
     0     0
```

```
d = (y_max - y_min)/n
```

```
d = 1×2
    0.2471    0.2471
```

```
d = d(1)
```

```
d = 0.2471
```

```
q = d.*[0:n-1]
```

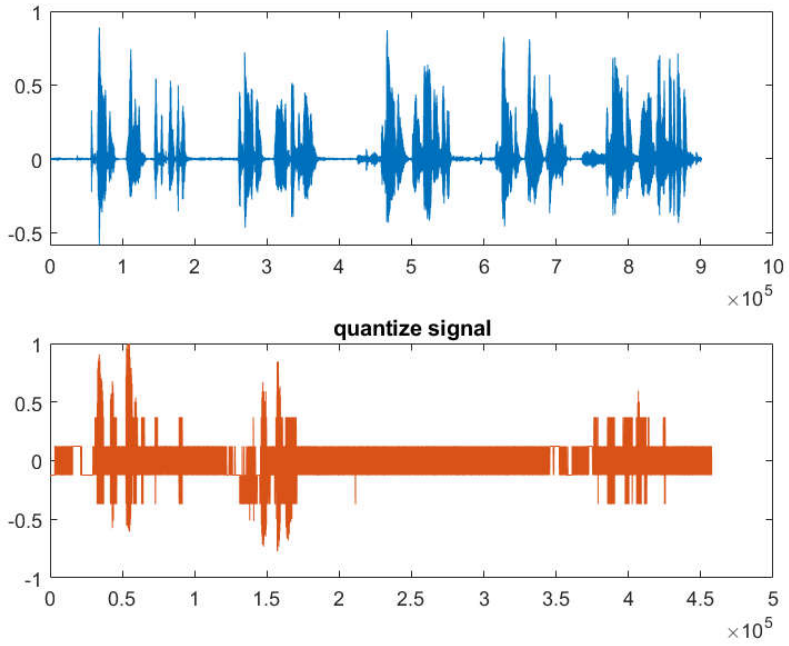
```
q = 1×4
         0    0.2471    0.4942    0.7413
```

```
q = q-((n-1)/2)*d
```

```
q = 1×4
   -0.3706   -0.1235    0.1235    0.3706
```

```
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_50000_4.wav';
audiowrite (filename,qu ,fs) ;
```

```
clc;
close all;

rectime = 10;
fs= 50000;
n=8;


filename = 'Voice 001.m4a';
[y,Fs] = audioread(filename);

sound (y, fs) ;%Play the sound
subplot(2,1,1)
plot(y)

pause (9)

filename_ori = 'mySpeech.wav';

[y,fso] = audioread(filename_ori);


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

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

 sound(mySpeech,fs);

y_max = max(y)
```

```
y_max = 1×2
    0.5575    0.5575
```

```
y_min = min(y)
```

```
y_min = 1×2
   -0.4309   -0.4309
```

```
qu = y/y_max(1)
```

```
qu = 457488×2
     0     0
     0     0
     0     0
     0     0
     0     0
     0     0
     0     0
     0     0
     0     0
     0     0
```

```
d = (y_max - y_min)/n
```

```
d = 1×2
    0.1235    0.1235
```

```
d = d(1)
```

```
d = 0.1235
```

```
q = d.*[0:n-1]
```

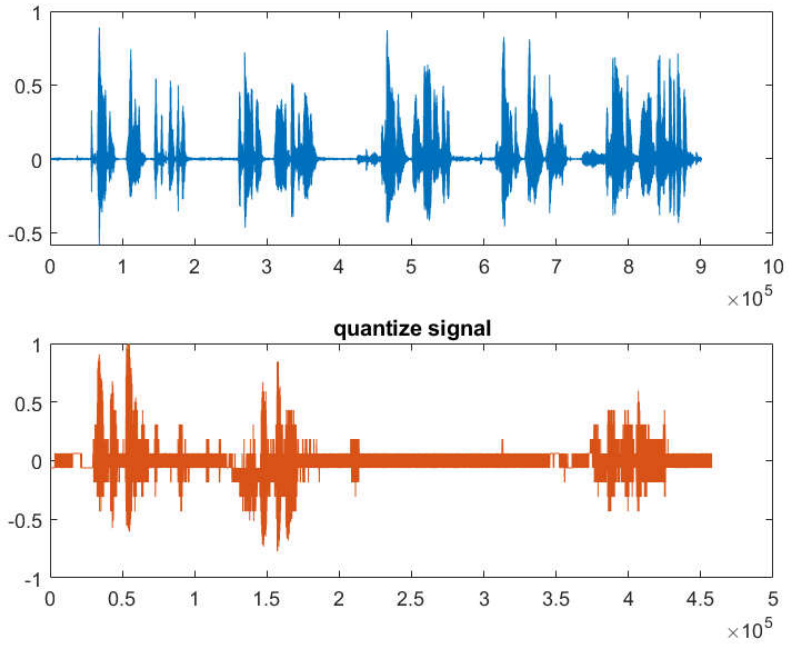
```
q = 1×8
         0    0.1235    0.2471    0.3706    0.4942    0.6177    0.7413    0.8648
```

```
q = q-((n-1)/2)*d
```

```
q = 1×8
   -0.4324   -0.3089   -0.1853   -0.0618    0.0618    0.1853    0.3089    0.4324
```

```
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_50000_8.wav';
audiowrite (filename,qu ,fs) ;
```

```
clc;
clear all;
close all;

rectime = 10;

fs= 50000;

filename = 'Voice 001.m4a';
[y,Fs] = audioread(filename);

sound (y, fs) ;%Play the sound

subplot(2,1,1)

plot(y)

pause (9)

filename_ori = 'mySpeech.wav';

[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
```

```
n = 16
```

```
y_max = max(y)
```

```
y_max = 1×2
    0.5575    0.5575
```

```
y_min = min(y)
```

```
y_min = 1×2
   -0.4309   -0.4309
```

```
qu = y/y_max(1)
```

```
qu = 457488×2
     0     0
     0     0
     0     0
     0     0
     0     0
     0     0
     0     0
     0     0
     0     0
     0     0
```

```
d = (y_max - y_min)/n
```

```
d = 1×2
    0.0618    0.0618
```

```
d = d(1)
```

```
d = 0.0618
```

```
q = d.*[0:n-1]
```

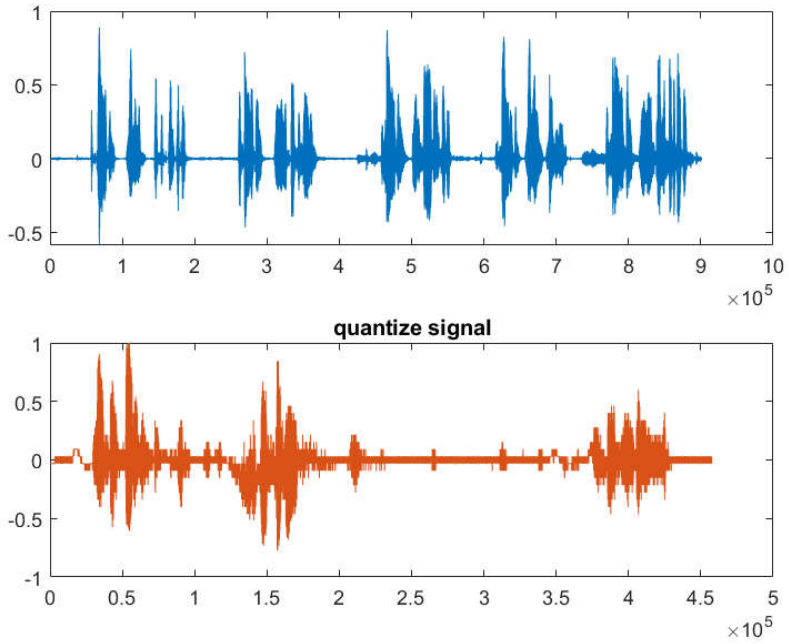
```
q = 1×16
         0    0.0618    0.1235    0.1853    0.2471    0.3089    0.3706    0.4324    0.4942    0.5560    0.6177    0.6795
```

```
q = q-((n-1)/2)*d
```

```
q = 1×16
   -0.4633   -0.4015   -0.3398   -0.2780   -0.2162   -0.1544   -0.0927   -0.0309    0.0309    0.0927    0.1544    0.2162
```

```
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_50000_16.wav';
audiowrite (filename,qu ,fs) ;
```

Sampling Freq / **No. of bits**	3000 Hz	7000 Hz	35000 Hz	50000 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

Analysis of the Results:

As can be observed, the low amplitude of the original sound was not retained in the linear quantization graph. In this output, only the high amplitude signal was properly kept. The A-law quantizer and the mu-law quantizer the quantizer, on the other hand, efficiently kept both low and high amplitude signals. In a nutshell, a non-linear A quantizer can produce lower-distortion audio than a linear quantizer.

Conclusion:

After conducting the experiment, some important fact was founded:

1. Audio at 3000 Hz is slightly distorted in every possible bit.

2. On the other hand, audio at 7000 Hz is slightly clear than 3000 Hz.

3. 35000 Hz in every possible bit, audio frequency is clearer than others.

4. But audio frequency at 50000 Hz is very much distorted.

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

References:

[1] https://en.wikipedia.org/wiki/Pulse-code_modulation.

[2] Aiub Lab Manual