Experiment Number:4

Roll Number	Class	Date of Performance	Date of Submission	Signature
42428	BE 8	18-03-2021		

Title: Write a program to simulate speech coding and decoding technique

used in Mobile Communication.

Aim: Write a program to simulate speech coding and decoding technique

Using PCM Modulation Technique.

Apparatus: PC with MATLAB Software,Algorithm,etc

Pre-requisites:

1. BPSK Modulation/Demodulation.
2. AWGN Model.
3. BER.

Theory:

Speech Coding and Decoding:

Speech coding is an application of data compression of digital audio signals containing speech. Speech coding uses speech-specific parameter estimation using audio signal processing techniques to model the speech signal, combined with generic data compression algorithms to represent the resulting modeled parameters in a compact bitstream.
The decoder circuit decodes the pulse coded waveform to reproduce the original signal. This circuit acts as the demodulator.

Pulse Code Modulation(PCM):

To get a pulse code modulated waveform from an analog waveform at the transmitter end (source) of a communications circuit, the amplitude of the analog signal samples at regular time intervals. The sampling rate or a number of samples per second is several times the maximum frequency. The message signal converted into the binary form will be usually in the number of levels which is always to a power of 2. This process is called quantization.

At the receiver end, a pulse code demodulator decodes the binary signal back into pulses with the same quantum levels as those in the modulator. By further processes, we can restore the original analog waveform.

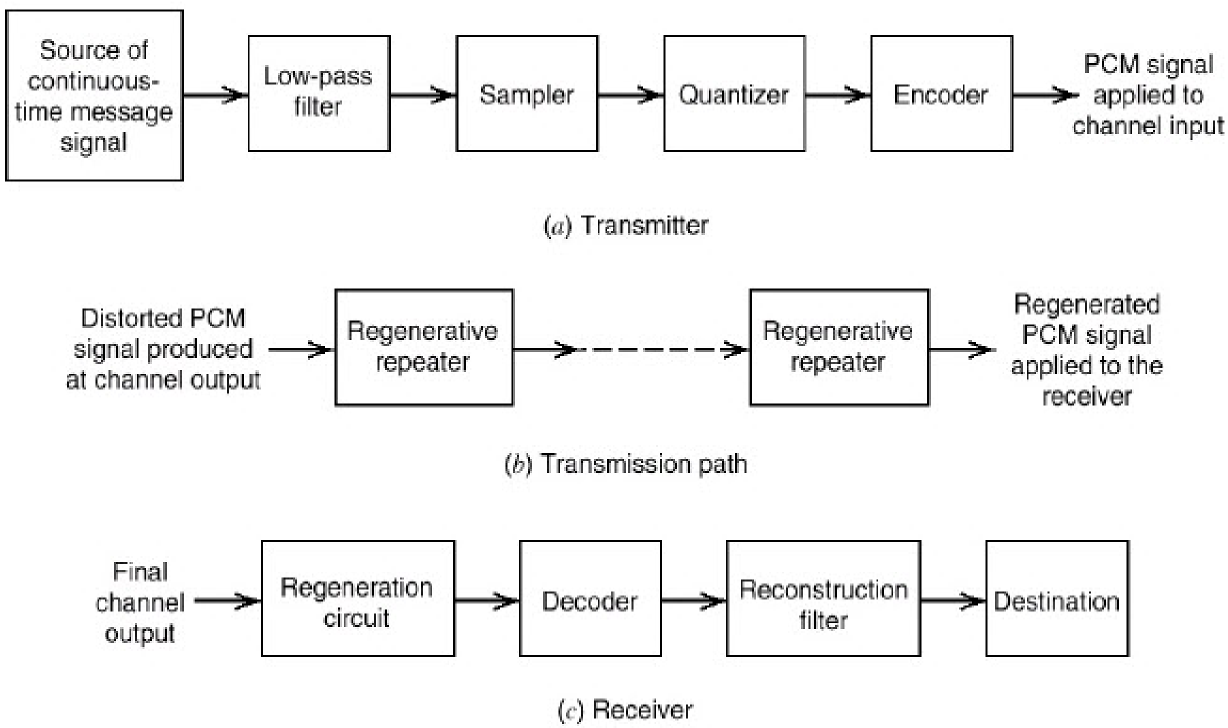


Fig.Basic Elements of PCM System

This above block diagram describes the whole process of PCM. The source of the continuous-time message signal is passed through a low pass filter and then sampling, Quantization, Encoding will be done.

We will see in detail step by step:

Low Pass Filter: This filter eliminates the high frequency components present in the input analog signal which is greater than the highest frequency of the message signal, to avoid aliasing of the message signal.

Sampling:

Sampling is a process of measuring the amplitude of a continuous-time signal at discrete instants, converts the continuous signal into a discrete signal. For example, conversion of a sound wave to a sequence of samples. The Sample is a value or set of values at a point in time or it can be spaced. Sampler extract samples of a continuous signal, it is a subsystem ideal sampler produces samples that are equivalent to the instantaneous value of the continuous signal at the specified various points. The Sampling process generates flat- top Pulse Amplitude Modulated (PAM) signal.

Sampling frequency, Fs is the number of average samples per second also known as the Sampling rate. According to the Nyquist Theorem sampling rate should be at least 2 times the upper cutoff frequency. Sampling frequency, Fs>=2*fmax to avoid Aliasing Effect. If the sampling frequency is very higher than the Nyquist rate it becomes Oversampling, theoretically a bandwidth-limited signal can be reconstructed if sampled at above the Nyquist rate. If the sampling frequency is less than the Nyquist rate it will become Undersampling.

Basically two types of techniques are used for the sampling process. Those are 1. Natural Sampling and 2. Flat- top Sampling.

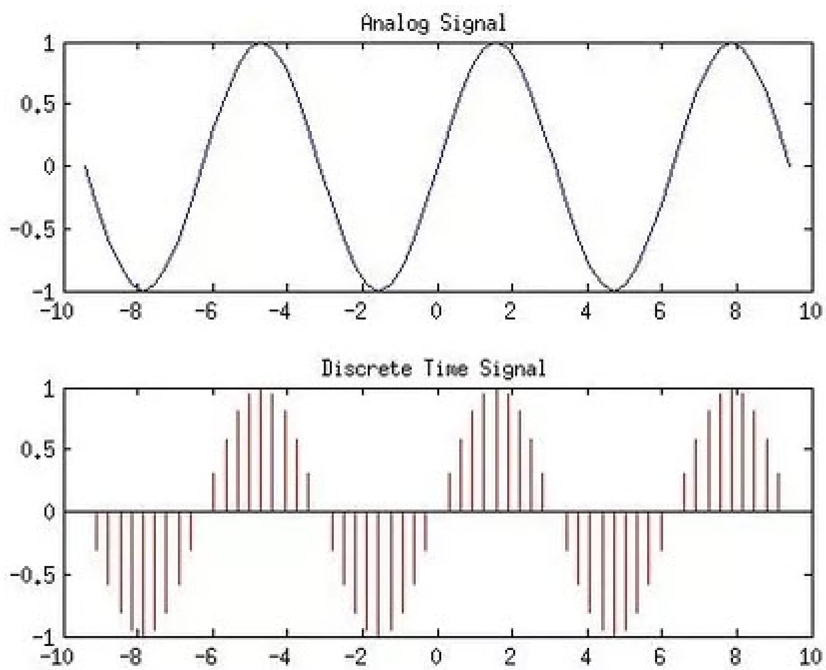


Fig:Analog and Sampled Signal

Quantization:

In quantization, an analog sample with an amplitude that converted into a digital sample with an amplitude that takes one of a specifically defined set of quantization values. Quantization is done by dividing the range of possible values of the analog samples into some different levels and assigning the center value of each level to any sample in the quantization interval. Quantization approximates the analog sample values with the nearest quantization values. So almost all the quantized samples will differ from the original samples by a small amount. That amount is called quantization error. The result of this quantization error is we will hear a hissing noise when playing a random signal. Converting analog samples into binary numbers that are 0 and 1.

In most cases, we will use uniform quantizers. Uniform quantization is applicable when the sample values are in a finite range (Fmin, Fmax). The total data range is divided into 2n levels, let it be L intervals. They will have an equal length Q. Q is known as Quantization interval or quantization step size. In uniform quantization, there will be no quantization error.

As we know,
L=2n, then Step size Q = (Fmax – Fmin) / L

Interval i is mapped to the middle value. We will store or send only the index value of quantized value.

An Index value of quantized value Qi (F) = [F – Fmin / Q]

Quantized value Q (F) = Qi (F) Q + Q / 2 + Fmin

But there are some problems raised in uniform quantization those are

Only optimal for the uniformly distributed signal.

- Real audio signals are more concentrated near zeros.
- The Human ear is more sensitive to quantization errors at small values.

The solution to this problem is using Non- uniform quantization. In this process, the quantization interval is smaller near zero.

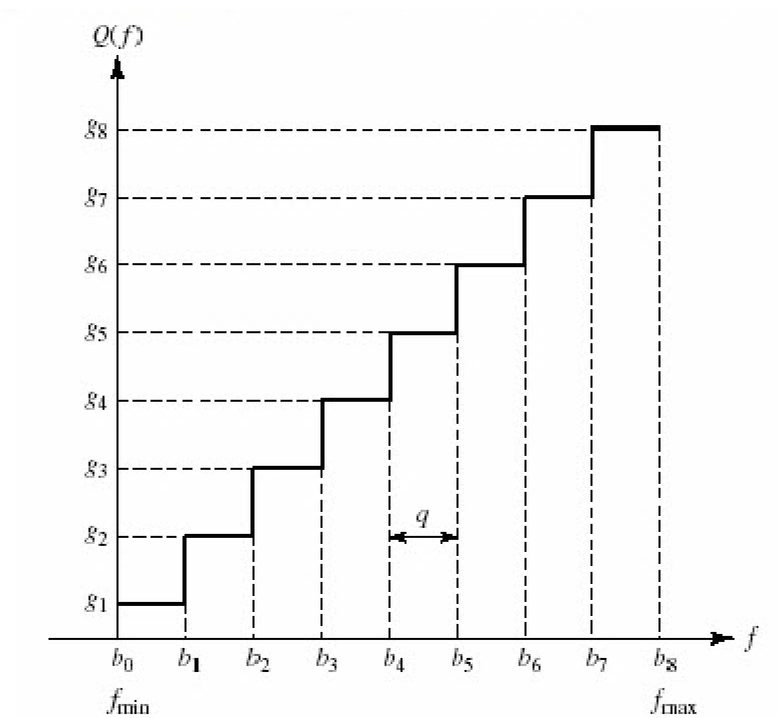


Fig:Uniformly Quantized Signal

Coding:

The encoder encodes the quantized samples. Each quantized sample is encoded into an 8-bit codeword by using A-law in the encoding process.

- Bit 1 is the most significant bit (MSB), it represents the polarity of the sample. “1” represents positive polarity and “0” represents negative polarity.
- Bit 2,3 and 4 will defines the location of the sample value. These three bits together form a linear curve for low level negative or positive samples.
- Bit 5,6,7 and 8 are the least significant bits (LSB) it represents one of the segments quantized value. Each segment is divided into 16 quantum levels.

PCM is two types of Differential Pulse Code Modulation (DPCM) and Adaptive Differential Pulse Code Modulation (ADPCM).

In DPCM only the difference between a sample and the previous value is encoded. The difference will be much smaller than the total sample value so we need some bits for getting the same accuracy as in ordinary PCM. So that the required bit rate will also reduce. For example, in 5-bit code 1 bit is for polarity and the remaining 4 bits for 16 quantum levels.

ADPCM is achieved by adapting the quantizing levels to analog signal characteristics. We can estimate the values with the preceding sample values. Error estimation is done as same as in DPCM. In 32Kbps ADPCM method difference between the predicted value and sample, value is coded with 4 bits, so that we’ll get 15 quantum levels. In this method data rate is half of the conventional PCM.

Regenerative Repeater: This section increases the signal strength. The output of the channel also has one regenerative repeater circuit, to compensate the signal loss and reconstruct the signal, and also to increase its strength.

Pulse Code Demodulation:

Pulse Code Demodulation will be doing the same modulation process in reverse. Demodulation starts with the decoding process, during transmission the PCM signal will be affected by noise interference. So, before the PCM signal sends into the PCM demodulator, we have to recover the signal into the original level for that we are using a comparator. The PCM signal is a series pulse wave signal, but for demodulation, we need a wave to be parallel.

By using a serial to parallel converter the series pulse wave signal will be converted into a parallel digital signal. After that the signal will pass through the n-bits decoder, it should be a Digital to Analog converter. Decoder recovers the original quantization values of the digital signal. This quantization value also includes a lot of high-frequency harmonics with original audio signals. For avoiding unnecessary signals we utilize a low-pass filter at the final part.

Reconstruction Filter:

After the digital-to-analog conversion is done by the regenerative circuit and the decoder, a low-pass filter is employed, called as the reconstruction filter to get back the original signal. Hence, the Pulse Code Modulator circuit digitizes the given analog signal, codes it and samples it, and then transmits it in an analog form. This whole process is repeated in a reverse pattern to obtain the original signal.

Algorithm:

Speech Coding and Decoding Using PCM:

- Define the analog sine wave signal with required frequency (2 KHz).
- Define the sampling frequency (20 KHz) greater than Nyquist frequency (4 KHz).
- Sample the original signal with the sampling frequency.
- Quantize the samples to the given number of quantization levels (8 levels).
- Encode the samples according to the respective quantization levels.
- Display the encoded signal.
- Decode the signal and obtained the voltage values for the samples.
- Use sinc interpolation to approximate the voltage values from the obtained quantized values.
- Display the signals

Conclusion:

In this experiment we understood how we can use PCM encoder and Decoder to convert an analog speech signal into a digital encoded signal and convert obtained digital signal into an approximated analog signal which is similar to the original signal with some errors. We demonstrated the same using a matlab code and verified the results using graphs.

Subject Teacher:

Date:

Code :

```
clc;
close all;
clear all;

cycles=3; %Number of cycles
original_frequency=2000; %Frequency of the signal = 2 kHz
t=[0:1/1000000:cycles/original_frequency]; %Here 1000000 is
the 'Matlab' Sampling frequency
original_signal=1*sin(2*pi*original_frequency*t);%Signal pk-
pk is 2 units

% Encoding and Sampling
sam_freq=20000; %sampling Frequency = 20 kHz
sam_time=1/sam_freq; %Sampling Time period
n=[0:1/sam_freq:cycles/original_frequency];
num_samples=length(n);
sam_signal=sin(2*pi*original_frequency*n);

%N is num of bits used for quantizing
N=3;
num_levels=2^N; %using N bits we get 2^N levels
width=2/(num_levels-1);
levels=[-1:width:1];
boundaries=[-1+(width/2):width:1-(width/2)];
codes=[0:num_levels-1];
quant=zeros(1,num_samples);
signal_after_coding=zeros(1,num_samples);

% Quantization and coding
for i=1:num_samples
    index=1;
    if(sam_signal(i)>=boundaries(end))
        signal_after_coding(i)=codes(end);
        quant(i)=levels(end);
    else
        for boundary=boundaries
            if(sam_signal(i)<=boundary)
                signal_after_coding(i)=codes(index);
                quant(i)=levels(index);
                break;
            end
            index=index+1;
        end
    end
end
```

```
%Signal after coding
binary_coded_signal=dec2bin(signal_after_coding)


% PCM Decoder

signal_after_coding=bin2dec(binary_coded_signal)
xa=zeros(1,num_samples);
ind=1;
for i = 1:num_samples
    x(i)=levels(signal_after_coding(i)+1);
end
x=x';
t=sam_time*(1:num_samples);
ts = [0:1/1000000:cycles/original_frequency];
[Ts,T] = ndgrid(ts,t);

% Here, Sinc interpolation is used
y = sinc((Ts - T)/sam_time)*x;

%plotting
subplot(2,2,1),
plot(ts,original_signal,'LineStyle','--');  %original Signal
hold on
stem(n,sam_signal);  %Sampled Signal
title('Original and Sampled Signals')
xlabel('Time/n'), ylabel('Signal');
legend('Original Signal','Sampled Signal');

subplot(2,2,2),
plot(ts,original_signal,'LineStyle','--');  %original Signal
hold on
stem(n,quant);  %Quantized signal
title('Original and Quantized Signals')
xlabel('Time/n'), ylabel('Signal');
legend('Original Signal','Quantized Signal');

subplot(2,2,3),
plot(ts, y)
title('Reconstructed Signal')
xlabel('Time'), ylabel('Reconstructed Signal');
```

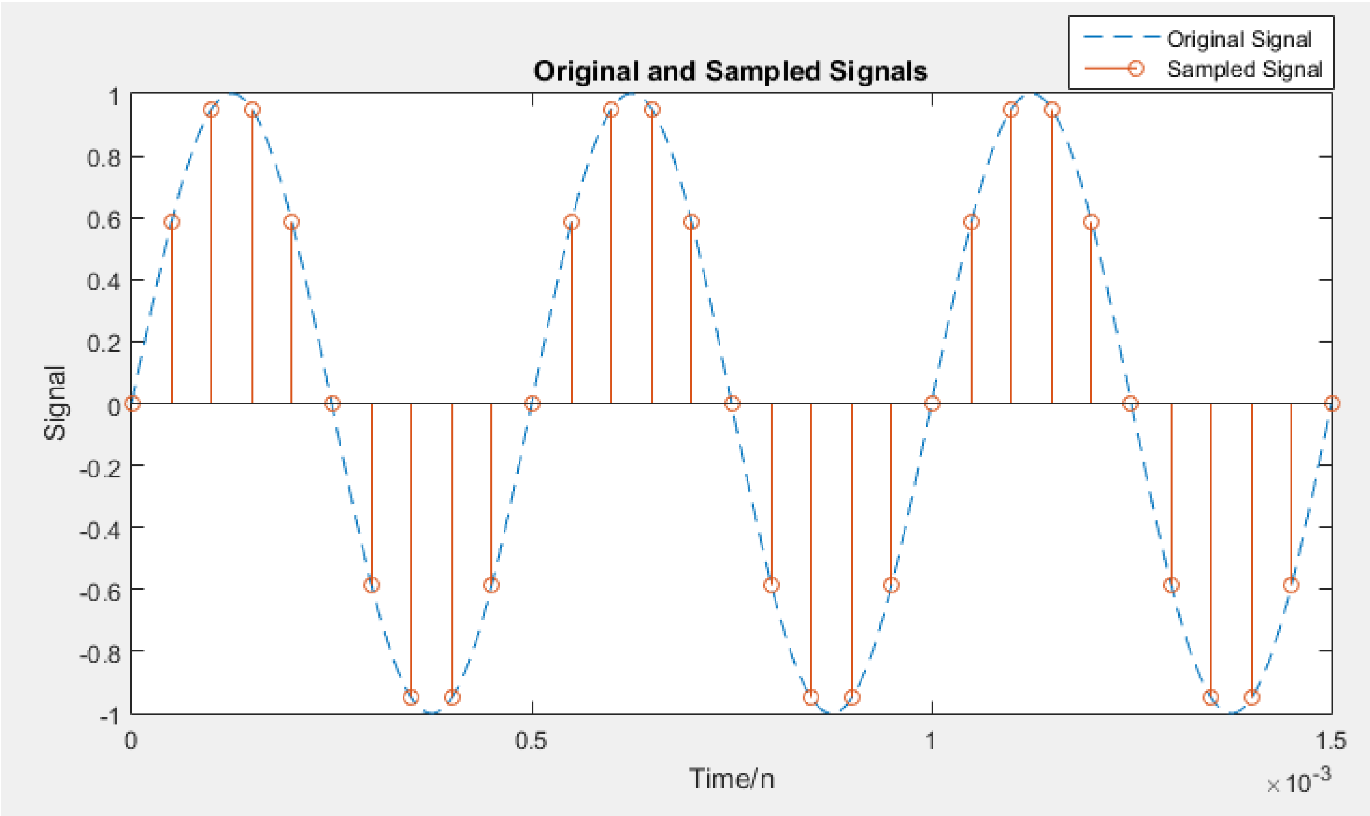
Output :

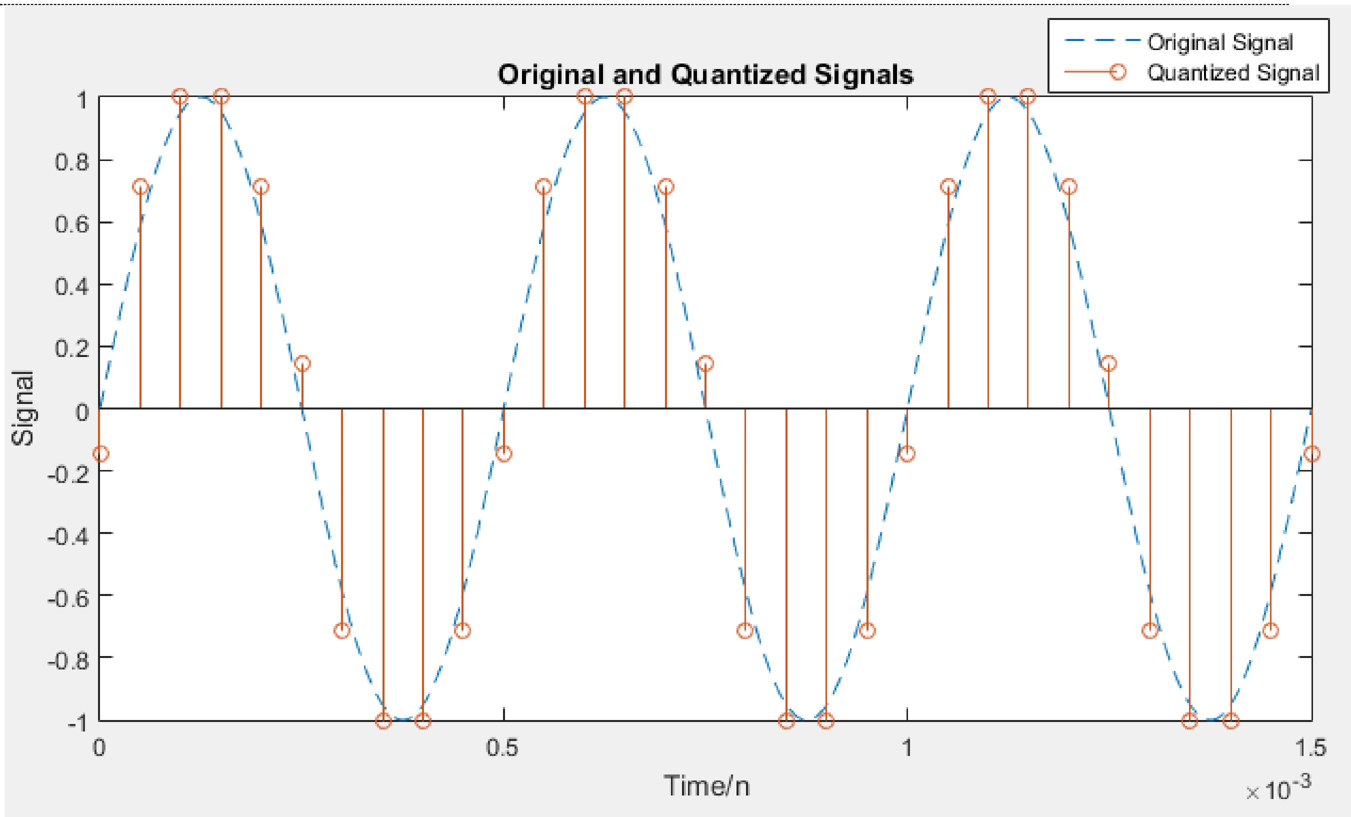
signal_after_coding =

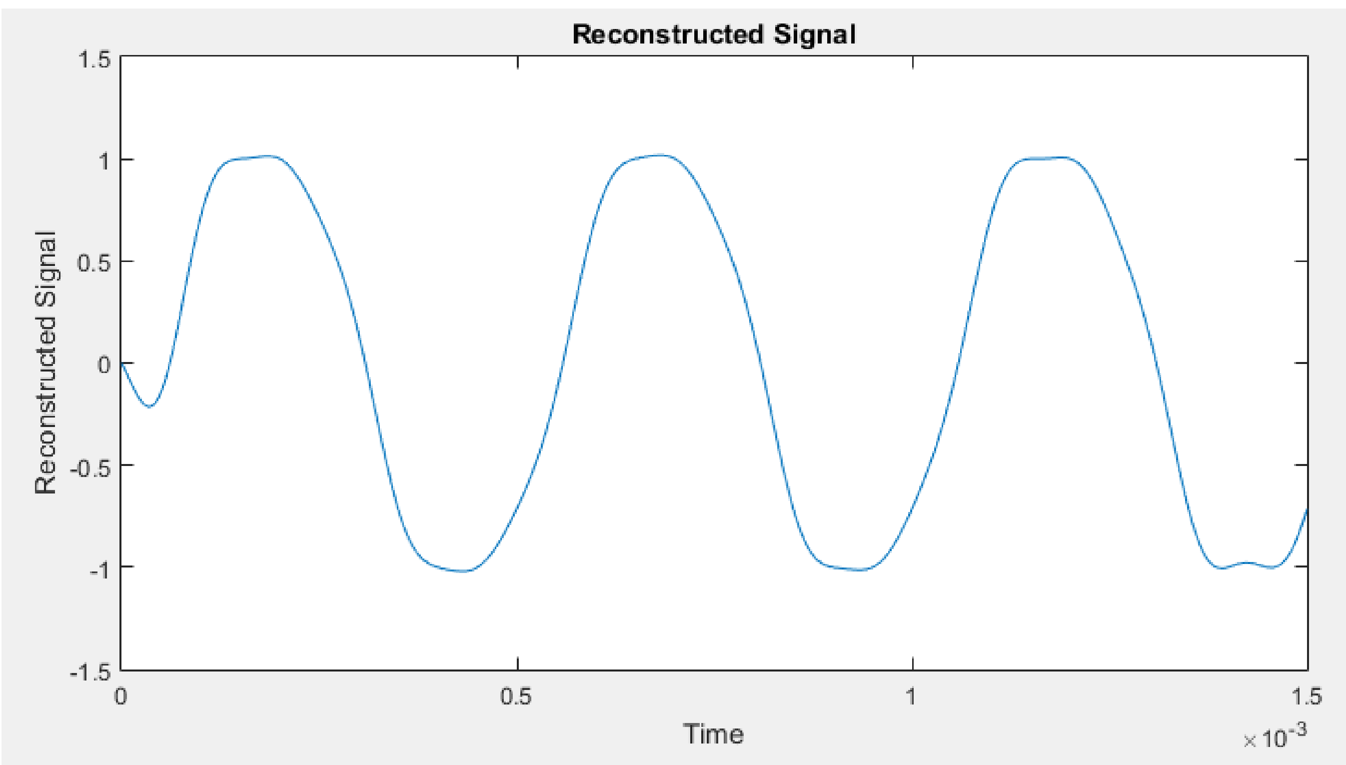
3	6	7	7	6	4	1	0	0	1	3	6
7	7	6	4	1	0	0	1	3	6	7	7
6	4	1	0	0	1	3					

binary_coded_signal =

011	110	111	111	110	100	001	000	000	001	011	110
111	111	110	100	001	000	000	001	011	110	111	111
110	100	001	000	000	001	011					




Original and Quantized Signals
Original Signal
Quantized Signal
Signal
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
Time/n
$\times 10^{-3}$


Reconstructed Signal
Reconstructed Signal
1.5
1
0.5
0
-0.5
-1
-1.5
0
0.5
1
1.5
Time
$\times 10^{-3}$

Answer the following Questions

1. The signals which are obtained by encoding each quantized signal into a digital word is called as
a) PAM signal
b) PCM signal
c) FM signal
d) Sampling and quantization

2. The length of the code-word obtained by encoding quantized sample is equal to
a) l=log(to the base 2)L
b) l=log(to the base 10)L
c) l=2log(to the base 2)L
d) l=log(to the base 2)L/2

3. Quantization noise can be reduced by _______ the number of levels.
a) Decreasing
b) Increasing
c) Doubling
d) Squaring

4. In PCM encoding, quantization level varies as a function of _______
a) Frequency
b) Amplitude
c) Square of frequency
d) Square of amplitude

5. What is bit depth?
a) Number of quantization level
b) Interval between two quantization levels
c) Number of possible digital values to represent each sample
d) None of the mentioned

6. Choosing a discrete value that is near but not exactly at the analog signal level leads to
a) PCM error
b) Quantization error
c) PAM error
d) Sampling error

7. In PCM the samples are dependent on _______
a) Time
b) Frequency
c) Quanization leavel
d) Interval between quantization level

8. DPCM encodes the PCM values based on
a) Quantization level
b) Difference between the current and predicted value
c) Interval between levels
d) None of the mentioned

9. Delta modulation uses _____ bits per sample.
a) One
b) Two
c) Four
d) Eight

10. Sample resolution for LPCM ____ bits per sample.
a) 8
b) 16
c) 24
d) All of the mentioned

11. Adaptive DPCM is used to
a) Increase bandwidth
b) Decrease bandwidth
c) Increase SNR
d) None of the mentioned