

Experiment on Pulse Code Modulation (PCM) using MATLAB software.

Objective: The objectives of this Experiment are:

1. Generating of a pulse Code Modulation (PCM).
2. To investigate the magnitude spectrum of the Pulse Code Modulation.

Theoretical Concept:

Modulation is the process of varying one or more parameters of a carrier signal in accordance with the instantaneous values of the message signal.

The message signal is the signal which is being transmitted for communication and the carrier signal is a high frequency signal which has no data but is used for long distance transmission.

There are many modulation techniques, which are classified according to the type of modulation employed. Of them all, the digital modulation technique used is **Pulse Code Modulation PCM**.

A signal is pulse code modulated to convert its analog information into a binary sequence, i.e., **1s** and **0s**. The output of a PCM will resemble a binary sequence. The following figure shows an example of PCM output with respect to instantaneous values of a given sine wave.

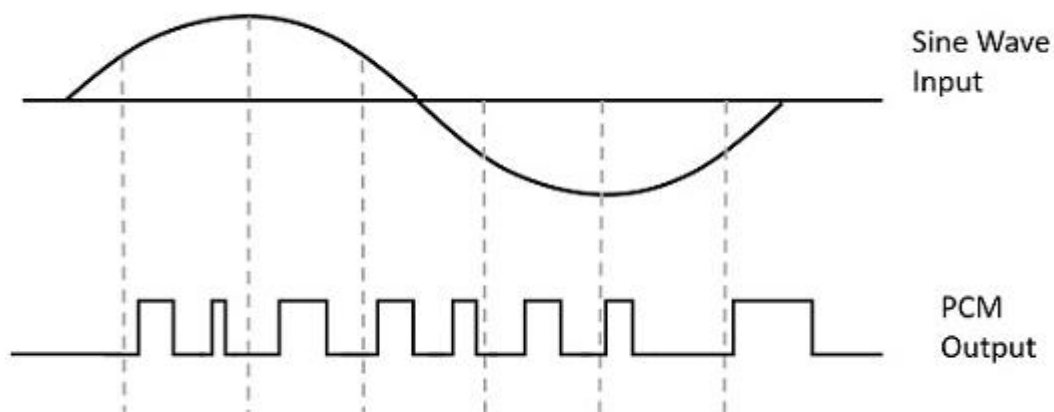


Figure 1: Pulse Code Modulation

Instead of a pulse train, PCM produces a series of numbers or digits, and hence this process is called as digital. Each one of these digits, though in binary code, represent the approximate amplitude of the signal sample at that instant.

In Pulse Code Modulation, the message signal is represented by a sequence of coded pulses. This message signal is achieved by representing the signal in discrete form in both time and amplitude.

Basic Elements of PCM

The transmitter section of a Pulse Code Modulator circuit consists of Sampling, Quantizing and Encoding, which are performed in the analog-to-digital converter section. The low pass filter prior to sampling prevents aliasing of the message signal.

The basic operations in the receiver section are regeneration of impaired signals, decoding, and reconstruction of the quantized pulse train. Following is the block diagram of PCM which represents the basic elements of both the transmitter and the receiver sections.

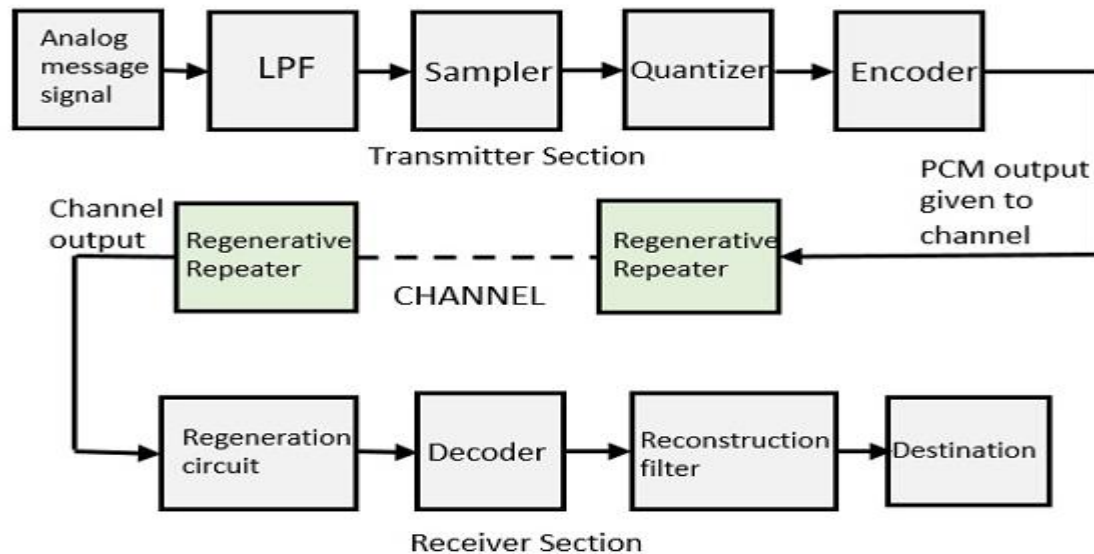


Figure 2: Block Diagram of PCM System.

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.

Sampler

This is the technique which helps to collect the sample data at instantaneous values of message signal, so as to reconstruct the original signal. The sampling rate must be greater than twice the highest frequency component W of the message signal, in accordance with the sampling theorem.

Sampling is defined as, "The process of measuring the instantaneous values of continuous-time signal in a discrete form."

Sample is a piece of data taken from the whole data which is continuous in the time domain.

When a source generates an analog signal and if that has to be digitized, having 1s and 0s i.e., High or Low, the signal has to be discretized in time. This discretization of analog signal is called as Sampling.

Quantizer

Quantizing is a process of reducing the excessive bits and confining the data. The sampled output when given to Quantizer, reduces the redundant bits and compresses the value.

The digitization of analog signals involves the rounding off of the values which are approximately equal to the analog values. The method of sampling chooses a few points on the analog signal and then these points are joined to round off the value to a near stabilized value. Such a process is called as Quantization.

The analog-to-digital converters perform this type of function to create a series of digital values out of the given analog signal. The following figure represents an analog signal. This signal to get converted into digital, has to undergo sampling and quantizing.

The quantizing of an analog signal is done by discretizing the signal with a number of quantization levels. Quantization is representing the sampled values of the amplitude by a finite set of levels, which means converting a continuous-amplitude sample into a discrete-time signal.

Encoder

The digitization of analog signal is done by the encoder. It designates each quantized level by a binary code. The sampling done here is the sample-and-hold process. These three sections LPF, Sampler, and Quantizer will act as an analog to digital converter. Encoding minimizes the bandwidth used.

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.

Decoder

The decoder circuit decodes the pulse coded waveform to reproduce the original signal. This circuit acts as the demodulator.

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.

Matlab Code

```
clc;
close all;
clear all;
n=input('Enter for n-bit PCM system : '); %Encodebook
Bit Length
n1=input('Enter Sampling Frequency : '); %Sampling
Frequency
L = 2^n; %Number of Quantisation Levels
%% Here we pplot the Analog Signal and its Sampled form
Vmax = 8;
x = 0:pi/n1:4*pi; %Construction of Signal
ActualSignal=Vmax*sin(x); %Actual input
subplot(3,1,1);
plot(ActualSignal);
title('Analog Signal');
subplot(3,1,2); %Sampled Version
stem(ActualSignal);grid on; title('Sampled Sinal');
%% Now perform the Quantization Process
Vmin=-Vmax; %Since the Signal is sine
StepSize=(Vmax-Vmin)/L; %
Diference between each quantisation level
QuantizationLevels=Vmin:StepSize:Vmax; %
Quantisation Levels - For comparison
codebook=Vmin-(StepSize/2):StepSize:Vmax+(StepSize/2); %
Quantisation Values - As Final Output of qunatiz
```

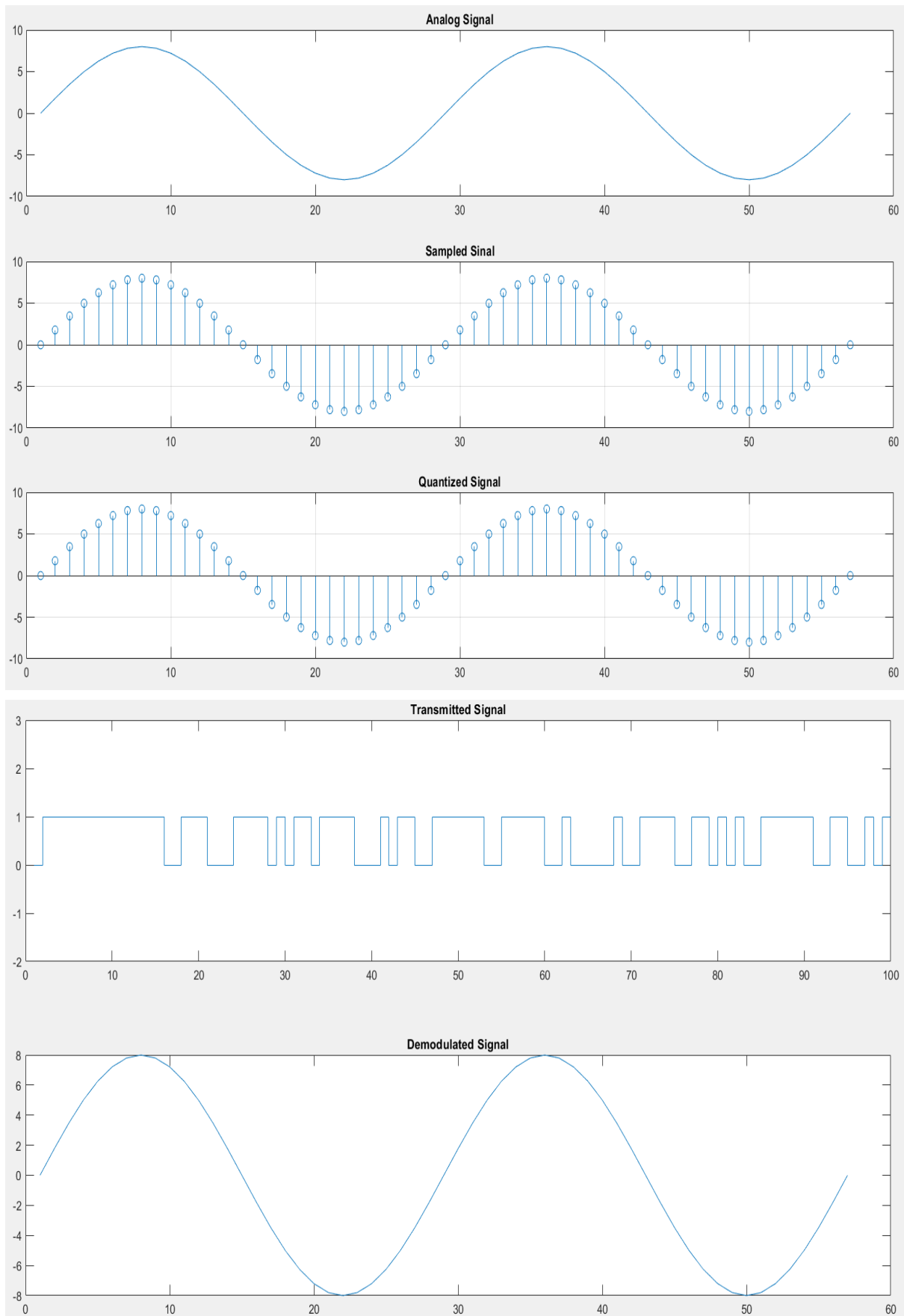
```

[ind,q]=quantiz(ActualSignal,QuantizationLevels,codebook);
% Quantization process
NonZeroInd = find(ind ~= 0);
ind(NonZeroInd) = ind(NonZeroInd) - 1;
    % MATLAB gives indexing from 1 to N. But we need
    indexing from 0, to convert it into binary codebook

BelowVminInd = find(q == Vmin-(StepSize/2));
q(BelowVminInd) = Vmin+(StepSize/2);
    %This is for correction, as signal values cannot go
    beyond Vmin
    %But quantiz may suggest it, since it return the Values
    lower than Actual
    %Signal Value
subplot(3,1,3);
stem(q);
grid on;                                % Display the Quantize values
title('Quantized Signal');
%% Having Quantised the values, we perform the Encoding
Process
figure
TransmittedSig = de2bi(ind,'left-msb');    % Encode the
Quantisation Level
SerialCode = reshape(TransmittedSig',[1
size(TransmittedSig,1)*size(TransmittedSig,2)]);
subplot(2,1,1);
grid on;
stairs(SerialCode);                    % Display the SerialCode
Bit Stream
axis([0 100 -2 3]);
title('Transmitted Signal');
%% Now we perform the Demodulation Of PCM signal
RecievedCode=reshape(SerialCode,n,length(SerialCode)/n);
%Again Convert the SerialCode into Frames of 1 Byte
index = bi2de(RecievedCode,'left-msb');
%Binary to Decimal Conversion
q = (StepSize*index);    %Convert into Voltage Values
q = q + (Vmin+(StepSize/2)); % Above step gives a DC
shfted version of Actual siganl
%Thus it is necessary to bring it to zero level
subplot(2,1,2);
grid on;
plot(q);    % Plot Demodulated signal
title('Demodulated Signal');

```

Output



Operation of PCM using MATLAB Simulink

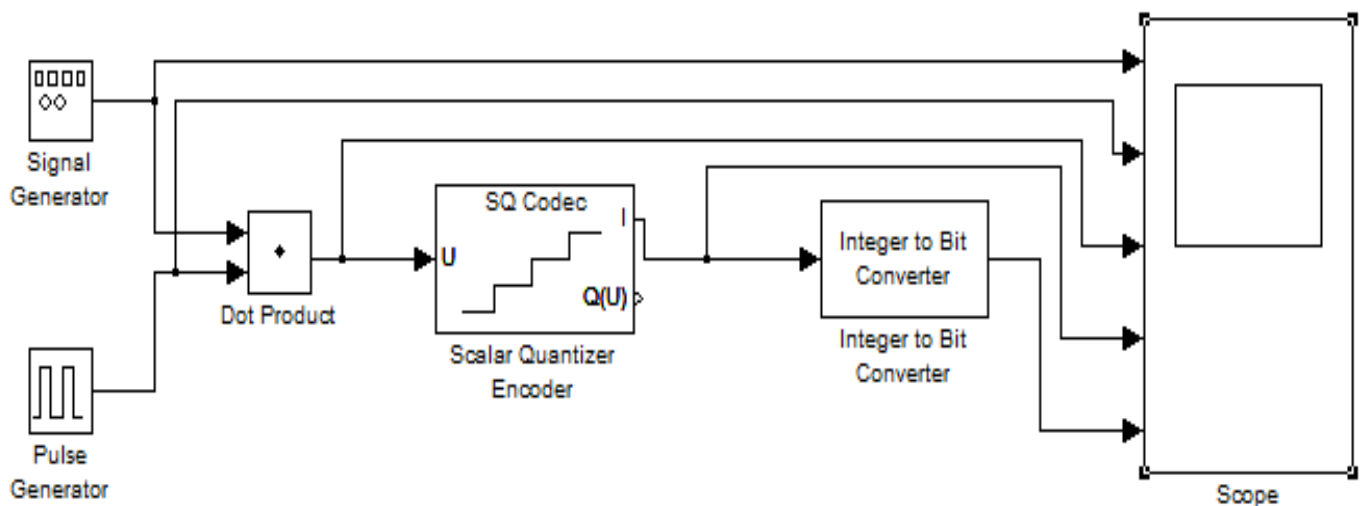


Figure 3: PCM block diagram

Output

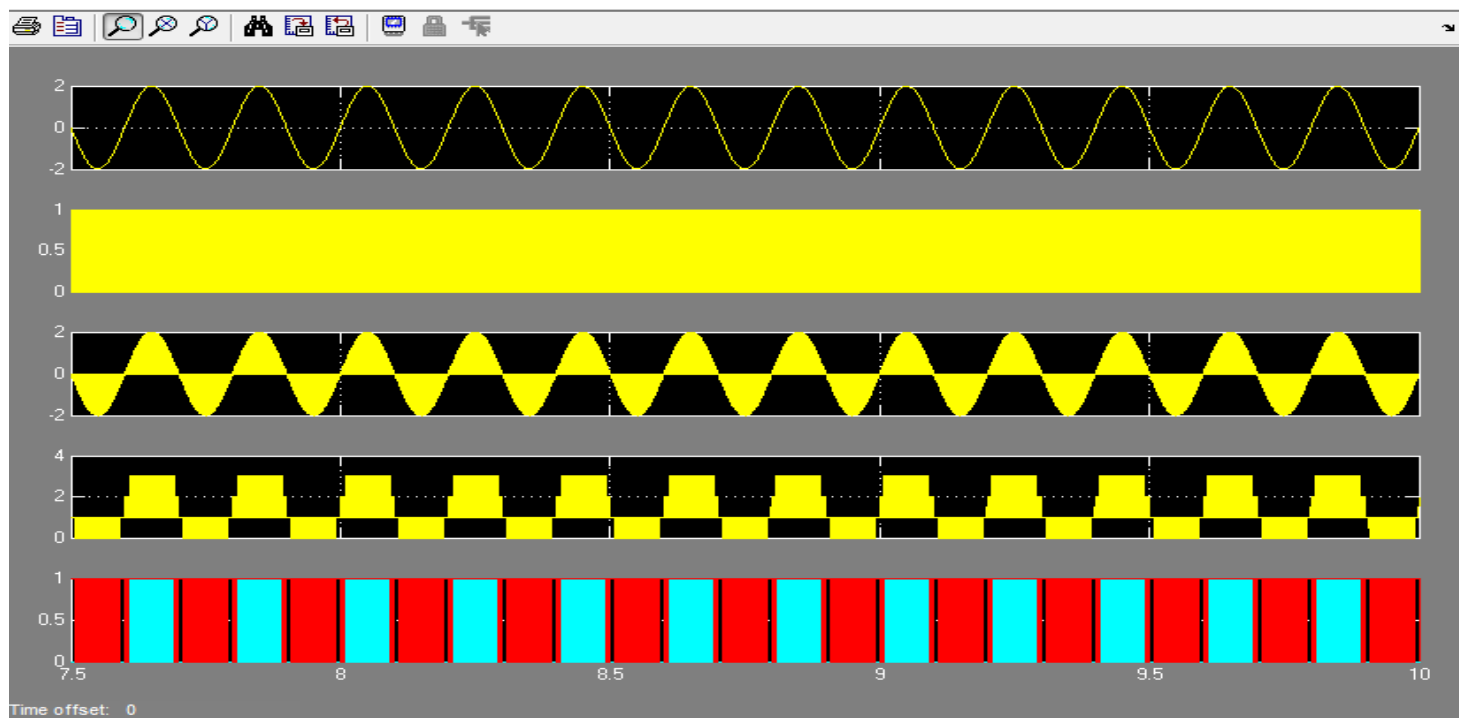


Figure 4: Output across PCM block diagram.

References:

1. https://www.tutorialspoint.com/digital_communication/digital_communication_pulse_code_modulation.htm
2. <https://www.mathworks.com/matlabcentral/fileexchange/73719-pulse-code-modulation>

Md. Humayun Kabir

Adjunct Lecturer

Dept. of Electronic and Telecommunication Engineering
International Islamic University Chittagong