Experiment 4

Performance of Waveform Coding using PCM

Objective

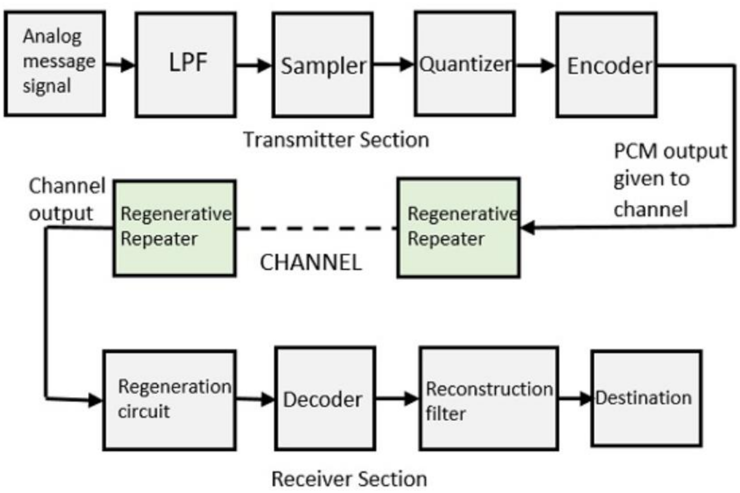
1. Generate a sinusoidal waveform with a DC offset so that it takes only positive amplitude value.
2. Sample and quantize the signal using an uniform quantizer with number of representation levels L. Vary L. Represent each value using decimal to binary encoder.
3. Compute the signal-to-noise ratio in dB.
4. Plot the SNR versus number of bits per symbol. Observe that the SNR increases linearly.

Software Used:

MATLAB R2017a

Theory:

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

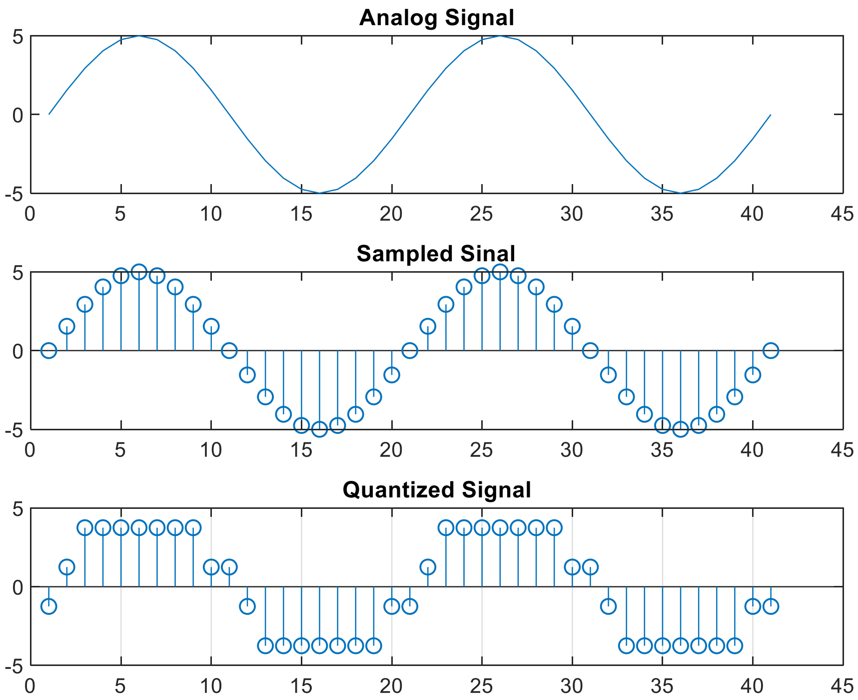


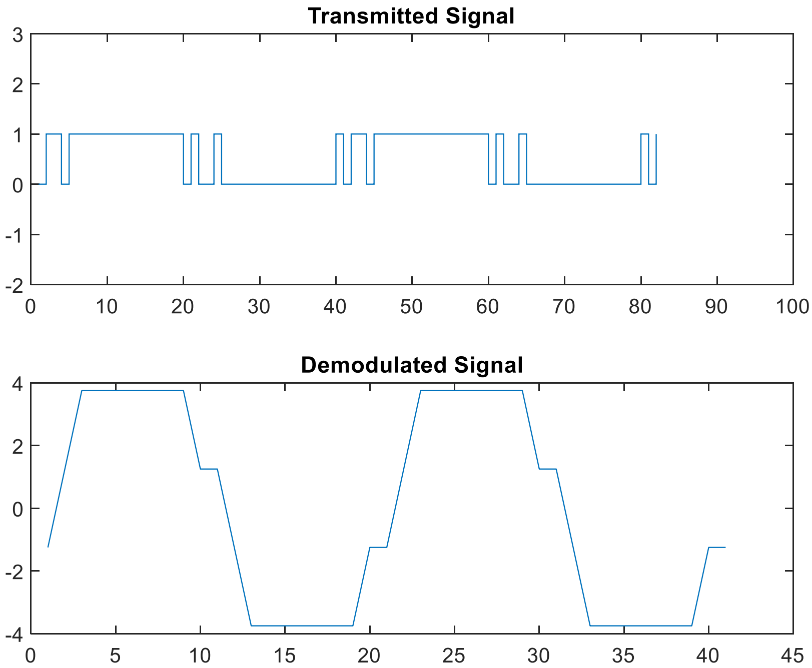
Program:

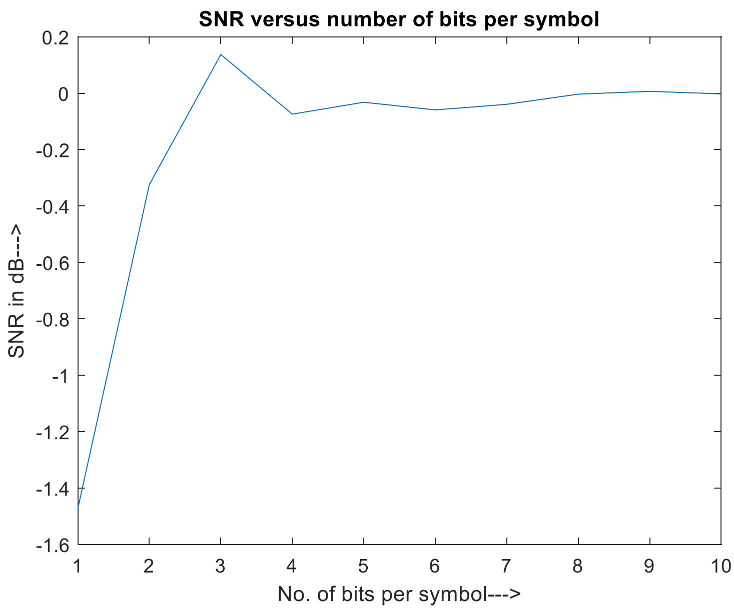
```
clc;
close all;
clear all;
n=input('Enter for n-bit PCM system :  '); %Encodebook Bit Length
n1=input('Enter Sampling Frequency : '); %Sampling Frequency
L = 2^n; %Number of Quantisation Levels %
% Here we plot the Analog Signal and its Sampled form
Vmax = 5;
x = 0:pi/n1:4*pi;  %Construction of Signal
ActualSignl=Vmax*sin(x); %Actual input
subplot(3,1,1);
plot(ActualSignl);
title('Analog Signal');
subplot(3,1,2); %Sampled Version
stem(ActualSignl);
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
% code = linspace(Vmin,Vmax,6);

[ind,q]=quantiz(ActualSignl,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');
% SNR vs no of bits per symbol
 for Nvar=1:10
   Lvar=2^Nvar;
   StepSizevar=(Vmax-Vmin)/Lvar;
   QLevelvar=Vmin:StepSizevar:Vmax;                        % level are
between vmin and vmax with difference of del
   code=Vmin-(StepSizevar/2):StepSizevar:Vmax+(StepSizevar/2);         %
Contaion Quantized valuses
   [ind,x]=quantiz(ActualSignl,QLevelvar,code);
   a(Nvar)=snr(x,ActualSignl);
 end
 figure
 Nvar=1:10;
 plot(Nvar,a);
  title('SNR versus number of bits per symbol');
 ylabel('SNR in dB--->');
 xlabel('No. of bits per symbol--->');
```

Output







Result

Studied and verified waveform coding using PCM.