EXPT. NO. 10	NAME:	Page No.: 1	Date: 28 Oct

EXPERIMENT 10 [U19CS012]

PULSE CODE MODULATION AND DEMODULATION

AIM: To demonstrate the pulse code modulation (PCM) and demodulation technique. To show the sampled, quantized / encoded and decoded time domain signal for different bit-codes.

SOFTWARE: MATLAB

THEORY: 1.> Pulse Code Modulation (PCM)

- PCM is a technique, which is used to convert an analog signal into digital signal.

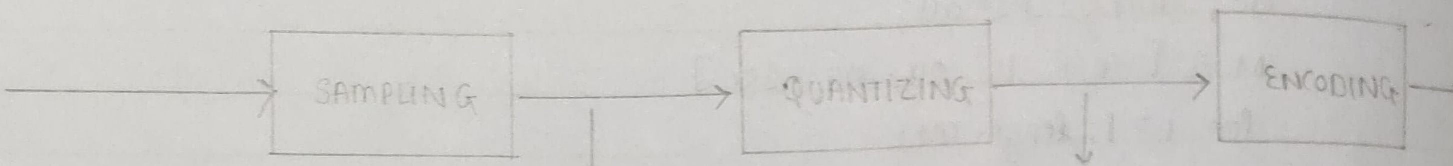
- PCM is a preferred method of communication within public switched telephone network (PSTN)

- A PCM stream is determined by two following steps:

  a) Sampling Rate: which the number of times per second that samples are taken.

  b) Bit depth: which determines the number of possible digital values that can be used to represent each sample.

- Hence, the output of PCM resembles a binary sequence.



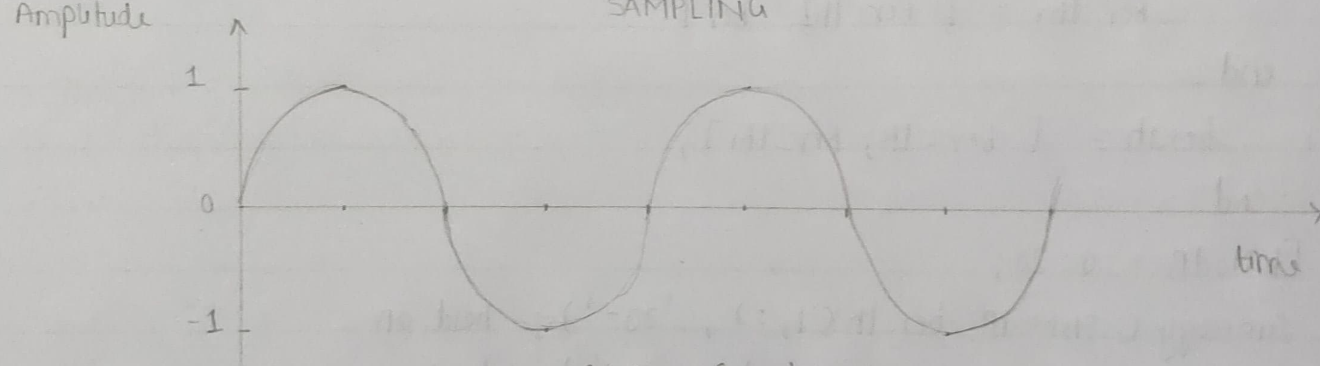
Continuous-time continuous Amplitude (analog) input signal

Discrete time continuous amplitude signal (PAM pulses)

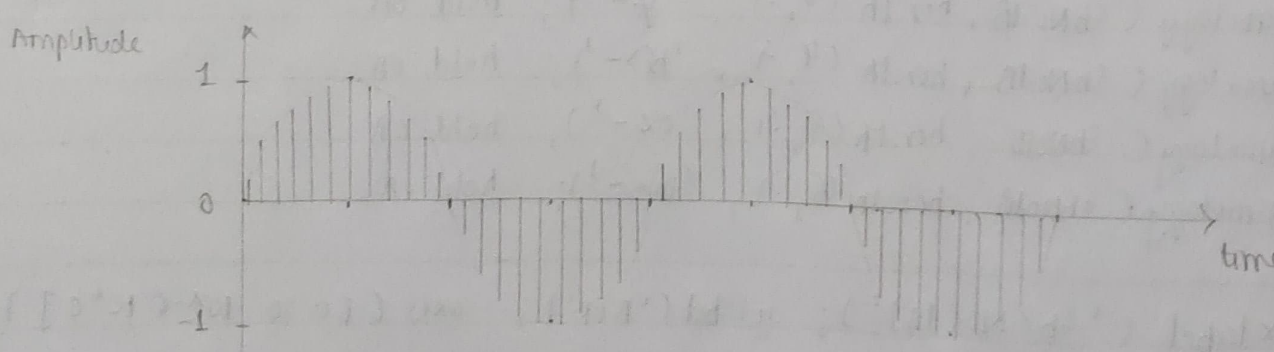
Discrete time Discrete amplitude signal (PCM pulses)

Digital Bit Stream Output

SAMPLING



Analog Signal



Discrete Time signal

2.) Reasons for Digital Transmission

- Less susceptible to interference cause by noise due to discrete level.

- Easy to detect errors due to discrete levels

- Easy to Encrypt (Higher Security)

- Simpler to store digital data

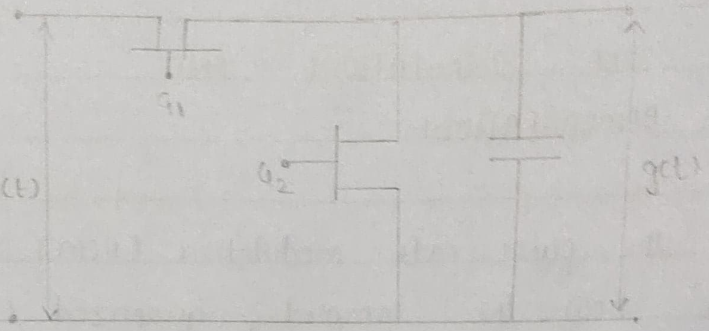
3.) Sampling

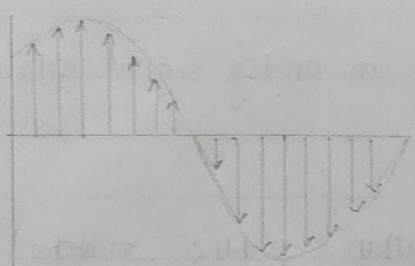
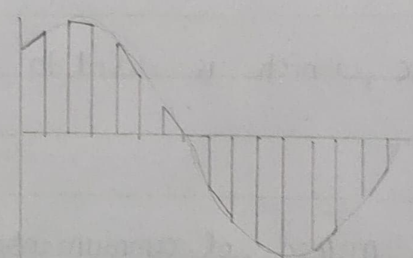
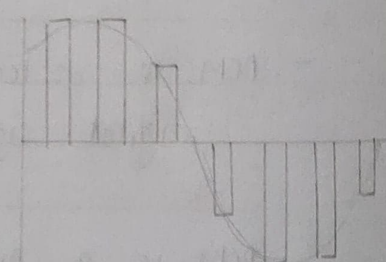
- Sampler extract samples of a continuous signal.
- Sampler produces samples that are equivalent to the instantaneous value of the continous signal at the specified various points.
- The sampling process generates Flat-top Pulse Amplitude Modulated (PAM) signal.

4.) Quantization

- 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 signal values with the nearest quantization values.

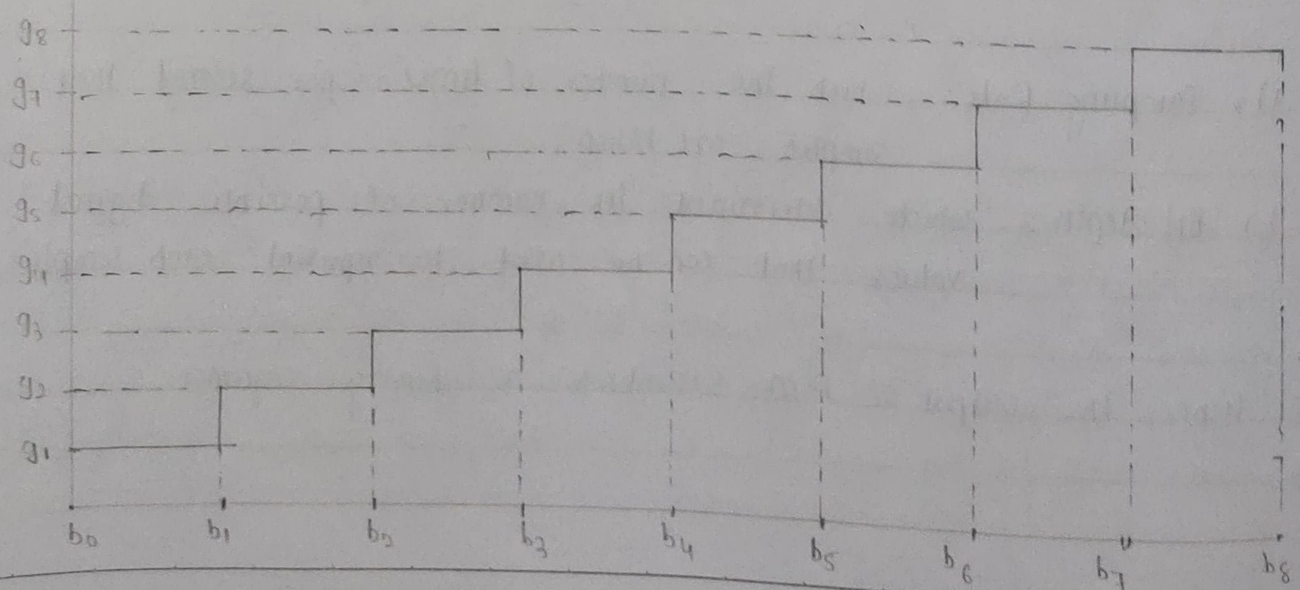
Flat-top PAM



Instantaneous Sampling	Natural Sampling	Flat-top Sampling
It is not practical method	This method is used practically	This is also used practically.
Sample rate = infinity	Sample Rate satisfied Nyquist	Sample rate satisfied Nyquist
		

Uniformly Quantized Signal

A/D output = n bits per sample (quantization level $M = 2^m$)



5) Pulse Code Modulation (PCM) is a method of converting an Analog signal into a digital signal. (A/D conversion)

- PCM produces a series of number or digits instead of pulse train.
- Each one of these digits, in binary code, represents the approximate amplitude of the signal sample at that instant.

6) Concluding Remark for PCM

- In PCM Transmitter, the signal $x(t)$ is first passed through the low pass filter of cut-off frequency F_m Hz.

- This low pass filter blocks all the Frequency components above F_m Hz. This means that now the signal $x(t)$ is band-limited to F_m Hz.

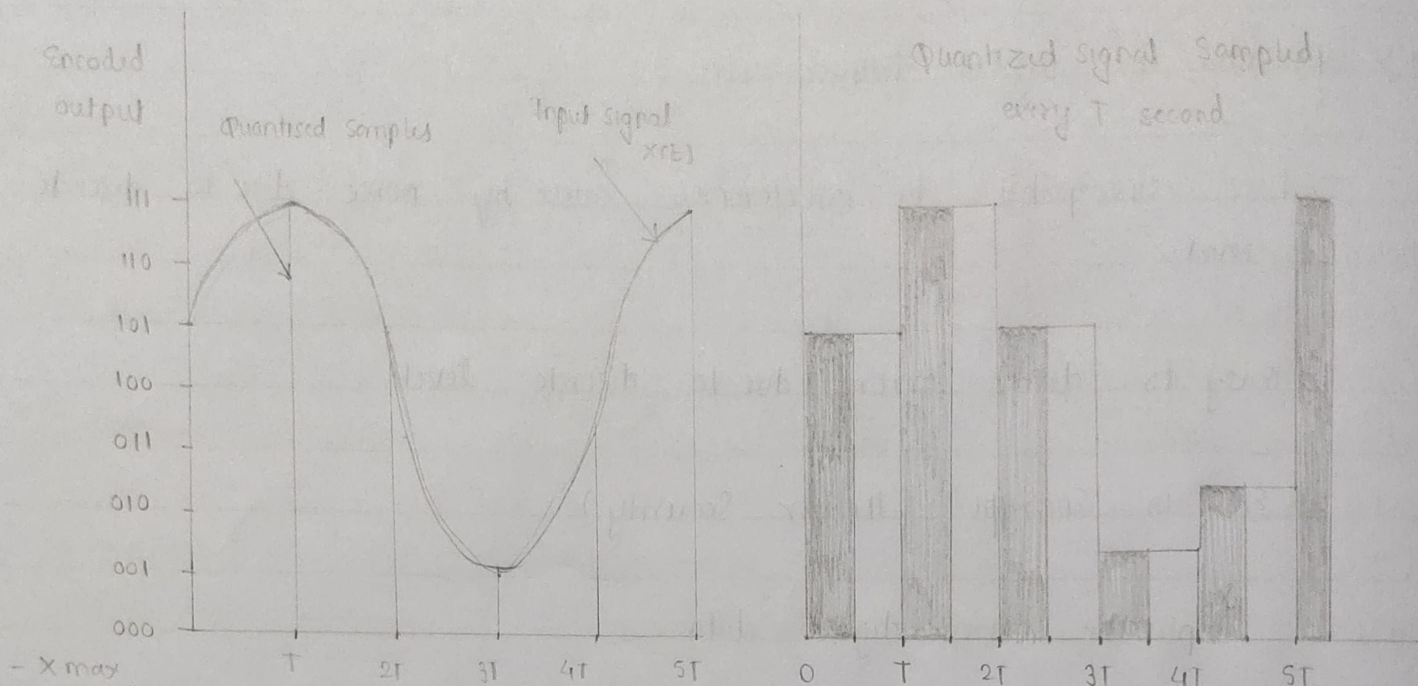
- The Sample and Hold circuit then samples this signal at the rate of F_s.

- Sampling Frequency F_s is selected sufficiently above Nyquist rate to avoid Aliasing.

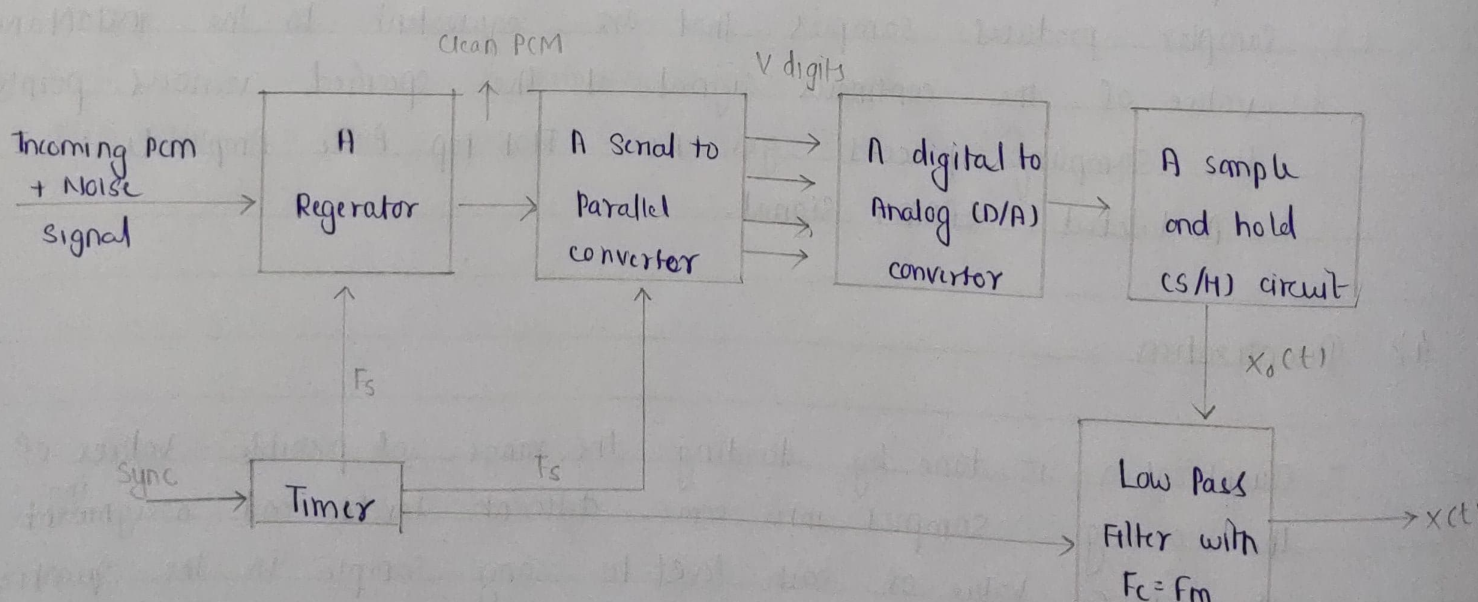
- The Output from sample and hold circuit is denoted by $x(nT_s)$

Transmitter

Figure : 1 Quantization of a sampled Analog signal



PCM Receiver



- This signal $x(nT_s)$ is discrete in time and continuous in Amplitude
- A q-level quantizer compares input $x(nT_s)$ with its fixed digital levels.
- Quantized signal is then encoded in PCM output using encoder.

7.) PCM Standards

- There are two standards of PCM ① The European standard
  ② The American Standard
- They differ slightly in the detail of their working but the principles are the same.

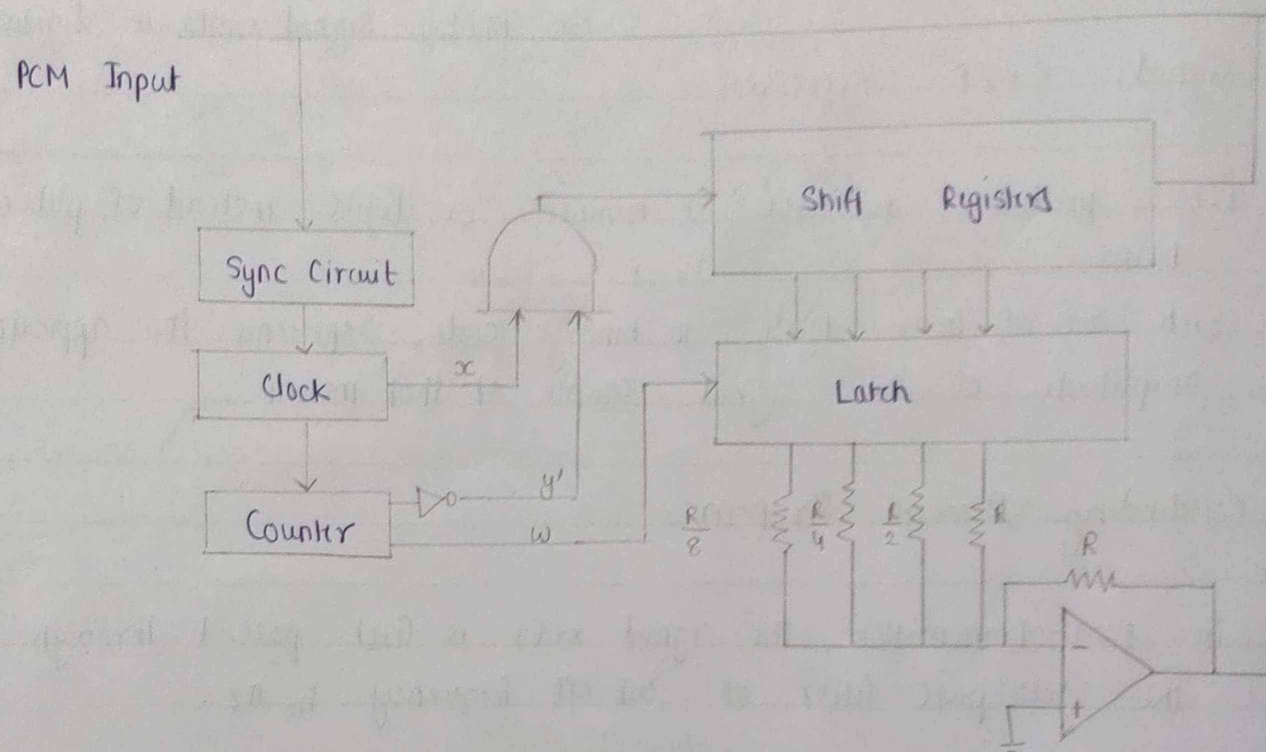
- European PCM = 30 channels
- North American PCM = 24 channels
- Japanese PCM = 24 channels

In India, we follow the European PCM of 30 channels system working.

8.) Applications

- In Compact Disks
- Digital Telephony
- Digital Audio Applications

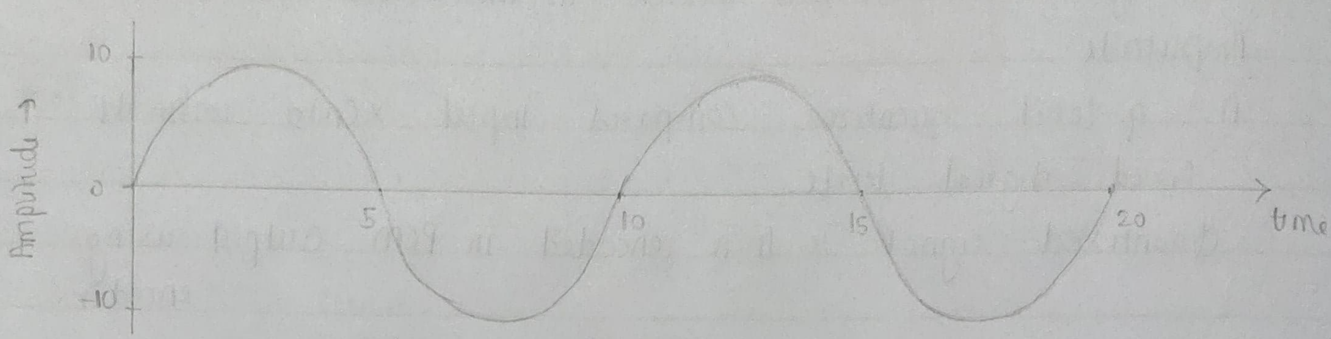
PCM Demodulator



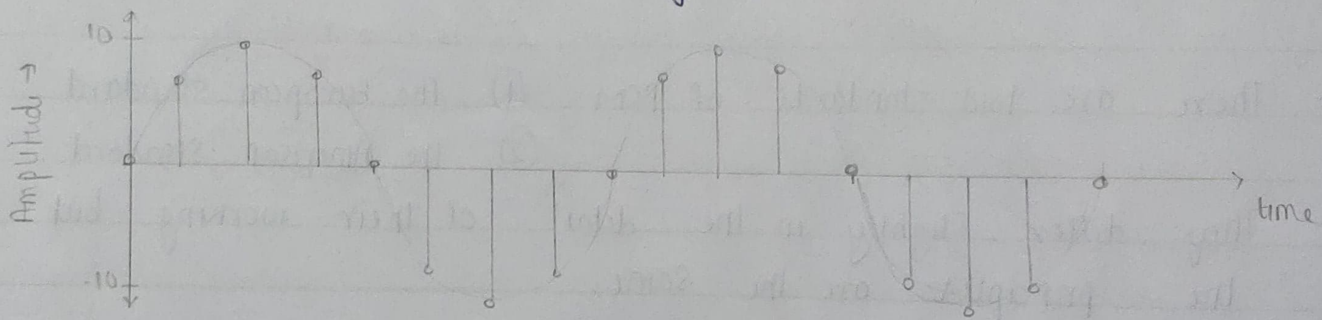
MATLAB CODE: ① Sampling

```
n= input('Enter n value for n-bit pcm system:');
n1 = input('Enter number of samples in a period :');
  L = 2^n;
%% Signal generation
% x = 0:1/100 : 4*pi;
% y = 8*sin(x)
% subplot(2,2,1);
% plot(x,y); grid on;
 % SAMPLING OPERATION
   x = 0: 2*pi/n1 : 4*pi;
   s = 8*sin(x);
   subplot(3,1,1);
   plot(s);
   title('Analog Signal');
   ylabel('Amplitude-->');
   xlabel('Time -->');
   subplot(3,1,2);
   stem(s);
   grid on;
   title('Sampled Signal');
   ylabel('Amplitude -->');
   xlabel('Time -->');
```

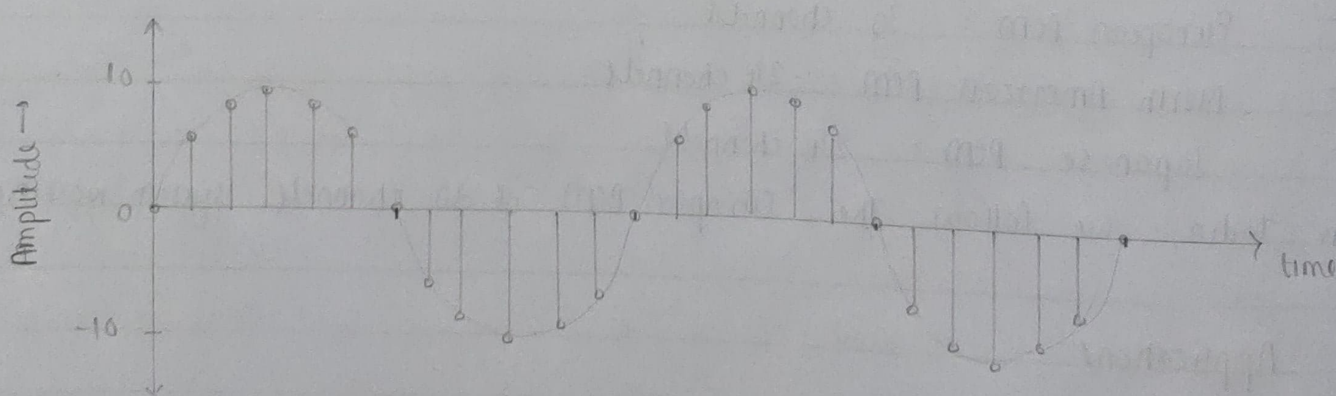
Outputs from MATLAB



Analog signal ↑



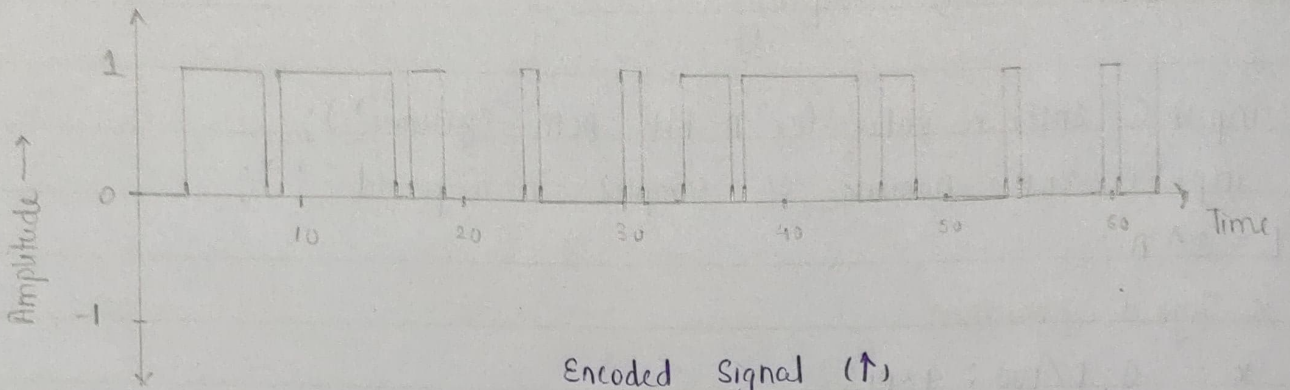
Sampled signal (↑)



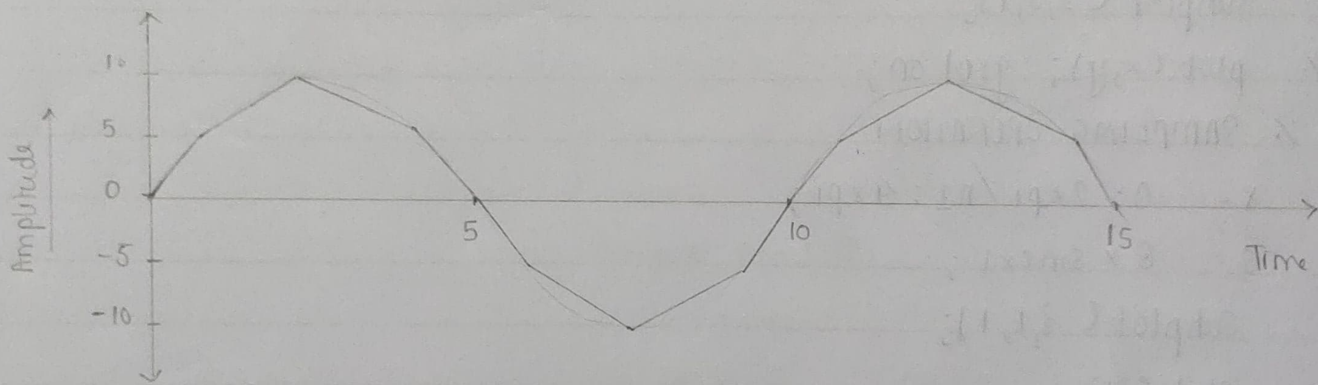
Quantized Signal

(2) Quantization Process

```
vmax = 8
vmin = - vmax;                    % level are between vmin & vmax with diff del
del = (vmax - vmin)/ L;
part = vmin: del : vmax;
code = vmin - (del/2) : del : vmax + (del/2);
[ind, q] = quantiz ( s, part, code);        % quantization Process

l1 = length (ind);
l2 = length (q);

for i = 1: l1                     % to make index as binary decimal so started
                                  0 to N
    if (ind(i)~= 0)
        ind(i) = ind(i) -1;
    end
    i = i+1;
end

for i = 1: l2                     % to make quantize value in between lvl
    if ( q(i) == vmin - (del/2) )
        q(i) = vmin + (del/2);
    end
subplot(3,1,3);                   % Display Quantize values
stem (q); grid on;
title ('Quantized Signal');
ylabel ('Amplitude -->');
xlabel ('Time -->');
```



Encoded Signal (↑)



Demodulated Signal

3 Encoding

```
% Encoding Process
figure
code = de2bi( ind, 'left-msb');          % convert decimal to binary
k=1;
for i=1: l1
    for j=1:n
        coded(k) = code(i,j);            % convert code matrix to coded row
        j=j+1;
        k=k+1;
    end
  i=i+1;
end
subplot(2,1,1); grid on;
stairs(coded);
axis([0 100 -2 3]);                      % Display the encoded signal
title('Encoded Signal');
ylabel('Amplitude -->');
xlabel('Time -->');
```

[U19CS012]

④ Demodulation of PCM signal

```
quant = reshape ( coded, n, length(coded)/n );
index = bi2de ( quant', 'left- msb' );          % getback index in decimal
q = del * index + Vmin + ( del/2 ) ;            % getback quantized values
subplot ( 2,1,2 );  grid on;
plot (q) ;
title ('Demodulated Signal');                   % Plot Demodulated signal
ylabel ('Amplitude -->');
xlabel ('Time -->');
```

> CONCLUSION: (1) We successfully demonstrated the Pulse Code Modulation (PCM) and demodulation technique.

(2) We observed block diagrams for receiver and transmitter of PCM signals. In later stage we also observed demodulation circuit which consist of Shift Registers, Latch and opamp.

(3) In the last phase, we executed MATLAB code and observed sampling, Quantization, Encoding and Demodulation waves and drawn them.

—×—