Govt. of N.C.T of Delhi

G.B PANT ENGINEERING COLLEGE

Okhla Industrial Estate Phase-III, New Delhi – 110020

Department of Electronics and Communication Engineering

B. Tech (CSE) – 5th Semester

ETEC-357 Digital Communication Lab Practical File

Submitted by:
RAVI PAWAR
03620902718
CSE 5th SEM

Submitted to:
Mr. Padam Saini

Index

S.no.	Title	Page No.	Remarks

Practical 1

Expt. No. 1

Date

Page No.

AIM

To study Sampling Theorem

APPARATUS REQUIRED

System Running with MATLAB software

THEORY

Sampling:

By sampling a continuous time signal at isolated equally spaced points in time, we obtain sequence of numbers

$n \in \{\ldots, -2, -1, 0, 1, 2, 3, \ldots\}$

T_s is the Sampling Period

Many Signal originate as continuous-time signal, eg-> Conventional music or voice

Frequency Domain:

- Replicates Spectrum of Continuous-Time Signal at offsets that are integer multiples of sampling frequency.
- Fourier Series of impulse train where $\omega_s = 2\pi f_s$
- As sampling rate increases, sampled waveform looks more and more like the original.
- Many applications (eg. communication systems) care more about frequency content in the waveform and not its shape
- Zero crossing: frequency content of sinusoid when it meet the axis.

Teacher's Signature:

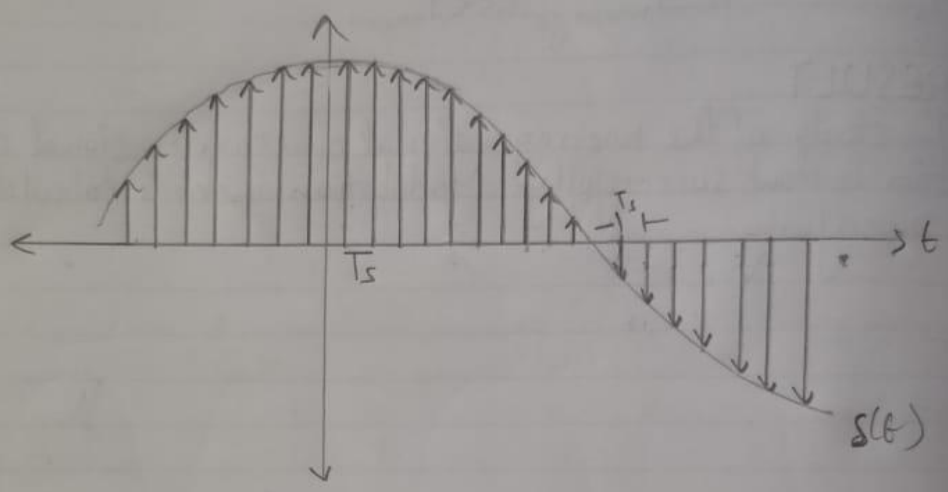


Fig: Sampled Analog Waveform

Expt. No. ____________ Date ____________

Page No. ________

Shanan Sampling Theorem

A continuous time signal $x(t)$ with frequency no higher than f_{max} can be reconstructed from its samples $x[n] = x(nT_s)$ by taking samples at a rate f_s which is greater than $2f_{max}$.

$$\text{Nquist rate} = 2 f_{max}$$
$$\text{Nqust frequency} = f_s/2$$

Consider a sinusoid $\sin(2\pi f_{max} t)$
use a sampling period of $T_s = 1/f_s = 1/2f_{max}$

Assumptions

1. Continuous Time signal has no frequency content above f_{max}
2. Sampling time is exactly the same b/w any two samples
3. Sequence of no. obtained by sampling is represented in exact precision.
4. Conversion of sequence to continuous time is ideal.

Aliasing

→ Analog sinusoid.
$x(t) = A\cos(2\pi f_0 t + \phi)$

→ Sampled at $T_s = 1/f_s$
$x[n] = x[T_s n] = A\cos(2\pi f_0 T_s n + \phi)$

→ Keeping the sampling period same, sample
$y(t) = A\cos(2\pi(f_0 + l f_s)t + \phi)$
where l is an integer

$$y[n] = y(T_s n)$$
$$= A\cos(2\pi(f_0 + l f_s)T_s n + \phi)$$
$$= A\cos(2\pi f_0 T_s n + 2\pi l f_s T_s n + \phi)$$

Teacher's Signature : ____________

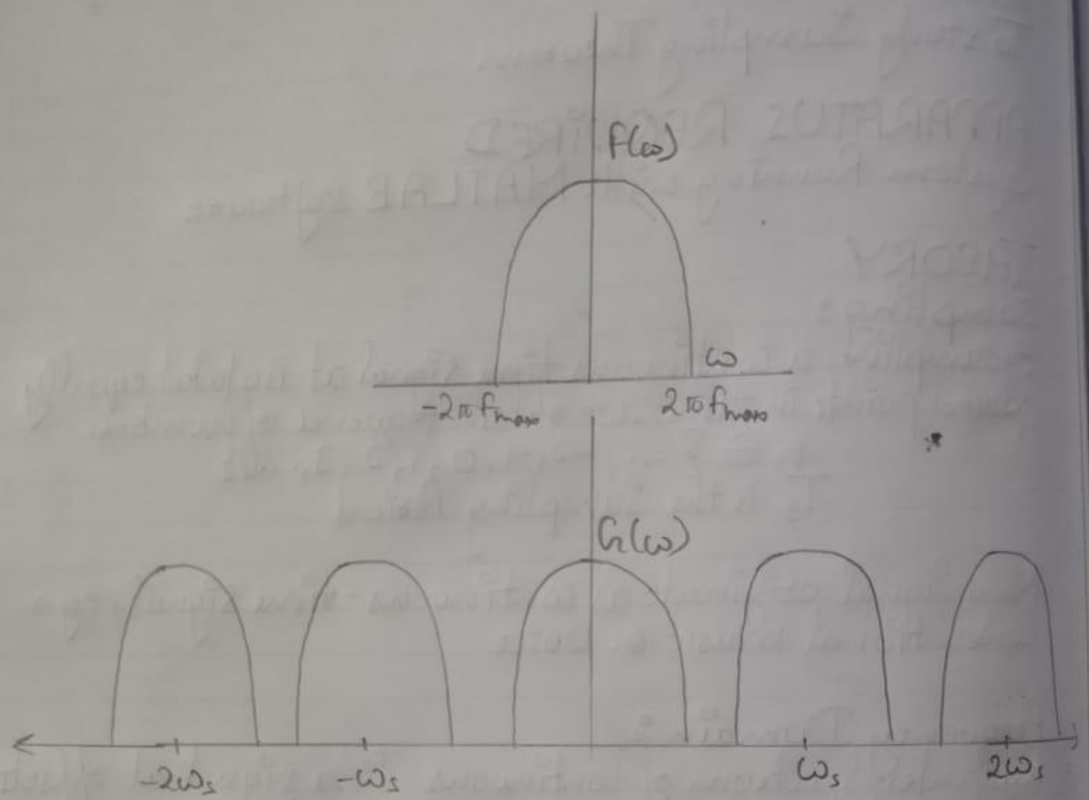


Fig:- Frequency Domains.

Expt. No. ____________ Date ____________

Page No. ________

$= A\cos(2\pi f_0 T_s n + 2\pi l n + \phi)$

$= A\cos(2\pi f_0 T_s n + \phi)$

$= x[n]$

Here, $F_s T_s = 1$

Since l is an integer

$\cos(n \pm 2\pi l) = \cos(n)$

$\Rightarrow$ $y[n]$ indistinguishable from $x[n]$.

CONCLUSION

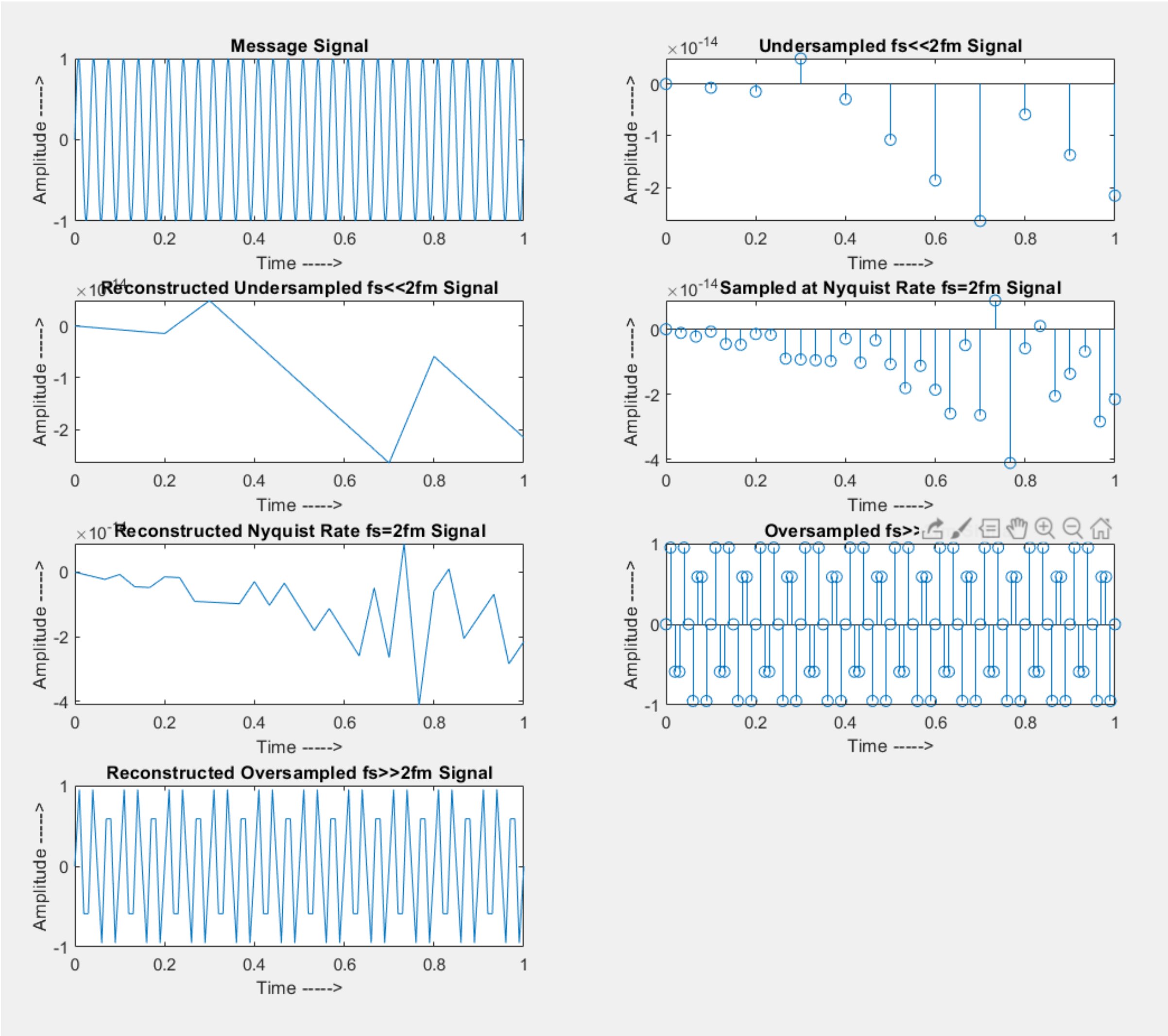
Hence, we successfully studied about Sampling Theorem.

Teacher's Signature : ____________

Program (Sampling Theorem)

MATLAB PROGRAM TO IMPLEMENT SAMPLING THEOREM

```
t = 0:0.001:1;
fm = input ('Enter the modulating signal frequency = ');
x = sin(2*pi*fm*t);
subplot (4,2,1);
plot(t,x);
xlabel('Time ----->');
ylabel('Amplitude ----->');
title('Message Signal');
fs1 = input('Enter Sampling Frequency < Modulating Signal Frequency = ');
fs2 = input('Enter Sampling Frequency = Modulating Signal Frequency = ');
fs3 = input('Enter Sampling Frequency > Modulating Signal Frequency = ');

%Sampling at fs<<2fm
n = 0:1/fs1:1;
x1 = sin(2*pi*fm*n);
subplot(4,2,2);
stem(n,x1);
xlabel('Time ----->');
ylabel('Amplitude ----->');
title('Undersampled fs<<2fm Signal');
subplot(4,2,3);
plot(n,x1);
xlabel('Time ----->');
ylabel('Amplitude ----->');
title('Reconstructed Undersampled fs<<2fm Signal');

%Sampling at fs=2fm
n = 0:1/fs2:1;
x2 = sin(2*pi*fm*n);
subplot(4,2,4);
stem(n,x2);
xlabel('Time ----->');
ylabel('Amplitude ----->');
title('Sampled at Nyquist Rate fs=2fm Signal');
subplot(4,2,5);
plot(n,x2);
xlabel('Time ----->');
ylabel('Amplitude ----->');
title('Reconstructed Nyquist Rate fs=2fm Signal');

%Sampling at fs>>2fm
n = 0:1/fs3:1;
x3 = sin(2*pi*fm*n);
subplot(4,2,6);
stem(n,x3);
xlabel('Time ----->');
ylabel('Amplitude ----->');
title('Oversampled fs>>2fm Signal');
subplot(4,2,7);
plot(n,x3);
xlabel('Time ----->');
ylabel('Amplitude ----->');
title('Reconstructed Oversampled fs>>2fm Signal');
```


Message Signal
Undersampled fs<<2fm Signal
Reconstructed Undersampled fs<<2fm Signal
Sampled at Nyquist Rate fs=2fm Signal
Reconstructed Nyquist Rate fs=2fm Signal
Oversampled fs>>
Reconstructed Oversampled fs>>2fm Signal
Amplitude ----->
Time ----->
×10^-14

Practical 2

Expt. No. 2

Date

Page No.

AIM

To study Pulse Code Modulation (PCM)

THEORY

Pulse Code Modulation (PCM) is a method used to digitally represent sampled analog signals. It is the standard form for digital audio in computers and various Blu-ray, Compact Disc and DVD formats, as well as other uses such as digital telephone systems. A PCM stream is a digital representation of an analog signal, in which the magnitude of the analog signal is sampled regularly at uniform intervals, with each sample being quantized to the nearest value within a range of digital steps.

Basis of Pulse Code Modulation.

The three steps for developing an equivalent PCM digital signal from an analog signal are -

1. Sampling
2. Quantization
3. Coding

1. Sampling

The foundation of PCM is based on Nyquist Sampling Theorem. If a unlimited signal is sampled at regular intervals of time and at a rate equal to or higher than twice the highest significant signal frequency, then the sample contains all the information of the original signal. The original signal may then be reconstructed by use of a low pass filter.

Teacher's Signature :

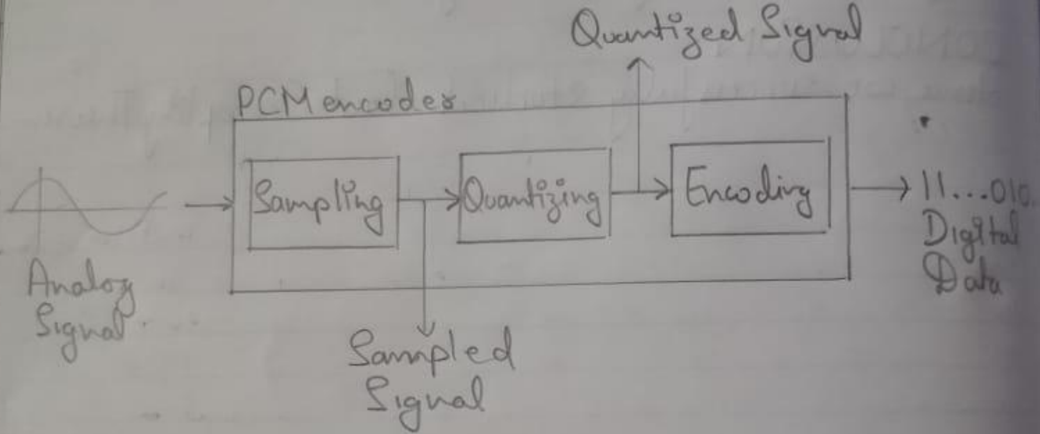


Fig: Block Diagram Pulse Code Modulation

Date

Expt. No.   Page No.

② Quantization.

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 of a near stabilized value. Such process is called Quantization.

③ Coding

Older PCM system uses 7 bit code, and modern systems use an 8 bit code with its improved quantizing distortion performance. The companding and ~~expanding~~ coding is done together simultaneously. The compression and later expansion functions are logarithmic. A pseudologarithmic curve made up of linear segments imparts finer granularity to low level signals and less granularity to the higher-level signals.

Advantages of Pulse Code Modulation

1. Immune to channel induced noise and distortion.
2. Repeaters can be employed along the transmitting channel
3. Encoders allow secured data transmission.
4. It ensures uniform transmission quality

Disadvantages of Pulse Code Modulation.

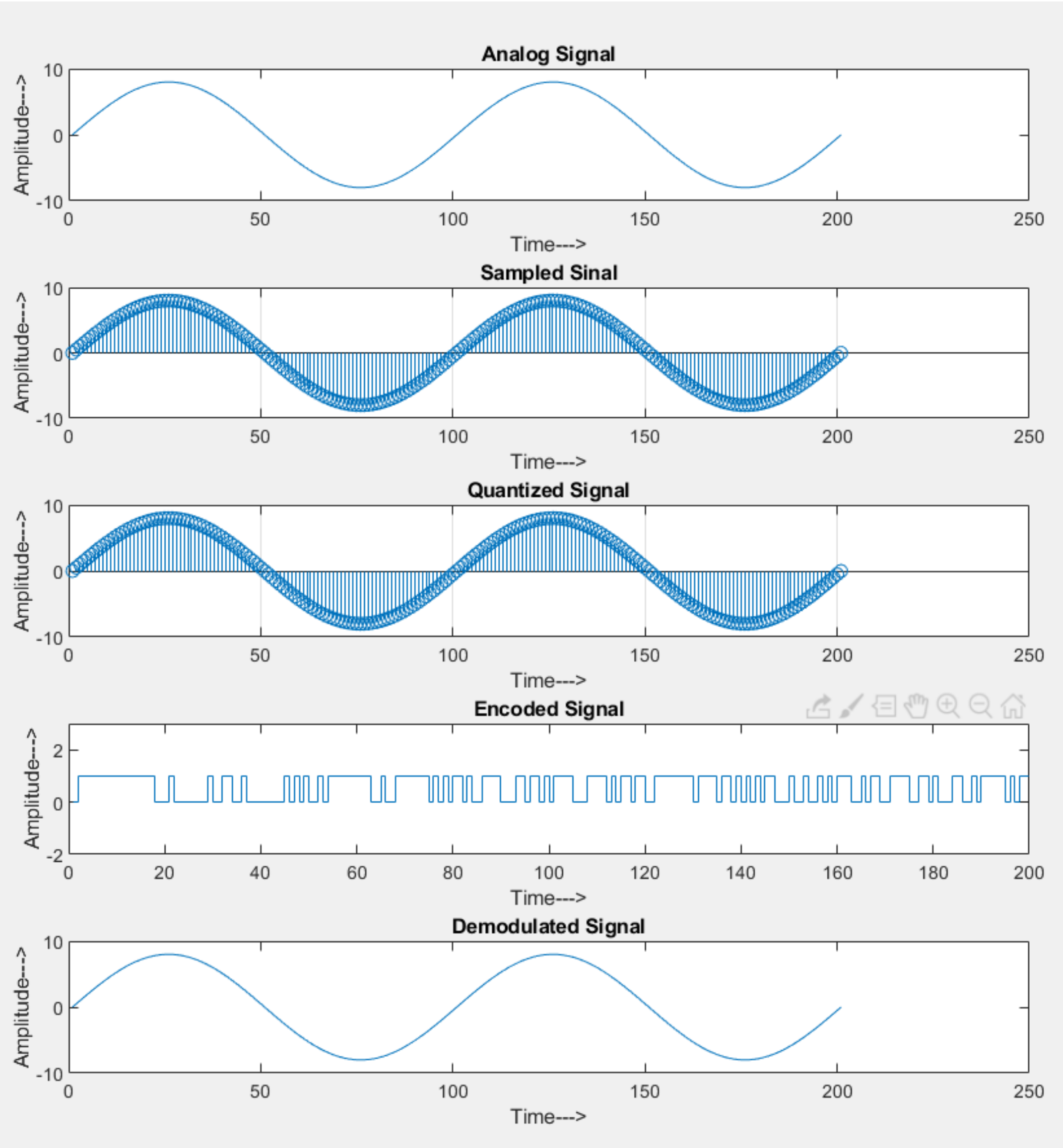
1. Pulse Code Modulation increases the transmission Bandwidth
2. A PCM system is somewhat more complex than other system.

Teacher's Signature :

MATLAB PROGRAM TO IMPLEMENT PULSE CODE MODULATION AND DEMODULATION

```
n=input('Enter n value for n-bit PCM system : ');
n1=input('Enter number of samples in a period : ');
L=2^n;

% Sampling Operation
x=0:2*pi/n1:4*pi;
s=8*sin(x);
subplot(5,1,1);
plot(s);
title('Analog Signal');
ylabel('Amplitude--->');
xlabel('Time--->');
subplot(5,1,2);
stem(s);grid on; title('Sampled Sinal'); ylabel('Amplitude--->'); xlabel('Time--->');

% Quantization Process
vmax=8;
vmin=-vmax;
del=(vmax-vmin)/L;
part=vmin:del:vmax;
code=vmin-(del/2):del:vmax+(del/2);
[ind,q]=quantiz(s,part,code);
l1=length(ind);
l2=length(q);
for i=1:l1
if(ind(i)~=0)
ind(i)=ind(i)-1;
end
i=i+1;
end
for i=1:l2
if(q(i)==vmin-(del/2))
q(i)=vmin+(del/2);
end
end
subplot(5,1,3);
stem(q);grid on;
title('Quantized Signal');
ylabel('Amplitude--->');
xlabel('Time--->');

% Encoding Process
code=de2bi(ind,'left-msb');
k=1;
for i=1:l1
for j=1:n
coded(k)=code(i,j);
j=j+1;
k=k+1;
end
i=i+1;
end
subplot(5,1,4); grid on;
```

```
stairs(coded);
axis([0 2*n1 -2 3]); title('Encoded Signal');
ylabel('Amplitude--->');
xlabel('Time--->');

% Demodulation Of PCM signal
qunt=reshape(coded,n,length(coded)/n);
index=bi2de(qunt','left-msb');
q=del*index+vmin+(del/2);
subplot(5,1,5); grid on;
plot(q);
title('Demodulated Signal');
ylabel('Amplitude--->');
xlabel('Time--->');
```



Practical 3