Estd : 2001

RN SHETTY TRUST®

RNS INSTITUTE OF TECHNOLOGY

Affiliated to VTU, Recognized by GOK, Approved by AICTE, New Delhi
(NAAC 'A+ Grade' Accredited, NBA Accredited (UG - CSE, ECE, ISE, EIE and EEE)
Channasandra, Dr. Vishnuvardhan Road, Bengaluru - 560 098
Ph:(080)28611880,28611881 URL: www.rnsit.ac.in

Department of Electronics and Communication

IV SEMESTER

Principles of Communication Systems –BEC402

LAB MANUAL 2023-24

(As per VTU course type-IPCC)

Compiled by

Department of ECE

RNS Institute of Technology

Bengalore-98

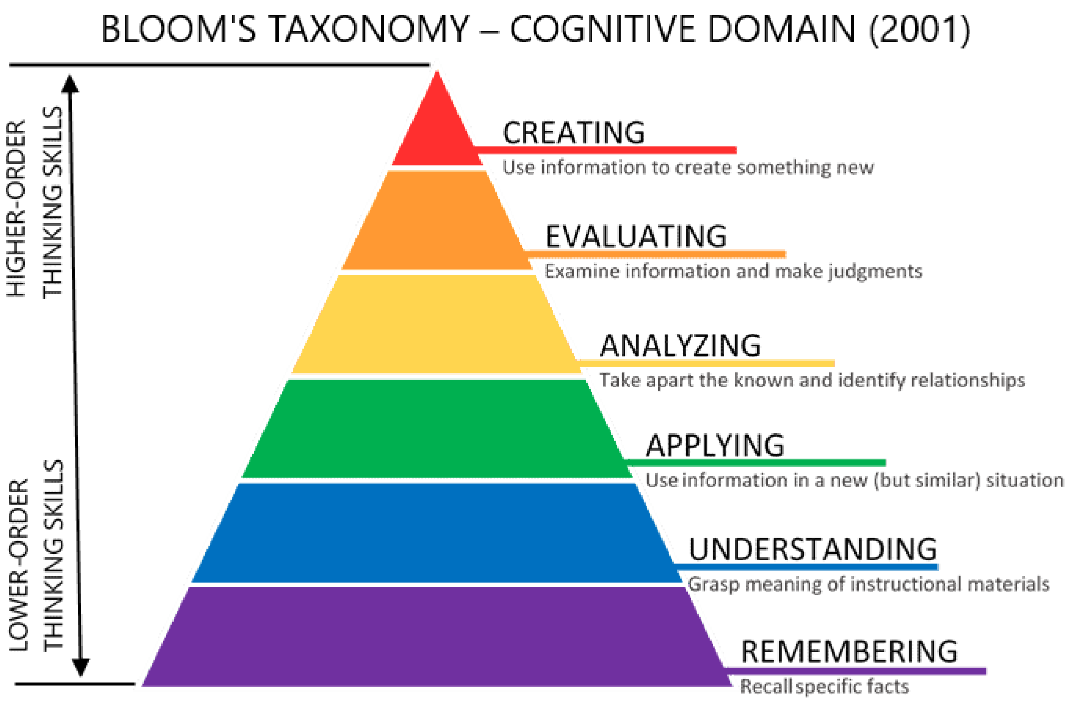
NAME____

USN____

SEM & SECTION______________________________

Revised Bloom's Taxonomy (RBT)

Bloom's cognitive Domain – 2001



Trademark

RNS INSTITUTE OF TECHNOLOGY

ESTD:2001

An Institute with a Difference

Edition 2023-24

Document owner

Author(s)	Dr. Uma S V	uma.s.v@rnsit.ac.in
	Dr. Rajini V honnungar	rajini.v.honnungar@rnsit.ac.in
	Dr. Prabhavathi C N	prabhavathi.cn@rnsit.ac.in
	Mrs. Ghousia Begum S	ghousia.begum.s@rnsit.ac.in
	Mrs. Chethana J	chethana.j@rnsit.ac.in
	Dr. Leena Chandrashekar	leena.chandrashekar@rnsit.ac.in
	Mr. Ugrasena Maharaj	ugrasenamaharaj@rnsit.ac.in
	Mrs. J S Shwetha	shwetha.js@rnsit.ac.in

VISION OF THE COLLEGE

Building RNSIT into a World - Class Institution.

MISSION OF THE COLLEGE

To impart high quality education in Engineering, Technology and Management with a difference, enabling students to excel in their career by

1. Attracting quality Students and preparing them with a strong foundation in fundamentals so as to achieve distinctions in various walks of life leading to outstanding contributions.
2. Imparting value based, need based, and choice based and skill based professional education to the aspiring youth and carving them into disciplined, World class Professionals with social responsibility.
3. Promoting excellence in Teaching, Research and Consultancy that galvanizes academic consciousness among Faculty and Students.
4. Exposing Students to emerging frontiers of knowledge in various domains and make them suitable for Industry, Entrepreneurship, Higher studies, and Research & Development.
5. Providing freedom of action and choice for all the Stake holders with better visibility.

VISION OF THE DEPARTMENT

Conquering technical frontiers in the field of Electronics and Communications.

MISSION OF THE DEPARTMENT

1. To achieve and foster excellence in core Electronics and Communication engineering with focus on the hardware, simulation and design.
2. To pursue Research, development and consultancy to achieve self-sustenance.
3. To create benchmark standards in electronics and communication engineering by active involvement of all stakeholders.

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

ECE Graduates within three-four years of graduation will have

- **PEO1**: Acquired the fundamentals of computers and applied knowledge of Information Science & Engineering and continue to develop their technical competencies by problem solving using programming.
- **PEO2**: Ability to formulate problems attained the Proficiency to develop system/application software in a scalable and robust manner with various platforms, tools and frameworks to provide cost effective solutions.
- **PEO3**: Obtained the capacity to investigate the necessities of the software Product, adapt to technological advancement, promote collaboration and interdisciplinary activities, Protecting Environment and developing Comprehensive leadership.
- **PEO4**: Enabled to be employed and provide innovative solutions to real-world problems across different domains.
- **PEO5**: Possessed communication skills, ability to work in teams, professional ethics, social responsibility, entrepreneur and management, to achieve higher career goals, and pursue higher studies.

PROGRAM OUTCOMES (POs)

Engineering Graduates will be able to:

- **PO1: Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization for the solution of complex engineering problems
- **PO2: Problem analysis:** Identify, formulate, research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences and engineering sciences.
- **PO3: Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for public health and safety, and cultural, societal, and environmental considerations.
- **PO4: Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
- **PO5: Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools, including prediction and modelling to complex engineering activities, with an understanding of the limitations.
- **PO6: The engineer and society:** Apply reasoning informed by the contextual knowledge to assess Societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
- **PO7: Environment and sustainability:** Understand the impact of the professional engineering

solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

- **PO8: Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
- **PO9: Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
- **PO10: Communication:** Communicate effectively on complex engineering activities with the engineering community and with the society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
- **PO11: Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
- **PO12: Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES (PSOs)

ECE Graduates will have

- **PSO1:** Apply fundamental knowledge of Electronics, Communications, Signal processing, VLSI, Embedded and Control systems etc., in the analysis, design, and development of various types of real-time integrated electronic systems and to synthesize and interpret the experimental data leading to valid conclusions.
- **PSO2:** Demonstrate competence in using Modern hardware languages and IT tools for the design and analysis of complex electronic systems as per industry standards along with analytical and managerial skills to arrive at appropriate solutions, either independently or in team.

CO-PO and CO-PSO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	2	2	1		2	1	1		2	1	1	1	1	
CO2	2	2	2		2	1	1	1	2	1	1	1	1	
CO3	2	2	2		2	1	1	1	2	1	2	1	1	
CO4	2	2	2		2	1	1	1	2	1	2	1	1	
CO5	2	2	2		2	1	1	1		1	2	1	1	
Avg. CO	**2**	**2**	**1.8**		**2**	**1**	**1**	**0.8**	**1.6**	**1**	**1.6**	**1**	**1**	

LIST OF EXPERIMENTS (AS PER VTU SYLLABUS)

PRACTICAL COMPONENT OF IPCC (Experiments can be conducted using a suitable circuit simulation software or hardware components)

Sl.NO	Experiments
1	Basic Signals and Signal Graphing: a) unit Step, b) Rectangular, c) standard triangle d) sinusoidal and e) Exponential signal.
2	Illustration of signal representation in time and frequency domains for a rectangular pulse.
3	Amplitude Modulation and demodulation: Generation and display the relevant signals and its spectrums.
4	Frequency Modulation and demodulation: Generation and display the relevant signals and its spectrums.
5	Sampling and reconstruction of low pass signals. Display the signals and its spectrum.
6	Time Division Multiplexing and demultiplexing.
7	PCM Illustration: Sampling, Quantization and Encoding
8	Generate a)NRZ, RZ and Raised cosine pulse, b) Generate and plot eye diagram
9	Generate the Probability density function of Gaussian distribution function.
10	Display the signal and its spectrum of an audio signal.

Course outcomes (Course Skill Set):

At the end of the course, the student will be able to:

1. Understand the principles of analog communication systems and noise modelling.
2. Identify the schemes for analog modulation and demodulation and compare their performance.
3. Design of PCM systems through the processes sampling, quantization and encoding.
4. Describe the ideal condition, practical considerations of the signal representation for baseband transmission of digital signals.
5. Identify and associate the random variables and random process in Communication system design.

Experiment no. 1: Basic Signals and Signal Graphing

AIM: Generate Basic Signals and their graphs: a) unit Step, b) Rectangular, c) standard triangle d) sinusoidal and e) Exponential signal.

APPARATUS: Matlab, Symbolic Math Toolbox

THEORY: The step signal or step function is that type of standard signal which exists only for positive time and it is zero for negative time. In other words, a signal x(t) is said to be step signal if and only if it exists for t > 0 and zero for t < 0. If a step signal has unity magnitude, then it is known as unit step signal or unit step function. It is denoted by u(t). The step signal is equivalent to applying a signal to a system whose magnitude suddenly changes and remains constant forever after application. If we want to obtain a signal which start at t = 0, so that it may have a value of zero for t < 0, then we only need to multiply the given signal with the unit step signal u(t). In practice, the unit step signal is used as a test signal because the response of a system for the unit step signal gives the information about how quickly the system responds to a sudden change in the input signal. A signal that produces a rectangular shaped pulse with a width of τ (where $\tau = 1$ for the unit rectangular function) centred at $t = 0$ is known as rectangular signal. The rectangular signal pulse also has a height of 1. The rectangular signal is also known as the unit pulse, gate function or normalised boxcar function. Also, the rectangular function is an even function of time. A function whose graph takes the shape of a triangle is known as triangular signal. The triangular signal is also known as hat function or tent function. A sine wave is a geometric waveform that oscillates (moves up, down, or side-to-side) periodically, and is defined by the function $y = \sin x$. In other words, it is an s-shaped, smooth wave that oscillates above and below zero. The exponential signal is a sequence of form x(n) = a^n for all n. When the value of a>1, the sequence grows exponentially and when the value is 0<a<1, the sequence decays exponentially. Note also that when a<0, the discrete-time exponential signal takes alternating signs.

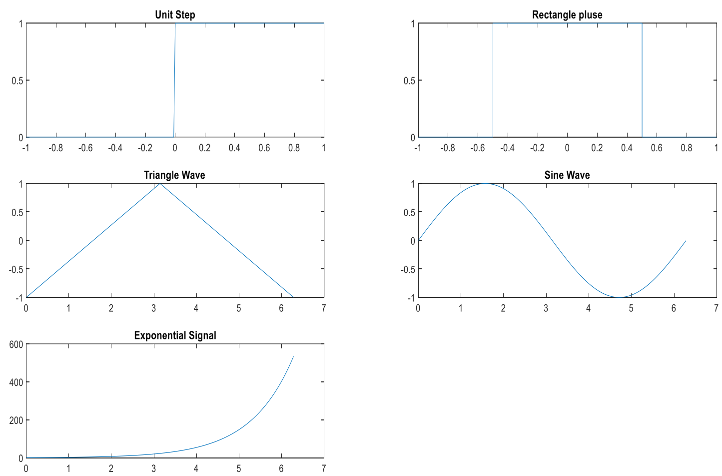
MATLAB CODE :

```
% Unit Step
t = (-1:0.01:1)';
%impulse = t==0;
unitstep = t>=0;
subplot(3,2,1)
plot(t,unitstep)
title('Unit Step');

% Rectangle
r1 = rectangularPulse(-1,1,-2:2);
syms x                              % Create symbolic scalar variables
subplot(3,2,2)
fplot(rectangularPulse(x), [-1 1])
title('Rectangle pluse');

% Generate triangular wave
t1 = 0:0.01:2*pi;
triangular_wave = sawtooth(t1, 0.5);
subplot(3,2,3)
plot(t1,triangular_wave)
title('Triangle Wave');

%sine wave
% Define the time vector
t2 = 0:0.01:2*pi; % Time from 0 to 2*pi with step size 0.01
% Generate sine wave
sin_wave = sin(t2);
subplot(3,2,4)
plot(t2, sin_wave);
title('Sine Wave');

%exponential signal
t3 = 0:0.01:2*pi; % Time from 0 to 2*pi with step size 0.01
% Generate sine wave
exp_wave = exp(t3);
subplot(3,2,5)
plot(t3, exp_wave);
title('Exponential Signal');
```

PROCEDURE:

1. Use the editor for entering the code
2. Save the file as .m and execute the code by clicking Run icon
3. Observe the waveform

EXPECTED WAVEFORM:



RESULT: Waveforms are observed and recorded.

VIVA QUESTIONS:

1. What is a step signal?
2. Define a rectangular pulse
3. Generate a 50Hz Sine wave
4. Generate a 50 Hz triangular wave
5. Write a code to generate and exponentially decaying signal.

References:

1. b061322022d65a4bc2ed2e2d3c54d3c5_MITRES_6_007S11_lec02.pdf
2. Exponential - MATLAB exp - MathWorks India

Experiment no. 2: Illustration of signal representation in time and frequency domains for a rectangular pulse.

AIM: To illustrate of signal representation in time and frequency domains for a rectangular pulse.

APPARATUS: Matlab Software

THEORY: The Rect Function is a function which produces a rectangular-shaped pulse with a width of 1 centered at t = 0. The Rect function pulse also has a height of 1. The Sinc function and the rectangular function form a Fourier transform pair. The Fourier Transform of our rectangular pulse is simply a sinc function parameterized by the width of our rectangle. There are three parameters that define a rectangular pulse: its height A, width T in seconds, and center t_o.

Mathematically, a rectangular pulse delayed by t_0 seconds is defined as

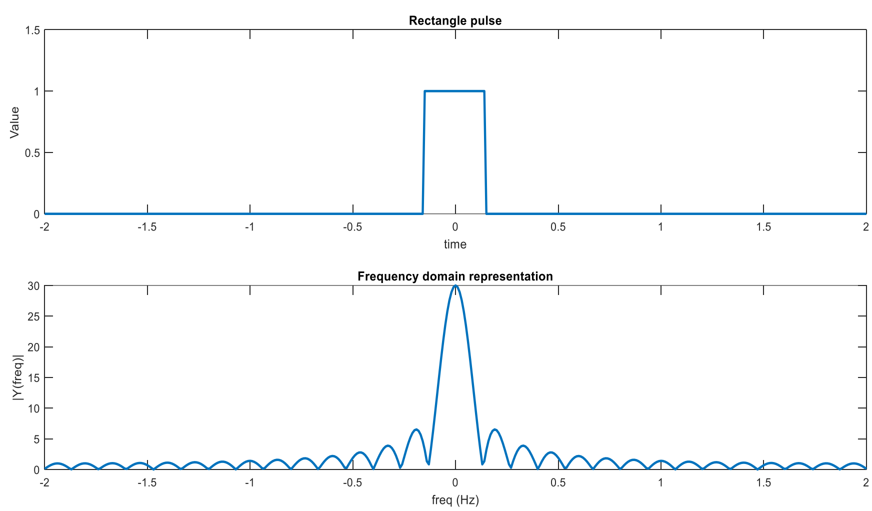
$$g(t-t_0) = A\,rect\left(\frac{t-t_0}{T}\right) = \begin{cases} A & \left|\frac{t-t_0}{T}\right| \leq \frac{1}{2} \\ 0 & otherwise \end{cases}$$

and its Fourier transform or spectrum is defined as $G(f) = AT\,sinc(\pi f T)\exp(-i2\pi f t_0)$. As the pulse becomes flatter (i.e., the width of the pulse increases), the magnitude spectrum loops become thinner and taller. In other words, the zeros (the crossings of the magnitude spectrum with the x axis) move closer to the origin. In the limit, as T becomes very large, the magnitude spectrum approaches a Dirac delta function located at the origin. As the height of the pulse become larger and its width becomes smaller, it approaches a Dirac delta function and the magnitude spectrum flattens out and becomes a constant of magnitude 1 in the limit. As t_0 changes, the pulse shifts in time, the magnitude spectrum does not change, but the phase spectrum does. Notice a 180º phase shift at each frequency defined by k/T , where k is an integer other than zero, and T is the pulse duration. These frequencies are the zeros of the magnitude spectrum.

Matlab Code:

```
a = 0.3;
t=[-2:0.01:2];
x = rectpuls(t, a); %rectangle pulse

subplot(2,1,1);    %subplot for rectangle pulse
plot(t,x,LineWidth=2);
axis([-2 2 0 1.5]);
title("Rectangle pulse");
xlabel('time');
ylabel('Value');

X=fft(x);

subplot(2, 1, 2);     %subplot for Frequency signal
plot(t,fftshift(abs(X)),LineWidth=2);
title("Frequency domain representation");
xlabel('freq (Hz)');
ylabel('|Y(freq)|');
```

EXPECTED WAVEFORM:



RESULT: The code is successfully executed and waveform is observed in the figure window.

VIVA QUESTIONS:

1. What is the change in the sinc pulse when you vary the width of the rectangular pulse?
2. Write the mathematical expression for rectangular pulse
3. Evaluate the frequency domain samples for a rect pulse (at least 5).
4. Find the magnitude and phase response for rect pulse.

REFERENCE:

1. Fourier Transform of Rectangular Function (tutorialspoint.com)
2. PulseAnalysis_mod.pdf (nmt.edu)
3. Rectangular Pulse and Its Fourier Transform - Wolfram Demonstrations Project

Experiment 3 : Amplitude Modulation and demodulation: Generation and display the relevant signals and its spectrums

AIM : To study the function of Amplitude Modulation & Demodulation

THEORY : Modulation is defined as the process of changing the characteristics (Amplitude, Frequency or Phase) of the carrier signal (high frequency signal) in accordance with the intensity of the message signal (modulating signal). Amplitude modulation is defined as a system of modulation in which the amplitude of the carrier is varied in accordance with amplitude of the message signal(modulating signal).

The message signal is given by the expression.

Em(t) =Em cosWmt

Carrier voltage

Ec(t)= Ec cosWct

E(t)=Ec + KaEm cosWmt

The amplitude modulated voltage is given by E=E(t) cosWct

E= (Ec+KaEm cosWmt) cosWct.

E= (1+KaEm/Ec cosWmt) Ec cosWct

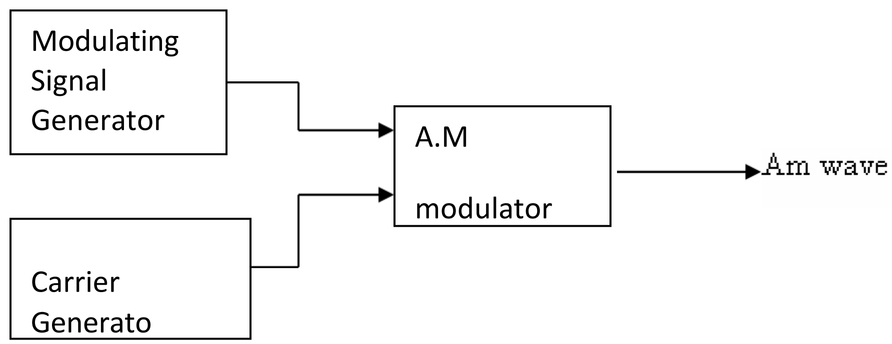
E= Ec(1+Ma cosWmt)cosWct

Ma=KaEm/Ec

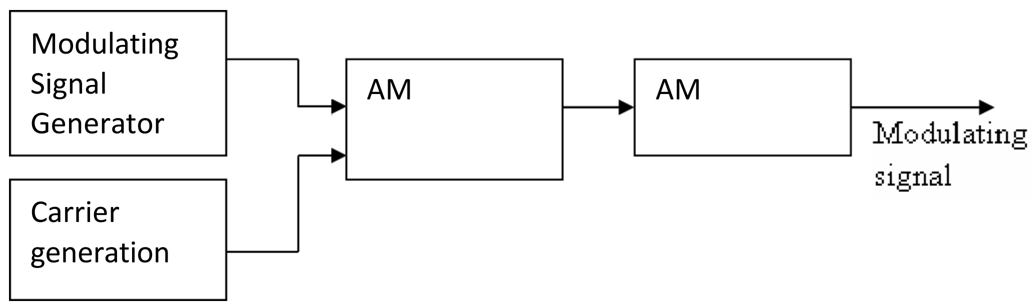
100* Ma gives the percentage of modulation

BLOCK DIAGRAM:

Modulation



DEMODULATION



PROGRAM:

AM Modulation and Demodulation

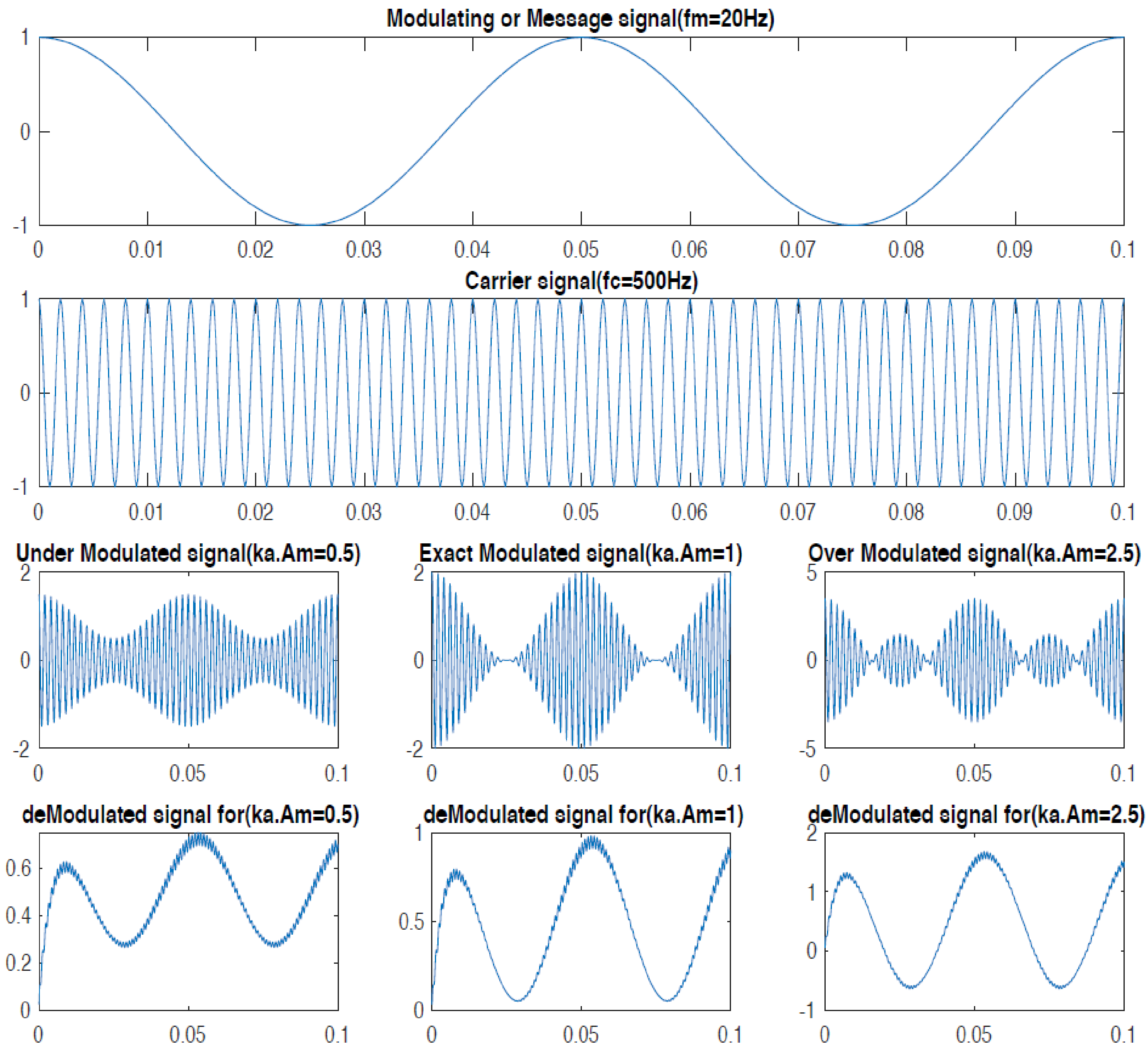
```
close all
clear all
clc
fs=8000;
fm=20;
fc=500;
Am=1;
Ac=1;
t=[0:0.1*fs]/fs;
m=Am*cos(2*pi*fm*t);
c=Ac*cos(2*pi*fc*t);
ka=0.5;
u=ka*Am;
s1=Ac*(1+u*cos(2*pi*fm*t)).*cos(2*pi*fc*t);
subplot(4,3,1:3);
plot(t,m);
title('Modulating or Message signal(fm=20Hz)');
subplot(4,3,4:6);
plot(t,c);
title('Carrier signal(fc=500Hz)');
subplot(4,3,7);
plot(t,s1);
title('Under Modulated signal(ka.Am=0.5)');
Am=2;
ka=0.5;
u=ka*Am;
s2=Ac*(1+u*cos(2*pi*fm*t)).*cos(2*pi*fc*t);
subplot(4,3,8);
plot(t,s2);
title('Exact Modulated signal(ka.Am=1)'); Am=5;
ka=0.5;
u=ka*Am;
```

```
s3=Ac*(1+u*cos(2*pi*fm*t)).*cos(2*pi*fc*t);
subplot(4,3,9);
plot(t,s3);
title('Over Modulated signal(ka.Am=2.5)');
r1= s1.*c;
[b a] = butter(1,0.01);
mr1= filter(b,a,r1);
subplot(4,3,10);
plot(t,mr1);
title(' deModulated signal for(ka.Am=0.5)');
r2= s2.*c;
[b a] = butter(1,0.01); mr2= filter(b,a,r2);
subplot(4,3,11);
plot(t,mr2);
title(' deModulated signal for(ka.Am=1)'); r3= s3.*c;
[b a] = butter(1,0.01); mr3= filter(b,a,r3);
subplot(4,3,12);
plot(t,mr3);
title(' deModulated signal for(ka.Am=2.5)');
```

Viva Questions:

1. Define modulation & demodulation?
2. Define AM and draw its spectrum?
3. Draw the phase’s representation of an amplitude modulated wave?
4. Give the significance of modulation index?
5. What are the different degrees of modulation?
6. What are the limitations of square law modulator?
7. Compare linear and nonlinear modulators?
8. Compare base modulation and emitter modulation?
9. Explain how AM wave is detected?
10. Define detection process?

EXPECTED WAVEFORM



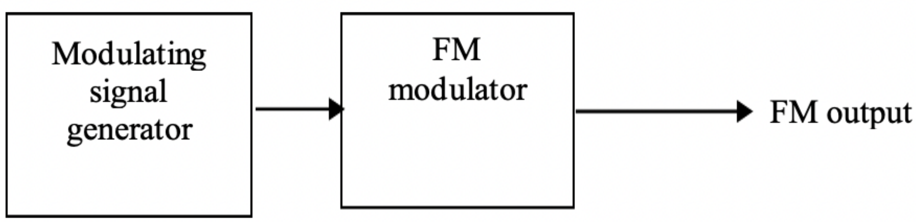
Experiment 4: Frequency Modulation and demodulation: Generation and display the relevant signals and its spectrums

AIM : To study the process of frequency modulation and demodulation

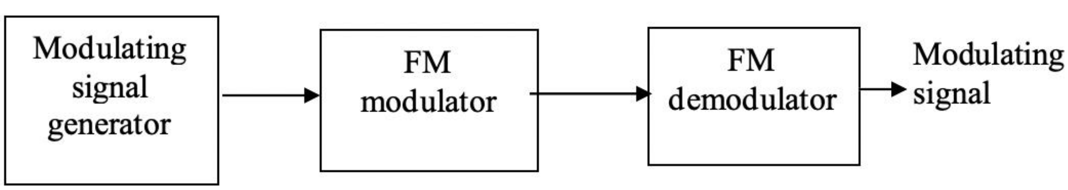
THEORY: The modulation system in which the modulator output is of constant amplitude, in which the signal information is super imposed on the carrier through variations of the carrier frequency. The frequency modulation is a non-linear modulation process. Each spectral component of the base band signal gives rise to one or two spectral components in the modulated signal. These components are separated from the carrier by a frequency difference equal to the frequency of base band component. Most
Importantly the nature of the modulators is such that the spectral components which produce decently on the carrier frequency and the base band frequencies. The spectral components in the modulated wave form depend on the amplitude. The modulation index for FM is defined as Mf= max frequency deviation/ modulating frequency.

BLOCK DIAGRAM:

Modulation



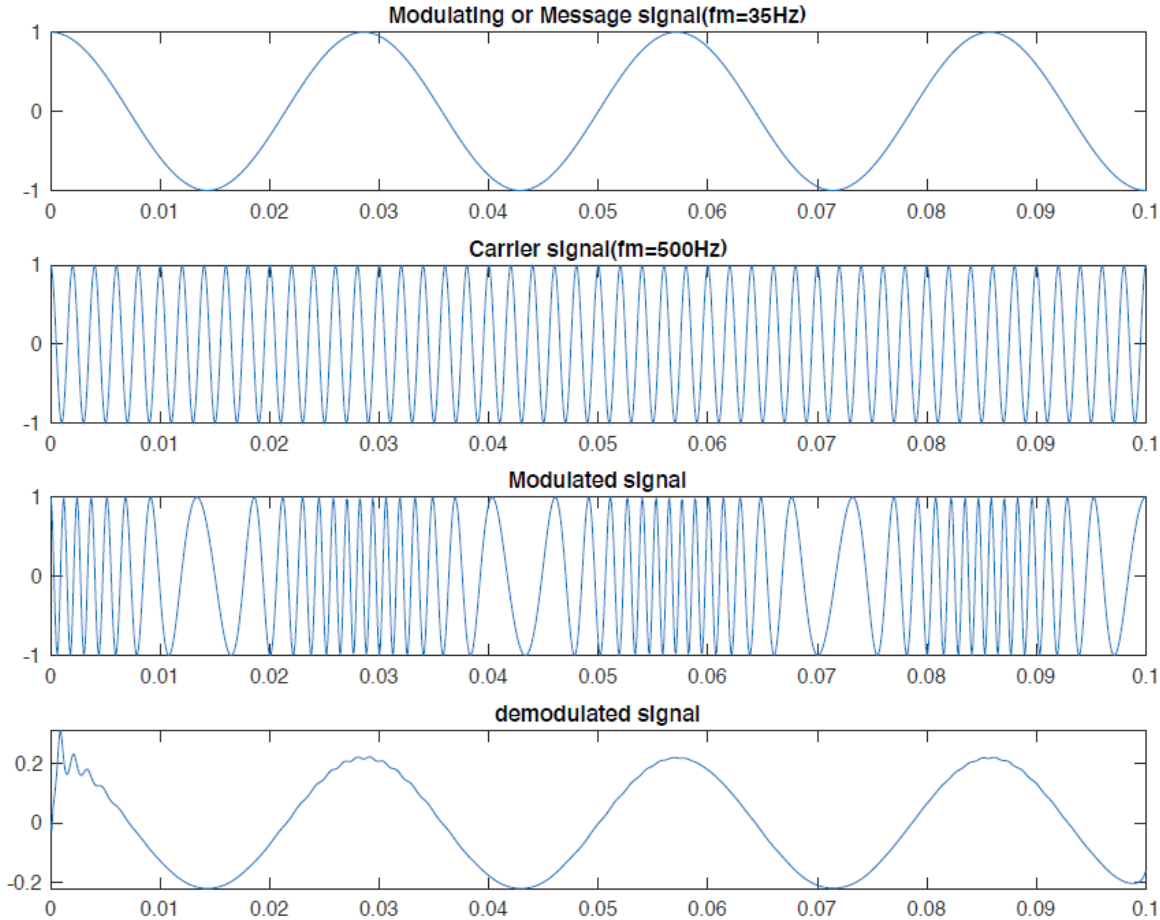
Demodulation



Program

```
close all
clear all
clc
%fm=35HZ,fc=500HZ,Am=1V,Ac=1V,B=10
fs=10000;
Ac=1;
Am=1;
fm=35;
fc=500;
B=10;
t=(0:.1*fs)/fs;
wc=2*pi*fc;
wm=2*pi*fm;
m_t=Am*cos(wm*t);
subplot(4,1,1);
plot(t,m_t);
title('Modulating or Message signal(fm=35Hz)'); c_t=Ac*cos(wc*t);
subplot(4,1,2);
plot(t,c_t);
title('Carrier signal(fm=500Hz)'); s_t=Ac*cos((wc*t)+B*sin(wm*t));
subplot(4,1,3);
plot(t,s_t);
title('Modulated signal');
d=demod(s_t,fc,fs,'fm');
subplot(4,1,4);
plot(t,d);
title('demodulated signal');
```

EXPECTED WAVEFORM



Viva Questions:

1. Define FM & PM.
2. What are the advantages of Angle modulation over amplitude modulation?
3. What is the relationship between PM and FM?
4. With a neat block diagram explain how PM is generated using FM

Experiment no 5: Sampling and reconstruction of low pass signals. Display the signals and its spectrum

AIM: To verify the Sampling theorem for a low pass signal and to reconstruct the same

THEORY: A continuous time signal can be represented in its samples and can be recovered back when sampling frequency f_s is greater than or equal to the twice the highest frequency component of message signal. Sampling is a process of converting a continuous time signal (analog signal), x(t) into a discrete time signal x[n], which is represented as a sequence of numbers. Converting back the discrete time samples x[n] into analog (resulting in process of reconstruction) which is **$f_s \geq 2f_m$.**

Right sampling: A minimum sampling rate is required so as to recover the original signal from its samples. This is referred as right or critical sampling.

Under sampling: When the sampling rate is less than the Nyquist rate, there is overlap of the frequency components and the higher frequency components and the other higher frequency terms of the continuous time signals appear in the low frequency region. This is referred to as aliasing

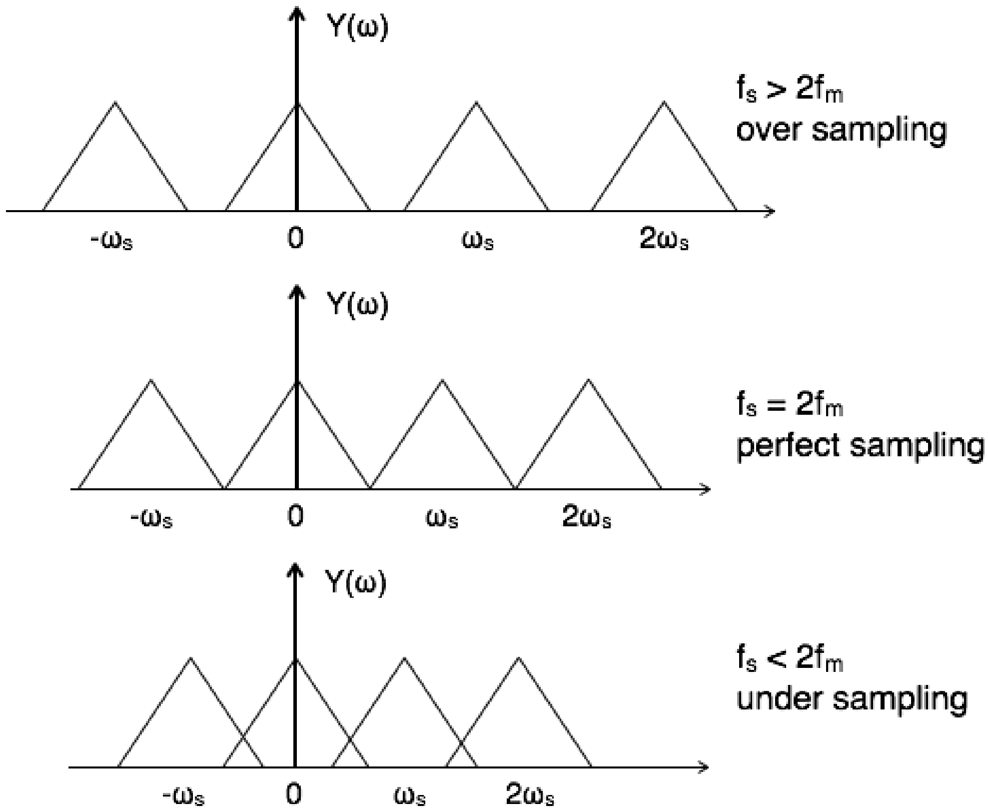
Over sampling: If the sampling rate is greater than the Nyquist rate, there is a lot of redundancy in the discrete time signal. This is referred to as over sampling.

The sampling frequency fs determines the spacing between samples. The minimum sampling rate of 2w samples per second, for a signal bandwidth of 'w'Hz is called Nyquist rate. $f_s = 2f_m$.

Right Sampling $\rightarrow f_s = 2f_m$

Under sampling $\rightarrow f_s \leq 2f_m$

Over sampling $\rightarrow f_s \geq 2f_m$



PROGRAM :

close all;

clear; tfinal=0.01;

t=0:0.00001:tfinal; **% define time vector .01/.00001=1000 sample pts**

Fa=input('enter the analog signal frequency');//Audio signal range which is from 300Hz to 3.4KHz(which is a low pass signal)

%Generate and plot analog signal xt=cos(2*pi*Fa*t);

subplot(4,1,1);

plot(t,xt,'r');

title('analog signal');

xlabel('time');

ylabel('amplitude');

%simulate condition for under sampling i.e., fs1<2*fd and plot the samples

```
Fs1=1.3*Fa;
n1=0:1/Fs1:tfinal;
x1n=cos(2*pi*Fa*n1);
subplot(4,1,2); stem(n1,x1n,'b');
 title('undersampled signal');
xlabel('discrete time');
ylabel('amplitude');
```

%condition for Nyquist sampling fs2=2*fd and plot the samples

```
Fs2=2*Fa;
 n2=0:1/Fs2:tfinal;
 x2n=cos(2*pi*Fa*n2);
subplot(4,1,3);
stem(n2,x2n,'r');
title('critically sampled signal');
xlabel('discrete time');
ylabel('amplitude');
```

%condition for oversampling

```
Fs3=10*Fa;
n3=0:1/Fs3:tfinal;
x3n=cos(2*pi*Fa*n3);
subplot(4,1,4);
stem(n3,x3n,'g');
title('oversampled signal');
xlabel('discrete time');
ylabel('amplitude');
```

%Plot the analog signal

```
figure
subplot(4,1,1);
 plot(t,xt);
title('analog signal');
xlabel('time');
ylabel('amplitude')
```

%Reconstruction from under sampled signal xr1=interp(x1n,2);

```
nr1=interp(n1,2);
```

```
subplot(4,1,2);
plot(nr1,xr1);
title('reconstruction from undersampled signal');
xlabel('time');
ylabel('amplitude');

%Reconstruction from critically sampled signal xr2=interp(x2n,2);
nr2=interp(n2,2);
subplot(4,1,3);
plot(nr2,xr2);
title('reconstruction from critically sampled signal');
xlabel('time');
ylabel('amplitude');

%Reconstruction from over sampled signal
xr3=interp(x3n,2); nr3=interp(n3,2);
subplot(4,1,4); plot(nr3,xr3);
title('reconstruction from oversampled signal');
xlabel('time');
ylabel('amplitude');
```

Output

Enter the analog signal frequency1000

The relevant outcomes are shown in fig 1.1 and Fig 1.2

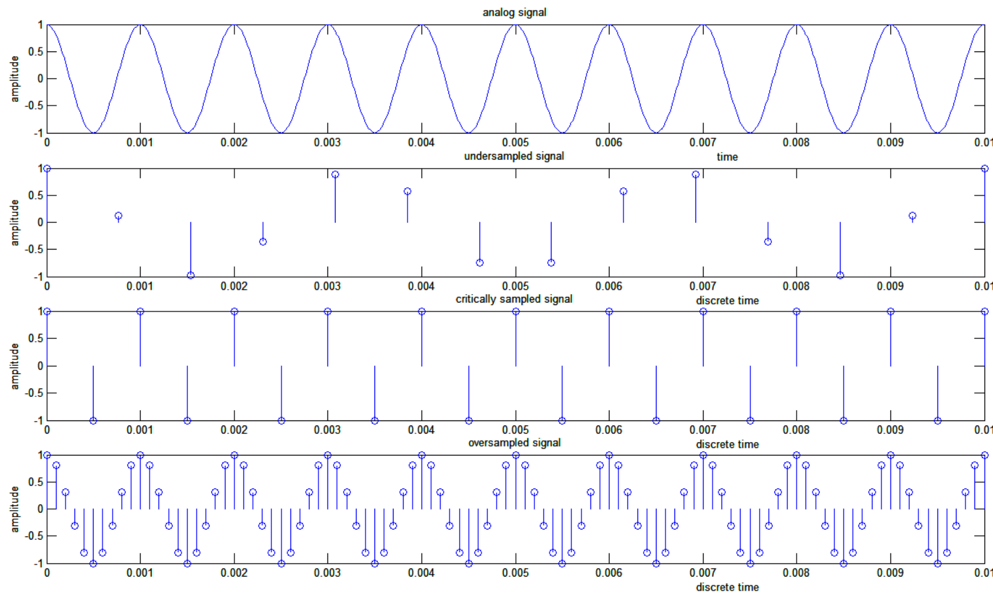


Fig.1.1 Plots of sampled sine wave of 1000Hz

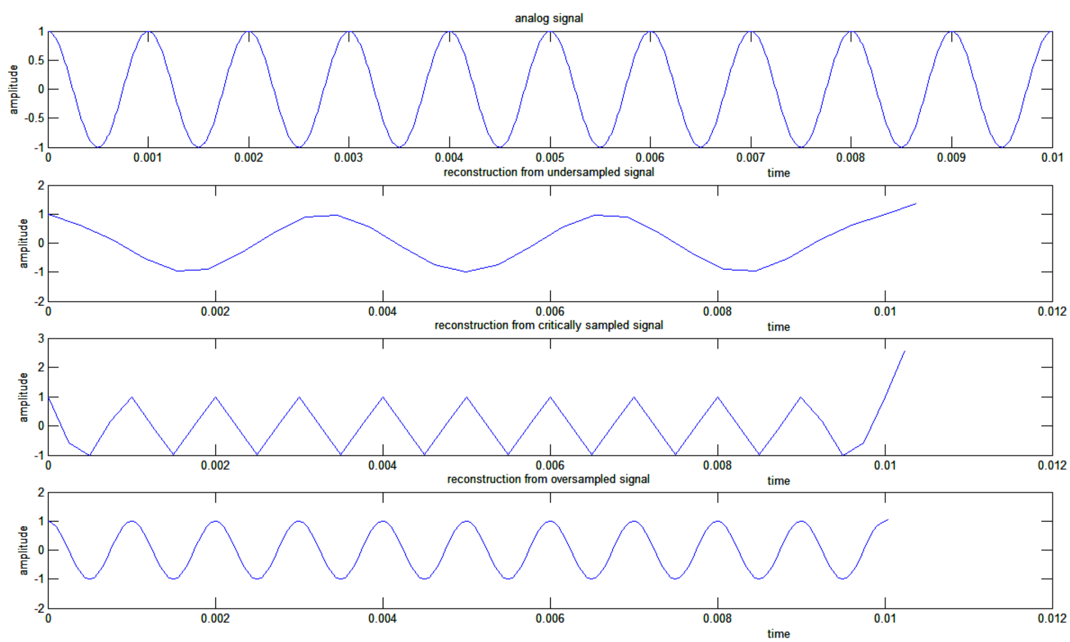


Fig.1.2 Plots of the reconstructed signal

Viva Questions:

a. State sampling theorem?

b. What do you mean by process of reconstruction?

c. What do you mean Aliasing? What is the condition to avoid aliasing for sampling?

d. What is an antialiasing filter?

e. Write the conditions of sampling.

f. Explain the statement, t= 0:0.000005:0.05. What does colon (:) and semicolon (;) denotes.

g. What is the relation between the continuous time angular frequencies and normalized discrete time angular frequency?

h. What is the relation between the bandwidth of continuous time signal and the bandwidth of corresponding discrete time signal after sampling process?

i. What are Nyquist frequency, Nyquist rate, Nyquist band?

j. What is the bandwidth of the continuous time signal?

Experiment 6: Time Division Multiplexing and de-Multiplexing

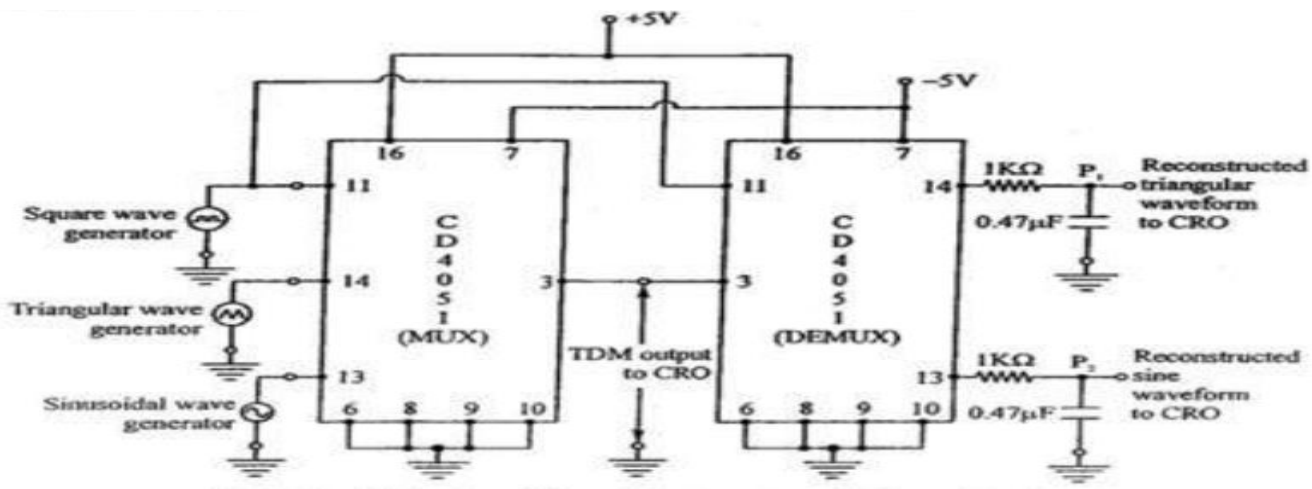
AIM: To simulate Time Division Multiplexing and Demultiplexing using MATLAB.

THEORY: More efficient communication system can be obtained if a station transmits more than one message" on the same carrier and on the same channel, or number of transmitters is transmitting simultaneously on the same channel. This process is known as "Multiplexing" and has been used for many years in long distance telephony. Multiplex transmissions have been used in commercial communications, not only for voice channels (wireless), but also for facsimile. Multiplexing has also been used since a long time for broadcasting. In practical it is to combine a set of low-bit-rate streams, each with a fixed and pre-defined bit rate, into a single high-speed bit stream that can be transmitted over a single channel. This technique is called time division multiplexing (TDM)
and has many applications, including wireline telephone systems and some cellular telephone systems. The main reason to use TDM is to take advantage of existing transmission lines. It would be very expensive if each low-bit-rate stream were assigned a costly physical channel (say, an entire fiber optic line) that extended over a long distance.

TDM is a technique used for transmitting several message signals over a communication channel by dividing the time frame into slots, one slot for each message signal. This is a digital technique in which the circuit is highly modularin nature and provides reliable and efficient operation. There is no cross talk in TDM due to circuit
non-linearity since the pulses are completely isolated. But it also has disadvantages, which include timing jitter and synchronization is required. Here, the carrier used is a square wave like in pulse modulation techniques. When the carrier is ON, one of the messages gets sampled and is output. When the carrier is OFF,another message gets sampled and is output. The amplitude of a periodic train of pulses is thus varied in proportion to the message signals. TDM provides an effective method for sharing a communication channel.

For demodulation, when the carrier is ON, one message is passed by the switching network to one of the output ports. When the carrier is OFF another message is passed to the second port. The PAM output can thus be obtained. The message signals can be reconstructed using a low pass filter. The reconstruction of message signals will be perfect if they are sampled at their respective Nyquist rates.

CIRCUIT DIAGRAM:



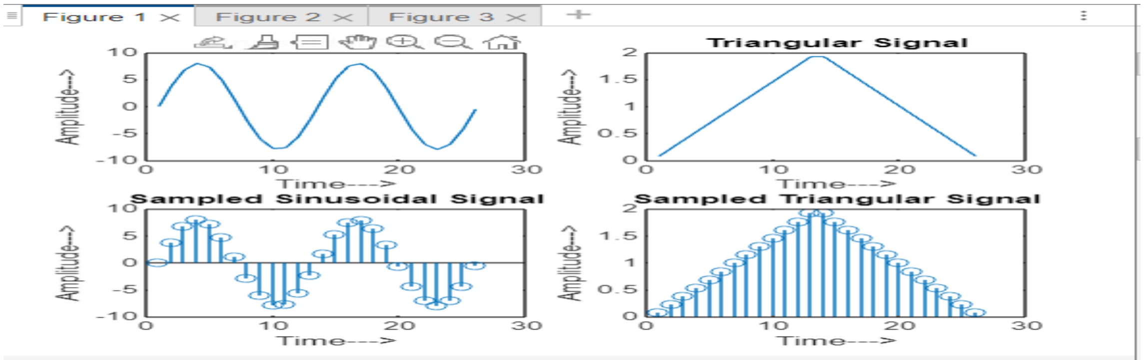
MATLAB CODE:

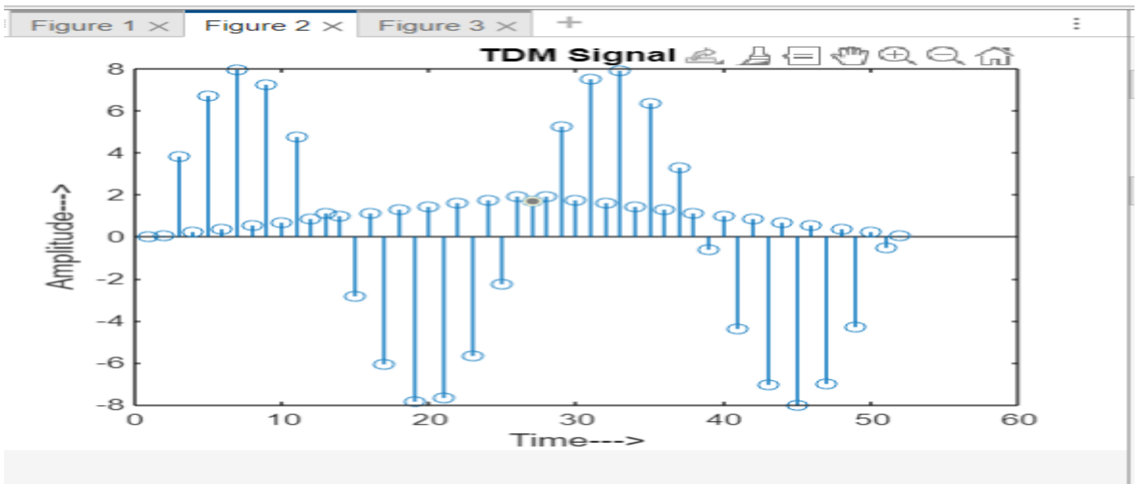
```
%TIME DIVISION MULTIPLEXING
clc;
close all;
clear all;
% Signal generation
x=0:.5:4*pi; % signal taken upto 4pi
sig1=8*sin(x); % generate 1st sinusoidal signal
l=length(sig1);
sig2=2*triang(l); % Generate 2nd triangular Signal
% Display of Both Signal
subplot(2,2,1);
```

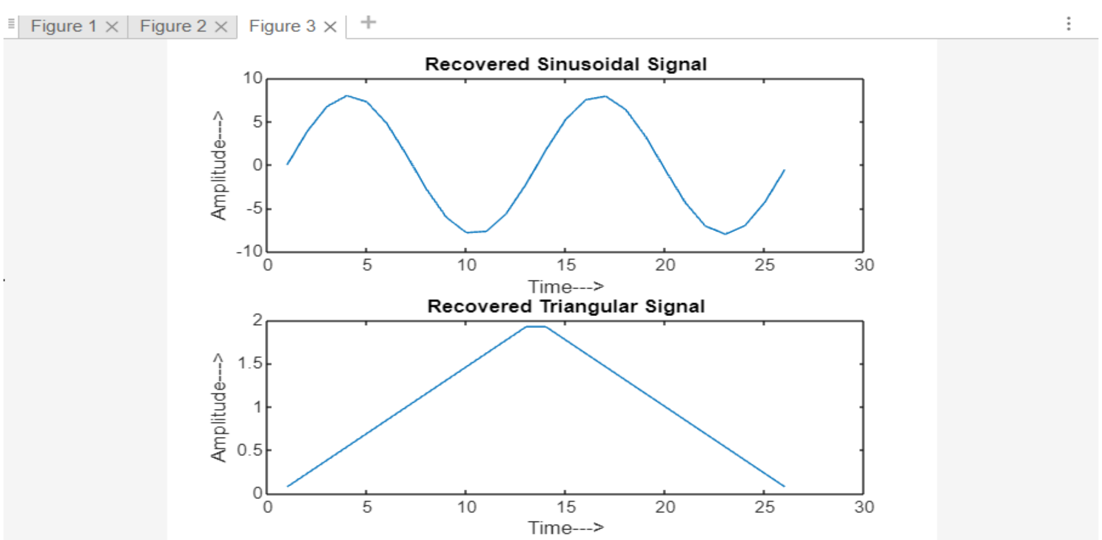
```
plot(sig1);
title('Sinusoidal Signal');
ylabel('Amplitude--->');
xlabel('Time--->');
subplot(2,2,2);
plot(sig2);
title('Triangular Signal');
ylabel('Amplitude--->');
xlabel('Time--->');
% Display of Both Sampled Signal
subplot(2,2,3);
stem(sig1);
title('Sampled Sinusoidal Signal');
ylabel('Amplitude--->');
xlabel('Time--->');
subplot(2,2,4);
stem(sig2);
title('Sampled Triangular Signal');
ylabel('Amplitude--->');
xlabel('Time--->');
l1=length(sig1);
l2=length(sig2);
for i=1:l1
sig(1,i)=sig1(i); % Making Both row vector to a matrix
sig(2,i)=sig2(i);
end
% TDM of both quantize signal
tdmsig=reshape(sig,1,2*l1);
% Display of TDM Signal
figure
stem(tdmsig);
```

```
title('TDM Signal');
ylabel('Amplitude--->');
xlabel('Time--->');
% Demultiplexing of TDM Signal
demux=reshape(tdmsig,2,l1);

for i=1:l1
signal3(i)=demux(1,i); % Converting The matrix into row vectors
signal4(i)=demux(2,i);
end
% display of demultiplexed signal
figure
subplot(2,1,1)
plot(signal3);
title('Recovered Sinusoidal Signal');
ylabel('Amplitude--->');
xlabel('Time--->');
subplot(2,1,2)
plot(signal4);
title('Recovered Triangular Signal');
ylabel('Amplitude--->');
xlabel('Time--->');
```

EXPECTED RESULT:







VIVA QUESTIONS:

1. Explain the concept of multiplexing/Demultiplexing?
2. What is TDM?
3. What is FDM?
4. Compare TDM and FDM?
5. What are the applications of TDM?
6. State the difference between Analog systems and digital systems.
7. Explain why digital systems are considered superior than Analog systems.

Experiment 7: Pulse Code Modulation Illustration :Sampling, Quantization and Encoding)

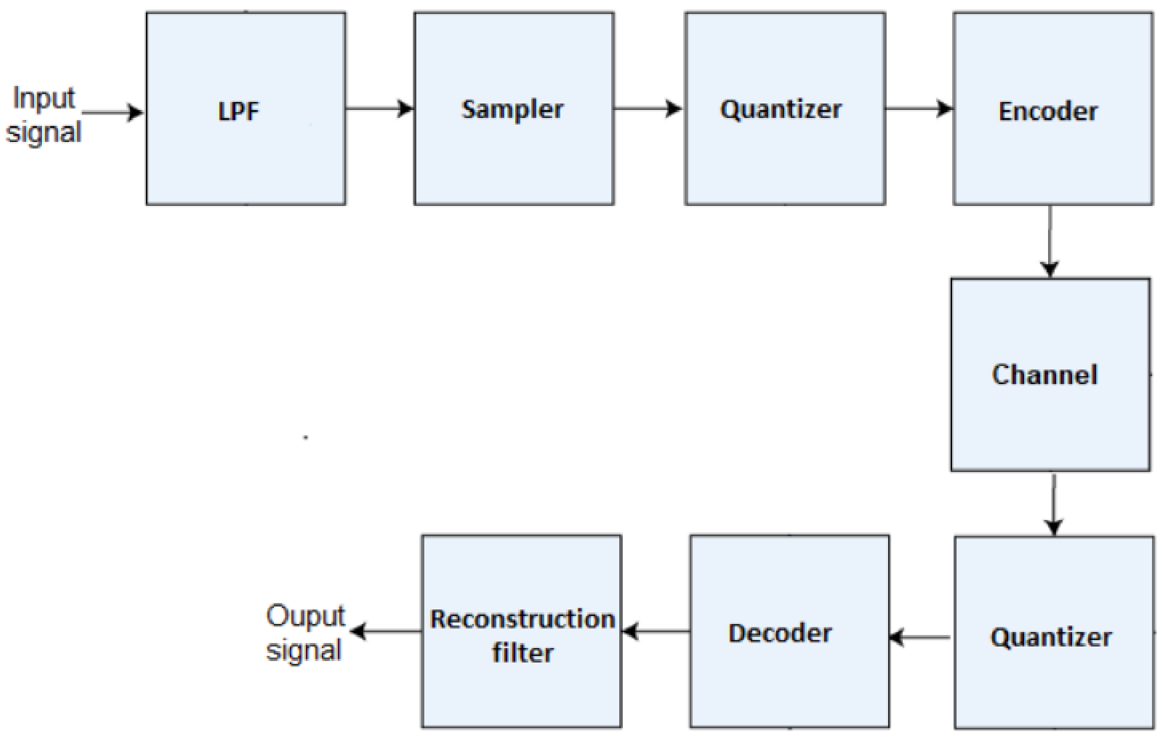
AIM: To simulate Pulse Code Modulation using MATLAB.

THEORY: Modulation is a process where different parameter (amplitude, frequency, and phase) of a carrier signal varies with the instantaneous value of the message signal. A carrier signal is a high-frequency signal, and the message signal is the original signal transmitted from the transmitter to the receiver. The message signal carries the information from one place to the other. The carrier signal is used with the message signal to make it suitable for **long-distance transmission**.

Different modulation techniques use different parameters to transmit a signal from the transmitter to the receiver. Here, we will discuss the **PCM** (Pulse Code Modulation) technique, one of the **digital modulation** methods to transmit the signal.

A PCM system converts an analog input signal to the **digital signal**, which is a combination of the binary sequence created from the binary digits **0 and 1**. An analog signal is a continuous wave, and the PCM signal is a wave with a series of digits. Thus, we can define PCM as the modulation method that transmits the pulses in the form of binary digits representing a **code number**.

BLOCK DIAGRAM:



MATLAB CODE:

```
%Pulse Code Modulation
clc;
close all;
clear all;
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); % Amplitude Of signal is 8v
% subplot(2,2,1);
```

```
% plot(x,y);grid on;
% Sampling Operation
x=0:2*pi/n1:4*pi; % n1 nuber of samples have tobe selected
s=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---
>');
% Quantization Process
vmax=8;
vmin=-vmax;
del=(vmax-vmin)/L;
part=vmin:del:vmax; % level are between vmin and vmax with difference of del
code=vmin-(del/2):del:vmax+(del/2); % Contaion Quantized valuses
[ind,q]=quantiz(s,part,code); % Quantization process
% ind contain index number and q contain quantized values
l1=length(ind);
l2=length(q);
for i=1:l1
if(ind(i)~=0) % To make index as binary decimal so started from 0 to N
ind(i)=ind(i)-1;
end
i=i+1;
end
for i=1:l2
if(q(i)==vmin-(del/2)) % To make quantize value inbetween the levels
q(i)=vmin+(del/2);
```

```
end
end
subplot(3,1,3);
stem(q);grid on; % Display the Quantize values
title('Quantized Signal');
ylabel('Amplitude--->');
xlabel('Time--->');
% Encoding Process
figure
code=de2bi(ind,'left-msb') % Cnvert the decimal to binary
k=1;5
for i=1:l1
for j=1:n
coded(k)=code(i,j); % convert code matrix to a coded row vector
j=j+1;
k=k+1;
end
i=i+1;
end
subplot(2,1,1); grid on;
stairs(coded); % Display the encoded signal
axis([0 100 -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'); % Getback the index in decimal form
q=del*index+vmin+(del/2); % getback Quantized values
subplot(2,1,2); grid on;
plot(q); % Plot Demodulated signal
title('Demodulated Signal');
```

```
ylabel('Amplitude--->');
xlabel('Time--->')
```

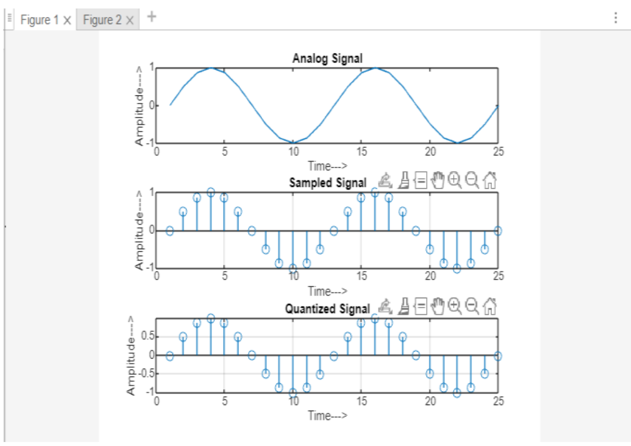
OUTPUT :

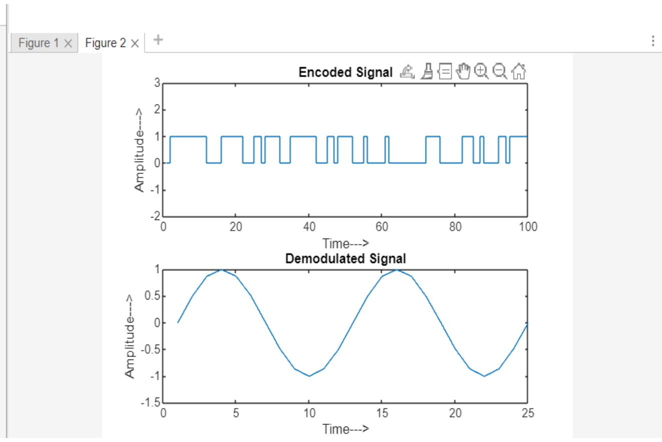
In the Command window

```
Enter n value for n-bit PCM system :
10
Enter number of samples in a period :
12
code =


     0     1     1     1     1     1     1     1     1     1
     1     0     0     0     0     1     1     1     1     1
     1     0     0     0     1     1     0     1     1     1
     1     0     0     0     1     1     1     1     1     1
     1     0     0     0     1     1     0     1     1     1
     1     0     0     0     1     0     0     0     0     0
     1     0     0     0     0     0     0     0     0     0
     0     1     1     1     1     0     0     0     0     0
     0     1     1     1     0     0     1     0     0     0
     0     1     1     0     1     1     1     1     1     1
     0     1     1     1     0     0     1     0     0     0
     0     1     1     1     0     1     1     1     1     1
     0     1     1     1     1     1     1     1     1     1
     1     0     0     0     0     1     1     1     1     1
     1     0     0     0     1     1     0     1     1     1
     1     0     0     0     1     1     1     1     1     1
     1     0     0     0     1     1     0     1     1     1
     1     0     0     0     0     1     1     1     1     1
     1     0     0     0     0     0     0     0     0     0
     0     1     1     1     1     0     0     0     0     0
```

```
0     1     1     1     0     0     1     0     0     0
0     1     1     0     1     1     1     1     1     1
0     1     1     1     0     0     1     0     0     0
0     1     1     1     1     0     0     0     0     0
0     1     1     1     1     1     1     1     1     1



Ans = 5
```

EXPECTED WAVEFORMS:





VIVA QUESTIONS:

1. Define Pulse code modulation? ...
2. How bits are needed to encode N different levels?
3. Define step size?
4. What is the max value of Quantization error?
5. What are the applications of PCM?
6. What are the disadvantages of Pulse code modulation?

Experiment no. 9: Probability density function(PDF) of Gaussian Distribution Function

AIM: To generate the Probability Density function of a Gaussian distribution function.

THEORY:

Probability density function is the function that represents the density of probability for a continuous random variable over the specified ranges.

In probability theory, a probability density function (PDF) is used to define the random variable's probability coming within a distinct range of values, as opposed to taking on any one value. The function explains the probability density function of normal distribution and how mean and deviation exists.

Normal distribution, also known as the Gaussian distribution, is a probability distribution that is symmetric about the mean, showing that data near the mean are more frequent in occurrence than data far from the mean. The normal distribution appears as a "bell curve" when graphed.

PROGRAM:

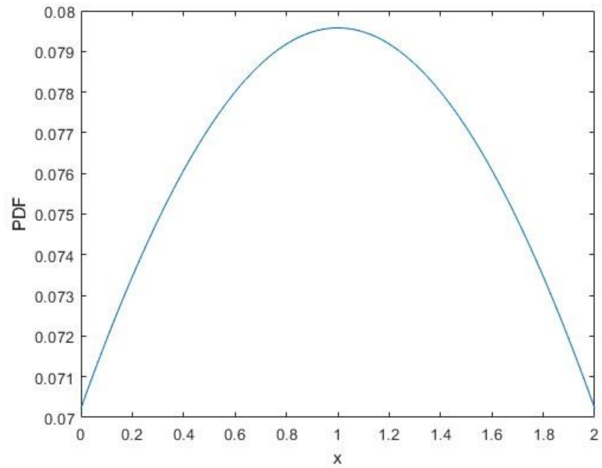
Step 1 : Generate the Gaussian Distribution function

```
 clear all;
 close all
 clc;
x= [0:0.01 : 2]
m=1 ;
sd = 2;
g= normpdf(x,m,sd);
plot (x,g);
xlabel('x') ;++
ylabel('Amplitude')
title ('Gaussian Distribution Function)
```

Step 2 : Generate the PDF of a Gaussian distribution function

```
y = 1/(2*pi*sd)*exp(-(x-m).^2/(2*sd^2));
plot(x,y);
```

EXPECTED OUTPUT:



VIVA QUESTIONS:

1. What is PDF?
2. What are random variables?
3. What are the different distributions?

REFERENCE:

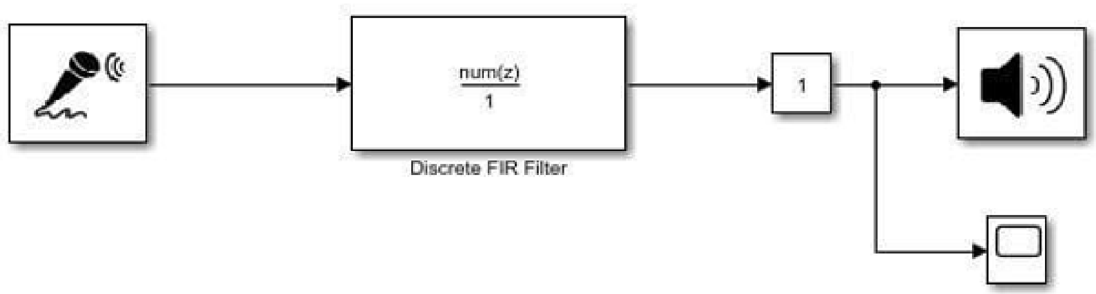
1. Simon Haykin & Michael Moher,Communication Systems, 5th Edition, John Wiley,2010.

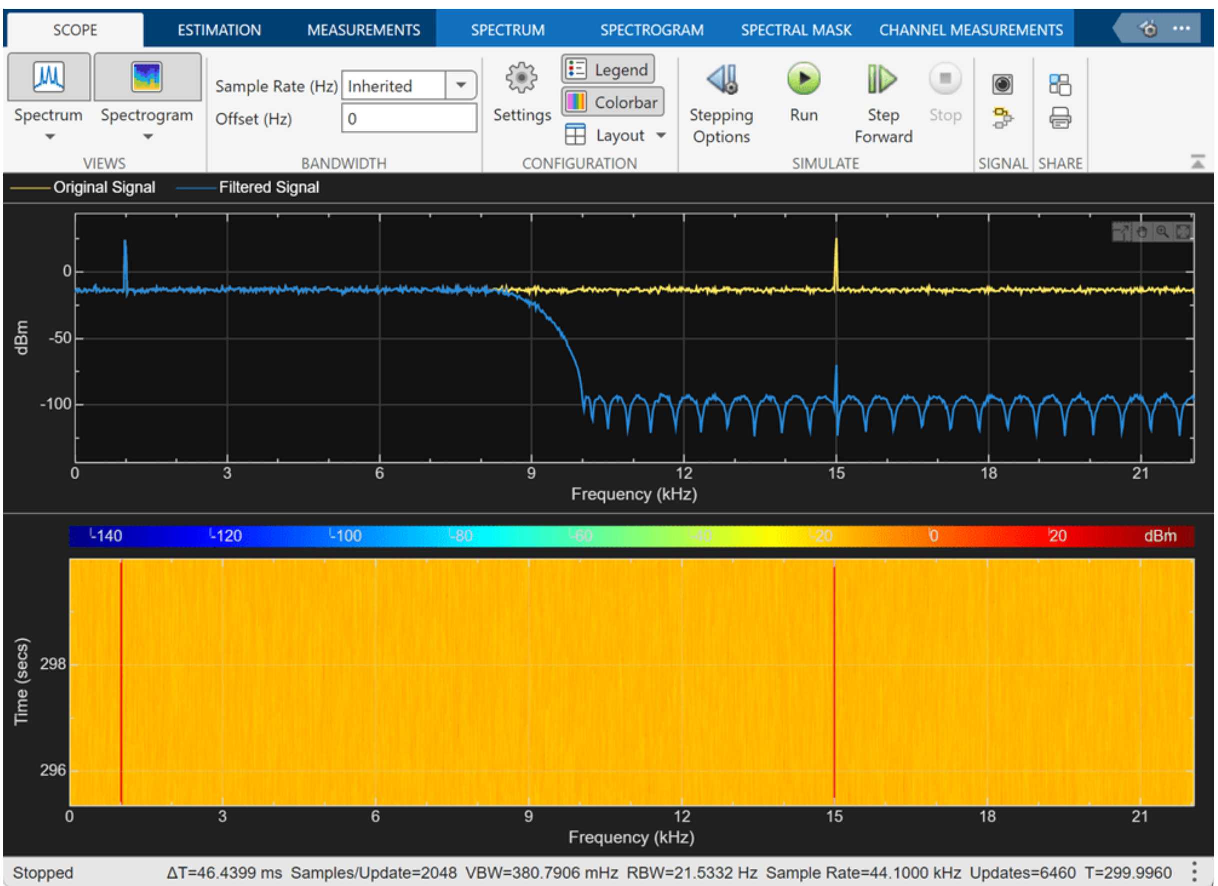
10. Display the signal and its spectrum of an audio signal

AIM: To read and display the an audio signal with a spectrum analyzer/ frequency plot

MATLAB TOOL: audio Toolbox

Simulink block:





a) **Matlab Syntax to read audio file:** First read the audio signal using wavread() function. as a result of reading, the signal will be vectorized. after that, you should use fft() function to get the fourier transform of vectorized signal. at the end plot() the fourier transform of signal

Read Complete Audio File

Create a WAVE file from the example file `handel.mat`, and read the file back into MATLAB®.

Create a WAVE (`.wav`) file in the current folder.

```
load handel.mat

filename = 'handel.wav';
audiowrite(filename,y,Fs);
clear y Fs
```

Read the data back into MATLAB using `audioread`.

```
[y,Fs] = audioread('handel.wav');
```

Play the audio.

```
sound(y,Fs);
```

b) **Program to display frequency and spectrum of audio signal:**

```
[xn fs]=wavread('signal_name.wav'); %can also use audioread command for .mp3 file
nf=1024; %number of point in DTFT
Y = fft(xn,nf);
f = fs/2*linspace(0,1,nf/2+1);
plot(f,abs(Y(1:nf/2+1)));
plot(psd(spectrum.periodogram,xn,'Fs',fs,'NFFT',length(xn)));
```