Ec lab journal

Name: Jigar Nainuji

Roll no: U18CO039

B.Tech 2nd year

Branch: Computer

Index

Sno.	Practical Name	
1.	Introduction to MATLAB and plot basic functions and signals like sine, cosine, tangent, unit impulse, unit step, unit ramp, and periodic signals like impulse train, square wave and triangular wave.	
2.	To perform Sampling and Reconstruction of signal (Hardware) and obtain its waveforms. Also Verify the Nyquist Criteria.	
3.	Write a program to compute exponential fourier series coefficients and plot the magnitude and phase spectrum. Also, plot the periodic signal using fourier series.	
4.	To perform Amplitude modulation and demodulation (Hardware) and obtain its waveforms. Also calculate the three different modulation indices.	
5.	a) To perform Pulse Amplitude Modulation: (Hardware) a. To modulate signal by Pulse Amplitude Modulation Scheme using Natural & Flat top sampling. b. To demodulate signal by Pulse Amplitude Modulation Scheme using Sample & Hold, Flat Top. c. Verify the sampling theorem by changing modulating & carrier frequency b) To perform Pulse Position Modulation and Demodulation and obtain its waveforms. (Hardware) c) To perform Pulse Width Modulation and Demodulation and obtain its waveforms(Hardware)	
6.	To study frequency modulation and demodulation and observe the waveforms. a) Observe the spectra of FM signal in labAlive virtual communication lab and Calculate the modulation index for FM b) To perform FM transmission via virtual lab labAlive for the audio signal c) To perform FM reception via virtual lab labAlive for the obtained recorded signal	
7.	a) To Generate and demodulate an amplitude shift keying(ASK) signal in MATLAB. b) To study Frequency Shift Keying (FSK) Modulation in MATLAB Simulink.	

	c) To study Binary Phase Shift Keying (BPSK) Modulation in MATLAB Simulink.	
8.	Write a program for amplitude modulation and demodulation considering input as sinusoidal wave and plot the various signals in time domain and frequency domain (MATLAB)	
9.	Write a MATLAB code to modulate and demodulate the given signal by Delta Modulation Technique.	
10.	Write a program for frequency modulation and demodulation considering input as sinusoidal wave and plot the various signals in time domain and frequency domain in MATLAB	
11.	To find the Numerical Aperture of given optical fiber in Virtual LAB.	

Write-up format for experiments using simulation tool

1. Include Introduction to MATLAB
    a. Basic commands- How to define variables, constants, integers, strings etc.
    b. Matrices
    c. Arithmetic operators
    d. Logical operators
    e. Looping, Branching and controlling
    f. Functions
    g. Plotting commands
2. Keep the font as times new roman for title(size 16) ,heading (size 14) and subheadings (size 12).
3. For code 'Monospaced' font to be used with size 11.
4. Include
    a. Aim (Descriptive)
    b. Theory- Basic Definition, signal representation (if any), equation, etc.
    c. Flowchart or Algorithm
    d. Code with proper margin and spaces (don't type messy). Also include comments for every commands and syntax you are using.
    e. Result and Observations
    f. Conclusion.
5. Include date (on every new experiment) on top right and page no. on the bottom middle with page layout as 'normal' (in header).
6. In footer include "EC208 – Communication System, Electronics engineering Department, SVNIT, Surat-07"
7. For hardware experiments follow the format as it has been in docs uploaded on moodle.
8. In Results include figures. It should be proper in size with labels and titles.
9. To save figure, in figure window go in file- export to- jpeg. Save in jpeg format.

INTRODUCTION TO MATLAB

Date: *08-01-2020*

Basic Commands in MATLAB:

1. **clc**-clear the command window. *Probably the best command of all time to reduce user anxiety.*
2. **clear all**-clear all variables in your workspace. *Trust me you'll be using this one.*
3. **plot()**-Plot curves by inserting vectors of the same length in the function. *Amazing.*
4. **subplot()**-Plot multiple figures in one window. *Godlike.*
5. **axis([**-1 1 -1 1**])**-Set the minimum x and y axis of your plot. *This can be set manually too.*
6. **legend('**string**')**-Name the data series of your figure. *Pretty sweet!*
7. **shg**-Display the figure window instantaneously. *No more frustrating clicks!*
8. **run** -directory/scriptnameRun another script within your script. *Helps you keep it clean and organised.*
9. **load** -directory/workspacefileLoad variables saved from the workspace directly to your script.
10. **help** -function access the documentation on the usage of the function directly in your command window.
11. **length(**vector**)**Returns the length of a vector. *Very useful when using for loops.*
12. **size(**matrix**)**Returns the size of a matrix.
13. **ones()**Create a vector or a matrix of ones. *This is awesome, forget building a vector like this: x=[1 1 1 1 1], just use ones(1,5) and you're good.*
14. **zeros()**Creates a vector or a matrix of zeros. *Same principle here but with zeros!*
15. **rand()**Create a random vector or a matrix. *So convenient when you'r just goofing around or testing basic things.*
16. **disp('**string**')**Display a string in the command window.
17. **tic**Start the invisible stopwatch.

18. **toc**Stop the stopwatch and get information on the time elapsed! *Useful if you want to know the duration of a simulation for example!*

19. **input()**Prompt the user for an input. *Can be very useful when using if conditions.*

Matrices:

A matrix is a two-dimensional array of numbers.

In MATLAB, you create a matrix by entering elements in each row as comma or space delimited numbers and using semicolons to mark the end of each row.

For example, let us create a 4-by-5 matrix

```
a = [ 1 2 3 4 5; 2 3 4 5 6; 3 4 5 6 7; 4 5 6 7 8]
```

Arithmetic Operators:

The most common arithmetic operations in MATLAB are

Operation	Symbol
Addition	+
Subtraction	-
Multiplication	*
Division	/
Powers	^

This is just like on a calculator, in spreadsheets (Excel) and most programming languages.

Logical Operators:

The logical data type represents true or false states using the numbers 1 and 0, respectively. Certain MATLAB® functions and operators return logical values to indicate fulfillment of a condition. You can use those logical values to index into an array or execute conditional code. Like & , | , ~ ,….

Looping , Branching and Controlling:

MATLAB Language Syntax:

if, elseif, else	Execute statements if condition is true
for	for loop to repeat specified number of times
parfor	Parallel for loop
switch, case, otherwise	Execute one of several groups of statements
try, catch	Execute statements and catch resulting errors
while	while loop to repeat when condition is true
break	Terminate execution of for or while loop
continue	Pass control to next iteration of for or while loop
end	Terminate block of code or indicate last array index
pause	Stop MATLAB execution temporarily
return	Return control to invoking script or function

With loop control statements, you can repeatedly execute a block of code. There are two types of loops:

- for statements loop a specific number of times, and keep track of each iteration with an incrementing index variable.

  For example, pre-allocate a 10-element vector, and calculate five values:

```
x = ones(1,10);
for n = 2:6
    x(n) = 2 * x(n - 1);
end
```

- while statements loop as long as a condition remains true.

  For example, find the first integer n for which factorial(n) is a 100-digit number:

```
n = 1;
nFactorial = 1;
while nFactorial < 1e100
    n = n + 1;
    nFactorial = nFactorial * n;
end
```

Each loop requires the end keyword.

Conditional statements:

The simplest conditional statement is an if statement. For example:

```
% Generate a random number
a = randi(100, 1);
```

```
% If it is even, divide by 2
if rem(a, 2) == 0
   disp('a is even')
   b = a/2;
end
```

if statements can include alternate choices, using the optional keywords elseif or else. For example:

```
a = randi(100, 1);

if a < 30
   disp('small')
elseif a < 80
   disp('medium')
else
   disp('large')
end
```

Control Flow:

%if	Conditional creation of code by the parser
break, _break	Terminate a loop or a Case switch prematurely
case, of, otherwise, end_case, _case	Switch statement
end	Close a block statement
for, from, to, step, end_for, _for_in, downto, _for_downto	For loop
if, then, elif, else, end_if, _if	If-statement (conditional branch in a program)
next, _next	Skip a step in a loop
repeat, until, end_repeat, _repeat	"repeat" loop
while, end_while, _while	"while" loop
return	Exit a procedure

Functions:

Syntax

function [y1,...,yN] = myfun(x1,...,xM)

Description

function [y1,...,yN] = myfun(x1,...,xM) declares a function named myfun that accepts inputs x1,...,xM and returns outputs y1,...,yN. This declaration statement must be the first executable line of the function. Valid function names begin with an alphabetic character, and can contain letters, numbers, or underscores.

You can save your function:

- In a function file which contains only function definitions. The name of the file must match the name of the first function in the file.
- In a script file which contains commands and function definitions. Functions must be at the end of the file. Script files cannot have the same name as a function in the file. Functions are supported in scripts in R2016b or later.

Files can include multiple local functions or nested functions. For readability, use the end keyword to indicate the end of each function in a file. The end keyword is required when:

- Any function in the file contains a nested function.
- The function is a local function within a function file, and any local function in the file uses the end keyword.
- The function is a local function within a script file.

Example:

```
function ave = average(x)
    ave = sum(x(:))/numel(x);
end
```

Plotting Commands:

Plot:

Syntax:

```
plot(X,Y)
plot(X,Y,LineSpec)
plot(X1,Y1,...,Xn,Yn)
plot(X1,Y1,LineSpec1,...,Xn,Yn,LineSpecn)
plot(Y)
plot(Y,LineSpec)
plot(___,Name,Value)
plot(ax,___)
h = plot(___)
```

Description:

`plot(X,Y)` creates a 2-D line plot of the data in `Y` versus the corresponding values in `X`.

- If `X` and `Y` are both vectors, then they must have equal length. The `plot` function plots `Y` versus `X`.
- If `X` and `Y` are both matrices, then they must have equal size. The `plot` function plots columns of `Y` versus columns of `X`.

To plot the graph of a function, you need to take the following steps −

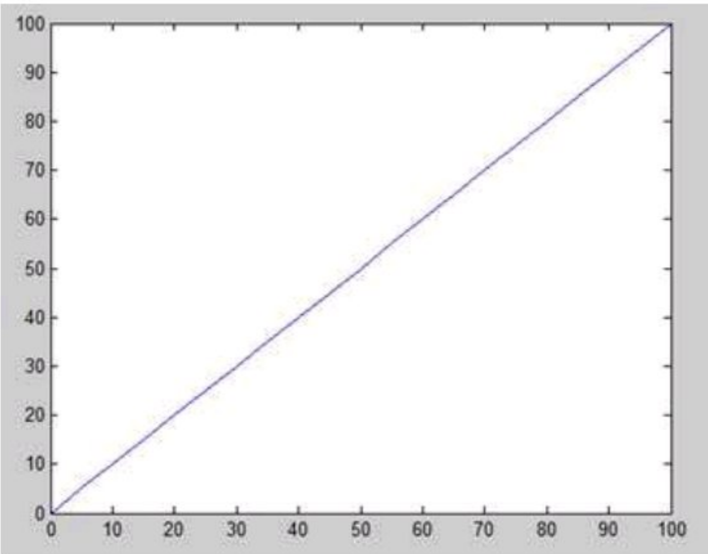
- Define **x**, by specifying the **range of values** for the variable **x**, for which the function is to be plotted
- Define the function, **y = f(x)**
- Call the **plot** command, as **plot(x, y)**

Following example would demonstrate the concept. Let us plot the simple function **y = x** for the range of values for x from 0 to 100, with an increment of 5.

Create a script file and type the following code −

```
x = [0:5:100];
y = x;
plot(x, y)
```

When you run the file, MATLAB displays the following plot :



Subplot:

subplot(m,n,p) divides the current figure into an m-by-n grid and creates axes in the position specified by p. MATLAB numbers subplot positions by row. The first subplot is the first column of the first row, the second subplot is the second column of the first row, and so on. If axes exist in the specified position, then this command makes the axes the current axes.

subplot(m,n,p,'replace') deletes existing axes in position p and creates new axes.

subplot(m,n,p,'align') creates new axes so that the plot boxes are aligned. This option is the default behavior.

Figure:

Create a figure graphics object

Syntax

- figure

- figure('*PropertyName*',PropertyValue,...)
- figure(h)
- h = figure(...)

Description

figure creates figure graphics objects. figure objects are the individual windows on the screen in which MATLAB displays graphical output.

figure creates a new figure object using default property values.

Labelling:

To label x-axis:

xlabel(txt)

To label y-axis:

ylabel(txt)

Conclusion: We learnt the basic syntax of MATLAB and various basic commands in it.

<u>EXPERIMENT 1:</u> Basic Functions/Signals in MATLAB

Date: *29-01-2020*

<u>Aim:</u> To plot basic functions sine, cosine, tangent and exponential in MATLAB. Plot basic signals such as unit impulse, unit step and unit ramp. Plot the periodic signals impulse train, square wave, sawtooth wave and triangular wave.

<u>Theory/Equations:</u>

Sine Wave:

A **sine wave** or **sinusoid** is a mathematical curve that describes a smooth periodic oscillation. A sine wave is a continuous wave. It is named after the function sine, of which it is the graph. Its most basic form as a function of time (t) is:

$$y(t) = A\sin(2\pi ft + \varphi) = A\sin(\omega t + \varphi)$$

where:

- ***A***, *amplitude*, the peak deviation of the function from zero.
- ***f***, *ordinary frequency*, the *number* of oscillations (cycles) that occur each second of time.
- ***ω* = 2*πf***, *angular frequency*, the rate of change of the function argument in units of radians per second
- **φ**, *phase*, specifies (in radians) where in its cycle the oscillation is at $t = 0$.

  When **φ** is non-zero, the entire waveform appears to be shifted in time by the amount **φ/*ω*** seconds. A negative value represents a delay, and a positive value represents an advance.

Cosine Wave:

A cosine wave is a signal waveform with a shape identical to that of a sine wave , except each point on the cosine wave occurs exactly 1/4 cycle earlier than the corresponding point on the sine wave. A cosine wave and its corresponding sine wave have the same frequency, but the cosine wave leads the sine wave by 90 degrees of phase .

$$f(t) = A\,cos\,cos\,(2\pi ft + \varphi) = Asin(\omega t + \varphi)$$

Tangent Wave:

$$f(t) = tan\,(t)$$

The tan function operates element-wise on arrays. The function accepts both real and complex inputs.

For real values of X, tan(X) returns real values in the interval [-∞, ∞].

For complex values of X, tan(X) returns complex values.

Exponential Function:

An exponential function can be defined as:

$$f(t) = e^t$$

It is expected to rise at a very fast rate within a short span of time. Usually, in electronics, a degrading exponential function is found common with the coefficient of 't' usually negative.

Unit Step Function:

$$f(t) = \{0, \quad x < 0 \\ \quad 1, \quad x > 0\}$$

Unit Impulse Function:

$$f(t) = \{0, \quad x = 0 \\ \quad 1, \quad x \neq 0\}$$

Ramp Function:

$$f(t) = \{0, \quad x < 0 \\ \quad t, \quad x \geq 0\}$$

Impulse Train function with a period T:

$$f(t) = \{0, \quad x \neq nT \\ \quad 1, \quad x = nT \;\; n\epsilon Z\}$$

Square Wave function with a period T:

$$f(t) = 1(-1)^{\lfloor 2t/T \rfloor}$$

Saw-tooth Wave function with a period T:

$$f(t) = t - \lfloor t \rfloor$$

Triangle Wave function with a period T:

The triangle wave can also be expressed as the integral of the square wave:

$$x(t) = \int_0^t \operatorname{sgn}(\sin(u))\, du.$$

Flowchart/Algorithm:

1. Clear console, screen and close windows using the commands clc, clear all, close all.
2. Initialise t with values from -10 to 10 with step as necessary.
3. Use in-built sin(), cos(), tan(), exp() while using plot function to plot respective graphs.

4. For Unit step function, create unit_step array with value 1 when t>0 and 0 otherwise, and plot it against t.

5. For Unit Impulse function, create unit_impulse array with value 1 when t=0 and 0 otherwise, and plot it against t.

6. For Ramp function, create ramp array with value t when t>0 and 0 otherwise and plot it against t.

7. For Impulse train function, create impulse_train array with value 1 when t = nT and 0 otherwise using for loop.

8. For square function, use in-built function square().

9. For sawtooth function, use in-built function sawtooth().

10. For triangle function, use in built function sawtooth() with width=0.5.

11. Use subplot command to create more than one graphs in one figure window. Use subplot command before plotting any graph such as subplot(<graph count x>,<graph count y>,<graph pos>) where graph count x = No. of graphs to be displayed horizontally, graph count y = No. of graphs to be displayed vertically, graph pos = Position of graph on the window which is usually row majored (1 – top left , 2 –top right , 3 – bottom left, 4 – bottom right in case of 2 by 2 subplot).

12. To plot graph, use plot() command with 1st parameter as the quantity for X axis, that is 't' and 2nd parameter as the quantity for Y axis, that is, any one of the above derived quantities.

13. To provide X axis label, use xlabel(<string>) and to provide Y axis label, use ylabel(<string>).

14. Use title(<string>) to provide title to the graph.

15. Use axis([]) to provide axis limits to the graph.

16. Do the above process for all required graphs.

17. It is recommended to use Sections as shown in the Code section so as to plot the graphs in a legible form. Sections can be used in the following fashion:- %% <title of section>

18. Run the written code using F5 or section using CTRL + F5.

19. Save the graphs from File > Save As.

20. The required experiment has been completed successfully

Code:

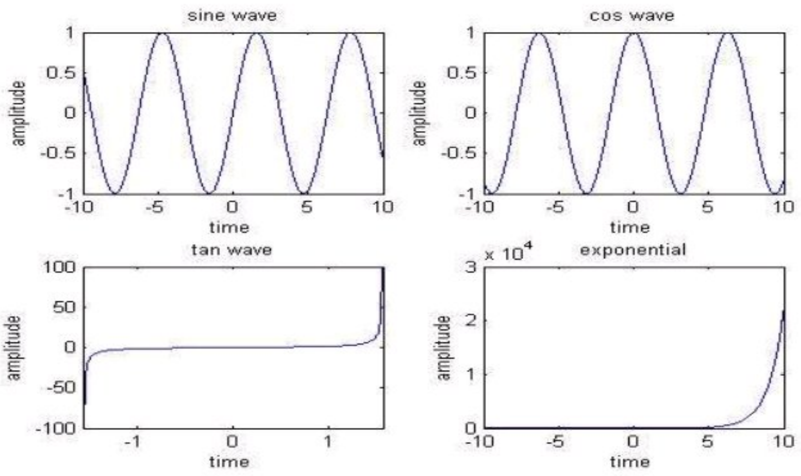
```
clc
clear all
close all
%question 1
t=-10:0.01:10

%sine wave
x=sin(t)
%figure:plot(t,x)
subplot(2,2,1):plot(t,x)
xlabel('time')
ylabel('amplitude')
title('sine wave')
```

```
%cosine wave
y=cos(t)
%figure:plot(t,y)
subplot(2,2,2):plot(t,y)
xlabel('time')
ylabel('amplitude')
title('cos wave')

%tangent wave
z=tan(t)
%figure:plot(t,z)
subplot(2,2,3):plot(t,z)
axis([(-pi/2) (+pi/2) -100 +100])
xlabel('time')
ylabel('amplitude')
title('tan wave')

%exponential wave
p=exp(t)
%figure:plot(t,p)
subplot(2,2,4):plot(t,p)
xlabel('time')
ylabel('amplitude')
title('exponential')

%question 2
%unit impulse function
d=zeros(1,length(t))
d(t==0)=1;
figure:subplot(3,1,1):plot(t,d)
xlabel('time')
ylabel('amplitude')
title('impulse function')

%unit step function
s=zeros(1,length(t))
s(t>=0)=1;
s(t<0)=0;
subplot(3,1,3):plot(t,s)
xlabel('time')
ylabel('amplitude')
title('step function')

%unit ramp function
r=zeros(1,length(t))
r(t>=0)=t(t>=0);
subplot(3,1,2):plot(t,r)
xlabel('time')
```

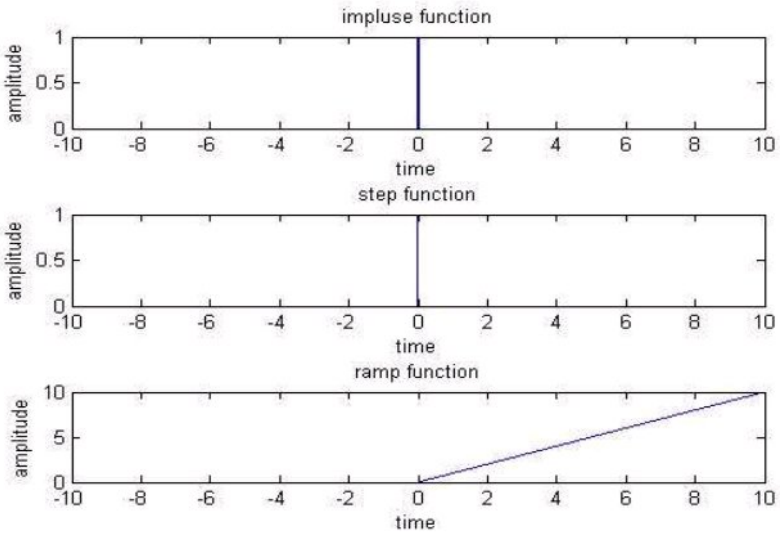
```
ylabel('amplitude')
title('ramp function')

%question 3
%impulse train
T=1;
c=((mod(t,T)==0)==1);
figure;
subplot(2,2,1);
plot(t,c);
xlabel('time');
ylabel('amplitude');
title('impluse train');
%sawtooth
T=1;
j=sawtooth(t);
subplot(2,2,2);
plot(t,j);
xlabel('time');
ylabel('amplitude');
title('saw tooth');

%square wave
i=square(t);
subplot(2,2,3);
plot(t,i);
xlabel('time');
ylabel('amplitude');
title('square wave');

%triangular wave
v=sawtooth(t,0.5)
subplot(2,2,4);
plot(t,v);
xlabel('time');
ylabel('amplitude');
title('trainglar wave');
```

Result/Output Waveforms:

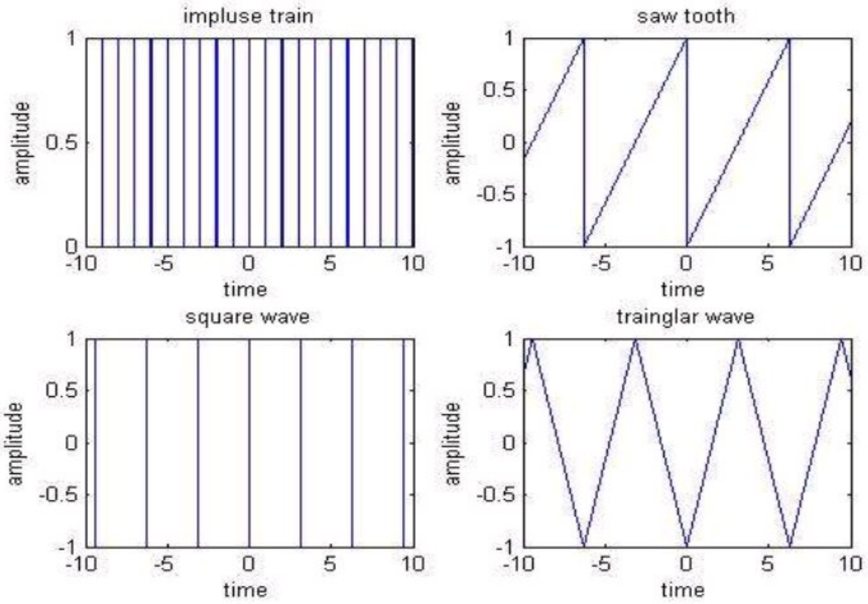
1.Basic functions:



2.Basic signals:



3.Periodic signals:



Conclusion: In this experiment we generated several *basic functions* like sine, cosine, tangent and exponential functions and *basic signals* like unit impulse, unit step and unit ramp and also the *periodic signals* like impulse train, square wave, sawtooth wave and triangular wave using ***MATLAB***.

Remarks: **Signature:**

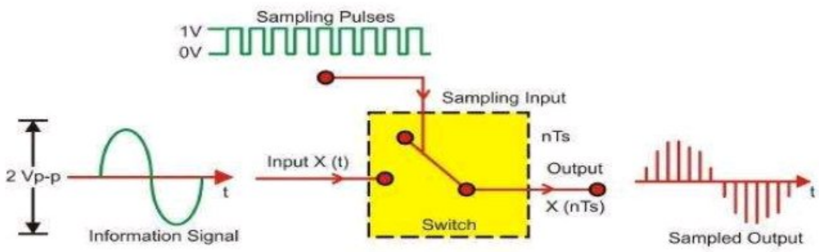
SAMPLING AND RECONSTRUCTION OF SIGNAL

Experiment No.: 2 **Date:30/01/2020**

Aim : Study of Sampling and Reconstruction of signal. Verify Nyquist criteria. Model ST21O1 W kit, connecting wires, CRO/DSO

Apparatus: Model ST 2151 W kit, connection wires, CRO/DSO

Sampling Theory:



Procedure:

A. Set up for Sampling and reconstruction of signal.

Initial set up of trainer:

Duty cycle selector switch position : Position 5

Sampling selector switch : Internal position

1. Connect the power cord to the trainer. Keep the power switch in 'Off' position.
2. Connect 1 KHz Sine wave to signal Input as shown in Fig.1.1.
3. Switch 'On' the trainer's power supply & Oscilloscope.
4. Connect BNC connector to the CRO and to the trainer's output port.

You can observe the process of step-by-step generating sine wave signal from Square wave of 1 KHz at TP3, TP4, TP5 and TP6 respectively.

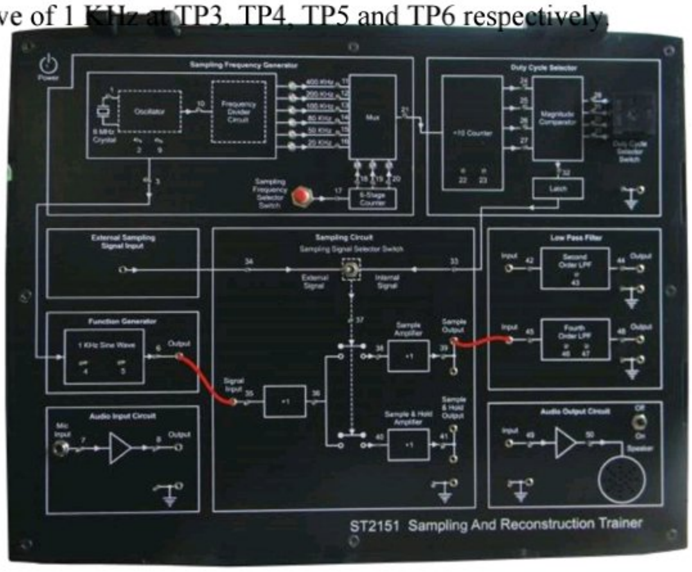


Fig. 1.1. Connection diagram for sampling a signal

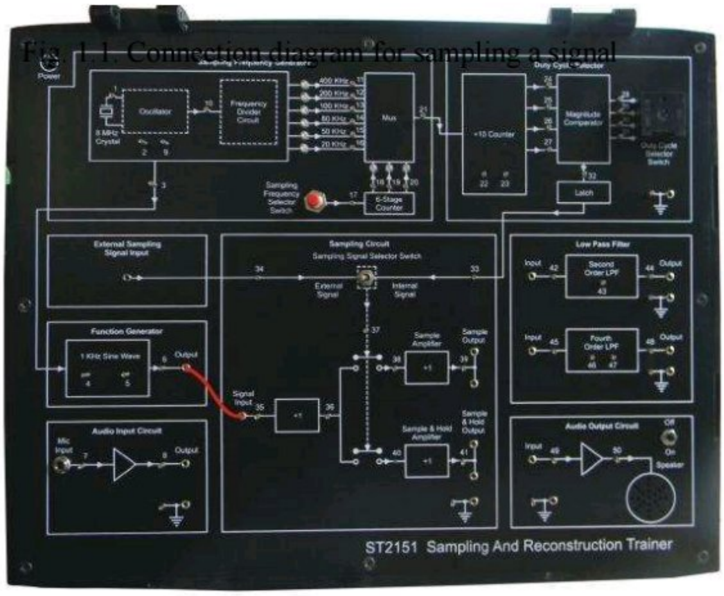


Fig. 1.2. Connection diagram for reconstruction of a sampled signal

B. Set up for effect of Sample Amplifier and Sample and Hold Amplifier on reconstructed signal.

Set up for effect of II order and IV order Low Pass Filter on reconstructed signal.

Initial set up of trainer:
Duty cycle selector switch position : Position 5
Sampling selector switch: Internal position

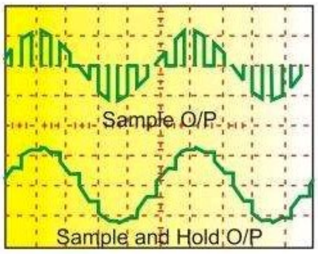
1. Connect the power cord to the trainer. Keep the power switch in 'Off' position.
2. Connect 1 KHz Sine wave to signal Input.
3. Switch 'On' the trainer's power supply & Oscilloscope.
4. Connect BNC connector to the CRO and to the trainer's output port.
5. Select sampling frequency of 8 KHz by Sampling Frequency Selector Switch pressed till 80 KHz signal LED glows.
6. Observe 1 KHz sine wave and Sample Output (TP39) on oscilloscope. The display shows 1 KHz sine wave being sampled at 8 KHz, so there are 8 samples for every cycle of the sine wave.
7. Connect Sample Output to Fourth Order low pass filter Input as shown in figure 1.2. Observe the filtered output (TP48) on the oscilloscope. The display shows the reconstructed 1 KHz sine wave.
8. Similarly observe the sampled 1 KHz sine wave at and Sample and Hold Output (TP41) on oscilloscope. The display shows 1 KHz sine wave being sampled and hold signal at 8 KHz. Connect Sample and Hold Output to Second Order low pass filter Input and observe the filtered output (TP44) on oscilloscope. The display shows the reconstructed1 KHz sine wave.

9. By pressing Sampling Frequency Selector Switch, change the sampling frequency from 2 KHz, 5 KHz, 10 KHz, 20 KHz up to 40 KHz (Sampling frequency is 1/10th of the frequency indicated by the illuminated LED). Observe how Sample output (TP39) and Sample and Hold Output (TP41) changes in each case\

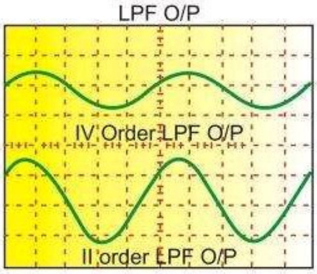
.

Sample Observations:

Sampling Signal 8 KHz (50%)



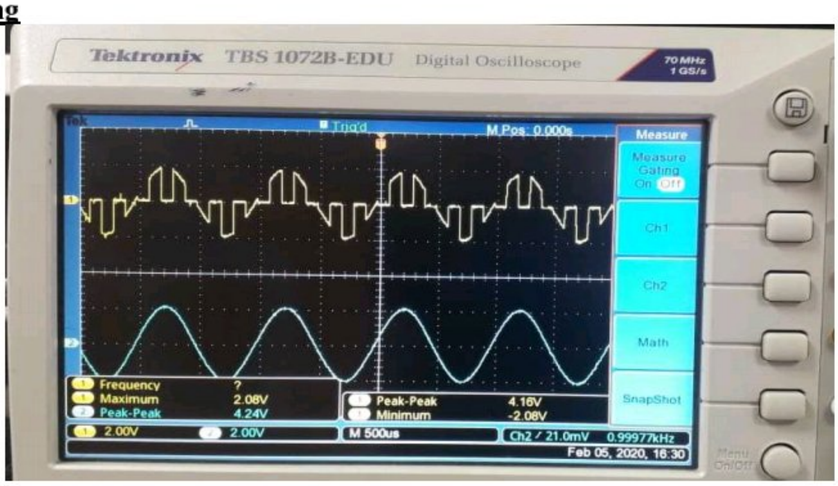
LPF O/P



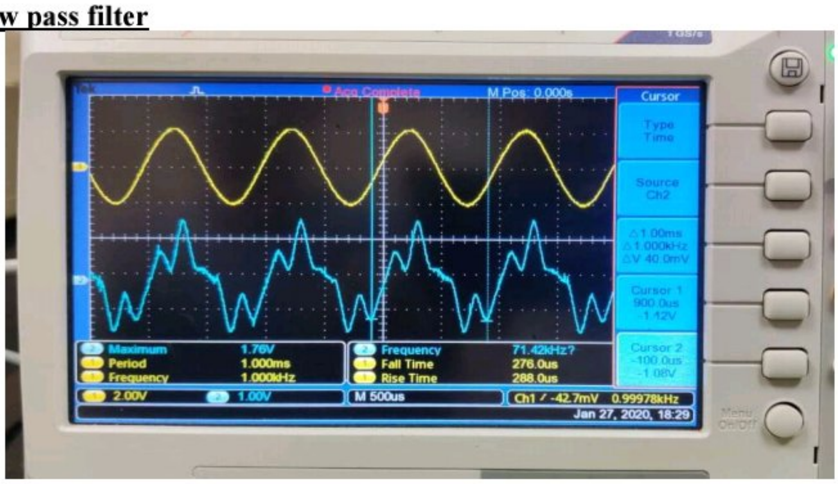
Output Waveforms:

sampling frequency:5 KHz **Duty cycle:50%**

Natural sampling

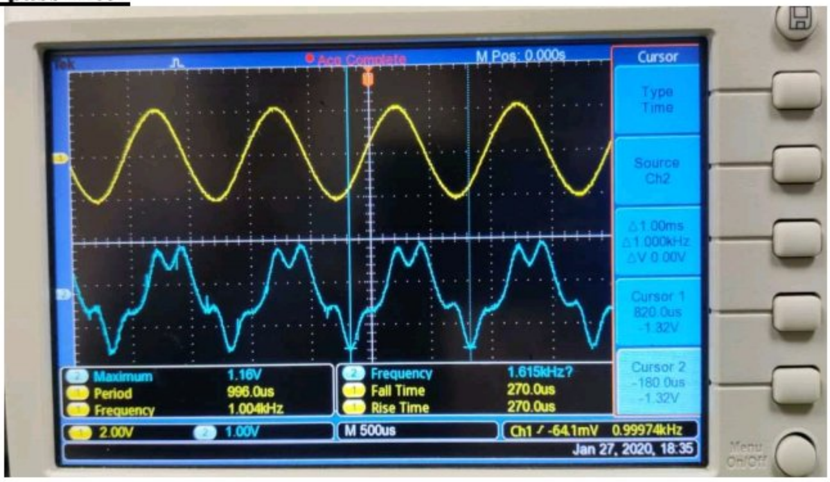


Second order low pass filter



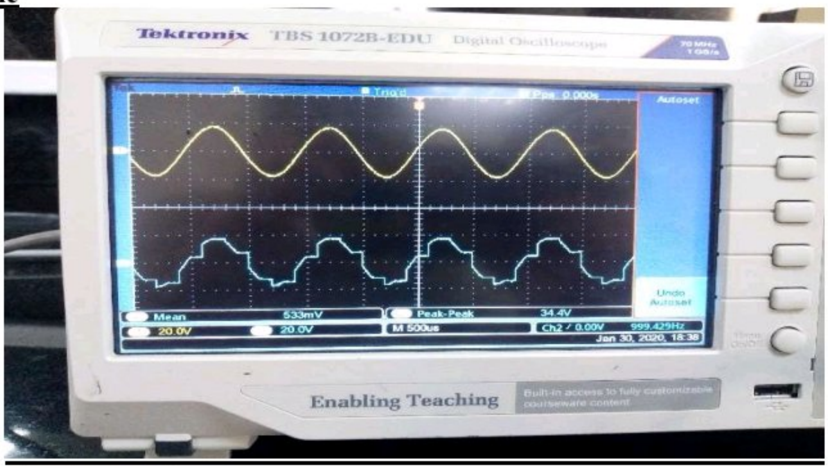
-

Forth order low pass filter

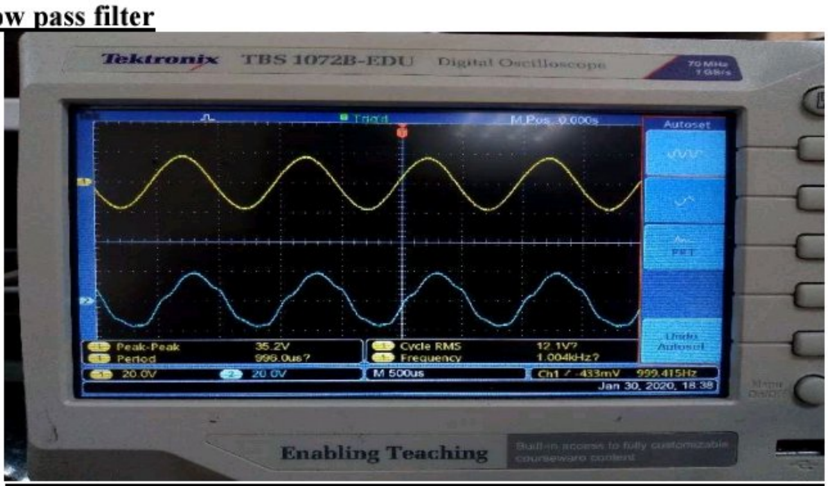


Sampling frequency:5KHz **Duty cycle: 50%**

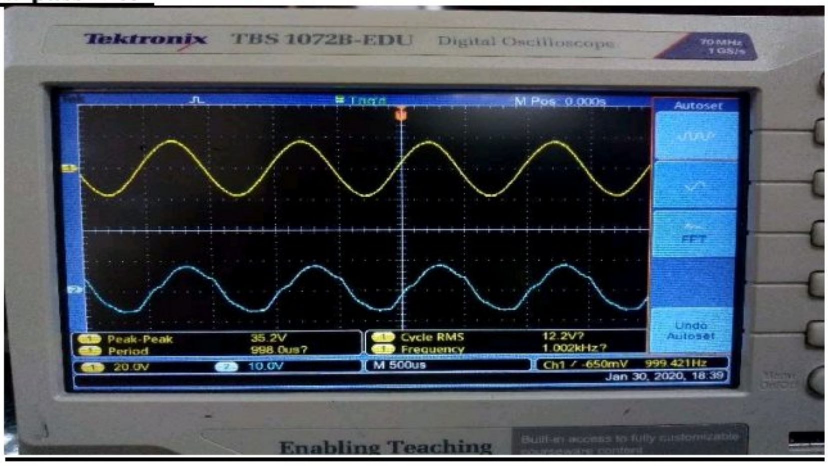
Hold and sample



Second order low pass filter

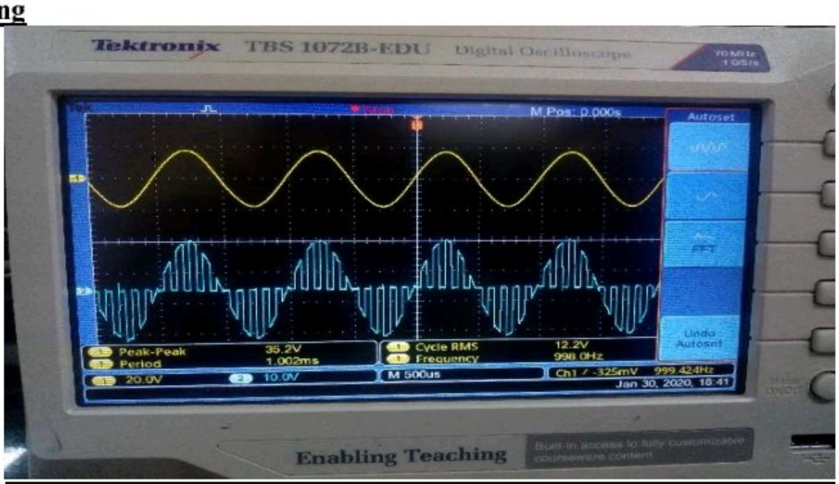


Fourth order low pass filter

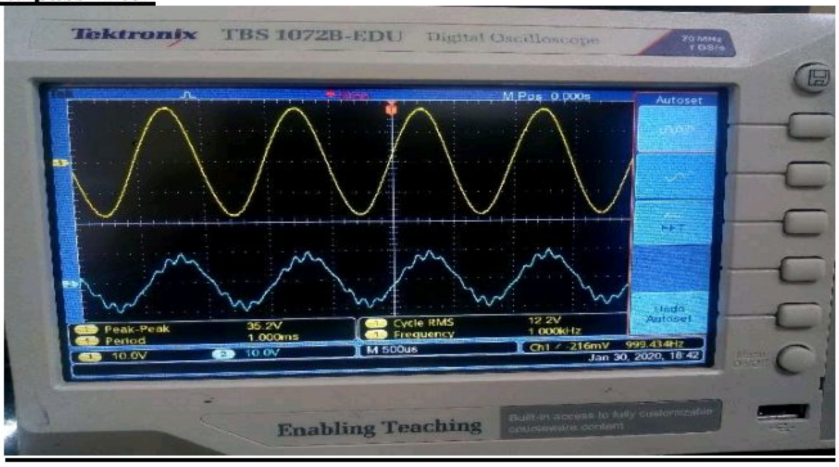


Sampling frequency:10 khz **Duty cycle: 50%**

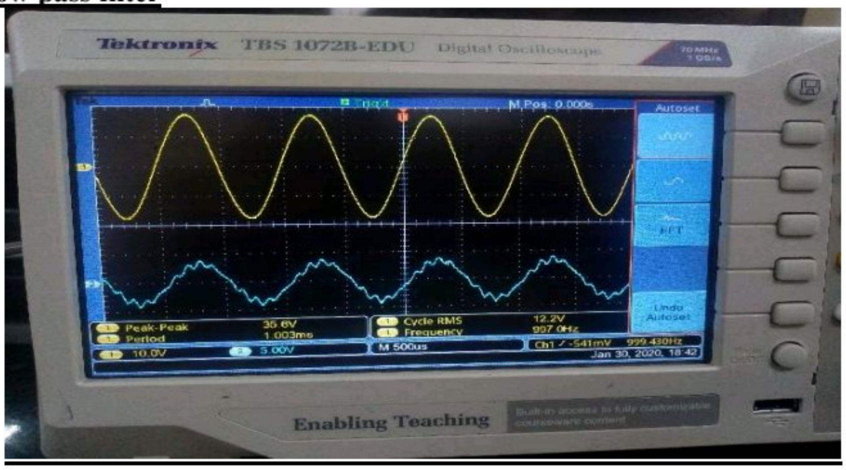
Natural sampling



Second order low pass filter

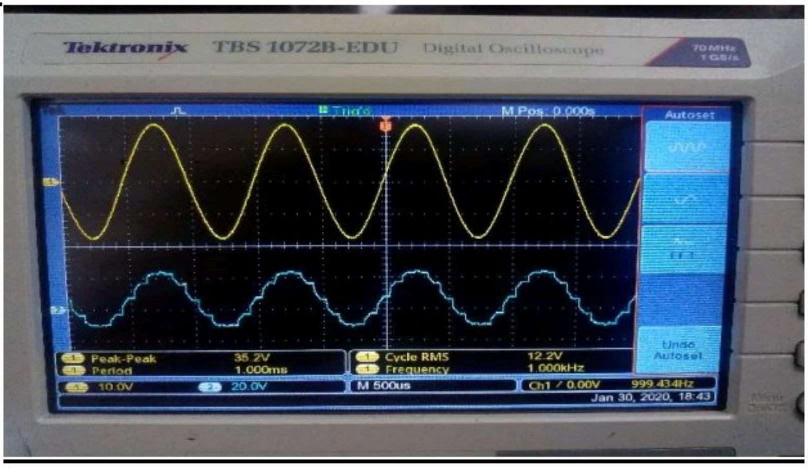


Fourth order low pass filter

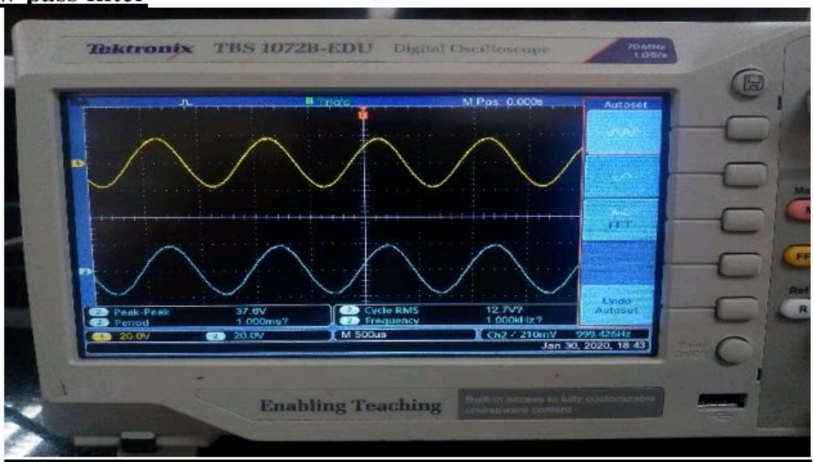


Sampling frequency:10 khz **Duty cycle: 50%**

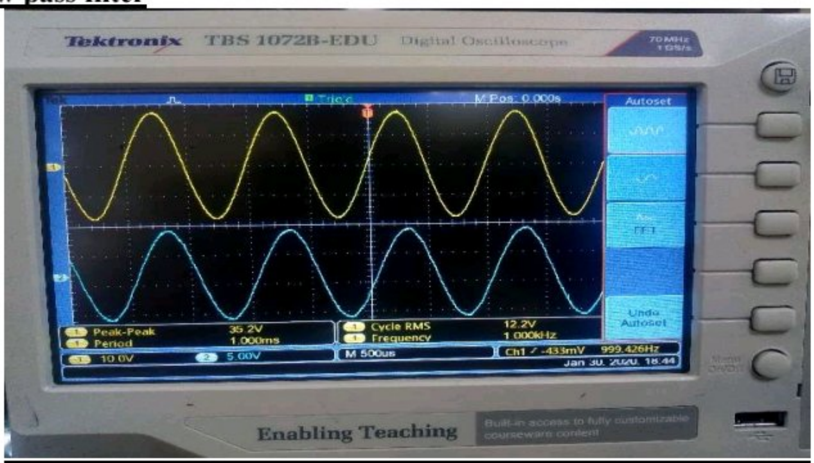
Sample and hold



Second order low pass filter



Fourth order low pass filter



Conclusion:

We have studied the Sampling and Reconstruction of signal. We also verified the Nyquist criteria. We also understood Under-Sampling, Over-Sampling and Perfect-Sampling. Waveforms were observed in DSO under different Duty Cycle values.

Remarks: **Signature:**

EXPERIMENT 3: Fourier Coefficients of Waveforms

Date: *12-02-2020*

Aim: To compute the fourier coefficients of exponential and square wave and plot their magnitude and phase spectra.

Theory:

Fourier Series:- A Fourier series is an expansion of a periodic function $f(x)$ in terms of an infinite sum of sines and cosines. Fourier series make use of the orthogonality relationships of the sine and cosine functions. The computation and study of Fourier series is known as harmonic analysis and is extremely useful as a way to break up an *arbitrary* periodic function into a set of simple terms that can be plugged in, solved individually, and then recombined to obtain the solution to the original problem or an approximation to it to whatever accuracy is desired or practical. Examples of successive approximations to common functions using Fourier series are illustrated above.

$$f(x) = \frac{1}{2} a_0 + \sum_{n=1}^{\infty} a_n \cos(n x) + \sum_{n=1}^{\infty} b_n \sin(n x),$$

where

$$a_0 = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x)\, d x$$
$$a_n = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \cos(n x)\, d x$$
$$b_n = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \sin(n x)\, d x$$

and $n = 1$, 2, 3, ... Note that the coefficient of the constant term a_0 has been written in a special form compared to the general form for a generalized Fourier series in order to preserve symmetry with the definitions of a_n and b_n.

The Fourier cosine coefficient a_n and sine coefficient b_n are implemented in the Wolfram Language as FourierCosCoefficient[*expr*, *t*, *n*] and FourierSinCoefficient[*expr*, *t*, *n*], respectively.

A Fourier series converges to the function $\bar{f}$ (equal to the original function at points of continuity or to the average of the two limits at points of discontinuity)

$$\bar{f} \equiv \begin{cases} \frac{1}{2}\left[\lim_{x \to x_0^-} f(x) + \lim_{x \to x_0^+} f(x)\right] & \text{for } -\pi < x_0 < \pi \\ \frac{1}{2}\left[\lim_{x \to -\pi^+} f(x) + \lim_{x \to \pi_-} f(x)\right] & \text{for } x_0 = -\pi, \pi \end{cases}$$

The Fourier series for a few common functions are summarized in the table below.

function	$f(x)$	Fourier series
Fourier series--sawtooth wave	$\frac{x}{2L}$	$\frac{1}{2} - \frac{1}{\pi}\sum_{n=1}^{\infty} \frac{1}{n} \sin\left(\frac{n\pi x}{L}\right)$
Fourier series--square wave	$2\left[H\left(\frac{x}{L}\right) - H\left(\frac{x}{L} - 1\right)\right] - 1$	$\frac{4}{\pi}\sum_{n=1,3,5,\ldots}^{\infty} \frac{1}{n} \sin\left(\frac{n\pi x}{L}\right)$
Fourier series--triangle wave	$T(x)$	$\frac{8}{\pi^2}\sum_{n=1,3,5,\ldots}^{\infty} \frac{(-1)^{(n-1)/2}}{n^2} \sin\left(\frac{n\pi x}{L}\right)$

Euler's formula: e^x=cosx+isinx

Using Eulers Equation, and a little trickery, we can convert the standard Rectangular Fourier Series into an exponential form. Even though complex numbers are a little more complicated to comprehend, we use this form for a number of reasons:

1. Only need to perform one integration
2. A single exponential can be manipulated more easily than a sum of sinusoids
3. It provides a logical transition into a further discussion of the Fourier Transform.

Next, for any signal f(t) over $[0,T_0]$, or any periodic f(t) with period T_0 we can compute the Exponential Fourier Series. We begin by writing

$$f(t) = \sum_{n=-\infty}^{\infty} D_n e^{jnw_0 t}$$

since our basis set is now $\left\{e^{jnw_0 t}\right\}$. We must find the D_0 and D_n coefficients to find the EFS of the signal f.

As before with trigonometric Fourier Series $D_0 = \frac{1}{T_0}\int_{T_0} f(t)dt$ and $D_n = \frac{1}{T_0}\int_{T_0} f(t)e^{-jnw_0 t}\,dt$

$w_o=2\pi/ T_o$

$w=n\, w_o=[-N,N]*\, 2\pi/ T_o$

T_o= Time period

Flowchart/Algorithm:

1. Clear console, screen and close windows using the commands clc, clear all, close all.

2. Initialise t with values from -4pi to 4pi with step as necessary and even To which is time period and N,w,wo also.

3. Use in-built exp(),square() while using plot function to plot respective graphs.

4. Run the for loop from 1 to the length of w and then calculate the integral of Dn as the formula shown above by using the syntax of integral in MATLAB.

5. Now for plotting magnitude spectra and phase spectra use stem(w, abs(D)) for magnitude spectra and stem(w, angle(D)) for phase spectra.

6. Use subplot command to create more than one graphs in one figure window. Use subplot command before plotting any graph such as subplot(<graph count x>,<graph count y>,<graph pos>) where graph count x = No. of graphs to be displayed horizontally, graph count y = No. of graphs to be displayed vertically, graph pos = Position of graph on the window which is usually row majored (1 – top left , 2 –top right , 3 – bottom left, 4 – bottom right in case of 2 by 2 subplot).

7. To plot graph, use plot() command with 1st parameter as the quantity for X axis, that is 't' and 2nd parameter as the quantity for Y axis, that is, any one of the above derived quantities.

8. To provide X axis label, use xlabel(<string>) and to provide Y axis label, use ylabel(<string>).

9. Use title(<string>) to provide title to the graph.

10. Use axis([]) to provide axis limits to the graph.

11. Do the above process for all required graphs.

12. Again run the loop for calculating summation function as above by using integral in MATLAB and again plotting it also as described above in the same manner(subplot).

13. Like this we can do for various functions like sine,cosine,tangent,sawtooth,square,exponential waves also just by replacing the function in integral by our desrired function and also changing the time period of that particular wave.

14. It is recommended to use Sections as shown in the Code section so as to plot the graphs in a legible form. Sections can be used in the following fashion:- %% <title of section>

15. Run the written code using F5 or section using CTRL + F5.

16. Save the graphs from File > Save As.

17. The required experiment has been completed successfully

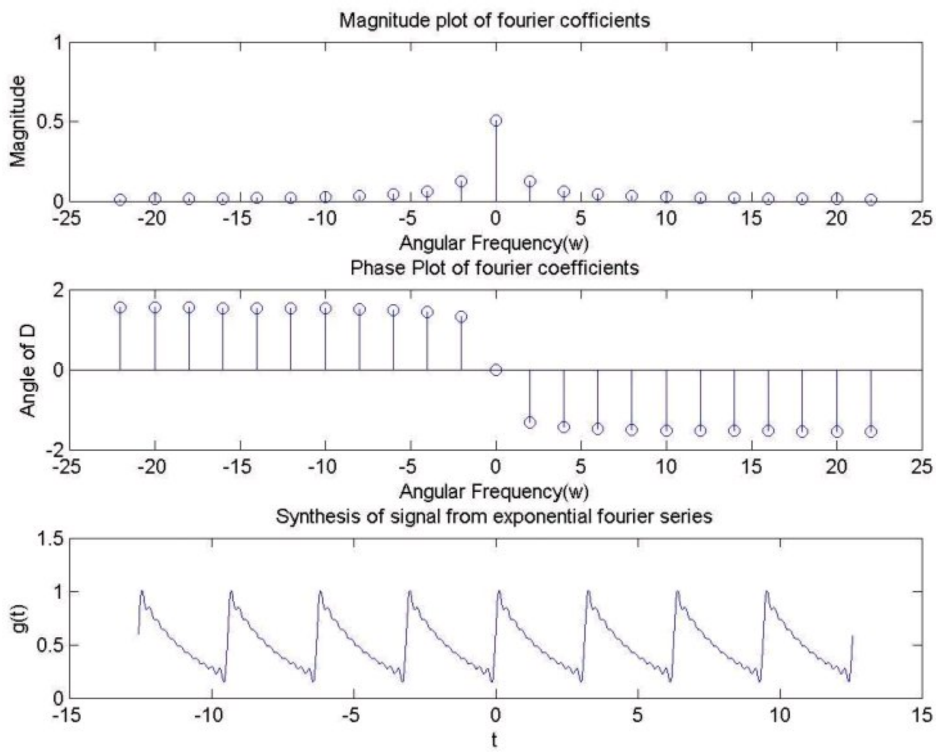
<u>Code:</u>

```
clc      %to clear command window
clear all
close all
%Aim:-To compute the fourier coefficients of exponential and square wave plot their magnitude and
phase spectra.

%exponential wave code
  T0 = pi;         %To=π
  N = 11;          %N=11
  w0 = 2*pi / T0;          %wo is defined
  t = -4 * pi : 0.01 : 4 * pi;         %t values are declared
  w = (-N: N) * (2 * pi / T0);      %w=nwo

  %Running loop upto the length of w

  for i = 1:length(w)
     D(i) = i / T0 * integral(@(t)exp(-t/2).*exp(-1j*w(i)*t), 0, T0);
  end
```

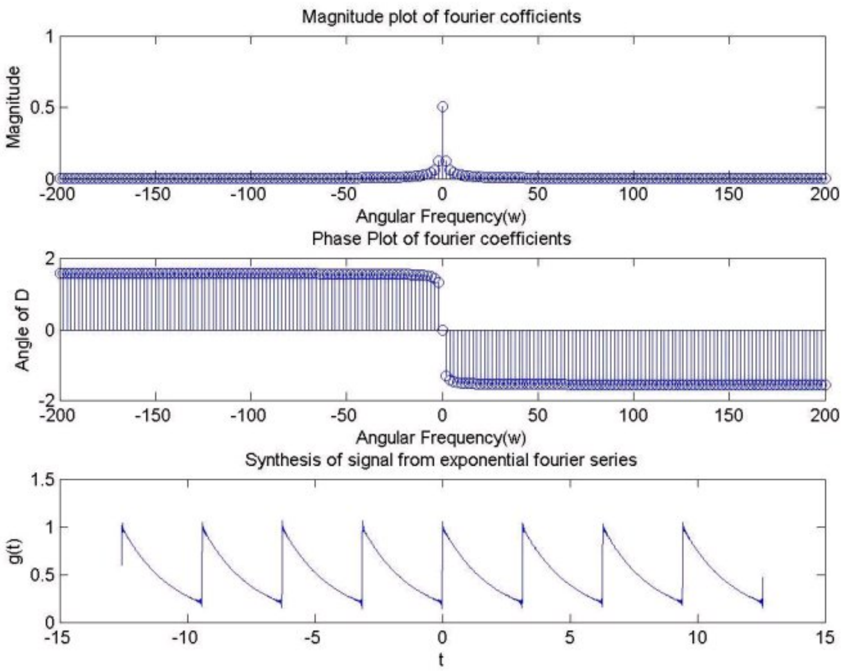
```
    %Plotting
    figure;
    subplot(3, 1, 1);
    stem(w, abs(D));
    xlabel('Angular Frequency (w)');
    ylabel('Magnitude');
    title('Magnitude plot of fourier coefficients');

    subplot(3, 1, 2);
    stem(w, angle(D));
    xlabel('Angular Frequency (w)');
    ylabel('Angle of D');
    title('Phase plot of fourier coefficients");

    sum = 0;%initialising the sum to zero

    %running the for loop again
    for i = 1:length(w)
       sum = sum + D(i)*exp(j*w(i)*t);
    end

    %Plotting
    subplot(3, 1, 3);
    plot(t, sum);
    xlabel('t');
    ylabel('g(t)');
    title('Synthesis of signal from exponential fourier series');

  %Square wave code

clc;         %to clear command window
clear all;
close all;

T0 = 2 * pi;         %time period of square wave
N = 11;              %initialising N
w0 = 2*pi / T0;
t = -4 * pi : 0.01 : 4 * pi;
w = (-N: N) * (2 * pi / T0);

%Loop
for i = 1:length(w)
   D(i) = i / T0 * integral(@(t)square(t).*exp(-1j*w(i)*t), 0, T0);
end

%Plotting
figure;
subplot(3, 1, 1);
stem(w, abs(D));
xlabel('Angular Frequency (w)');
ylabel('Magnitude |Dn|');
```

```
title('Magnitude Spectrum');

subplot(3, 1, 2);
stem(w, angle(D));
xlabel('Angular Frequency (w)');
ylabel('Angle of D');
title('Phase Spectrum');

%initialising sum to zero
sum = 0;
for i = 1:length(w)
   sum = sum + D(i)*exp(j*w(i)*t);
end

%Plotting
subplot(3, 1, 3);
plot(t, sum);
xlabel('t');
ylabel('G(t)');
title('Synthesis of Signal from Square Wave Fourier Series');
```

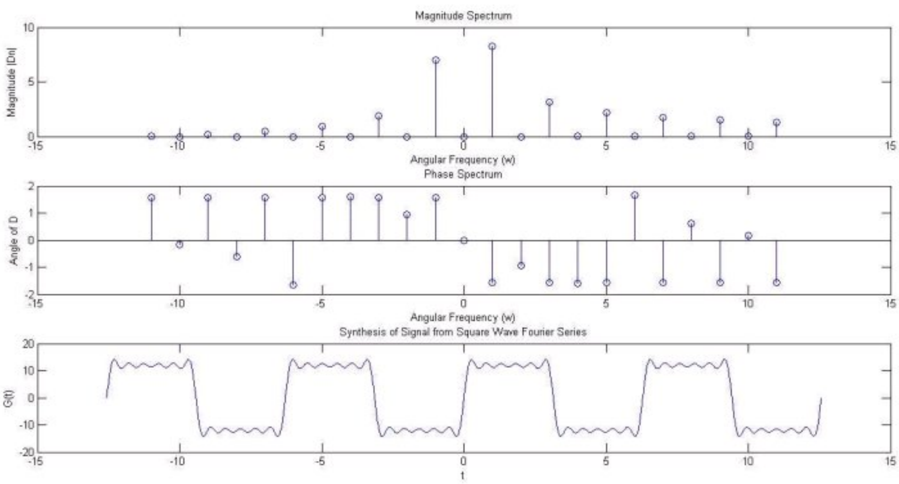
Output Waveforms:



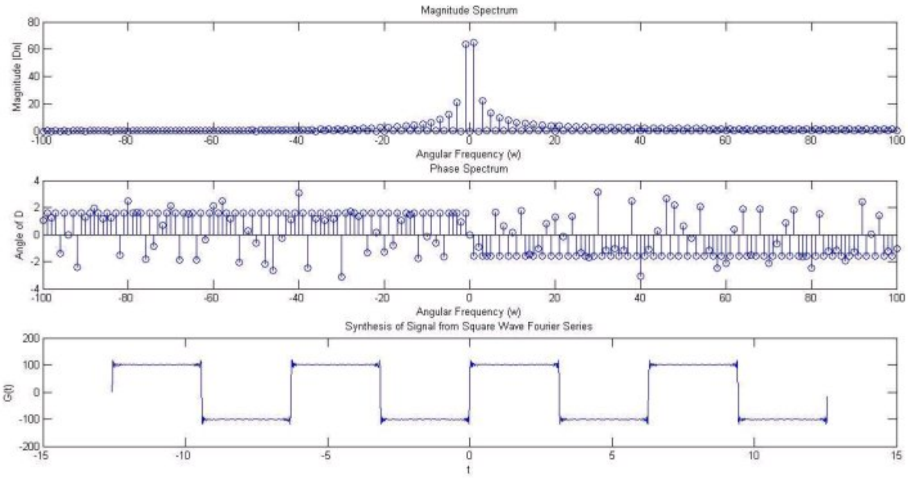
Exponential wave magnitude and phase spectra(n=11)



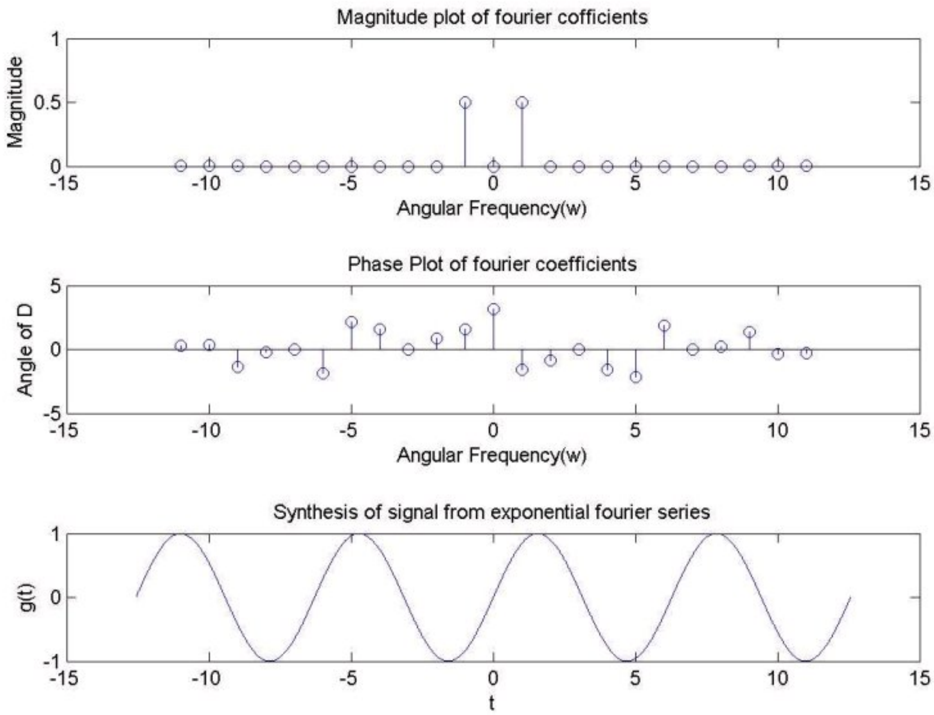
Exponential wave magnitude and phase spectra(n=100)



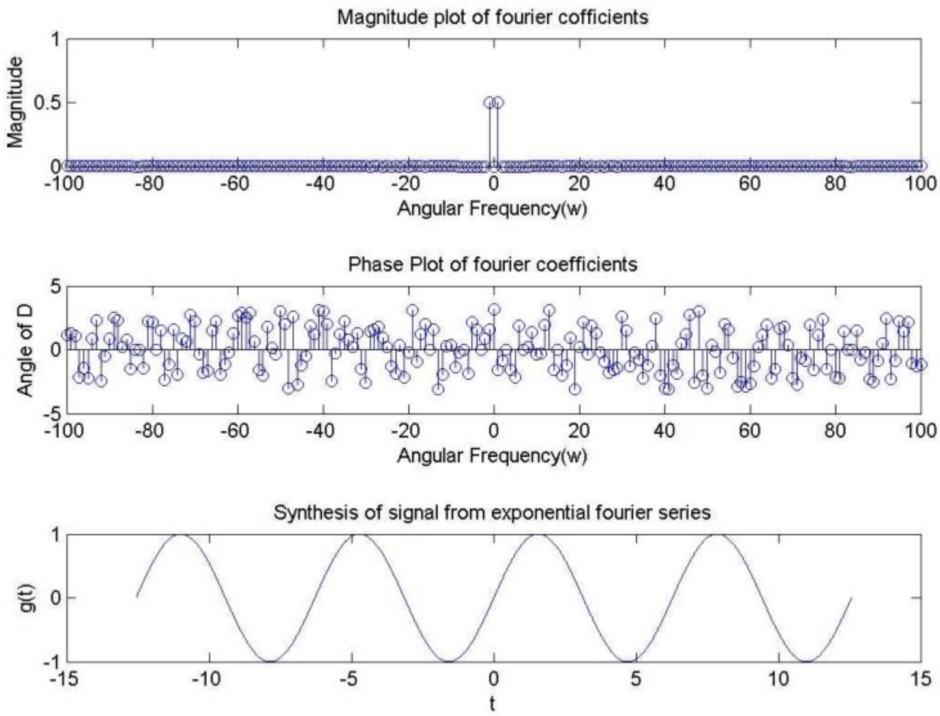
Square wave magnitude and phase spectra(n=11)



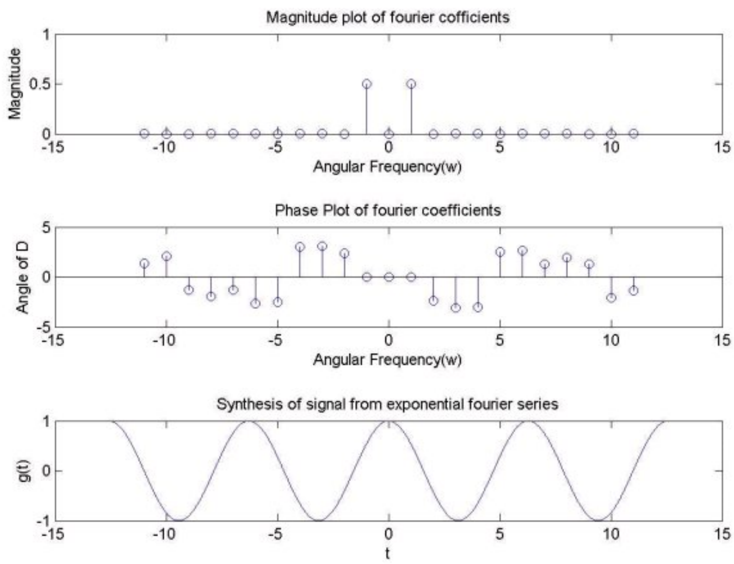
Square wave magnitude and phase spectra(n=100)



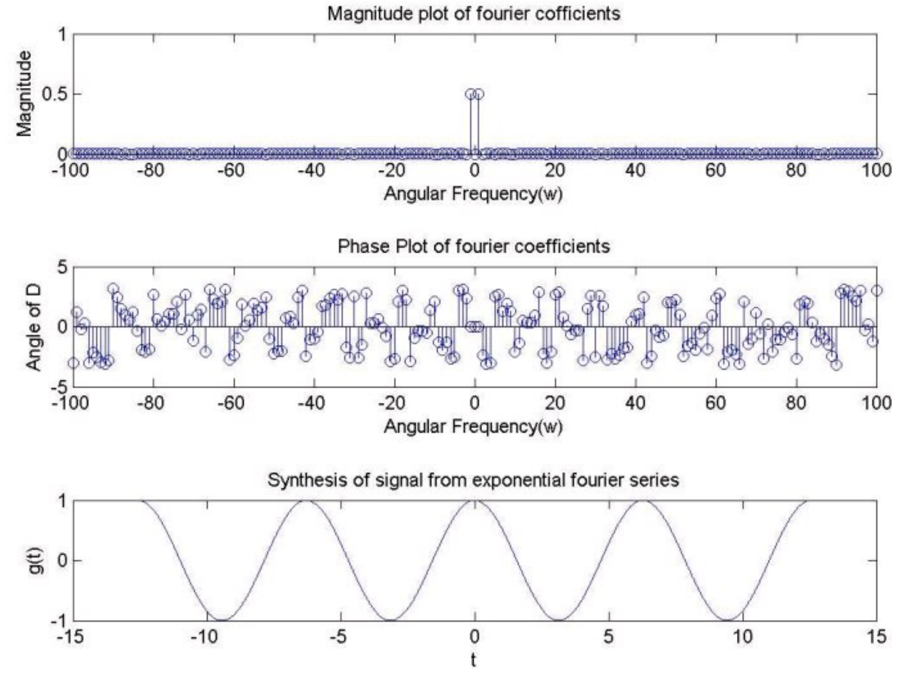
Sine wave magnitude and phase spectra(n=11)



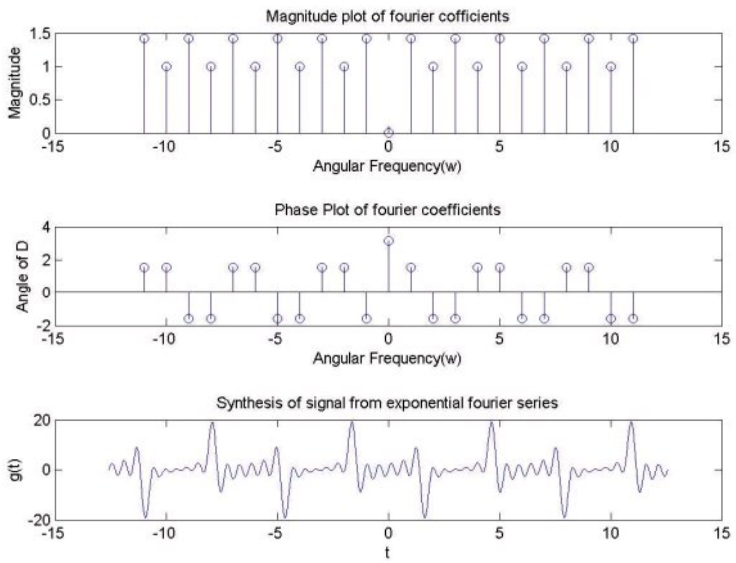
Sine wave magnitude and phase spectra(n=100)



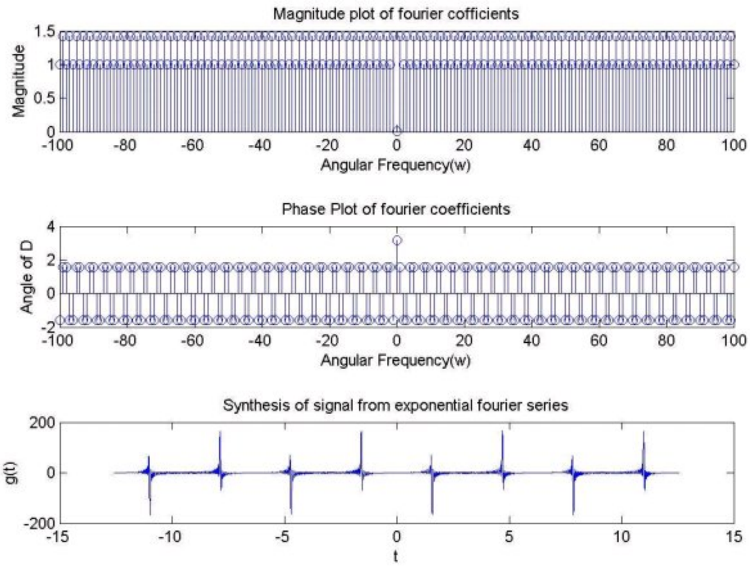
Cosine wave magnitude and phase spectra(n=11)



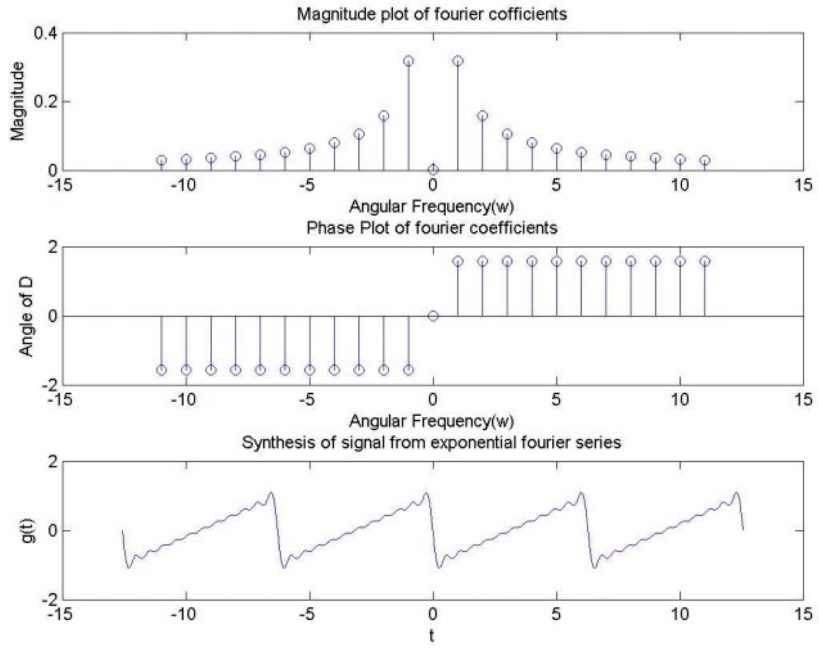
Cosine wave magnitude and phase spectra(n=100)



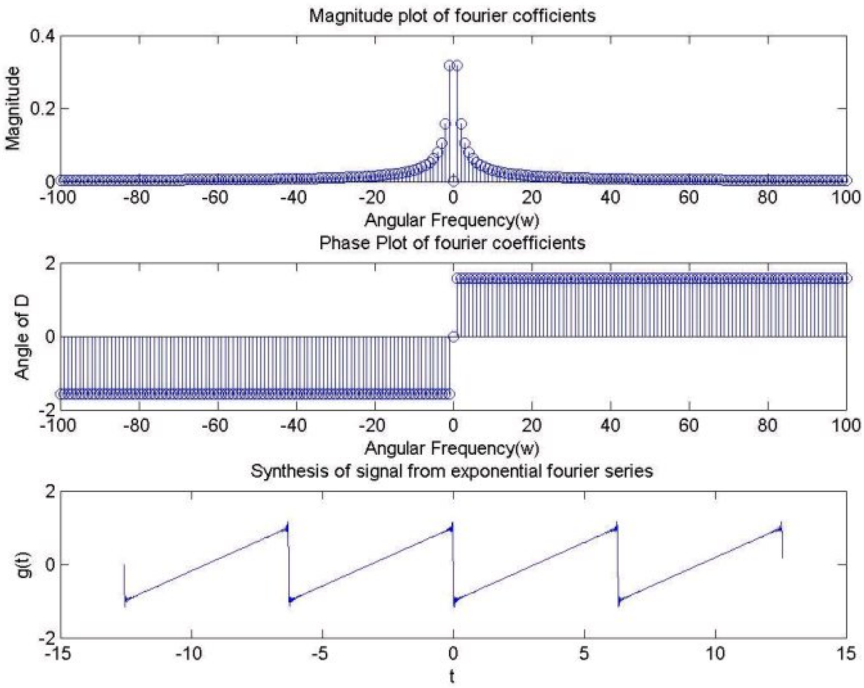
Tangent wave magnitude and phase spectra(n=11)



Tangent wave magnitude and phase spectra(n=100)



Sawtooth wave magnitude and phase spectra(n=11)



Sawtooth wave magnitude and phase spectra(n=100)

Conclusion: In this experiment we computed the fourier coefficients of exponential wave and square wave and plotted their magnitude and phase spectra using ***MATLAB***.

Remarks: **Signature:**

AMPLITUDE MODULATION

Experiment No.: 4 **Date:20/02/2020**

Aim: To study amplitude modulation and observe the waveforms for three different modulation indices.

Apparatus: Trainer board ST 2201 & 2202, power supply, connecting wires, CRO, function generator, carrier generator.

Theory: In communications, modulation means to vary some parameter of the high frequency carrier wave in proportion to the amplitude of the baseband or the modulating signal. A parameter of the carrier wave means, either of the following:

5. amplitude
6. frequency
7. phase

Now when amplitude of the carrier wave is varied with respect to the amplitude of the baseband signal, the modulation incorporated is termed as amplitude modulation. The Fig 1.1 shows an AM wave

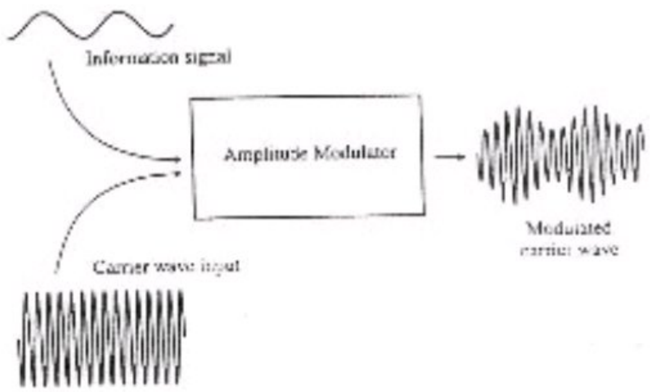


Fig. 1.1 Concept of AM wave

We go in for modulation of the carrier signal because if we want to transmit the baseband low frequency signal without any modulation the size of the antenna required will be too large.

Depth of Modulation :

The amount by which the amplitude of the carrier wave increases and decreases depends on the amplitude of the information signal and is called the 'depth of modulation'. The depth of modulation can be quoted as a fraction or as a percentage.

$$\text{Percentage modulation} = \frac{V\max - V\min}{V\max + V\min} \times 100\%$$

Description of different blocks of the circuit:

(1) Input audio amplifier section:

This section is used to amplify low level audio signal coming from Mic/Loudspeaker and give it to A.M. Modulator section for live A.M. modulation. It consist pre-amplifier stage and output amplifier stage. Transistor BC148B is used for the pre-amplification. Input signal from the Mica is given to BC148B through coupling capacitor. Output of BC148B is given to pin 10 of IC 810 which contains audio amplifier, driver & output stage. The amplified output is obtained at pin 16 which can be used as modulating signal.

(2) Modulating audio signal generator section:

IC 8038 is used to generate sine wave signal. Pot P2 is used to vary its frequency .The range is 20 Hz to20KHz.Two 100k pots are used to adjust the peaks of the sine wave and 1K preset is used for duty cycle adjustment. The sine wave signal is available at pin 2 of IC8038. This signal is amplified by IC LM356.Pot P2 is used to vary the amplitude of sine wave signal.

(3) RF Carrier oscillator section:

Transistor BC107B is used to generate RF sine wave signal. Pot P1 is used to vary its frequency from 200 kHz to1MHz.Here transistor Q2, Q3, Q4 and Q5 is used to amplify the RF signal of Q1. Pot P2 is used to vary the amplitude of sine wave from 0 to 10 Vpp.

(4) Double balanced amplitude modulator section:

IC 1496 is used as balanced modulator. The modulating audio signal is connected at pin 1 through buffer transistor Q1.This IC has two inputs as it works as balanced modulator. The Second input can be connected at pin 4 through buffer transistor Q2.The RF carrier signal is connected at pin 8 through coupling capacitor from RF carrier oscillator section. The modulated outputs are available at pin 12 and 6 of this IC which are then balanced amplified by Q3, Q4, Q5 and Q6.The final balanced modulated output is available at output terminals. Bal-A preset is used to balance carrier signal while Bal-B preset is used to balance input audio signal. 1K preset is used to adjust output zero DC level.

(5) DC voltage generating section:

To observe the effect of dc voltages on AM modulating signal +1V dc and -8V to +8V dc voltage is required which is generated using IC741 and presets.

(6) Filter section:

Here notch filter of 455 KHz is designed using crystal. This filter is used to obtain suppressed carrier double side band modulated signal from DSB signal.

(7) AM demodulators:

(a) Diode detector circuit:

This circuit consists detector diode OA79 and capacitor C1, C2,C3 and load resistor R1. It works as an envelope detector circuit.R1 and C forms a low pass filter meant to reduce the carrier frequency ripple in the output.

(b) Product detector:

This section is similar to AM balance modulator section. the difference is only that input pin 8 is given RF carrier oscillator signal from RF carrier oscillator and pin 1 is given

AM modulated signal from balanced modulator section. The output is product of these two signals which contains basic audio modulating signal which can be filtered by low pass filter.

(8) Output audio amplifier section:

This section is same as input audio amplifier section except pre-amplifier section.

(9) Power supply section:

The regulated power supply is used for different supply voltages. Using step down transformer, diode bridge and IC7805, 7815, 7915 we can obtain different DC supply voltages required for the operation of different blocks.

Procedure:

Modulation:

1. Fig. 1.2 shows the AM transmitter panel. Ensure that the following initial conditions exist on the board.
Audio input select switch in INT position:
Mode switch in DSB position.
Output amplifier's gain pot in full clockwise position.
Speakers switch in OFF position.

2. Turn on power to the **ST2201** board.

3. Turn the audio oscillator block's amplitude pot to its full clockwise (MAX) position, and examine the block's output (t.p.14) on an oscilloscope

4. Monitor, in turn, the two inputs to the balanced modulator & band pass filter circuits 1 block, at t.p.1 and t.p.9

5. Next, examine the output of the balanced modulator & band pass filter circuit 1 block (at t.p.3), together with the modulating signal at t.p.1 Trigger the oscilloscope on the t.p. 1 signal.

6. To determine the depth of modulation, measure the maximum amplitude (Vmax) and the minimum amplitude (V min) of the AM waveform at t.p.3, and use the following formula:

$$\text{Percentage Modulation} = \frac{Vmax - Vmin}{Vmax + Vmin}$$

Where Vmax and Vmin are the maximum and minimum amplitudes.

7 Now vary the amplitude and frequency of the audio-frequency sinewave, by adjusting the amplitude and frequency present in the audio oscillator block. Note the effect that varying each pot has on the amplitude modulated waveform. The amplitude and frequency amplitudes of the two sidebands can be reduced to zero by reducing the amplitude of the modulating audio signal to zero. Do this by turning the amplitude pot to its MIN position, and note that the signal at t.p. 3 becomes an un-modulated sine wave of frequency 1 MHz, indicating that only the carrier component now remains.

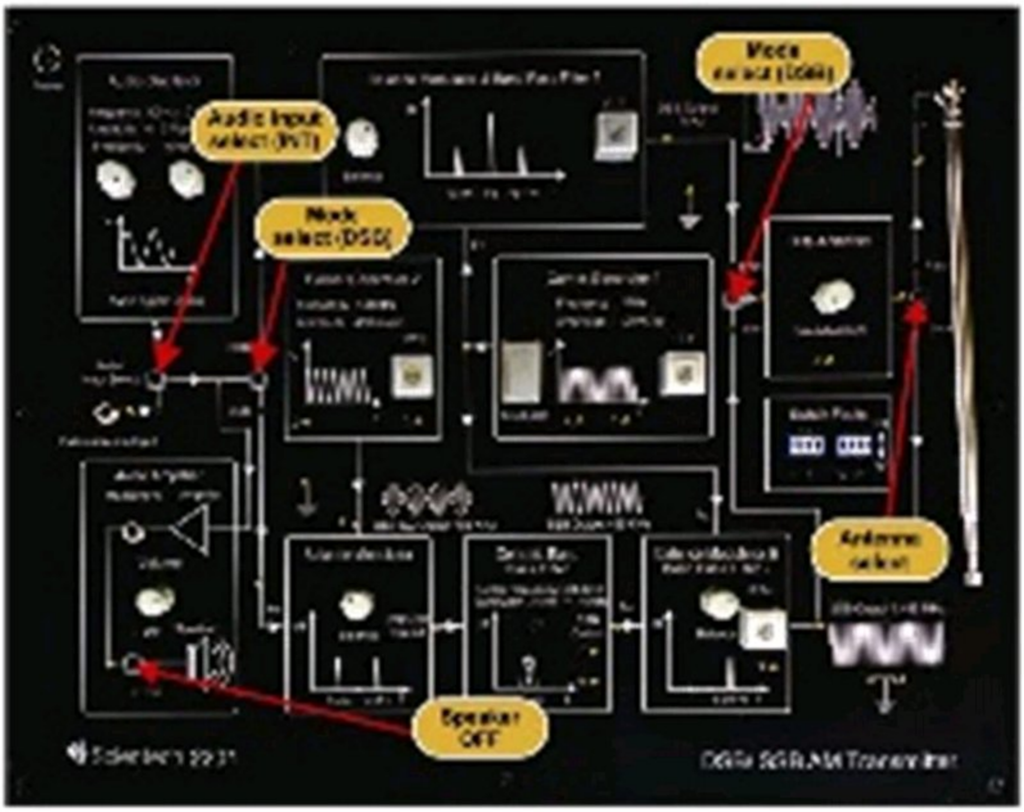


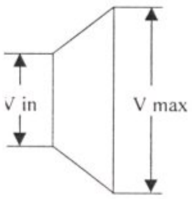
Fig. 1.2 AM transmitter

To calculate modulation index of DSB wave by trapezoidal pattern.

1. Repeat from step no. 1 to step no. 6

2. Now apply the modulated waveform to the Y input of the oscilloscope and the modulating signal to the X input.

3. Press the XY switch, you will observe the waveform similar to the one given below:



Calculate the modulation index by substituting in the formula

$$\text{Percentage Modulation} = \frac{V\max - V\min}{V\max + V\min}$$

4 Some common trapezoidal patterns for different modulation indices are as shown:

Demodulation:

1. Fig. 1.3 shows the AM receiver panel. Position the **ST2201** & **ST2202** modules, with the **ST2201** board the left, and a gap of about three inches between them.

2. Ensure that proper initial conditions exist on the **ST2201 & ST2202** board

3. Turn on power to the modules. We will now transmit the SSB waveform to the **ST2202** receiver. The mode of transmission can be selected by a selection switch (i.e. by an antenna or by a link).

4. On the **ST2202** module, monitor the output of the IF amplifier 2 block (t.p. 28) and turn the tuning dial until the amplitude of the monitored signal is at its greatest. Check that you have tuned into the SSB signal, by turning **ST2201**'s amplitude pot (in the audio oscillator block) to its MIN position, and checking that the monitored signal amplitude drops to zero. Return the amplitude pot to its MAX position.

5. On the **ST2202** module, monitor the output of the product detector block (at t.p. 37), together with the output of the audio amplifier block (t.p. 39), triggering the scope with the later signal.

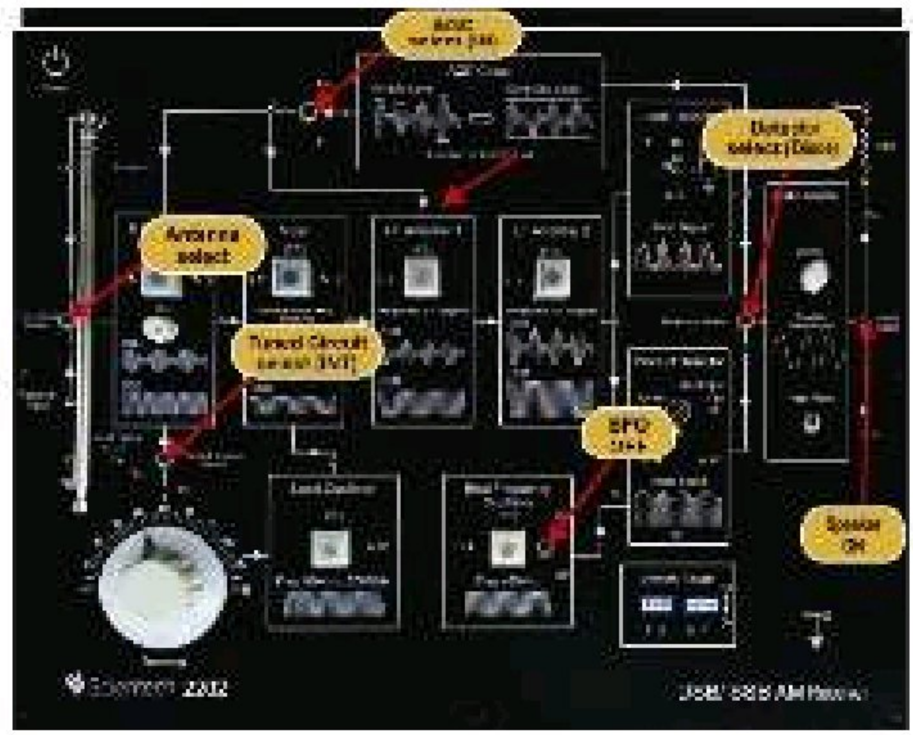


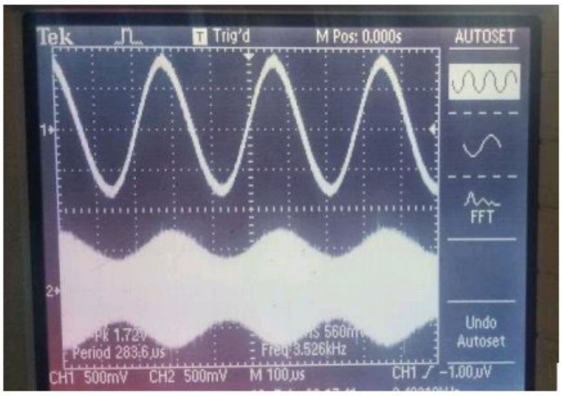
Fig. 1.3. AM Receiver

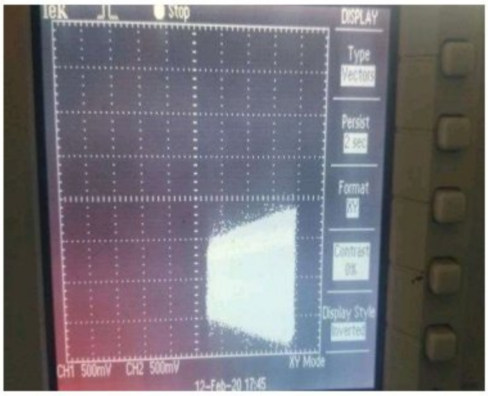
Observation Table:

In YT Plane			In XY Plane		
Modulation Index	V_{max}(volt)	V_{min}(volt)	Modulation Index	V_{max}(volt)	V_{min}(volt)
0.33	1.68	0.84	0.33	1.8	0.9
1	0.64	0	1	0.7	0
1.84	0.54	-0.16	2	0.6	-0.2

Output Waveforms:

A. $m_a < 1$

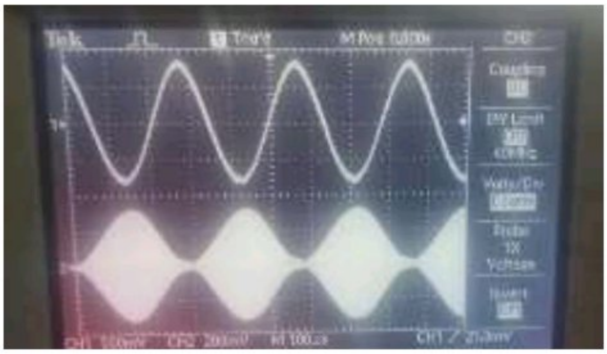


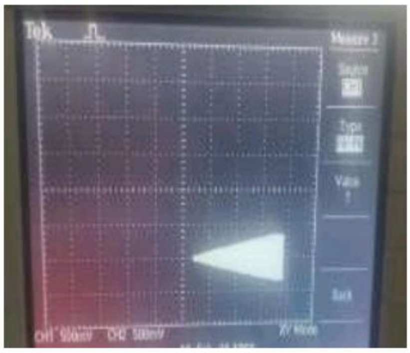


In YT Plane

In XY Plane

B. $m_a = 1$

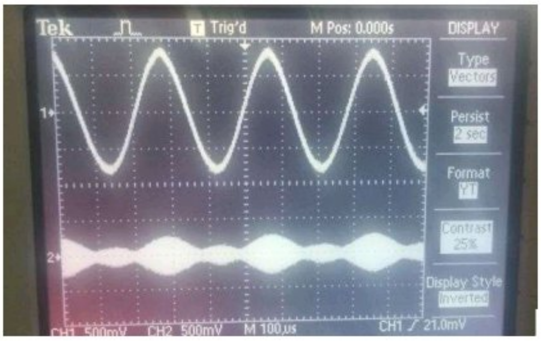




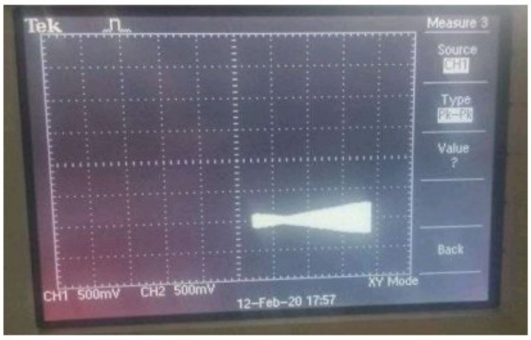
In YT Plane

In XY Plane

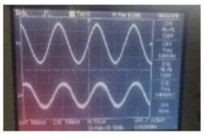
C. $m_a > 1$



In YT Plane



In XY Plane



Output From Demodulator

<u>Conclusion</u>:

After performing this experiment, we understood about role of Modulation Index in Amplitude Modulation and How Signal vary at Different Value of Modulation Index And at receiver Side Intermediate Frequency.

<u>Remarks</u>:

<u>Signature</u>:

PULSE AMPLITUDE MODULATION/DEMODULATION

Experiment No.: 5 **Date:20/02/2020**

Aim: To study Pulse Amplitude Modulation:

8. To modulate signal by Pulse Amplitude Modulation Scheme using Natural & Flat top sampling.
9. To demodulate signal by Pulse Amplitude Modulation Scheme using Sample & Hold, Flat Top.
10. Verify the sampling theorem by changing modulating & carrier frequency.

Apparatus: ST2110 PAM trainer kit, CRO, Connecting Leads, Probes Etc

Procedure:

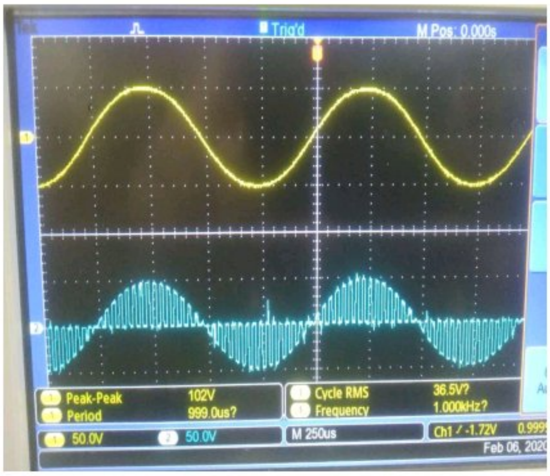
7. Connect the circuit as shown in Fig. 1.1.

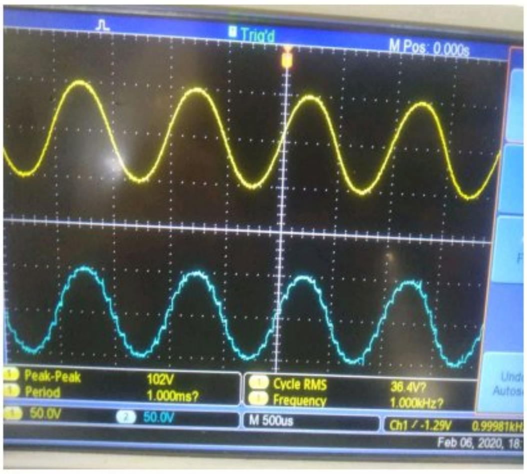
    Output of Sine wave to Modulation Signal in PAM block keeping the switch in 1 kHz position

    8 kHz pulse output to Pulse IN
8. Switch On power supply.
9. Monitor the outputs at tp. 3, 4 and 5, these are natural, Sample and Hold flat top outputs respectively.
10. Observe the difference between the two outputs and try giving reasons behind them.
11. Try Varying the amplitude and frequency of sine wave by amplitude pot and frequency change over switch. Observe the effect on all the two outputs.
12. Also, try varying the frequency of pulse, by connecting the pulse input to the 4 frequencies available i.e. 8.16.32. 64 kHz in pulse output look.
13. For demodulation part Connect the sample output low pass filter input and Output of low pass filter to input of AC amplifier. Keep the gain pot in AC amplifier block in max position.
14. Follow the steps as of modulation part.
15. Monitor the output of AC amplifier. It should be a pure sine wave similar to input.
16. Similarly connect the sample and hold and flat top outputs to Law Pass Filter and see the demodulated waveform at the output of AC amplifier.

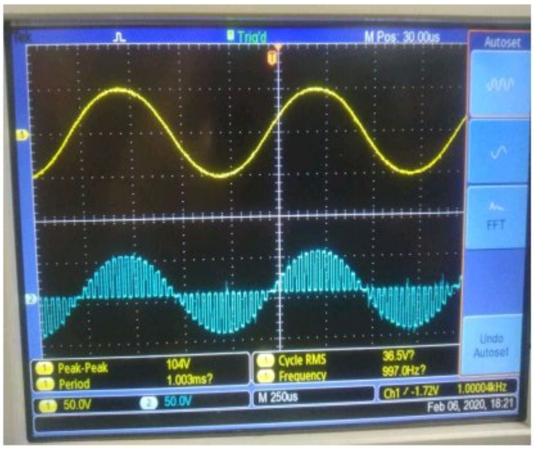
Observation: Output Voltage:10.2 V peak to peak.

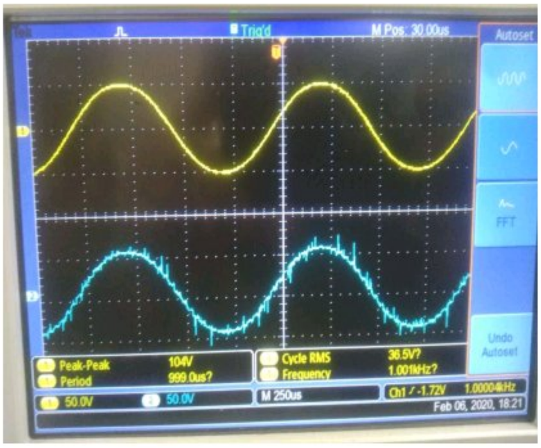
Output Waveforms: Message frequency:1Khz
sample frequency:16Khz





Sample amplifier output **Sample and hold amplifier output**





Flat top Output **Recovered Signal**

<u>Conclusion:</u>

We performed Pulse Amplitude Modulation/Demodulation using Sample , Sample and Hold , Flat top Amplifiers and Recovery of Original signal using Low Pass Filter.

<u>Remarks</u>: **<u>Signature</u>:**

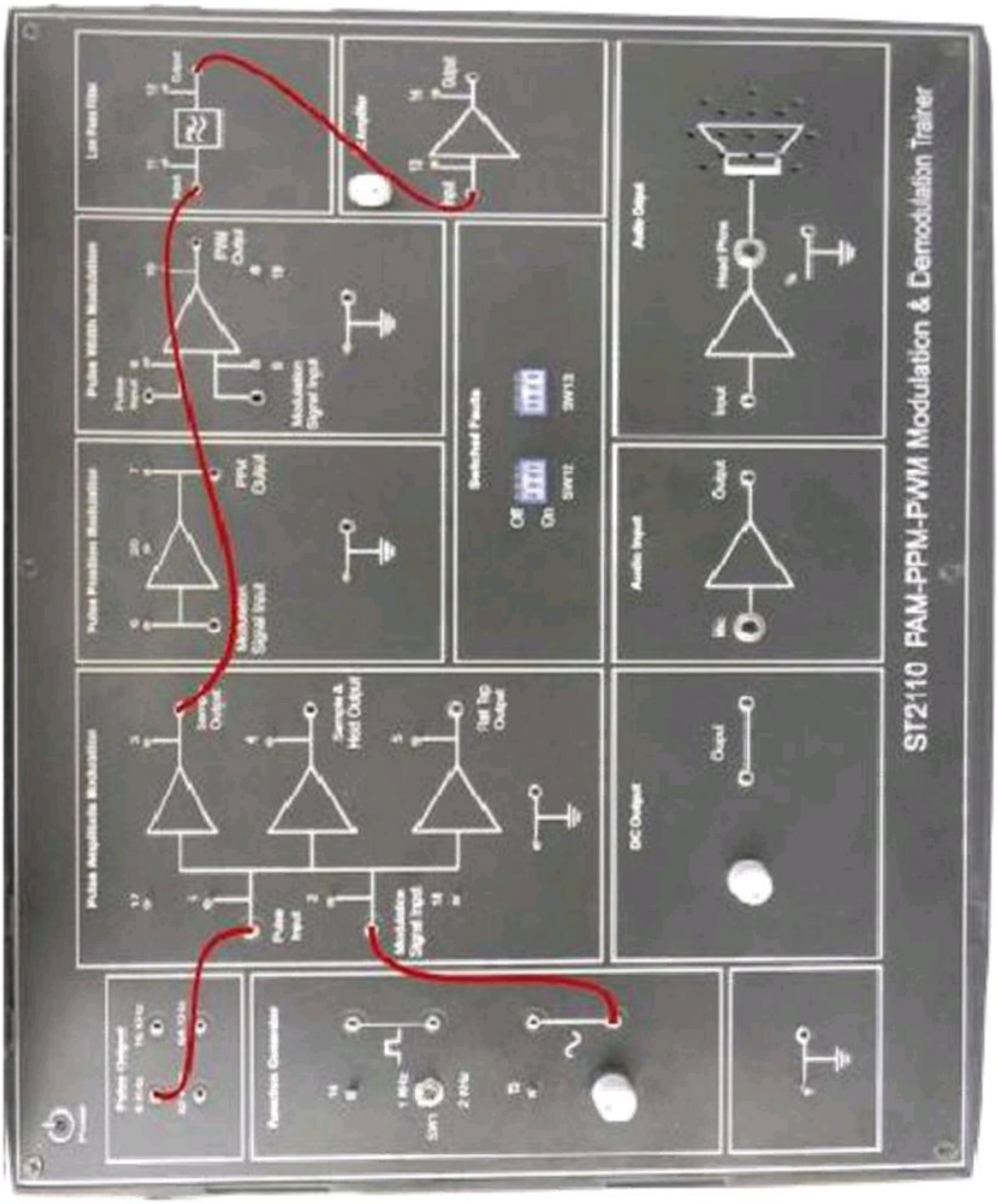


Fig. Connection diagram for Pulse Amplitude Modulation and Demodulation

PULSE POSITION MODULATION/DEMODULATION

Experiment No.: 5 **Date:20/02/2020**

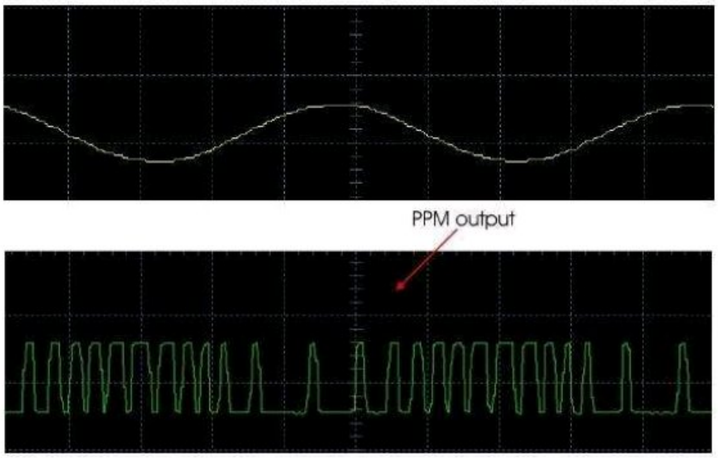
Aim: To study Pulse Position Modulation/Demodulation:

Apparatus: ST2110 PPM Trainer kit, CRO, Connecting Leads, Probes Etc

PART A: PPM using DC input
Procedure:

1. Connect the power supplies of ST2110. Make the connections as shown in the Fig.
1.1 Connect the DC output to input of PPM block .Switch 'ON' the power.
2. Observe the output of PPM block at TP7.
3. Vary the DC output while observing the output of PPM block.
4. Switch 'On' the switched faults No. 1, 2, & 6 one by one & observe their effects PPM input and try to locate them.

Observation:



PPM output waveform

PART B: PPM using sine wave input
Procedure:

1. Connect the power supplies of ST2110. Make the connections as shown in the Fig. 1.2. Connect the sine wave output of FG block to input of PPM block .Switch 'ON' the power.
2. Keep the oscilloscope at 0.5mS / div, time base speed and in X-5 mode, and observe the pulse position modulated waveform at the pulse position modulation block output at TP 7.
3. Vary the amplitude of sine wave and observe the pulse position modulation, keep the amplitude preset in centre. Here you can best observe the pulse modulation.
4. Switch 'On' the switched faults No. 1, 2, & 6 one by one & observe their effects PPM input and try to locate them.

Observation:
Same as in Part A

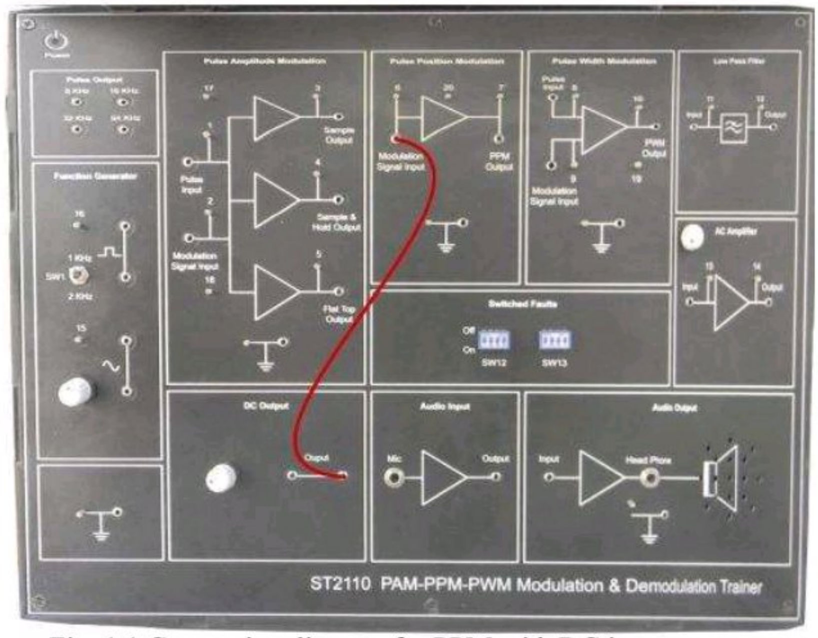


Fig. 1.1 Connection diagram for PPM with DC input

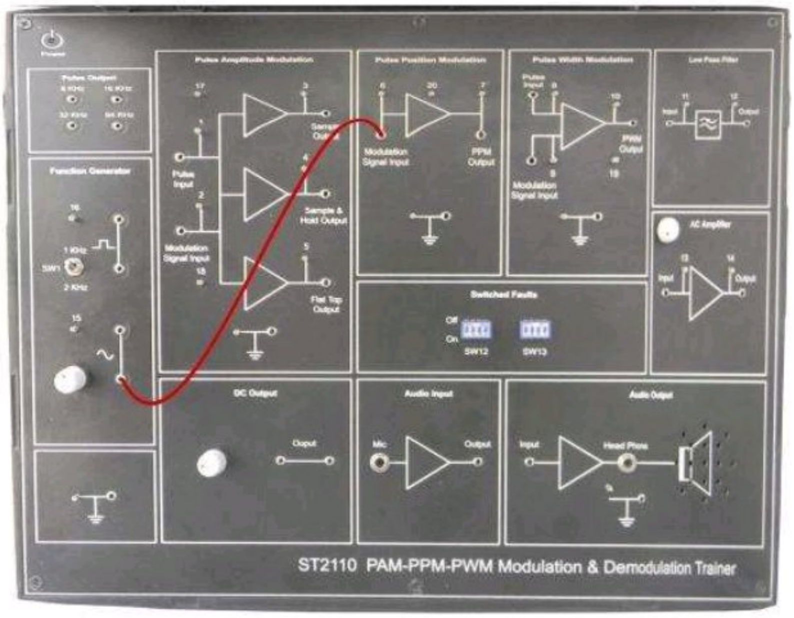


Fig. 1.2 Connection diagram for PPM with Sinusoidal input

Part C: Pulse Position Demodulation:

1 Make the connections as shown in Fig. 1.3. Switch 'On' the power supply & oscilloscope.

2 Observe the waveform at the TP12 output of low pass filter block.

3 Then observe the demodulated output at TP14 output of AC amplifier.

4 Switch 'On' the switched faults No. 1, 2, & 6 one by one & observe their effects PPM input and try to locate them.

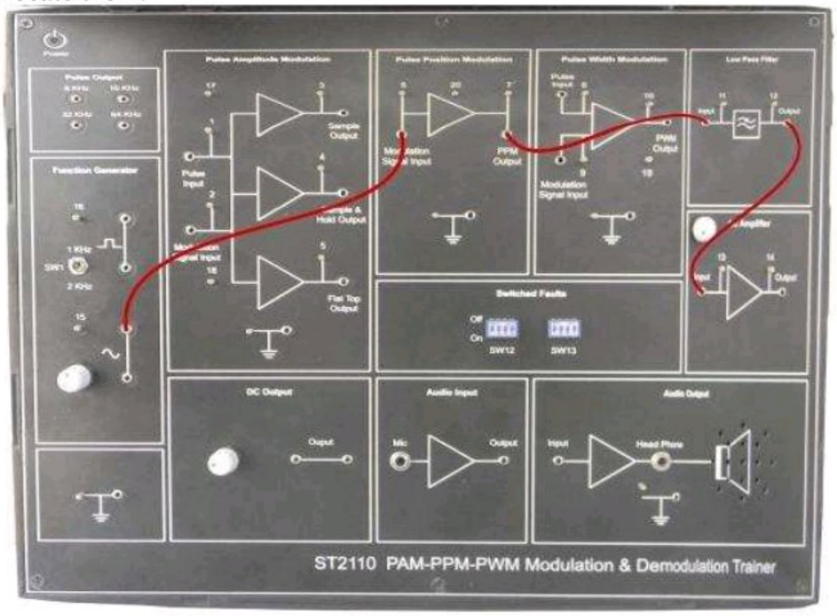
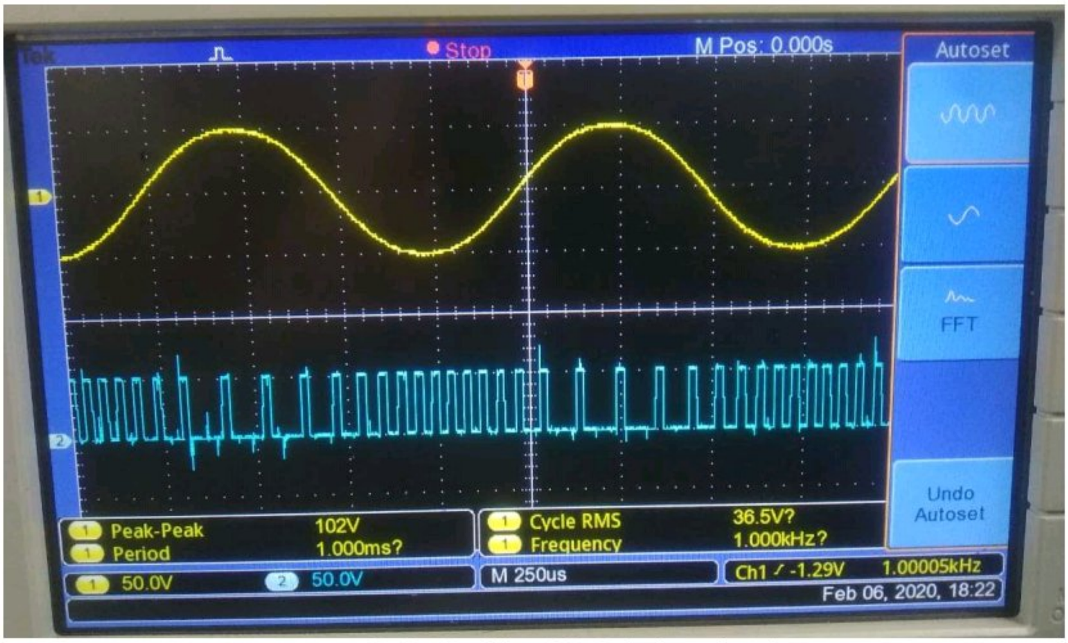


Fig. 1.3 Connection diagram for PPM demodulation

Output Waveforms:



Pulse Position Modulation signal

Recovered Signal

Conclusion:

We performed Pulse Position Modulation/Demodulation and Recovery of Original signa using Low Pass Filter and AC Rectifier

Remarks:

Signature:

PULSE WIDTH MODULATION/DEMODULATION

Experiment No.: 5 **Date:20/02/2020**

Aim: To study Pulse Width Modulation/Demodulation:

Apparatus: ST 2110 PWM Trainer kit, connecting chords, power supply, DSO/CRO.

Procedure:

4. Connect the circuit as shown in Figure 1.1. Switch on the power supply.
5. Observe the output on PWM output block.
6. Vary the amplitude of sine wave and see its effect on pulse output.
7. Vary the sine wave frequency by switching the frequency selector switch to 2 KHz.
8. Also, change the frequency of the pulse by connecting the pulse input to different pulse frequencies viz. 8 KHz, 16 KHz, 32 KHz and see the variations in the PWM output.
9. Switch 'On' fault No. 1, 2, & 5 one by one & observes their effect on PWM output and try to locate them.
10. Connect the output of PWM to the low pass filter and the output of the low pass filter to the AC amplifier. Observe the demodulated output.

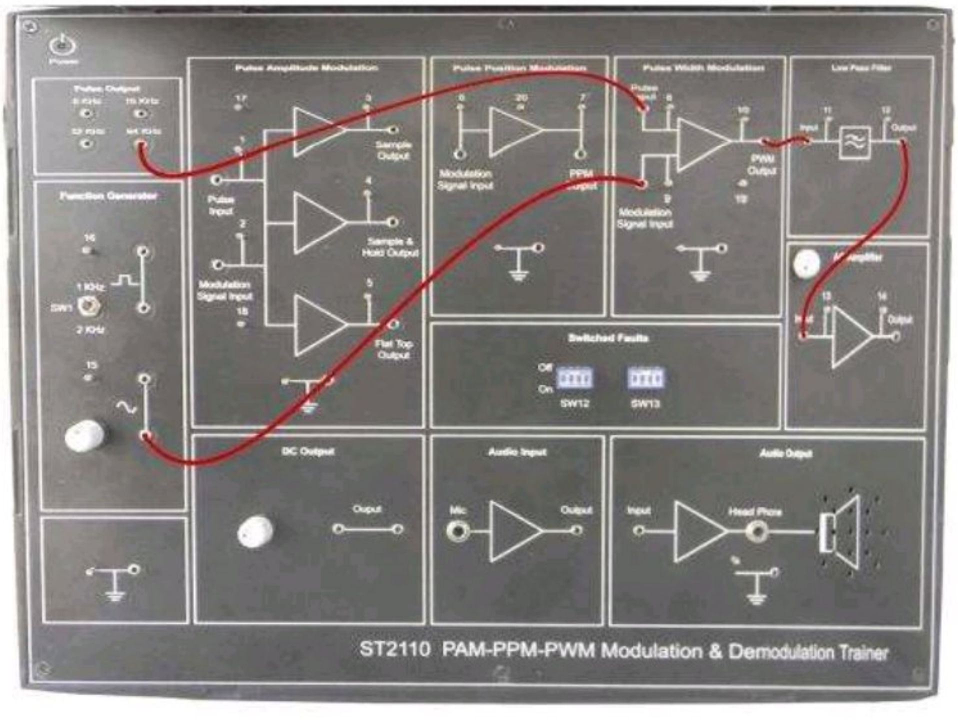
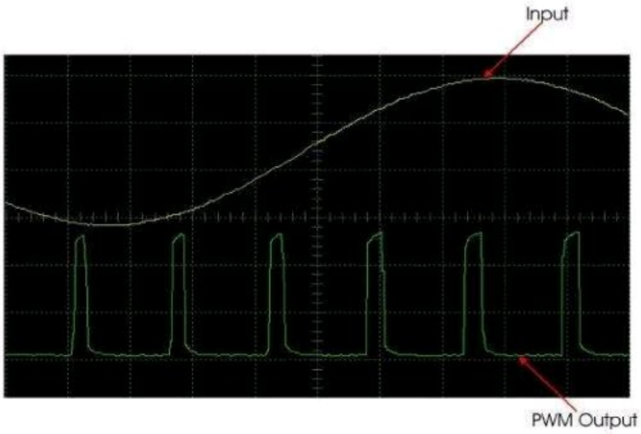
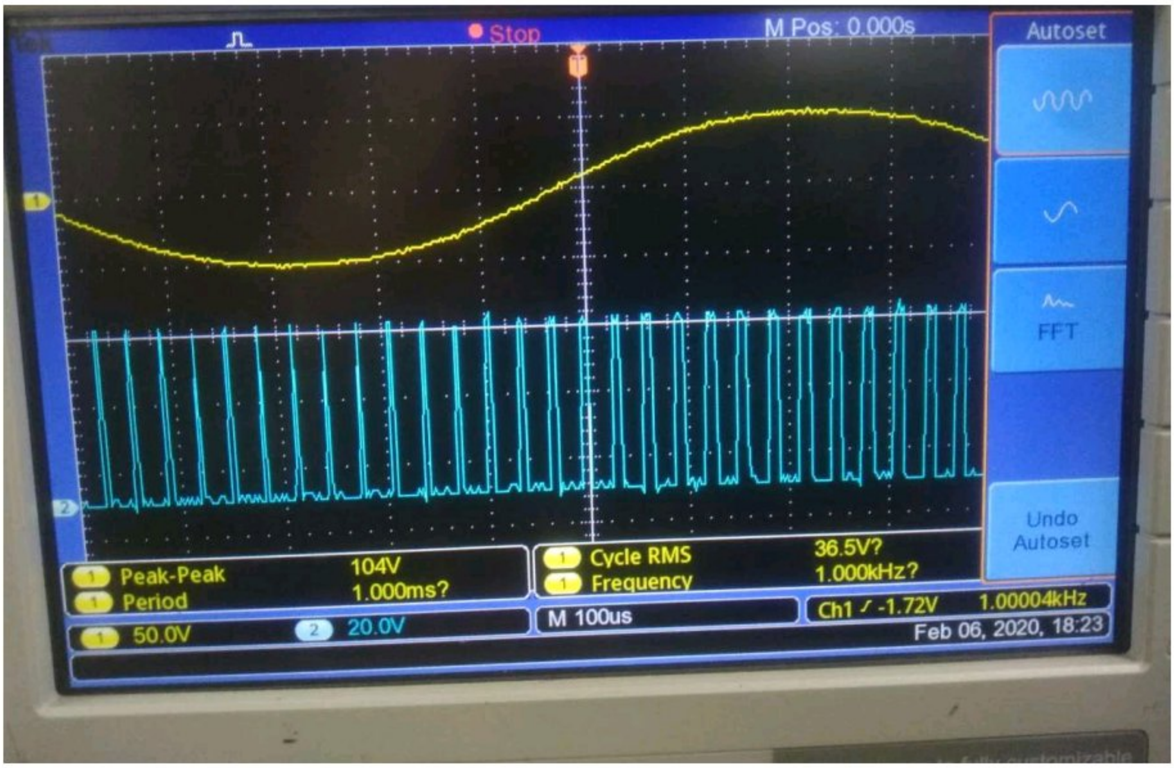


Fig. 1.1 Connection diagram for PWM modulation and demodulation

Observation:



Output Waveforms:



Pulse Width Modulation

Recovered Signal

Conclusion:

We performed Pulse Width Modulation/Demodulation and Recovery of Original signal using Low Pass Filter and AC Rectifier .

Remarks: **Signature**:

FREQUENCY MODULATION (FM)

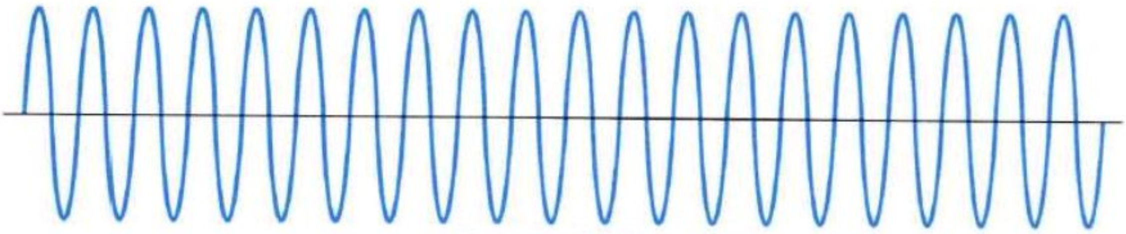
Experiment No: 6 **Date:**

Aim: To study frequency modulation and demodulation and observe the waveforms.

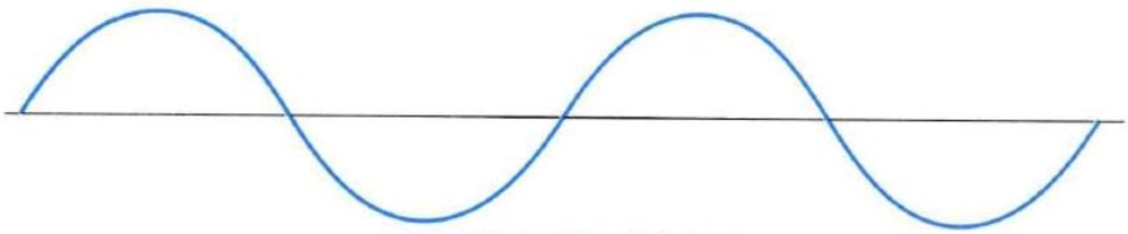
a) **Observe the spectra of FM signal in labAlive virtual communication lab and Calculate the modulation index for FM**
b) **To perform FM transmission via virtual lab labAlive for the audio signal**
c) **To perform FM reception via virtual lab labAlive for the obtained recorded signal**

Frequency Modulation (FM):

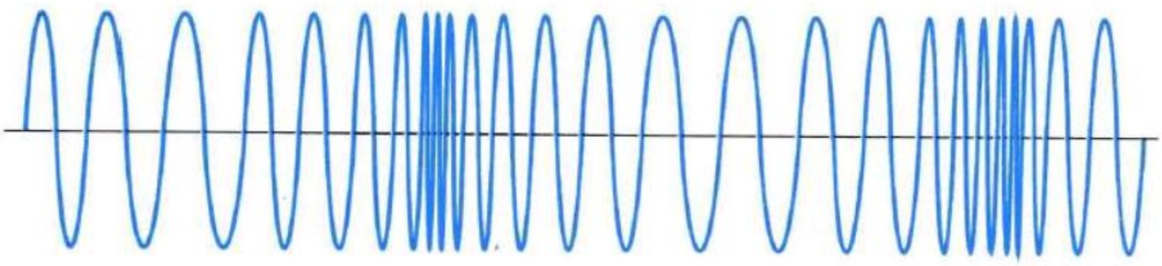
The frequency of the carrier waveform varies with the information signal



Carrier Wave



Modulating Wave



Frequency Modulated Wave

Frequency modulation is a system in which the amplitude of the modulated carrier is kept constant, while its frequency is varied by the modulating signal, the modulating signal is sinusoidal. This signal has two important parameters which must be represented by the modulation process without distortion: namely its amplitude and frequency.

If carrier signal, $e_c = E_c \, sin \, sin \, \omega_c t$ and modulating signal, $e_m = E_m \, sin \, sin \, \omega_m t$ then, the peak or maximum frequency deviation:

$\Delta f \; \alpha \;\; e_m$

$\Delta f = k_f e_m$

Where, k_f is proportionality constant[V/Hz], and e_m is the instantaneous value of the modulating signal amplitude. Thus the frequency of the FM signal is:

$$e_s(t) = e_c + \Delta f = e_c + k_f e_m(t)$$

Then, $e_s(t) = e_c + k_f E_m \, sin \, sin \, \omega_m t$

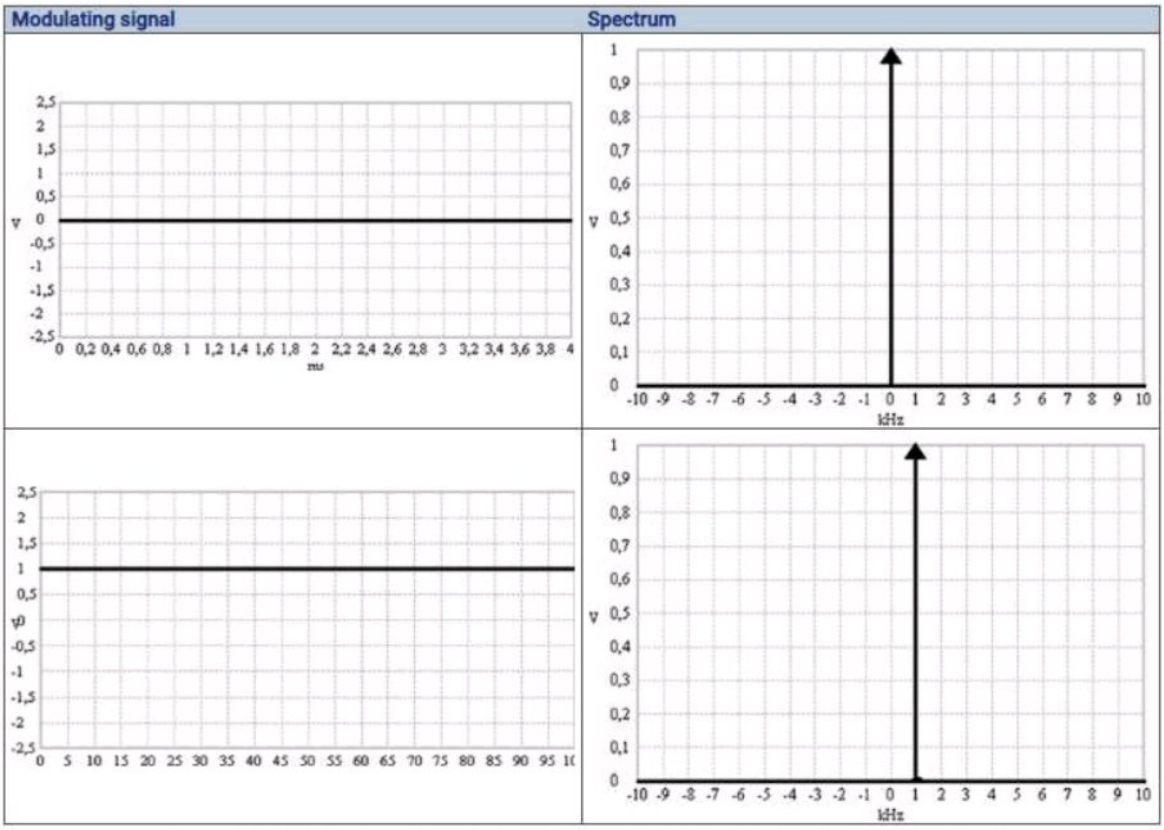
Then the equation for the FM signal is:

$$e_s(t) = E_c \, sin \, sin \, (\omega_c t + \beta \;\; sin \, sin \, \omega_m t\,)$$

Where, β = modulation index, which can be greater than 1. It is measured in radians

β = Freq. Deviation / Modulating Freq.

$$\beta = \frac{\Delta f}{f_m}$$

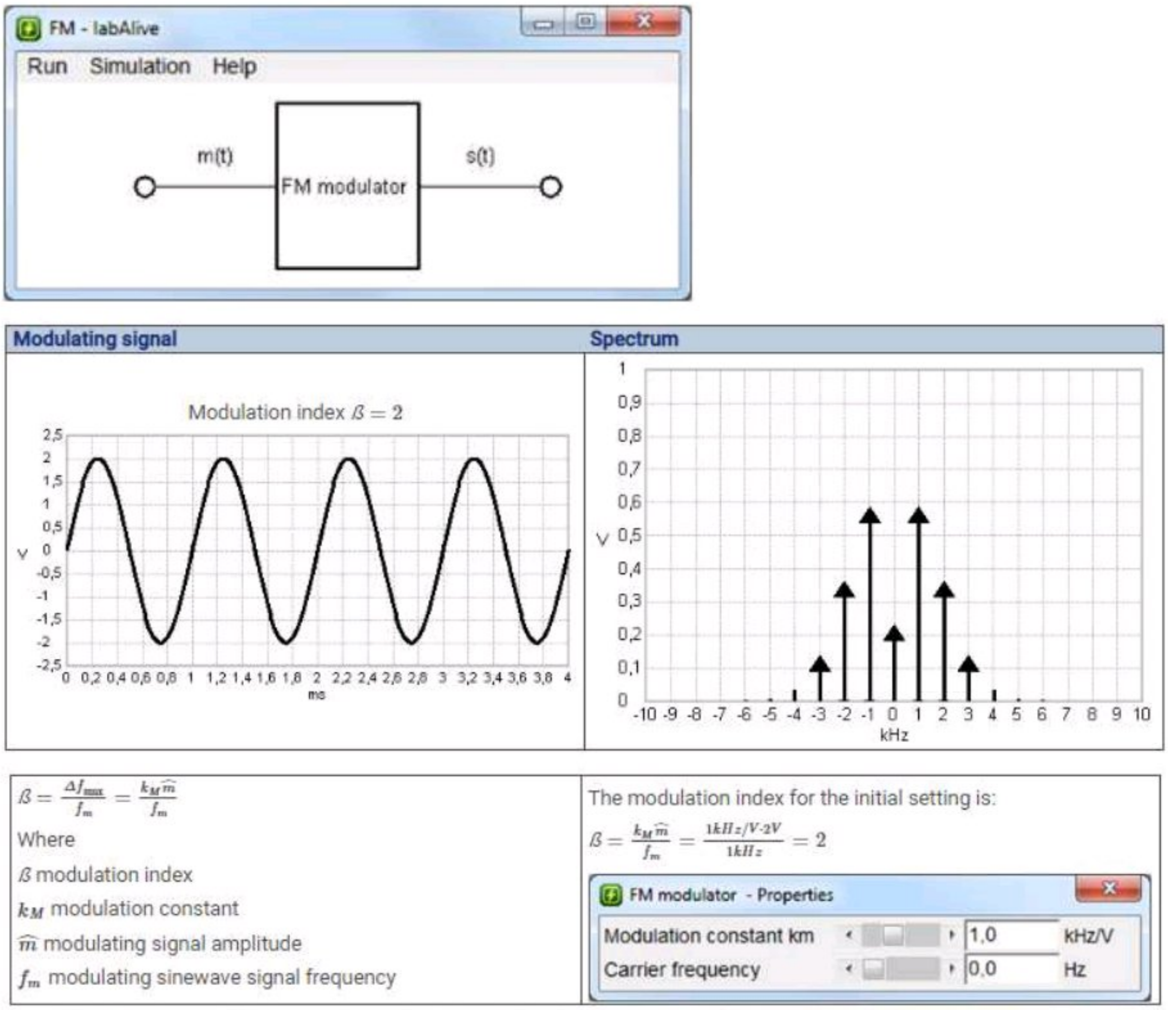


Frequency modulation example - frequency deviation is 1 kHz for a 1V-DC modulating signal

Part a): In this experiment a sinewave signal is frequency modulated. Modulating signal and modulator parameters determine the spectrum of the resulting FM transmission signal.

Procedure:

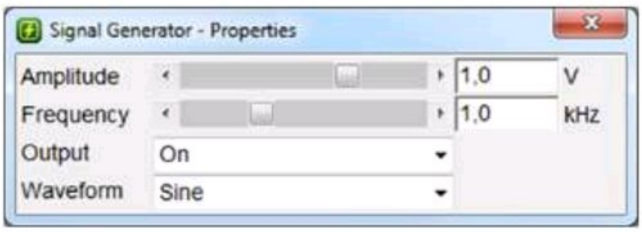
- On launching the experiment, you will see the following windows:



The modulation index ß is the ratio of the maximum frequency deviation of the carrier to the frequency of the sinewave modulating signal.

The Bessel function values at the resulting modulation index determine the spectrum of the FM signal.

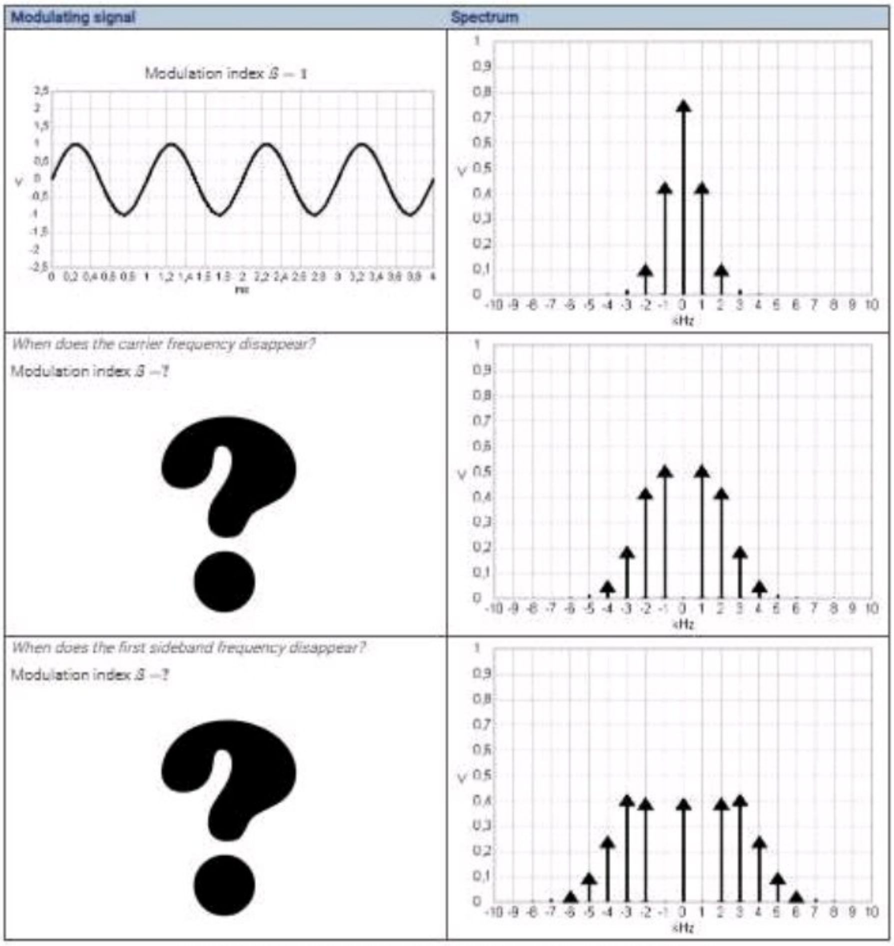
- Vary the modulating signal amplitude $\widehat{m}$.



- The modulation index is proportional to the modulating signal amplitude. In this setting the amplitude in Volts is the modulation index:

$$\beta = \frac{\Delta f}{f_m} = \frac{k_f \widehat{m}}{f_m} = \frac{\frac{1kHz}{V} \times \widehat{m}}{1kHz} = \frac{\widehat{m}}{V}$$

- The adjusted modulating signal amplitude determines the spectral amplitudes of the carrier and sideband frequencies. For some values the carrier or specific sideband frequencies disappear. This relates to zero crossings of the respective Bessel function at the corresponding modulation index.



FM signal spectra for sinewave modulation with different modulation indices.

The carrier frequency is 0 Hz in this setting. It might be changed via the modulator properties.

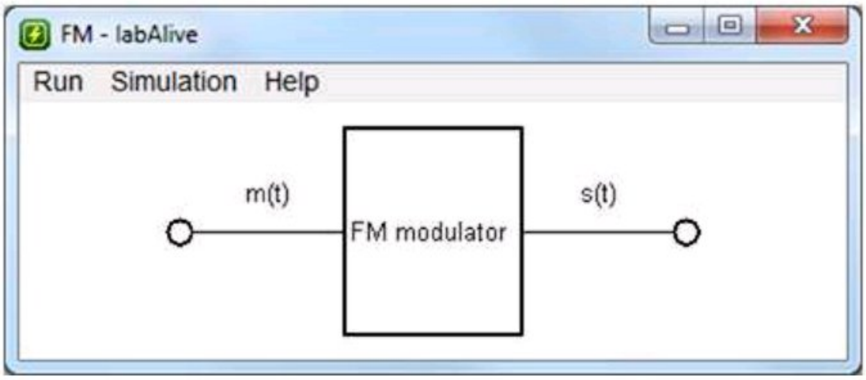
NEXT STEPS

- When do the 2nd and 3rd sideband frequencies disappear?

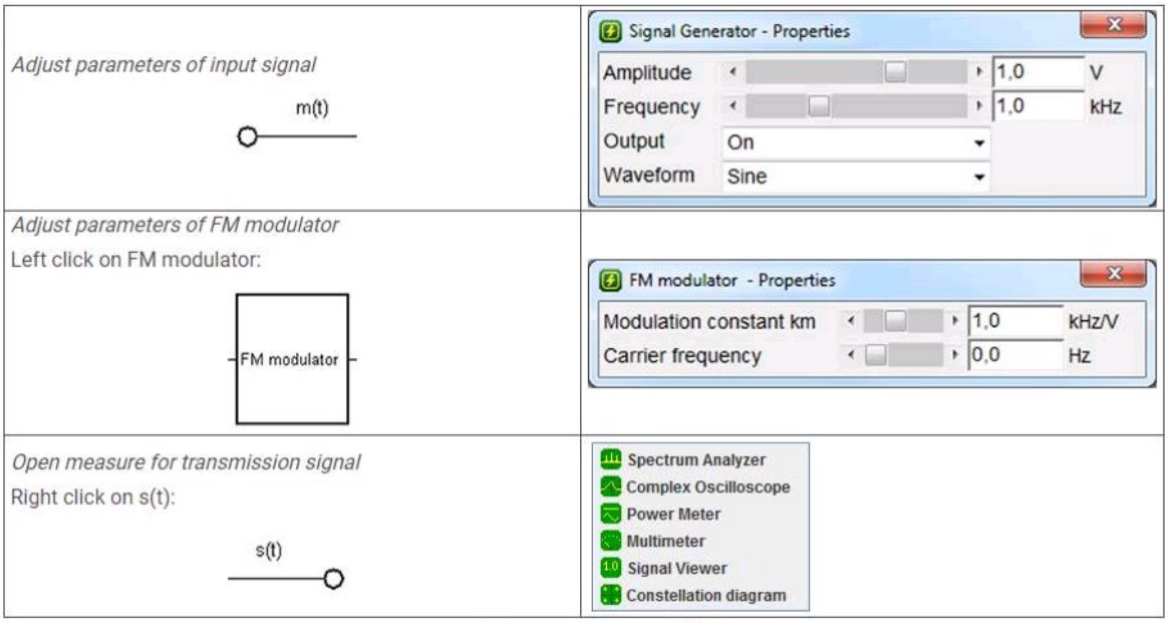
  ____

- Vary the modulating sinewave signal frequency.
- Select different waveforms (signal generator properties) and regard the FM spectrum.
- Use the Bessel functions to determine the spectrum of an FM signal with $\beta = 3$

This simulation implements frequency modulation. The FM signal is generated for the chosen modulating signal. Its spectrum is shown in a spectrum analyzer. All parameters of the modulating signal and modulator can be adjusted.



To change the different settings click on the corresponding wiring:



Adjust parameters of input signal m(t)	Signal Generator - Properties Amplitude 1,0 V Frequency 1,0 kHz Output On Waveform Sine
Adjust parameters of FM modulator Left click on FM modulator: FM modulator	FM modulator - Properties Modulation constant km 1,0 kHz/V Carrier frequency 0,0 Hz
Open measure for transmission signal Right click on s(t): s(t)	Spectrum Analyzer Complex Oscilloscope Power Meter Multimeter Signal Viewer Constellation diagram

Adjust parameters of FM

- Click on the launch tab on the link https://www.etti.unibw.de/labalive/experiment/fm/
- Change the amplitude and frequency of the modulating signal, observe the spectra (attach the output waveforms you observe) and complete the following observation table.
- Note down the frequency deviation from the spectra of FM signal as suggested in the figures below

Reference: Communication Systems by Simon Haykin, 4th edition. Refer Example 2.2 on page 116-117.

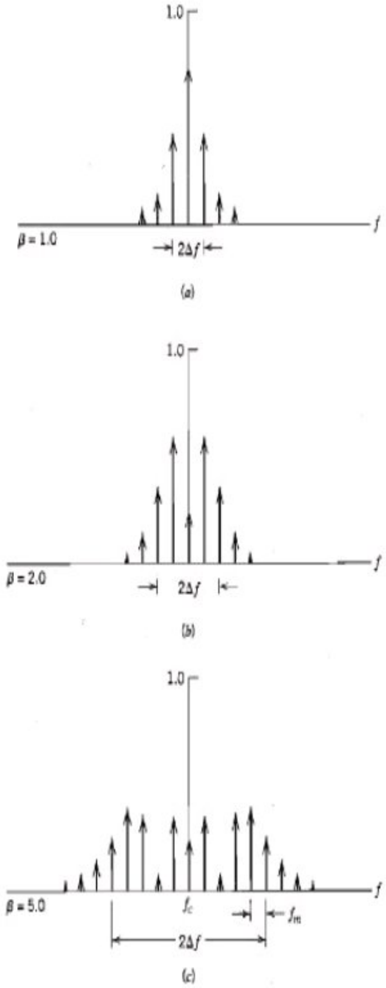


FIGURE 2.24 Discrete amplitude spectra of an FM signal, normalized with respect to the carrier amplitude, for the case of sinusoidal modulation of fixed frequency and varying amplitude. Only the spectra for positive frequencies are shown.

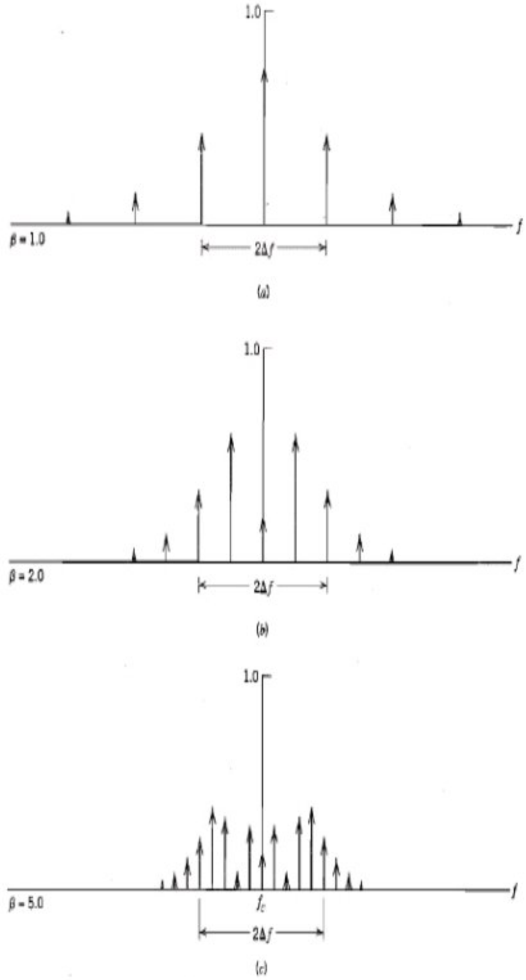
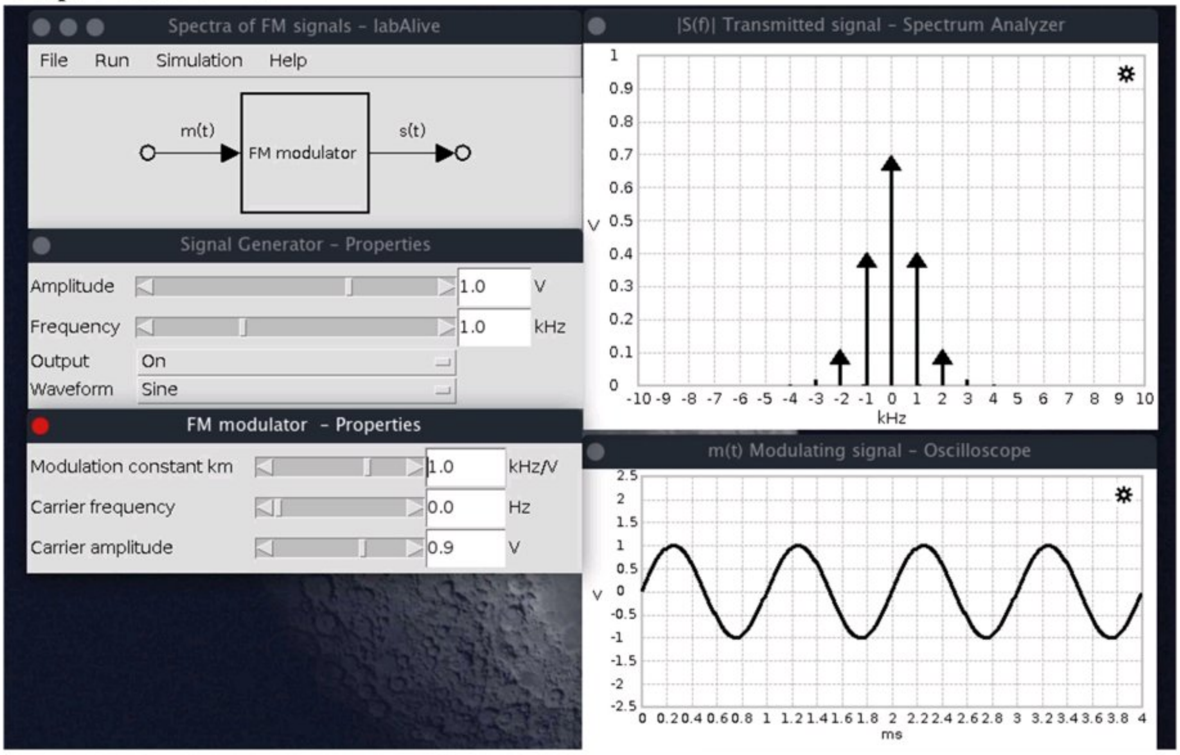


FIGURE 2.25 Discrete amplitude spectra of an FM signal, normalized with respect to the carrie amplitude, for the case of sinusoidal modulation of varying frequency and fixed amplitude. Only the spectra for positive frequencies are shown.

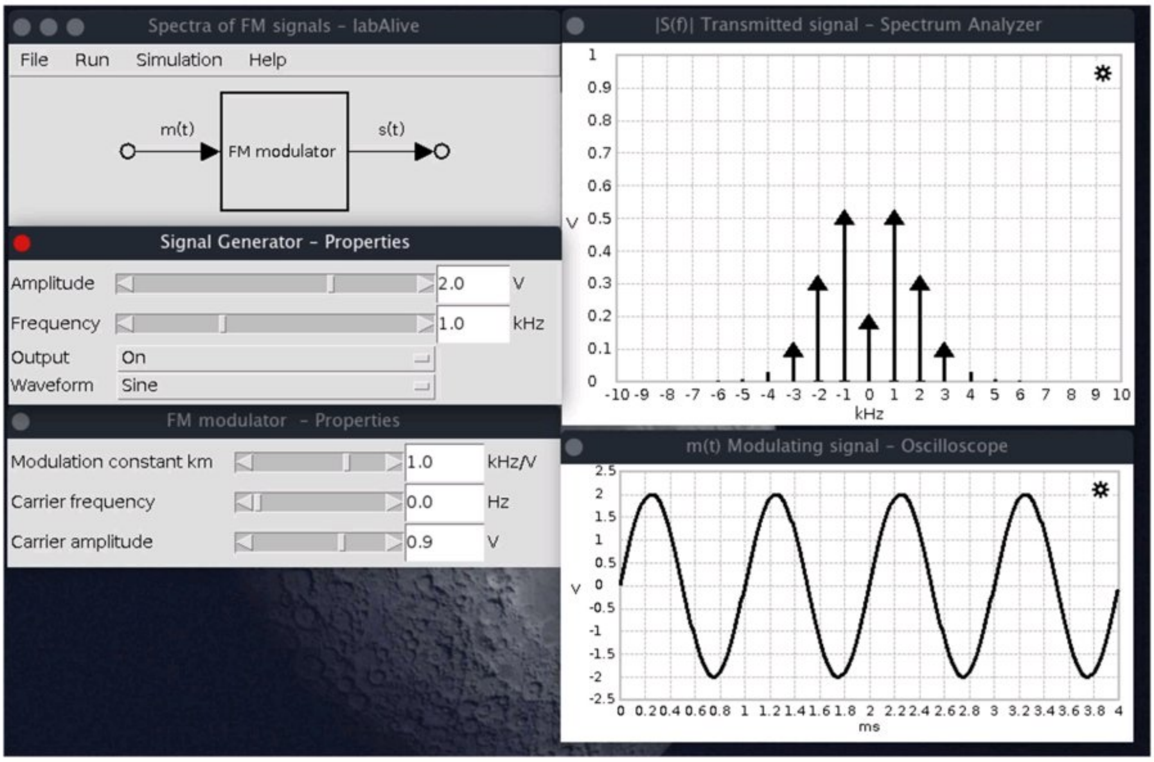
Sr. no	**Modulating Signal frequency, f_m**	**Modulating signal amplitude, $\hat{m}$**	**Frequency Deviation, Δf**	**Modulation Index, $\beta = \frac{\Delta f}{f_m}$**
1	1kHz	1.0V	1.0	1
		2.0V	2.0	2
2	2kHz	2.0V	3.0	1.5
		3.0V	4.5	2.25
3	3kHz	3.0V	9	3
		2.0	6	2

Follow this link for detailed procedure and setup https://youtu.be/THzJ6bjf1HI, https://www.etti.unibw.de/labalive/experiment/fm/

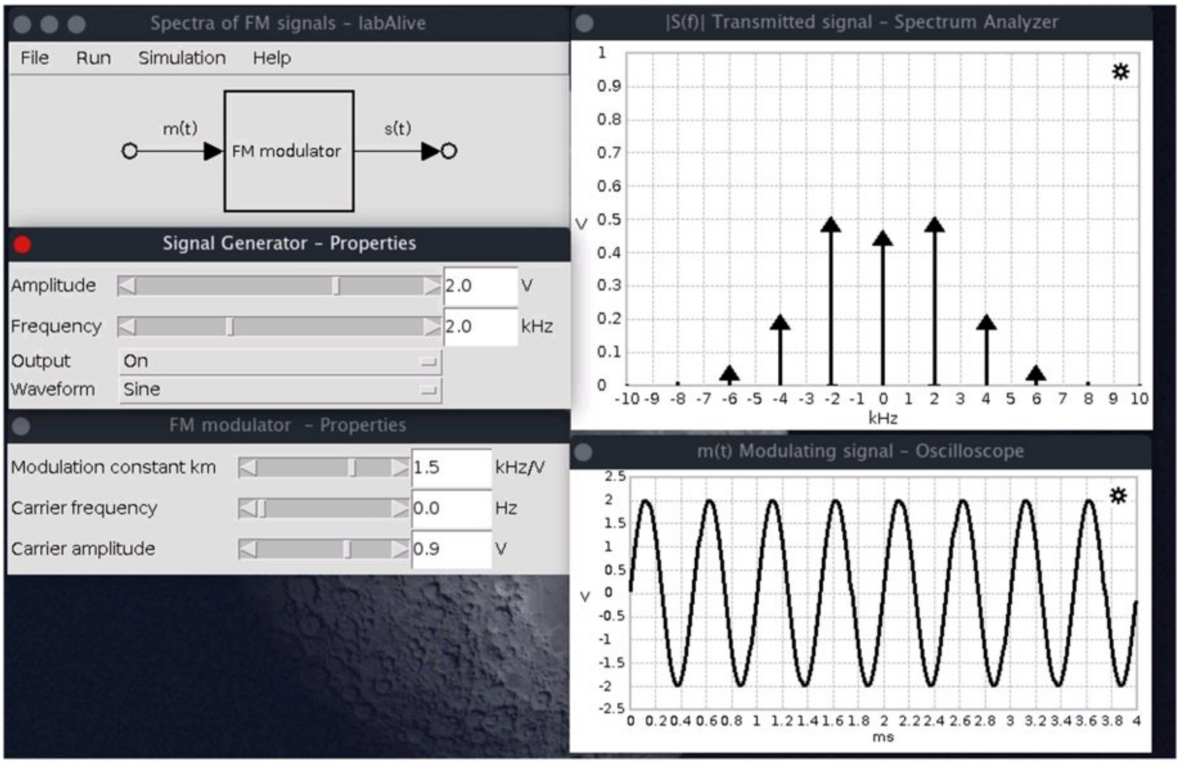
Output Waveforms:



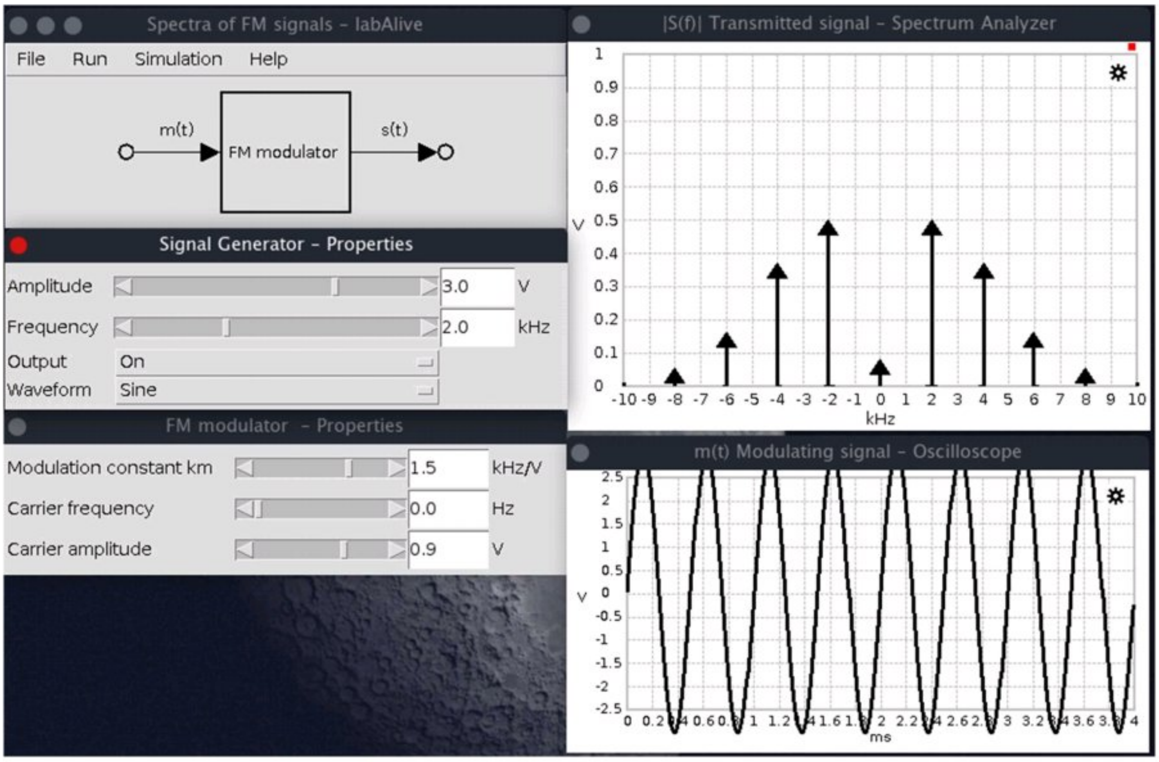
fm = 1kHz m=1.0V Beta=1



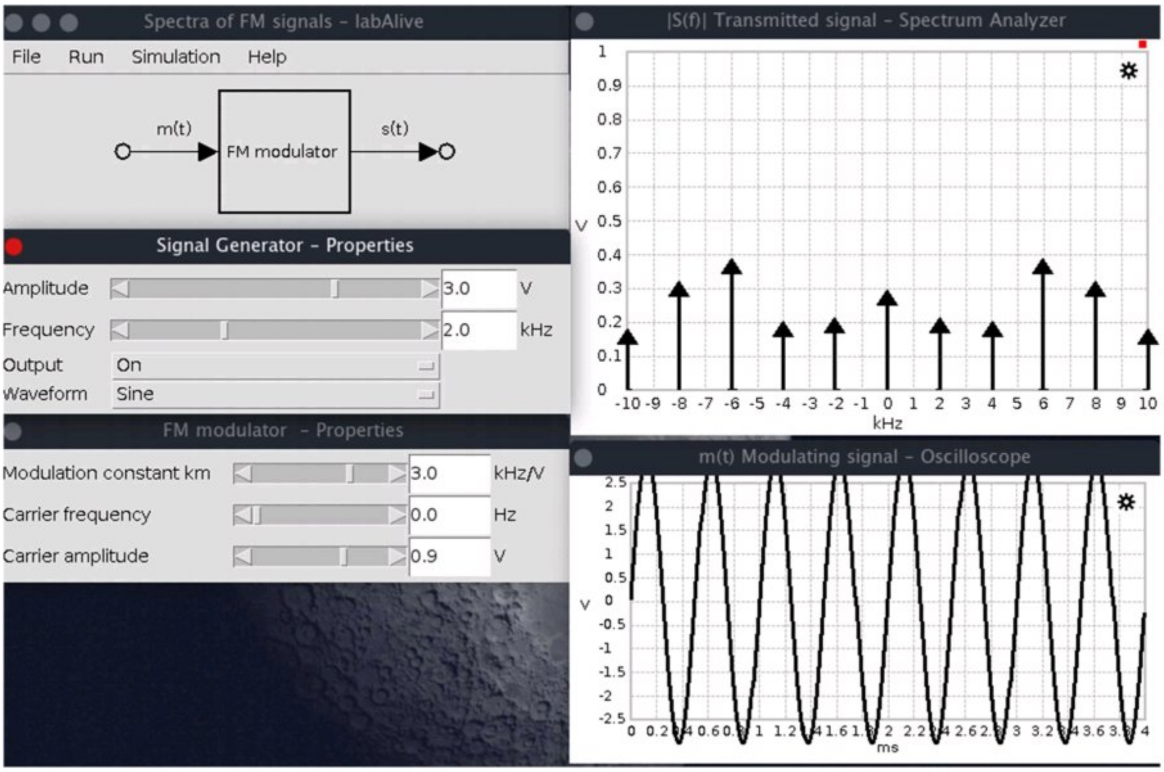
fm = 1kHz m=2.0V Beta=2



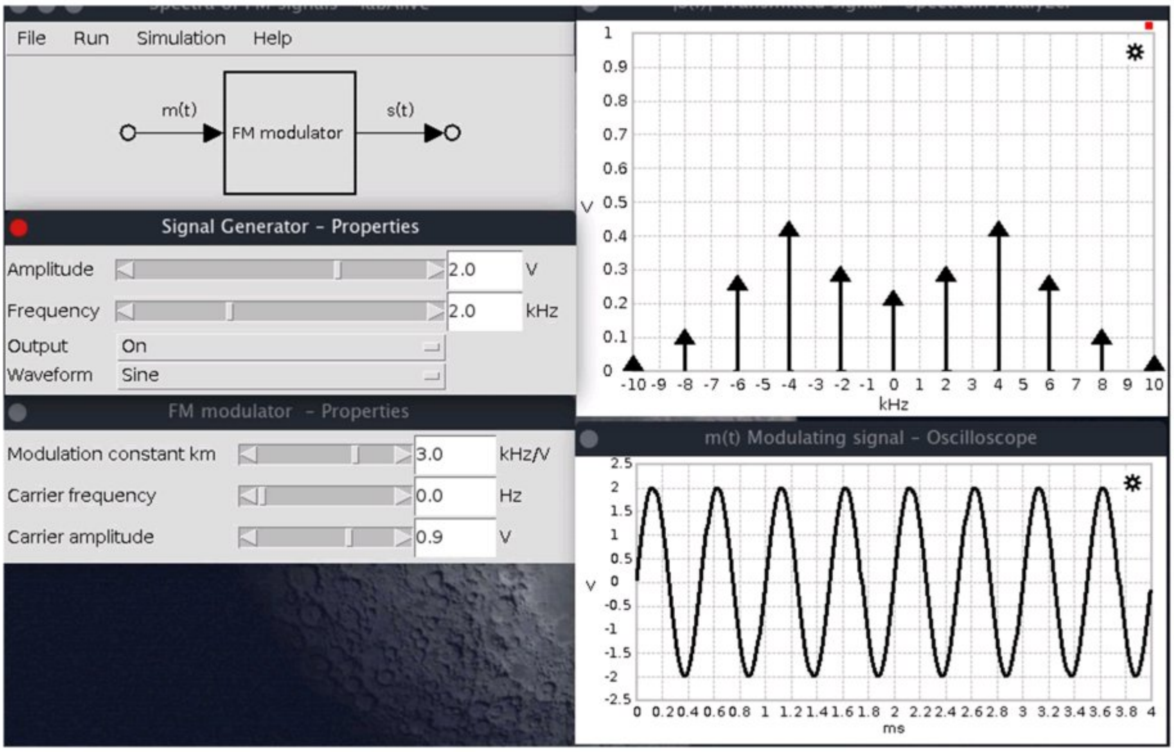
fm = 2kHz m=2.0V Beta=1.5



fm = 2kHz m=3.0V Beta=2.25



fm = 3kHz m=3.0V Beta=3

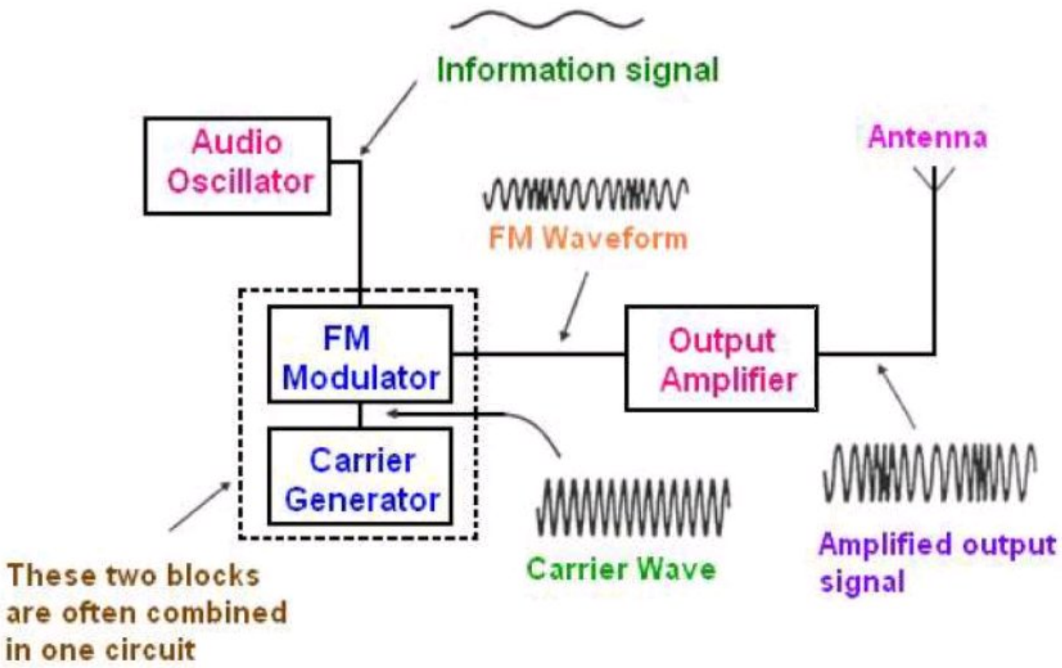


fm = 3kHz m=2.0V Beta=2

Part b) To send an audio file via FM transmission link

FM Transmitter:

The block diagram is shown in figure.



Procedure:

START

Initially a music signal provided by the server is frequency modulated. You might select your own audio file in the format 44.1kHz, 16 bit, stereo or the microphone *Line in*.

SET FREQUENCY DEVIATION

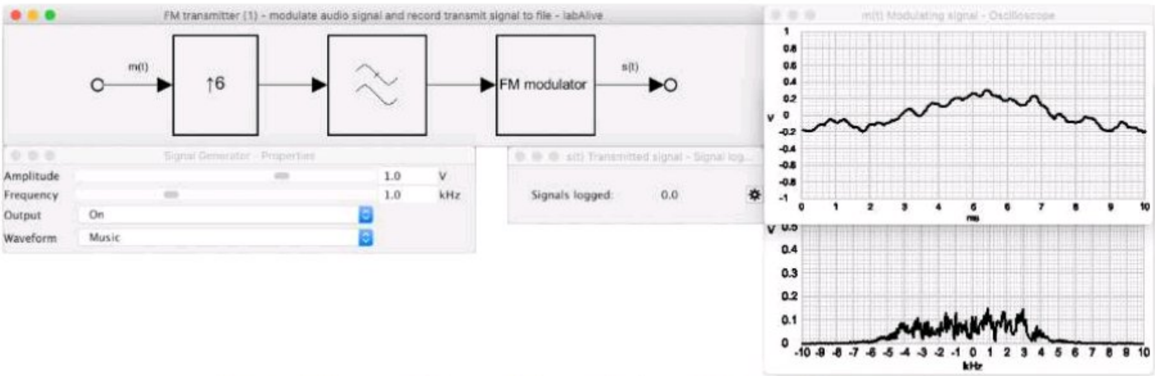
Increase the signal generator's amplitude so that the frequency deviation, i.e. the maximum shift away from the carrier frequency, is 75 kHz. In this base band simulation the carrier frequency is set to 0 Hz.

RECORD THE FM MODULATED SIGNAL TO A FILE

Open the settings of the signal logger - click the gear wheel settings icon. Click *Start save samples to file* to record the transmitted signal to a file.

DEMODULATE THE FM MODULATED SIGNAL

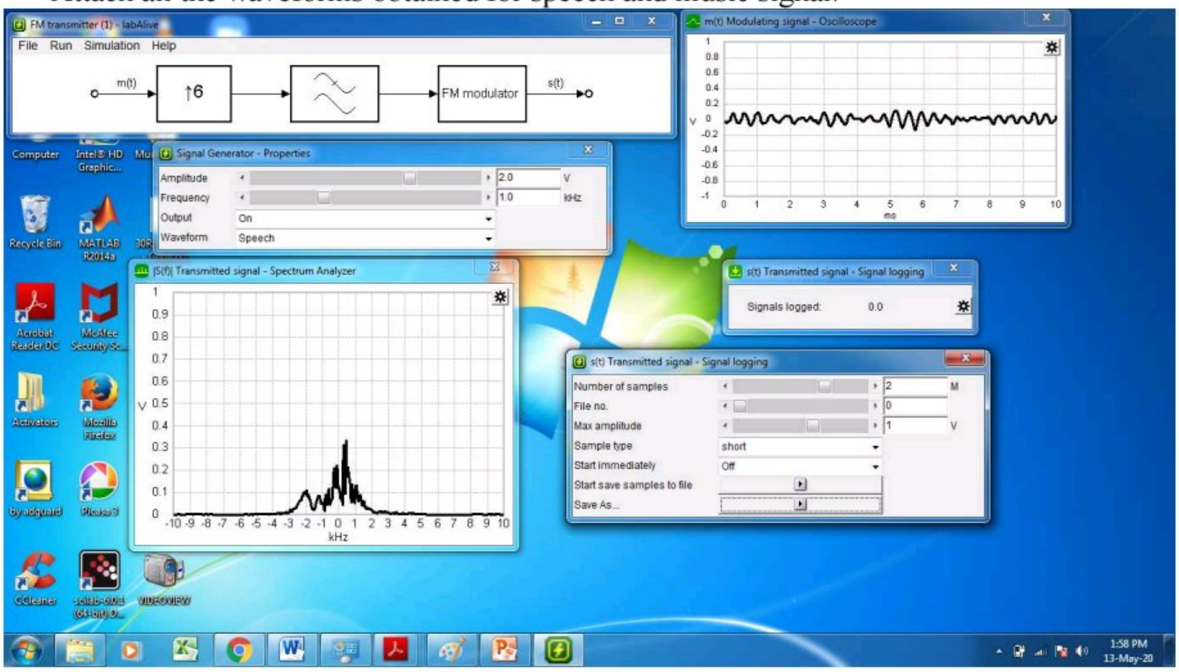
Go the the FM receiver and demodulate the FM transmitted signal.



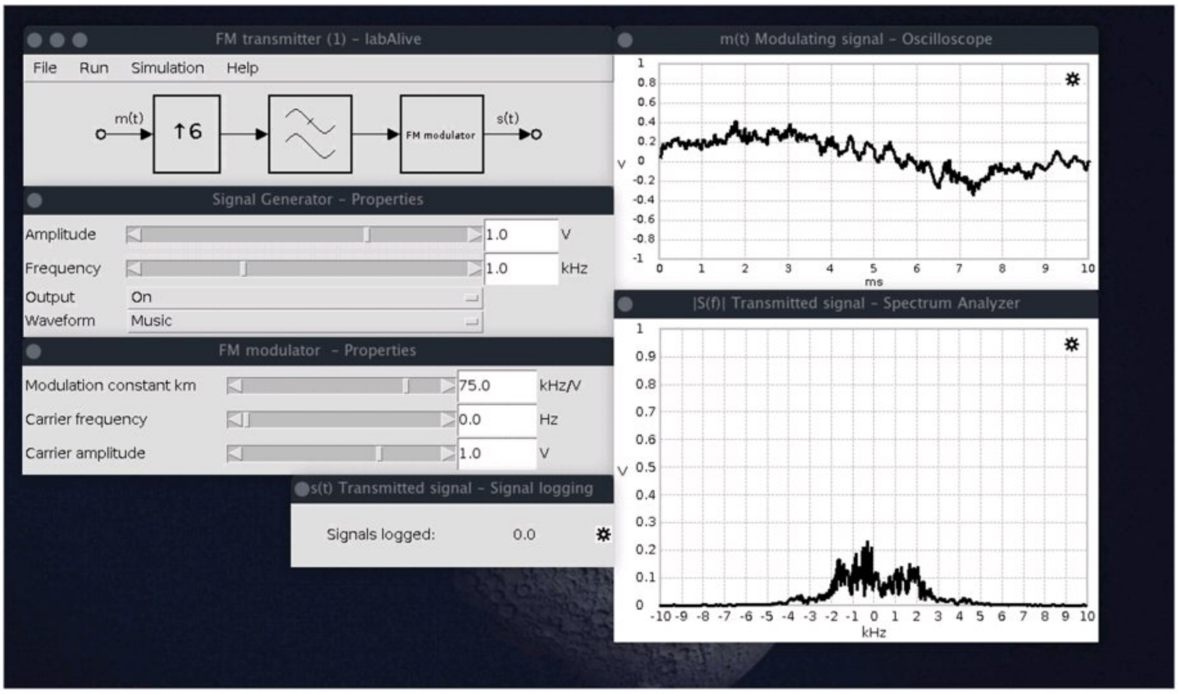
FM transmitter - modulate an audio signal and record the transmit signal to file.

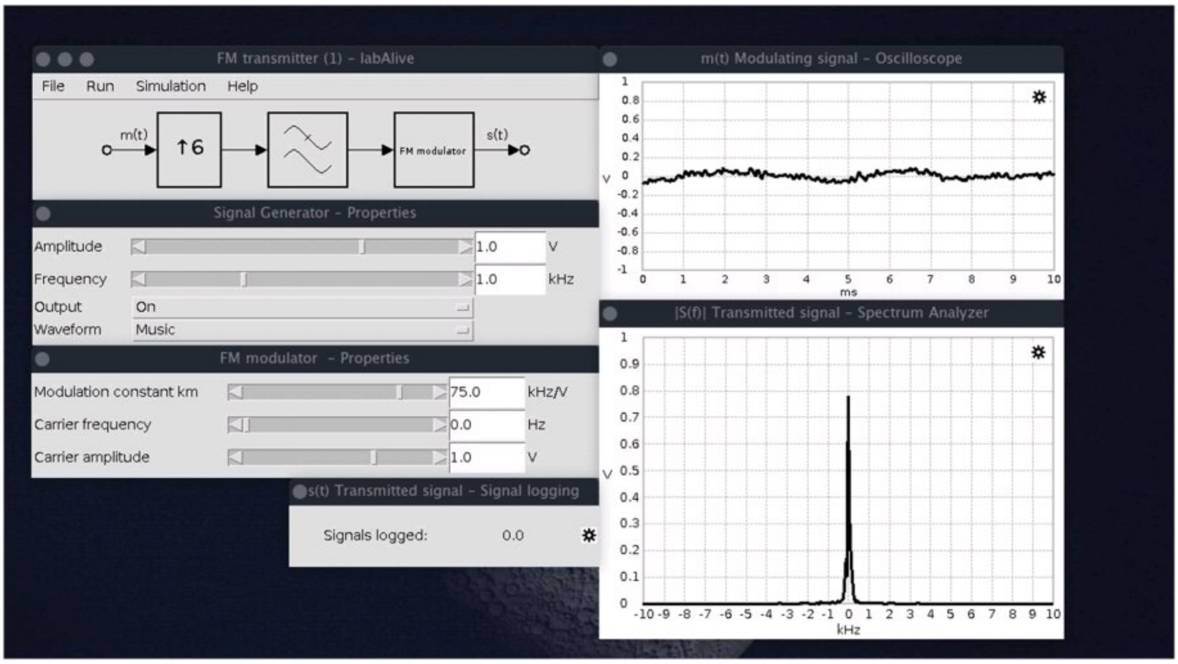
Notes:

- You may modify the input signal from the signal generator properties
- Varying the frequency and amplitude of input signal will change frequency deviation.
- To save the samples, click on the settings icon in signal logging window and save it. You will use this same file for reception.
- You can scale the graph by clicking on setting icon in modulating signal window and frequency spectrum analyser window.
- Attach all the waveforms obtained for speech and music signal.



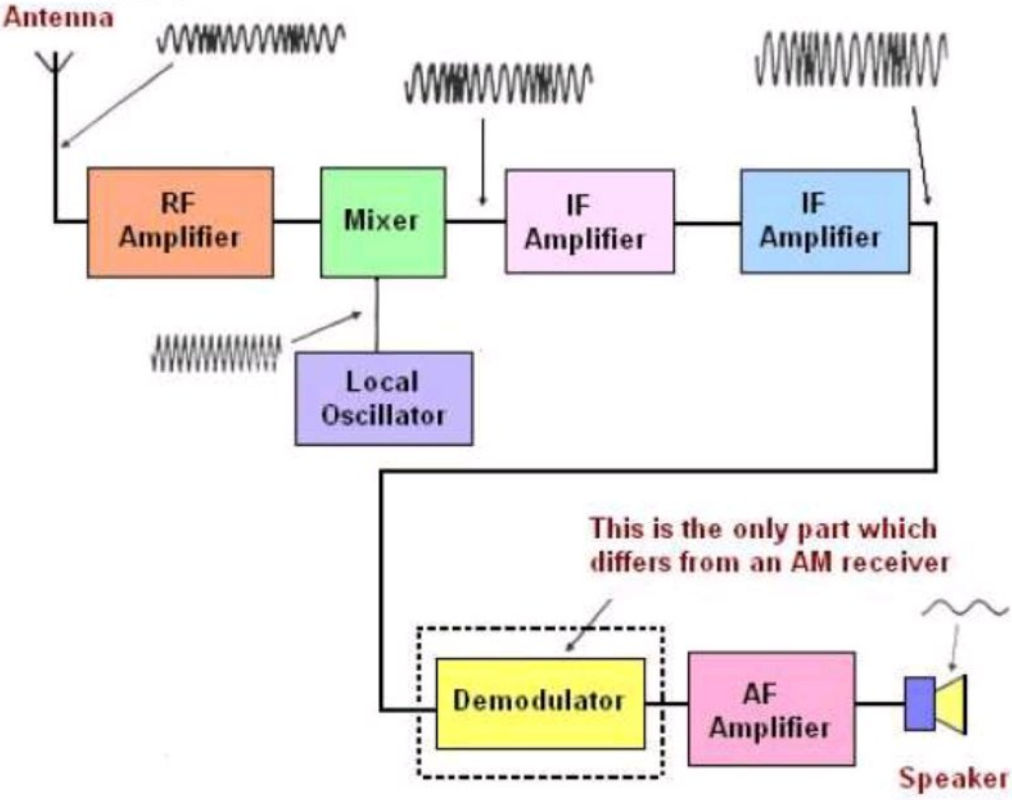
Output Waveforms:





Part C): To receive an audio file via FM receiver.

FM receiver:



FM Receiver

Procedure:

START

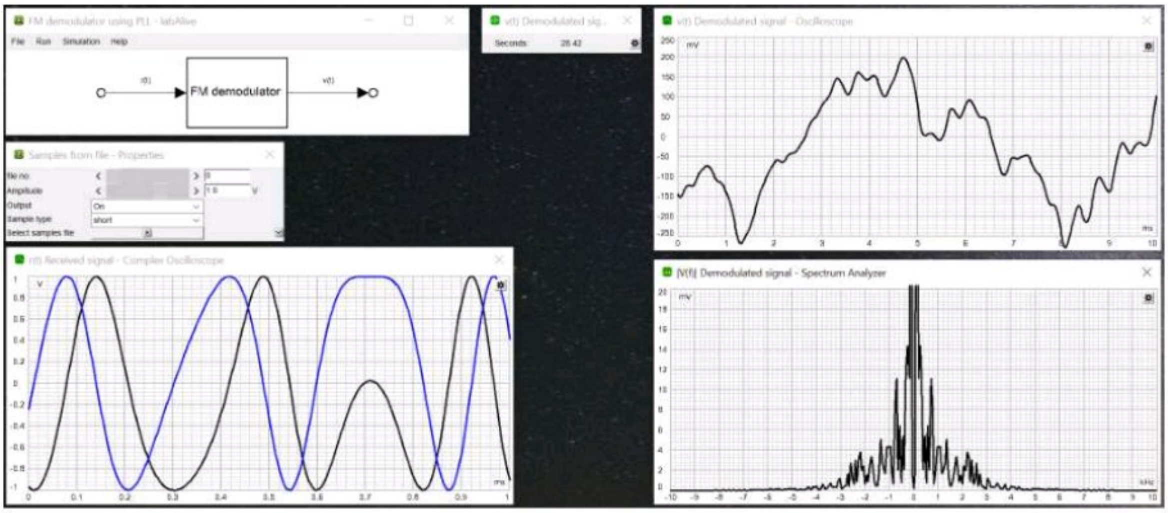
Select the file of the FM modulated signal you created with the FM transmitter.

LISTEN TO THE DEMODULATED SIGNAL

Enjoy the demodulated audio signal. It should be fine if you modulated the audio signal properly. If it's too quiet or distorted analyze if the frequency deviation is too large or too small.

VARY FREQUENCY DEVIATION AND CREATE DIFFERENT FM MODULATED SIGNALS

Vary the modulating signal's amplitude and thus also the frequency deviation using the FM transmitter.

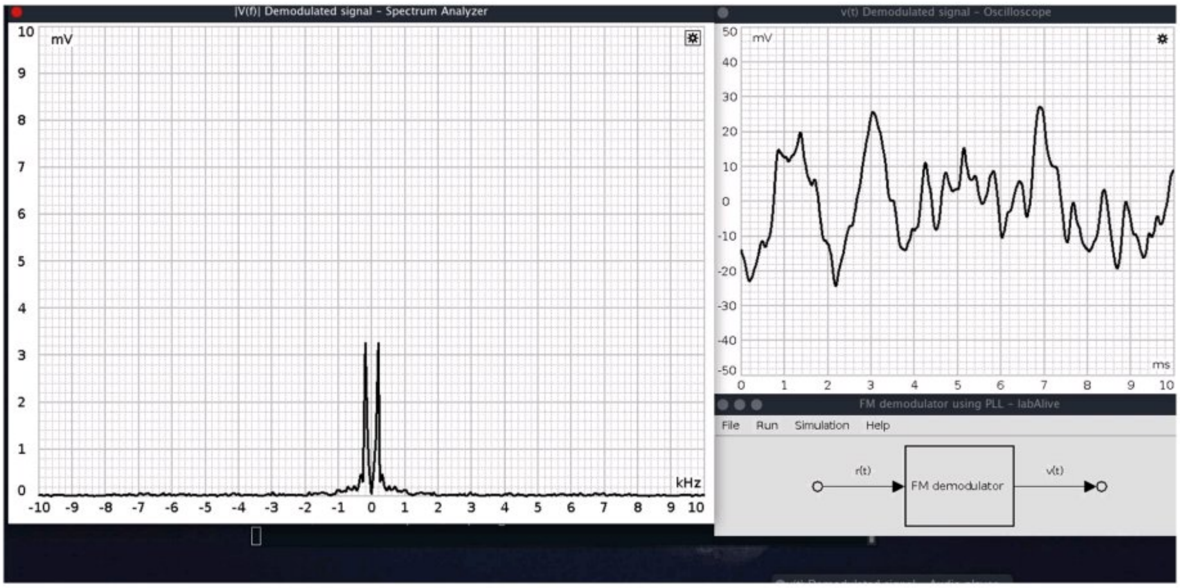


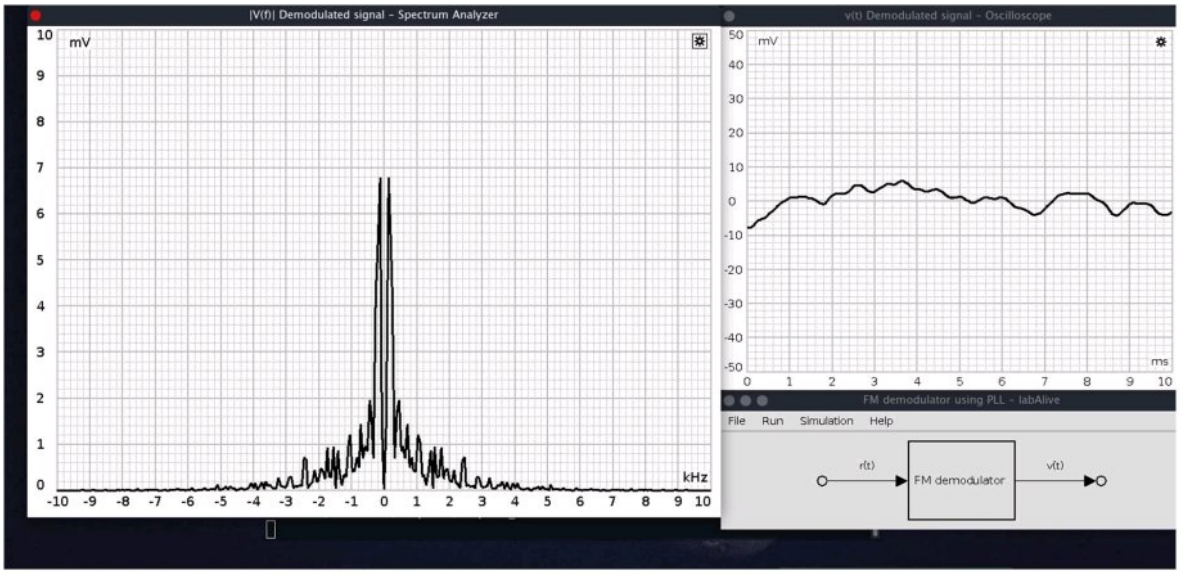
FM receiver using a PLL demodulator.

Notes:

- In the signal logging window, change the sample type to double and number of samples to 1G if you're not able to save the file.
- Listen to received audio signal and change the parameter values in modulator and demodulator block to listen the beat if hissing sound is coming.
- Take the screenshots of the graphs observed and paste it in the output waveforms

Output Waveforms:





Conclusion:

In this experiment, we have observed and studied the Spectrum of an FM signal using labAlive. We also performed FM Modulation and Demodulation of audio signals for various transmission parameters like Fm, Modulation Index, Message Signal Amplitude and observed their generated spectra.

Remark **Signature**

ASK signal generation and demodulation

Experiment No.: 7 **Date:**

Aim:

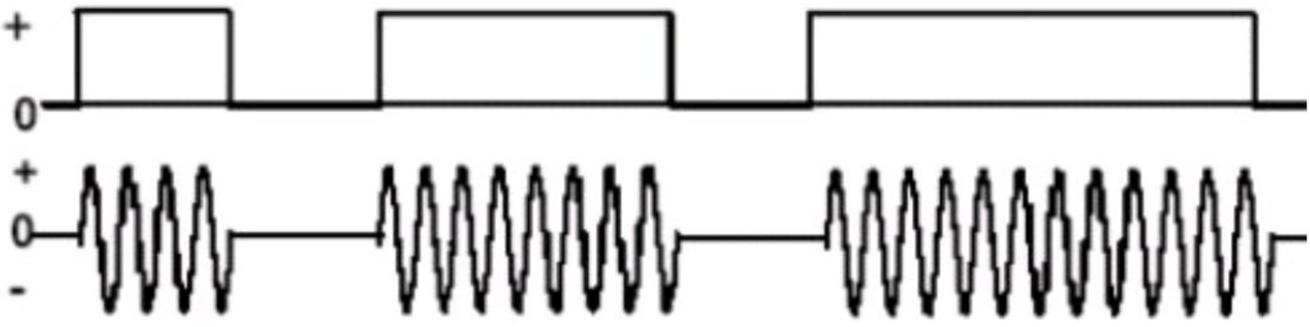
To Generate and demodulate an amplitude shift keying (ASK) signal.

Theory:

Amplitude Shift Keying (ASK) is the digital modulation technique. In amplitude shift keying, the amplitude of the carrier signal is varied to create signal elements. Both frequency and phase remain constant while the amplitude changes. In ASK, the amplitude of the carrier assumes one of the two amplitudes dependent on the logic states of the input bit stream. This modulated signal can be expressed as:

$$x_c(t) = \begin{cases} 0 & \text{symb} \\ A\cos\omega_c t & \text{symb} \end{cases}$$

Amplitude shift keying (ASK) in the context of digital signal communications is a modulation process, which imparts to a sinusoid two or more discrete amplitude levels. These are related to the number of levels adopted by the digital message. For a binary message sequence there are two levels, one of which is typically zero. Thus the modulated waveform consists of bursts of a sinusoid. Figure 1 illustrates a binary ASK signal (lower), together with the binary sequence which initiated it (upper). Neither signal has been band limited.

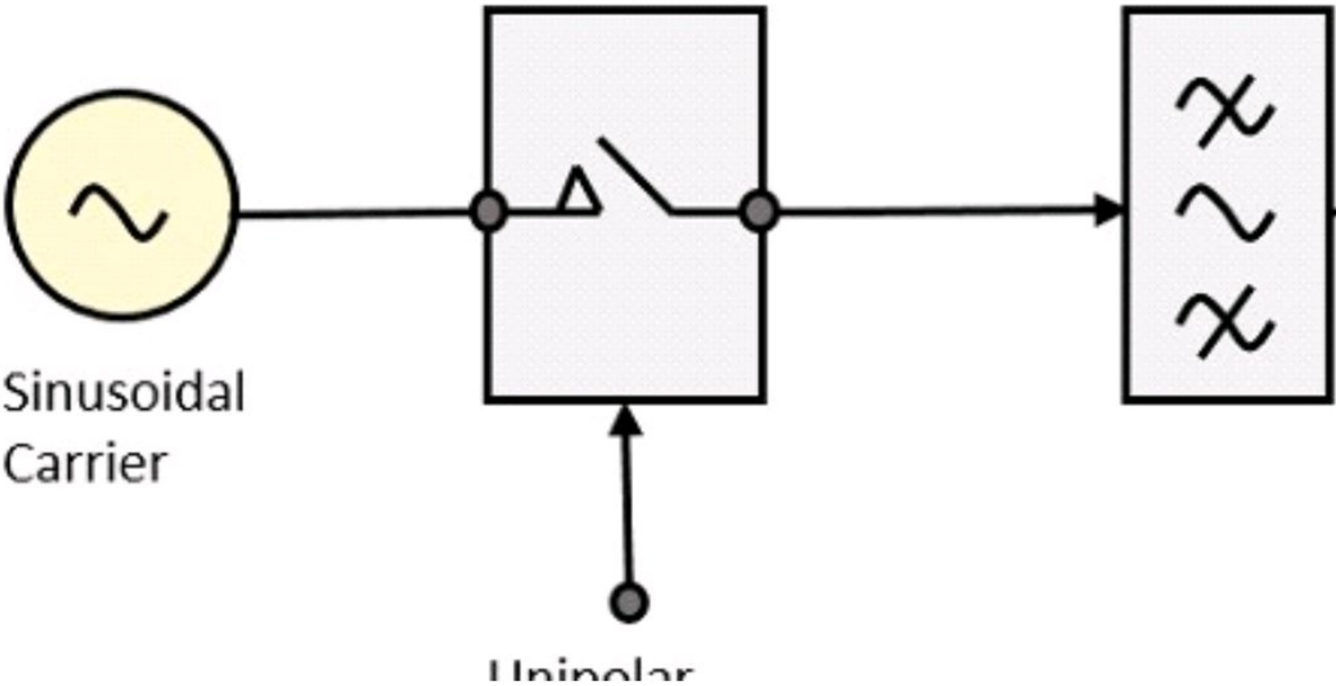


Message Signal & ASK Signal

There are sharp discontinuities shown at the transition points. These result in the signal having an unnecessarily wide bandwidth. Band limiting is generally introduced before transmission, in which case these discontinuities would be 'rounded off'. The band limiting may be applied to the digital message, or the modulated signal itself. The data rate is often made a sub-multiple of the carrier frequency.

ASK Modulator:

The ASK modulator block diagram comprises of the carrier signal generator, the binary sequence from the message signal and the band-limited filter. Following is the block diagram of the ASK Modulator.



The carrier generator, sends a continuous high-frequency carrier. The binary sequence from the message signal makes the unipolar input to be either High or Low. The high signal closes the switch, allowing a carrier wave. Hence, the output will be the carrier signal at high input. When there is low input, the switch opens, allowing no voltage to appear. Hence, the output will be low.

The band-limiting filter shapes the pulse depending upon the amplitude and phase characteristics of the band-limiting filter or the pulse-shaping filter.

ASK Demodulator:

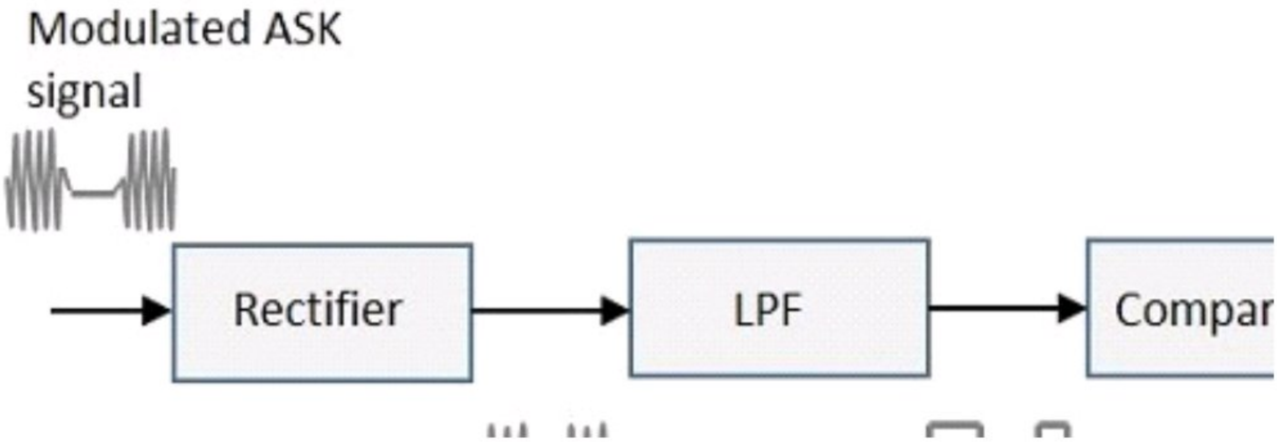
There are two types of ASK Demodulation techniques.

- Asynchronous ASK Demodulation/detection
- Synchronous ASK Demodulation/detection

The clock frequency at the transmitter, when matches with the clock frequency at the receiver, it is known as a Synchronous method, as the frequency gets synchronized. Otherwise, it is known as Asynchronous.

Asynchronous ASK Demodulator:

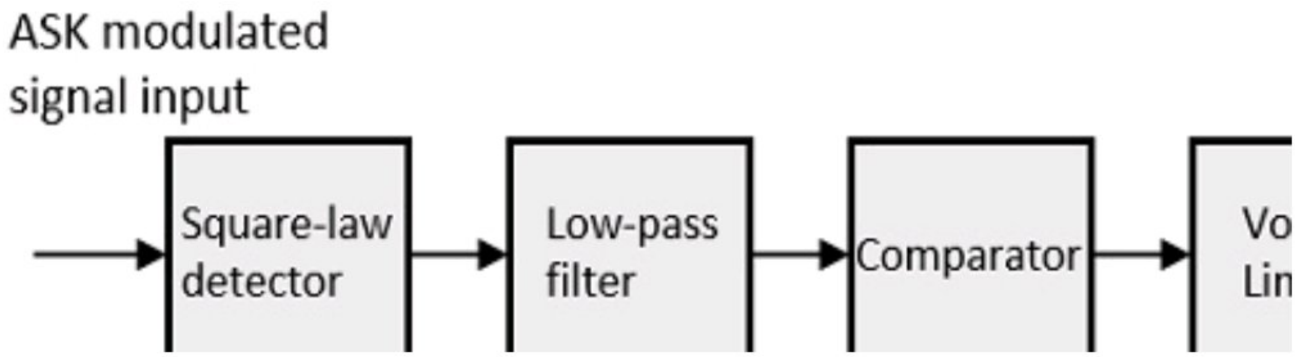
The Asynchronous ASK detector consists of a half-wave rectifier, a low pass filter, and a comparator. Following is the block diagram for the same.



The modulated ASK signal is given to the half-wave rectifier, which delivers a positive half output. The low pass filter suppresses the higher frequencies and gives an envelope detected output from which the comparator delivers a digital output.

Synchronous ASK Demodulator:

Synchronous ASK detector consists of a Square law detector, low pass filter, a comparator, and a voltage limiter. Following is the block diagram for the same.



The ASK modulated input signal is given to the Square law detector. A square law detector is one whose output voltage is proportional to the square of the amplitude modulated input voltage. The low pass filter minimizes the higher frequencies. The comparator and the voltage limiter help to get a clean digital output.

Algorithm to implement ASK modulation & demodulation on MATLAB:

ASK Modulation:

- Generate a carrier signal of frequency fc.
- Start a FOR Loop.
- Generate a binary sequence, a message signal.
- Generate ASK modulated signal, which will transmit carrier signal for logic 1 and zero signal for logic 0.
- Plot message signal and ASK modulated signal.
- End FOR Loop.
- Plot binary data and carrier.

ASK Demodulation:

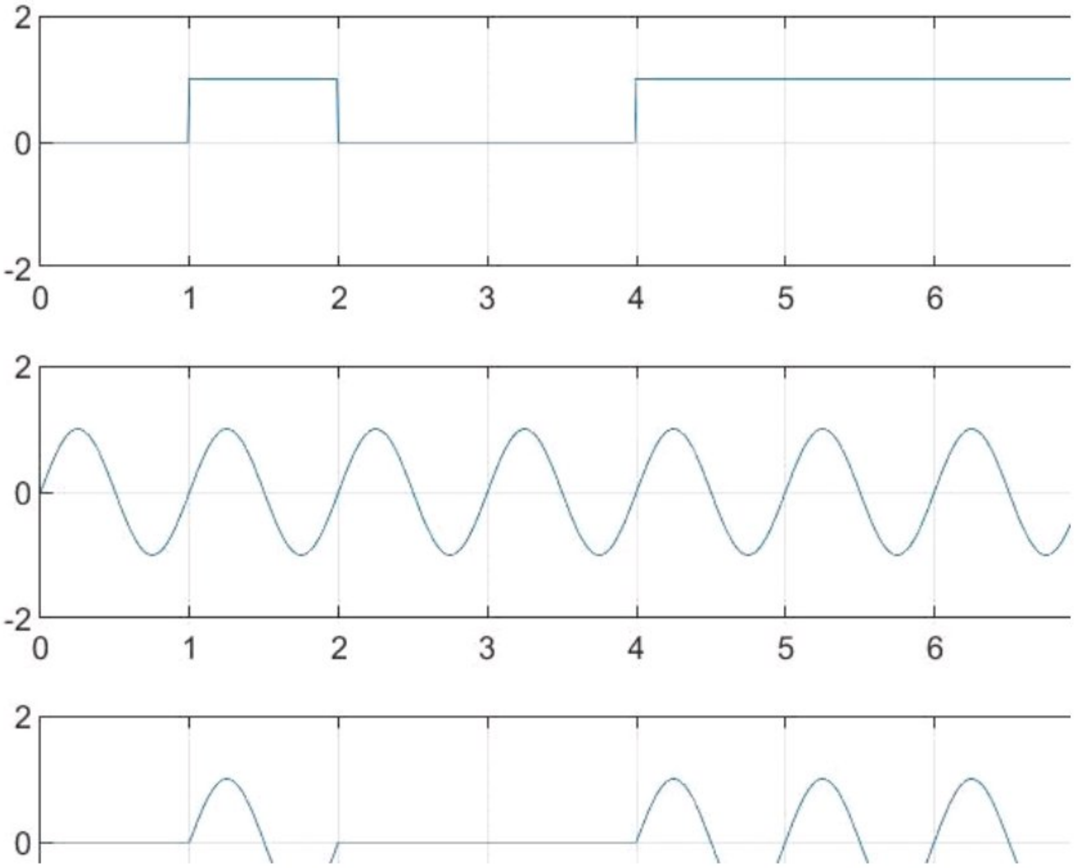
- Start FOR Loop.
- Perform correlation of ASK signal with carrier to get decision variable.

- Make decision to get demodulated binary data. If x > 0, choose ‘1’ else choose ‘0’.
- Plot the demodulated binary data.

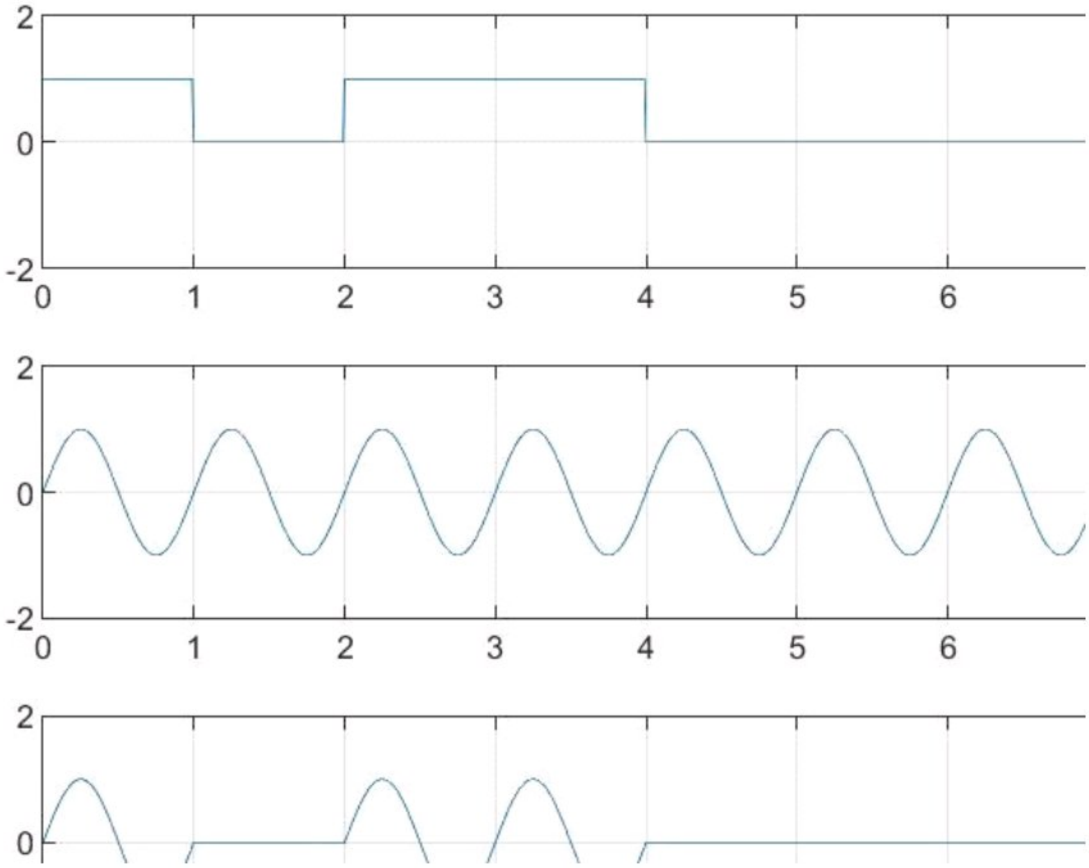
Result:

[1. Message signal 2. Carrier signal 3. ASK signal]

- Observation waveform for the bit stream [0 1 0 0 1 1 1 0]



- Observation waveform for the bit stream [1 0 1 1 0 0 0 1]



MATLAB Code:

```
clc
clear all
b = input('Enter the Bit stream \n '); %b = [0 1 0
1 1 1 0];
n = length(b);%defining n
t = 0:.01:n;%defining t
x = 1:1:(n+1)*100;%defining x
%using for loop
for i = 1:n
 for j = i:.1:i+1
 bw(x(i*100:(i+1)*100)) = b(i);
```

```
 end
end
bw = bw(100:end);
sint = sin(2*pi*t);
st = bw.*sint;
%Plotting all using subplot
subplot(3,1,1)
plot(t,bw)
grid on ; axis([0 n -2 +2])
subplot(3,1,2)
plot(t,sint)
grid on ; axis([0 n -2 +2])
subplot(3,1,3)
plot(t,st)
grid on ; axis([0 n -2 +2])
```

Application of ASK:

- Low-frequency RF applications
- Home automation devices
- Industrial networks devices
- Wireless base stations
- Tire pressuring monitoring systems

Conclusion:

In this experiment we learnt how to generate and demodulate amplitude shift keying signal using ***MATLAB.***

Signature **Remarks**

AMPLITUDE MODULATION IN MATLAB

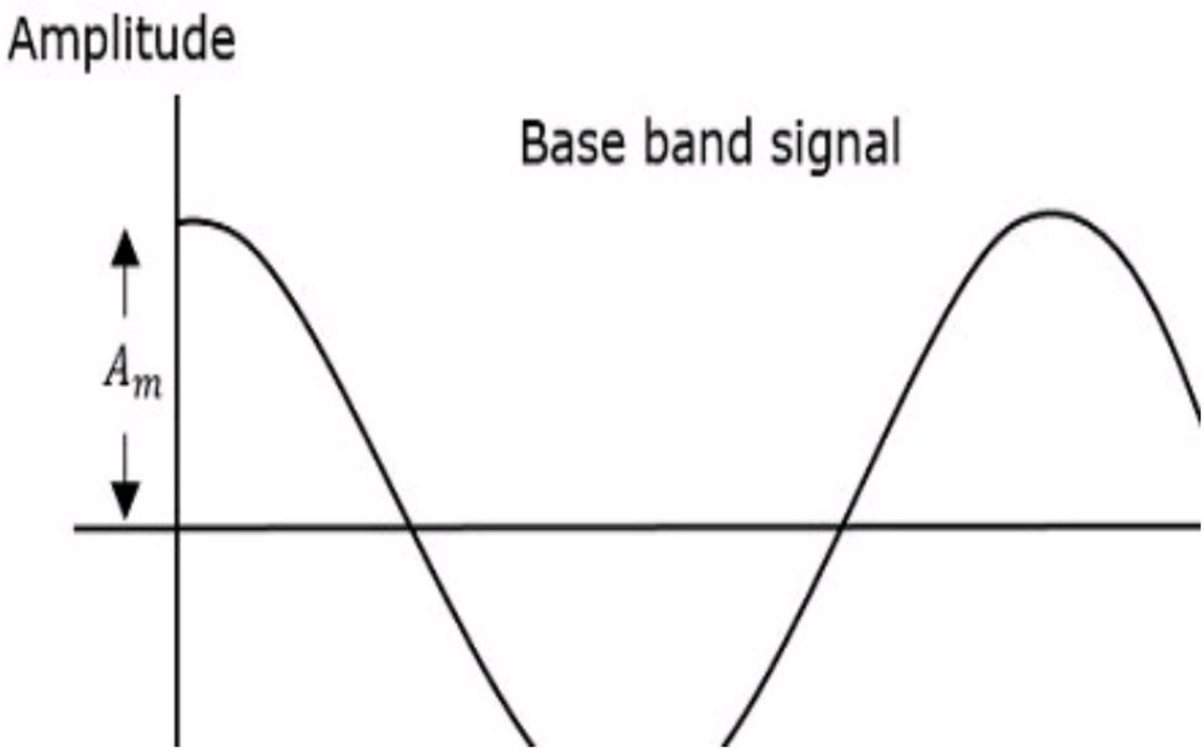
Experiment No: ***08*** **Date:**

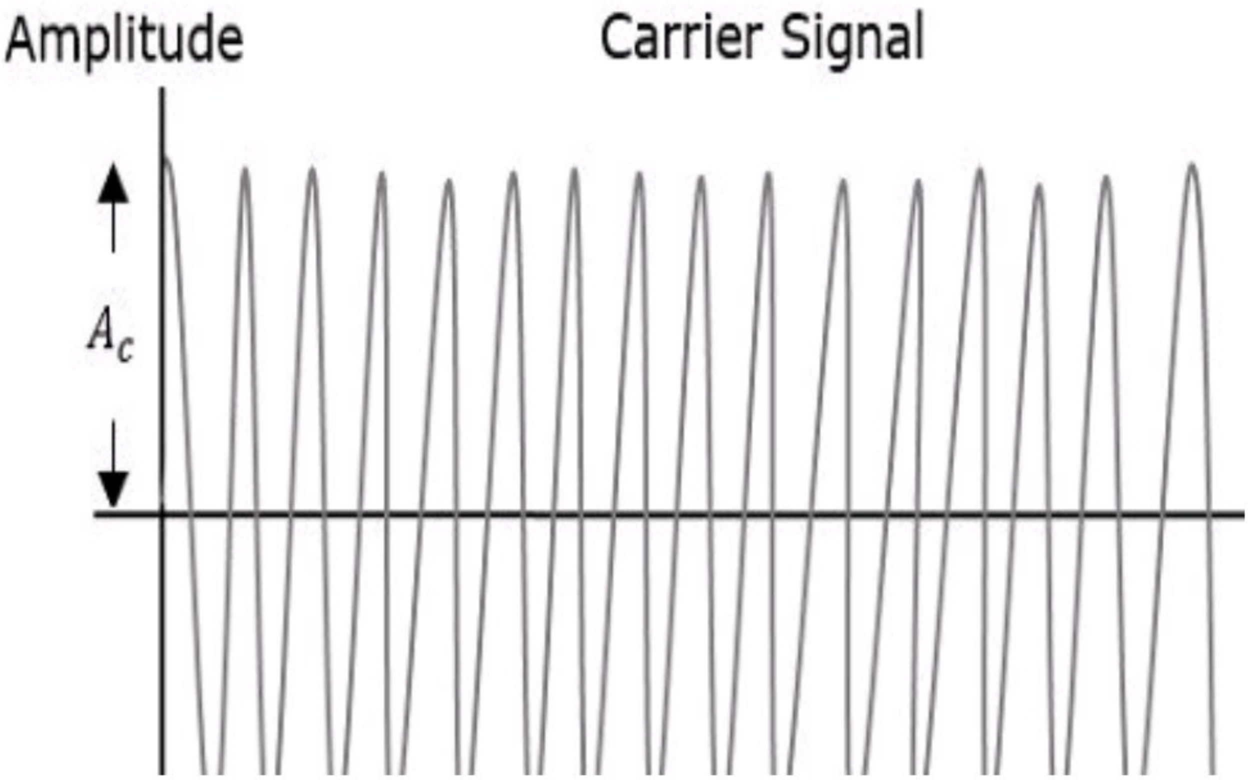
Aim: To implement amplitude modulation and demodulation using MATLAB.

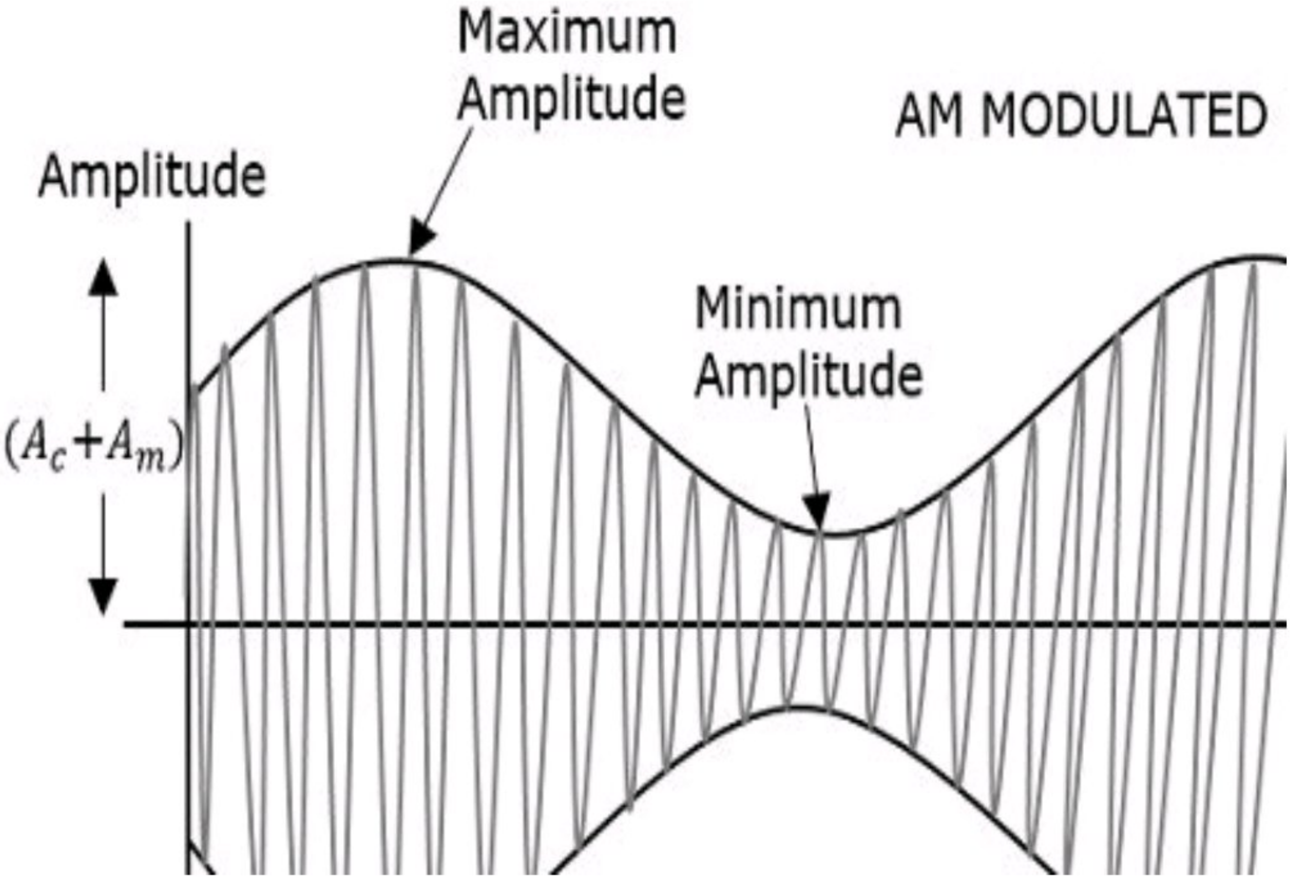
Brief Theory/Equations:

A continuous-wave goes on continuously without any intervals and it is the baseband message signal, which contains the information. This wave has to be modulated.

According to the standard definition, "The amplitude of the carrier signal varies in accordance with the instantaneous amplitude of the modulating signal." Which means, the amplitude of the carrier signal containing no information varies as per the amplitude of the signal containing information, at each instant. This can be well explained by the following figures.




Amplitude
Carrier Signal
A_c



The first figure shows the modulating wave, which is the message signal. The next one is the carrier wave, which is a high frequency signal and contains no information. While, the last one is the resultant modulated wave.

It can be observed that the positive and negative peaks of the carrier wave, are interconnected with an imaginary line. This line helps recreating the exact shape of the modulating signal. This imaginary line on the carrier wave is called as **Envelope**. It is the same as that of the message signal.

Mathematical Expressions:

Following are the mathematical expressions for these waves.

Time-domain Representation of the Waves

Let the modulating signal be,

$$m(t)=A_m\cos(2\pi f_m t)$$

and the carrier signal be,

$$c(t)=Ac\cos(2\pi fct)$$

Where,

Am and Ac are the amplitude of the modulating signal and the carrier signal respectively.
fm and fc are the frequency of the modulating signal and the carrier signal respectively.

Then, the equation of Amplitude Modulated wave will be

$s(t)=[Ac+Am\cos(2\pi fmt)]\cos(2\pi fct)$ (Equation 1)

Modulation Index:

A carrier wave, after being modulated, if the modulated level is calculated, then such an attempt is called as **Modulation Index** or **Modulation Depth**. It states the level of modulation that a carrier wave undergoes.

Rearrange the Equation 1 as below.

$s(t)=Ac[1+(AmAc)\cos(2\pi fmt)]\cos(2\pi fct)$
$\Rightarrow s(t)=Ac[1+\mu\cos(2\pi fmt)]\cos(2\pi fct)$ (Equation 2)
Where, μ is Modulation index and it is equal to the ratio of Am and Ac Mathematically, we can write it as
$\mu=Am/Ac$ (Equation 3)

Hence, we can calculate the value of modulation index by using the above formula, when the amplitudes of the message and carrier signals are known.

Now, let us derive one more formula for Modulation index by considering Equation 1. We can use this formula for calculating modulation index value, when the maximum and minimum amplitudes of the modulated wave are known.

Let Amax and Amin be the maximum and minimum amplitudes of the modulated wave.
We will get the maximum amplitude of the modulated wave, when $\cos(2\pi fmt)$ is 1.
$\Rightarrow Amax=Ac+Am$ (Equation 4)
We will get the minimum amplitude of the modulated wave, when $\cos(2\pi fmt)$ is -1.
$\Rightarrow Amin=Ac-Am$ (Equation 5)

Add Equation 4 and Equation 5.

$$Amax+Amin=Ac+Am+Ac-Am=2Ac$$

$\Rightarrow Ac=Amax+Amin/2$ (Equation 6)

Subtract Equation 5 from Equation 4.

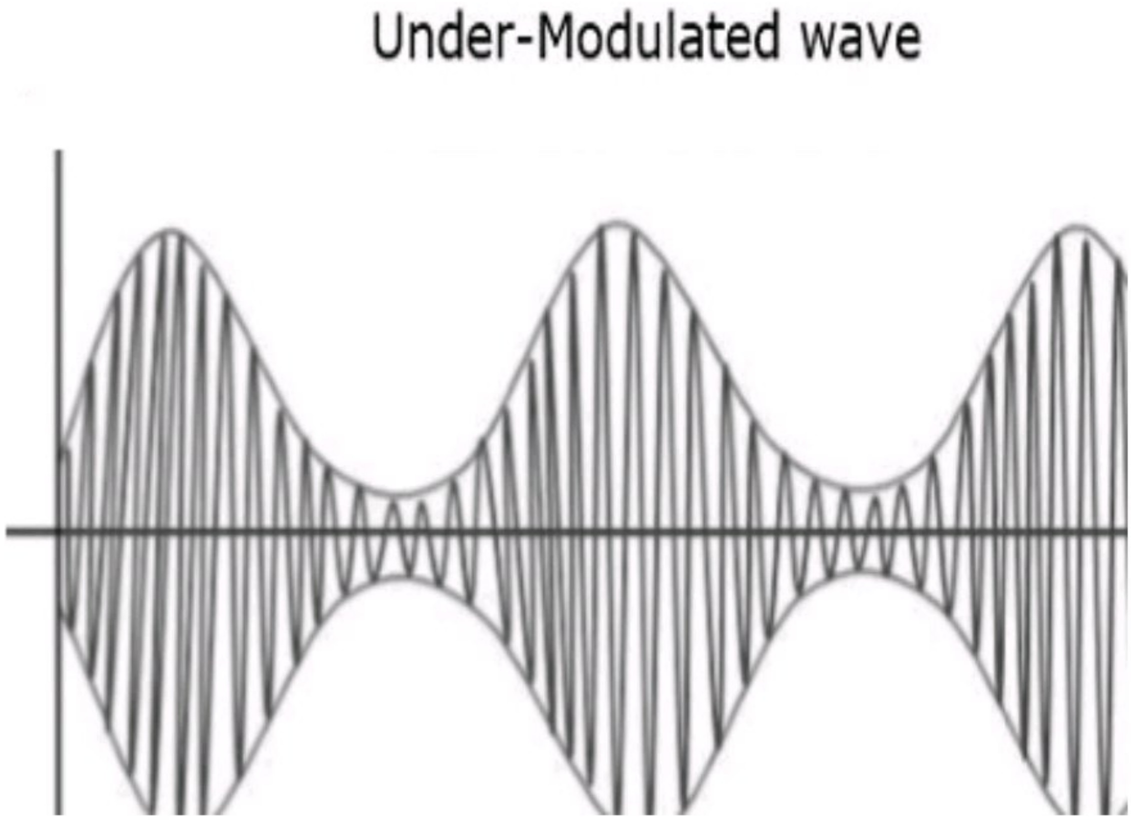
$$Amax-Amin=Ac+Am-(Ac-Am)=2Am$$

$\Rightarrow Am=Amax-Amin/2$ (Equation 7)
$\mu=Amax-Amin/Amax+Amin$ (Equation 8)

Therefore, Equation 3 and Equation 8 are the two formulas for Modulation index. The modulation index or modulation depth is often denoted in percentage called as Percentage of Modulation. We will get the **percentage of modulation**, just by multiplying the modulation index value with 100.

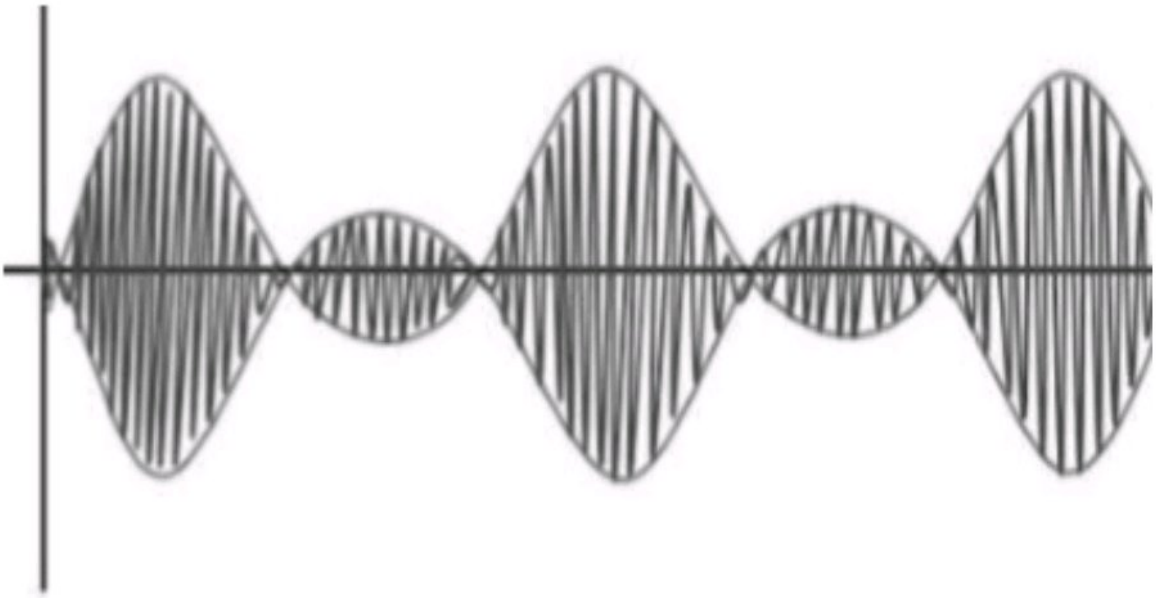
For a perfect modulation, the value of modulation index should be 1, which implies the percentage of modulation should be 100%.

For instance, if this value is less than 1, i.e., the modulation index is 0.5, then the modulated output would look like the following figure. It is called as **Under-modulation**. Such a wave is called as an **under-modulated wave**.

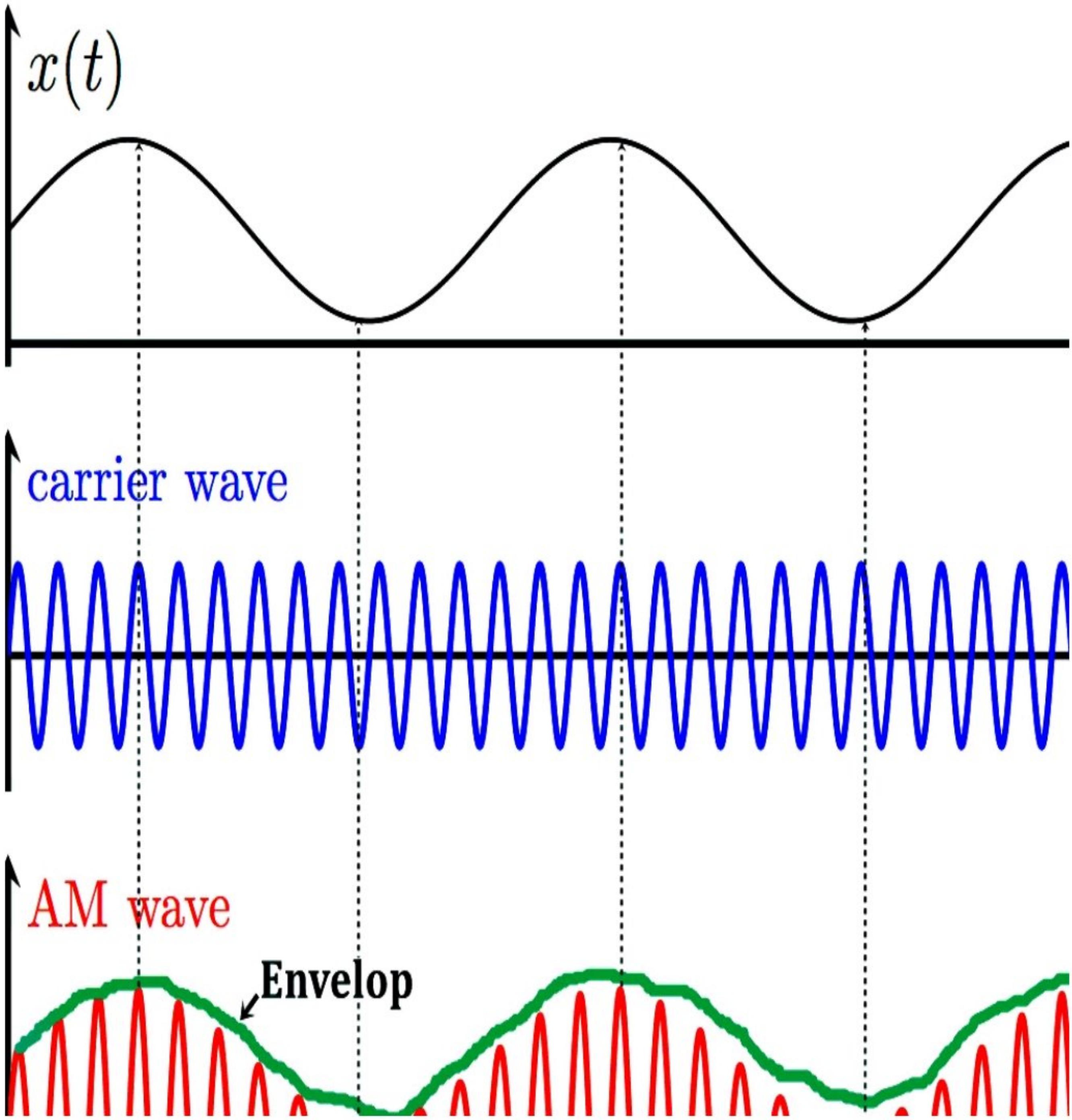


If the value of the modulation index is greater than 1, i.e., 1.5 or so, then the wave will be an **over-modulated wave**. It would look like the following figure.

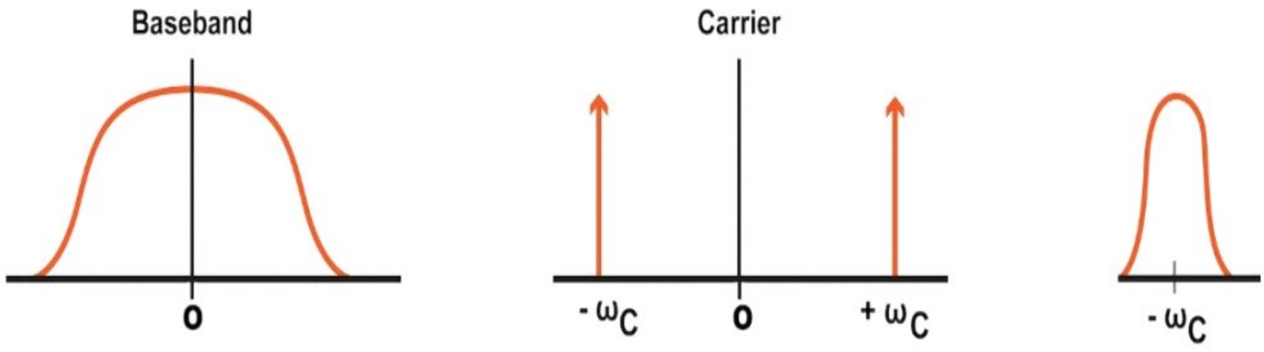
Over-Modulated wave



As the value of the modulation index increases, the carrier experiences a 180° phase reversal, which causes additional sidebands and hence, the wave gets distorted. Such an over-modulated wave causes interference, which cannot be eliminated.

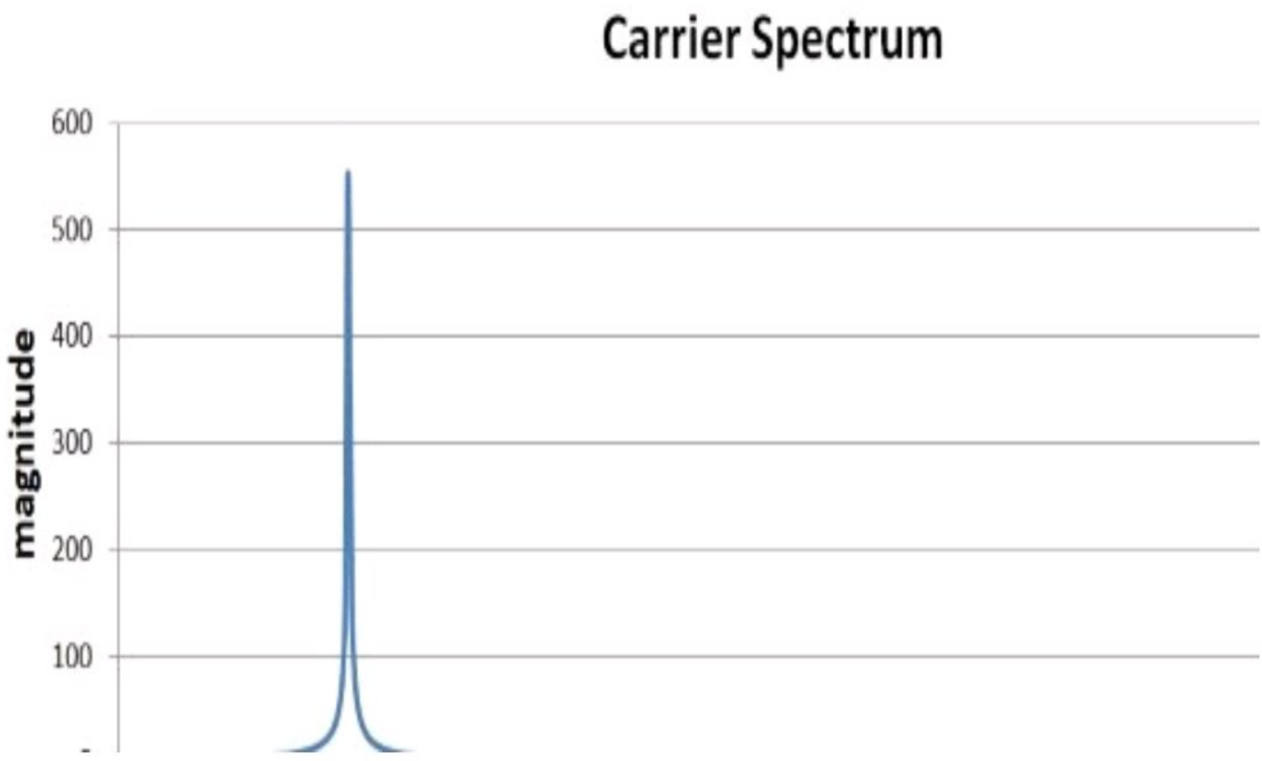


$x(t)$
carrier wave
AM wave
Envelop

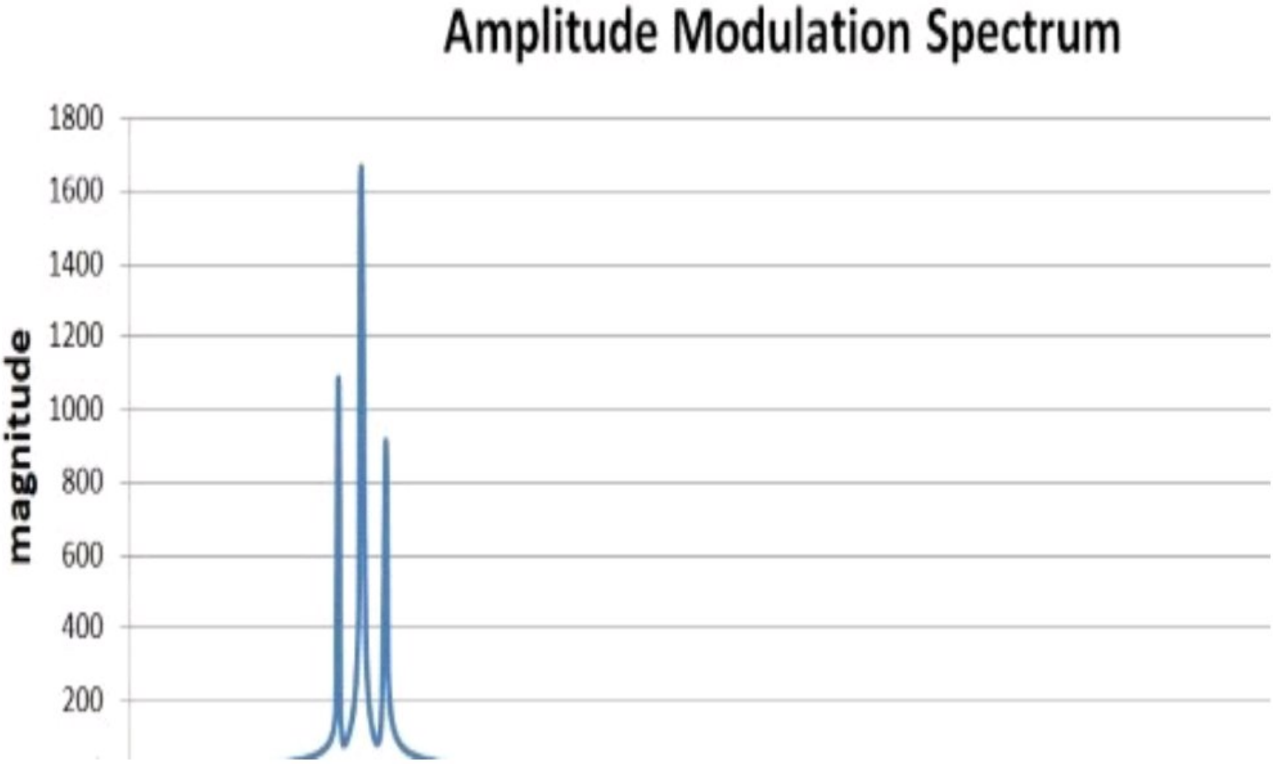


In Frequency Domain

Let's start with the frequency-domain representation of a carrier signal:

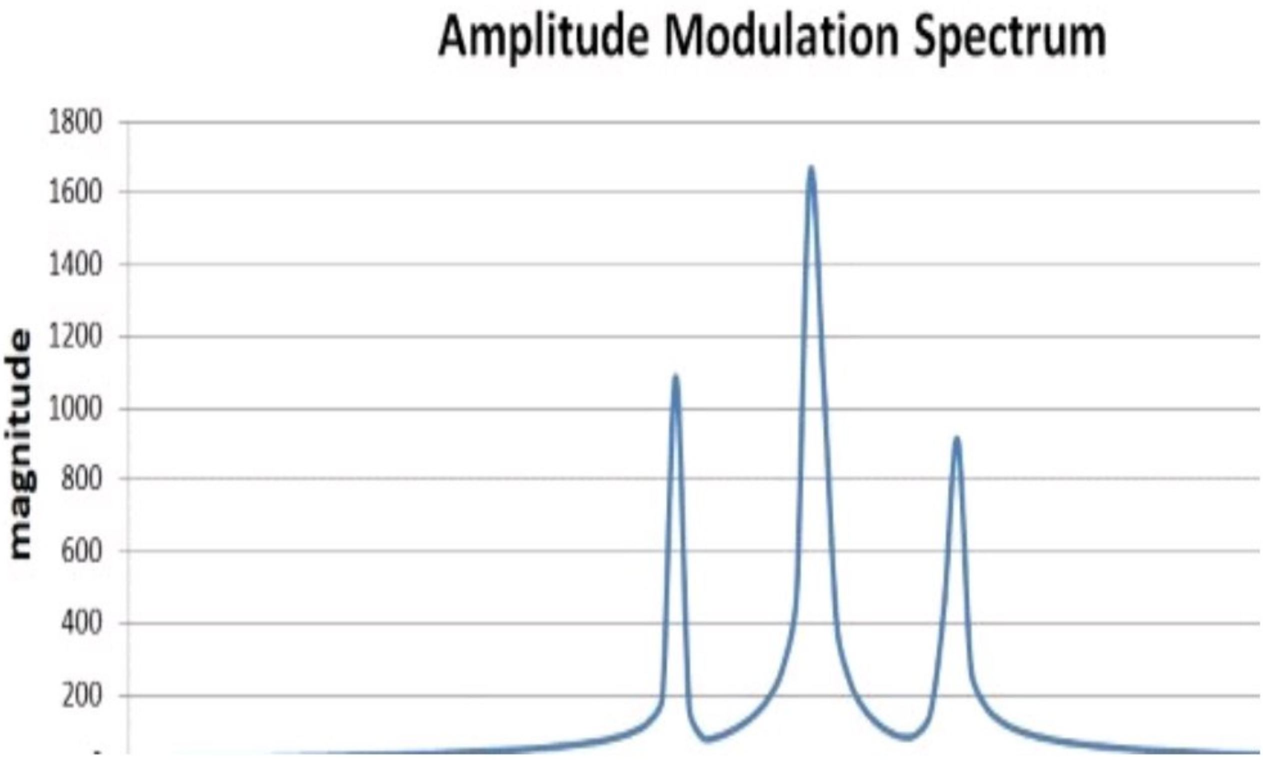


This is exactly what we expect for the unmodulated carrier: a single spike at 10 MHz. Now let's look at the spectrum of a signal created by amplitude modulating the carrier with a constant-frequency 1 MHz sinusoid.



Here you see the standard characteristics of an amplitude-modulated waveform: the baseband signal has been shifted according to the frequency of the carrier. You could also think of this as "adding" the baseband frequencies onto the carrier signal, which is indeed what we're doing when we use amplitude modulation—the carrier frequency remains, as you can see in the time-domain waveforms, but the amplitude variations constitute new frequency content that corresponds to the spectral characteristics of the baseband signal.

If we look more closely at the modulated spectrum, we can see that the two new peaks are 1 MHz (i.e., the baseband frequency) above and 1 MHz below the carrier frequency:



AM Demodulation: Amplitude Modulation Detection:

Demodulation is a key process in the reception of any amplitude modulated signals whether used for broadcast or two way radio communication systems.

Demodulation is the process by which the original information bearing signal, i.e. the modulation is extracted from the incoming overall received signal.

The process of demodulation for signals using amplitude modulation can be achieved in a number of different techniques, each of which has its own advantage.

The demodulator is the circuit, or for a software defined radio, the software that is used to recover the information content from the overall incoming modulated signal.

AM demodulators are found in many items of radio equipment: broadcast receivers, professional radio communication equipment, walkie talkies - AM is still used for air-band radio communications.

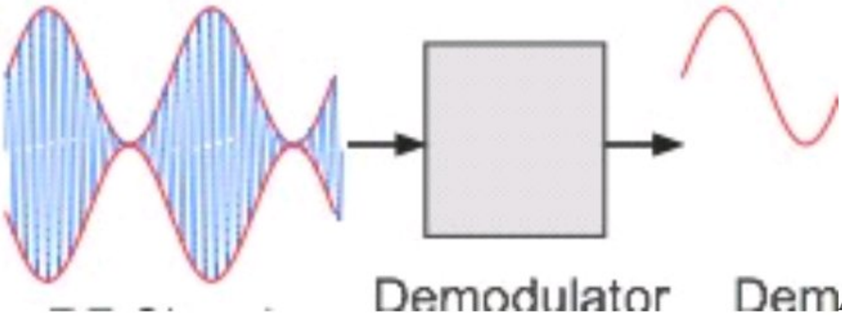
Detection or demodulation:

The terms detection and demodulation are often used when referring to the overall demodulation process. Essentially the terms describe the same process, and the same circuits.

As the name indicates the demodulation process is the opposite of modulation, where a signal such as an audio signal is applied to a carrier.

In the demodulation process the audio or other signal carried by amplitude variations on the carrier is extracted from the overall signal to appear at the output.

As the most common use for amplitude modulation is for audio applications, the most common output is the audio. This may be broadcast entertainment for broadcast reception, and for two way radio communications, it is often used for land communications for aeronautical associated applications - often within walkie talkies.



AM amplitude modulation demodulation principle

Terms like diode detector, synchronous detector and product detector are widely used. But the term demodulation tends to be used more widely when referring to the process of extracting the modulation from the signal.

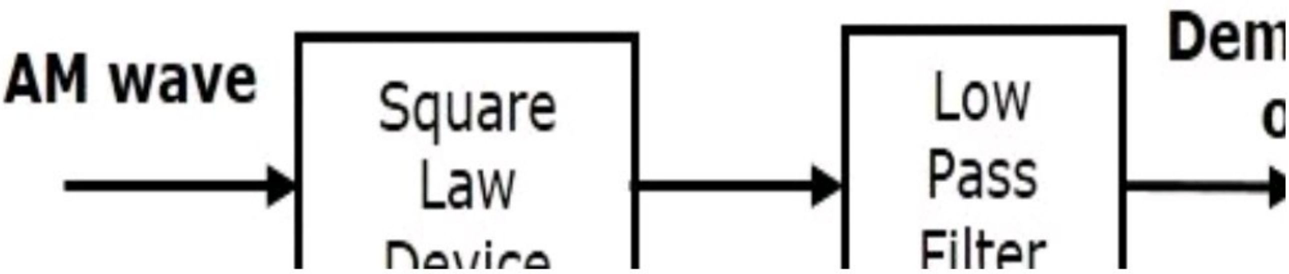
The term detection is the older term dating back to the early days of radio. The term demodulation is probably more accurate in that it refers to the process of demodulation, i.e. extracting the modulation from the signal.

The following demodulators (detectors) are used for demodulating AM wave.

- Square Law Demodulator
- Envelope Detector

Square Law Demodulator:

Square law demodulator is used to demodulate low level AM wave. Following is the block diagram of the **square law demodulator**.



This demodulator contains a square law device and low pass filter. The AM wave $V_1(t)$V1(t) is applied as an input to this demodulator.

The standard form of AM wave is

$$V_1(t)=A_c[1+k_am(t)]\cos(2\pi f_ct)$$

We know that the mathematical relationship between the input and the output of square law device is

$V_2(t)=k_1V_1(t)+k_2V_1^{2}(t)$ (Equation 1)

Where,

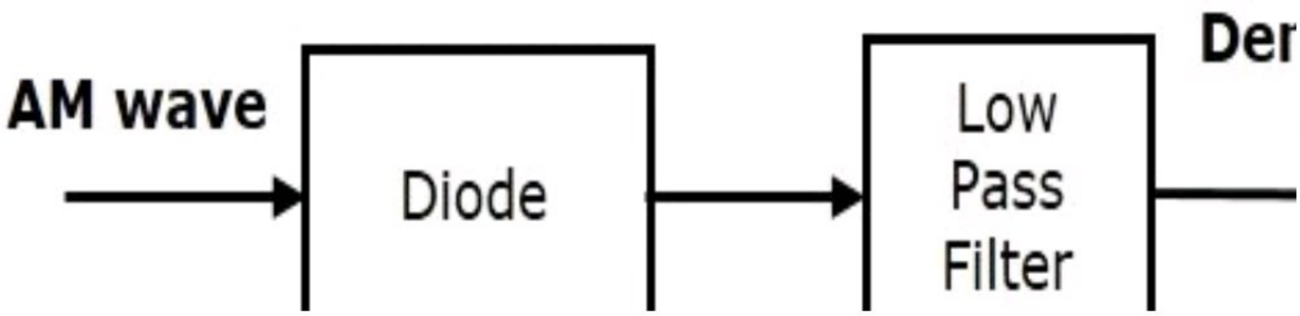
$V_1(t)$ is the input of the square law device, which is nothing but the AM wave
$V_2(t)$ is the output of the square law device
k_1 and k_2 are constants
Substitute $V_1(t)$ in Equation 1

$$V_2(t)=k_1(A_c[1+k_am(t)]\cos(2\pi f_ct))+k_2(A_c[1+k_am(t)]\cos(2\pi f_ct))^2$$

On solving,the term $k_2A_c^2k_am(t)$ is the scaled version of the message signal. It can be extracted by passing the above signal through a low pass filter and the DC component $k_2A_c^2/2$ can be eliminated with the help of a coupling capacitor.

Envelope Detector:

Envelope detector is used to detect (demodulate) high level AM wave. Following is the block diagram of the envelope detector.



This envelope detector consists of a diode and low pass filter. Here, the diode is the main detecting element. Hence, the envelope detector is also called as the **diode detector**. The low pass filter contains a parallel combination of the resistor and the capacitor.

The AM wave s(t)s(t) is applied as an input to this detector.

We know the standard form of AM wave is

$$s(t)=Ac[1+kam(t)]\cos(2\pi fct)$$

In the positive half cycle of AM wave, the diode conducts and the capacitor charges to the peak value of AM wave. When the value of AM wave is less than this value, the diode will be reverse biased. Thus, the capacitor will discharge through resistor **R** till the next positive half cycle of AM wave. When the value of AM wave is greater than the capacitor voltage, the diode conducts and the process will be repeated.

We should select the component values in such a way that the capacitor charges very quickly and discharges very slowly. As a result, we will get the capacitor voltage waveform same as that of the envelope of AM wave, which is almost similar to the modulating signal.

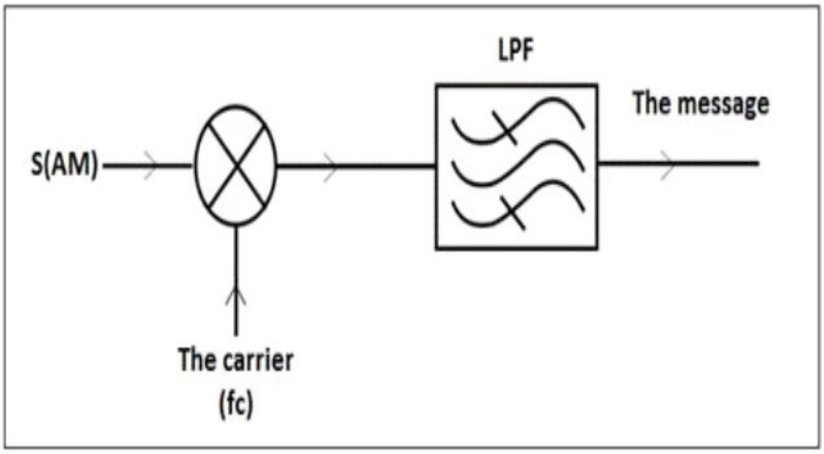
Product Detector:

A **product detector** is a type of demodulator used for AM and SSB signals. Rather than converting the envelope of the signal into the decoded waveform like an envelope detector, the product detector takes the product of the modulated signal and a local oscillator, hence the name. A product detector is a frequency mixer.

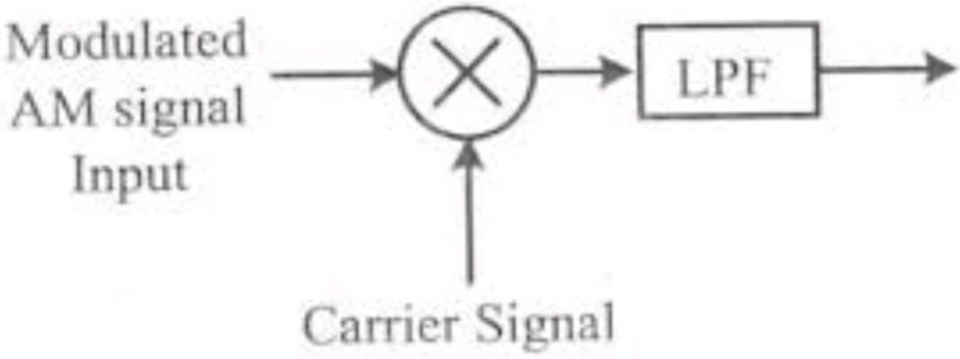
Product detectors can be designed to accept either IF or RF frequency inputs. A product detector which accepts an IF signal would be used as a demodulator block in a superheterodyne receiver, and a detector designed for RF can be combined with an RF amplifier and a low-pass filter into a direct-conversion receiver.

Product detector

The General Block diagram of the product is as shown below:



$$output = Ac\cos(2\pi fct) \times S_{AM}$$



Flowchart/Algorithm:

1. Clear console, screen and close windows using the commands clc, clear all, close all.

2. Initialise vm,vc,fm,fc and fs with some values. Initialise t from -2 to 2 with an interval of (1/fs).Initialise df=fs/n where n is length of t. Initialise f=(-fs/2):df:((fs/2)-df).

3. Now we know wm,wc also because we know fm,fc. Then m is message signal which is vm*cos(wm*t) and c is the carrier signal vc*cos(wc*t).And the modulated signal s is product of message and carrier signals.

4. Use subplot command to create more than one graphs in one figure window. Use subplot command before plotting any graph such as subplot(<graph count x>,<graph count y>,<graph pos>) where graph count x = No. of graphs to be displayed horizontally, graph count y = No. of graphs to be displayed vertically, graph pos = Position of graph on the window which is usually row majored (1 – top left , 2 –top right , 3 – bottom left, 4 – bottom right in case of 2 by 2 subplot).

- **5.** Now by using subplot plot the message signal, carrier signal and modulated signal in the time domain.

6.Now by using in-built functions fft and fftshift which is fast fourier transform, we can find fourier transform of various signals.

7.Plotting the message signal, carrier signal and modulated signal in frequency domain using subplot.

8.To provide X axis label, use xlabel(<string>) and to provide Y axis label, use ylabel(<string>).

9.Use title(<string>) to provide title to the graph.

10. Use axis([]) to provide axis limits to the graph.

11. Do the above process for all required graphs.

12. Now comes the demodulation, multiplying the modulated signal with carrier signal takes place.

13. Then use the in-built functions butter,filter for getting the required part of signal. As we need low pass filter in finding coefficients A,B in butter syntax keep 'low'.And use filter for getting demodulated signal which is "y".Then again using subplot, plot the output of filter,demodulated signal in both time domain and frequency domain.Then again for comparing message signal and demodulated signal pot them in one graph using plot command and scaling it accordingly, then we can compare them and conclude.

14. It is recommended to use Sections as shown in the Code section so as to plot the graphs in a legible form. Sections can be used in the following fashion:- %% <title of section>

15. Run the written code using F5 or section using CTRL + F5.

16. Save the graphs from File > Save As.

17. The required experiment has been completed successfully

•

Code:

```
Clc     %to clear command window

clear all

close all

vm=5;  %initialising vm

vc=5;  %initialising vc

fm=1;  %initialising fm
```

```
wm=2*pi*fm;
fc=10;  %initialising fc
wc=2*pi*fc;
fs=100;         %initialising fs
t=-2:(1/fs):2;  %initialising t
n=length(t);    %Finding length of t
df=fs/n;        %df=fs/n
f=(-fs/2):df:((fs/2)-df);  %initialising f
m=vm*cos(wm.*t);     %Message signal
c=vc*cos(wc.*t);     %Carrier signal
s=m.*c;              %Modulated signal
figure:subplot(3,2,1):plot(t,m);  %Plotting in time domain
xlabel('Time');
ylabel('Amplitude');
title('Time domain');

subplot(3,2,3):plot(t,c);
xlabel('Time');
ylabel('Amplitude');
title('Time domain');

subplot(3,2,5):plot(t,s);
xlabel('Time');
ylabel('Amplitude');
title('Time domain');
```

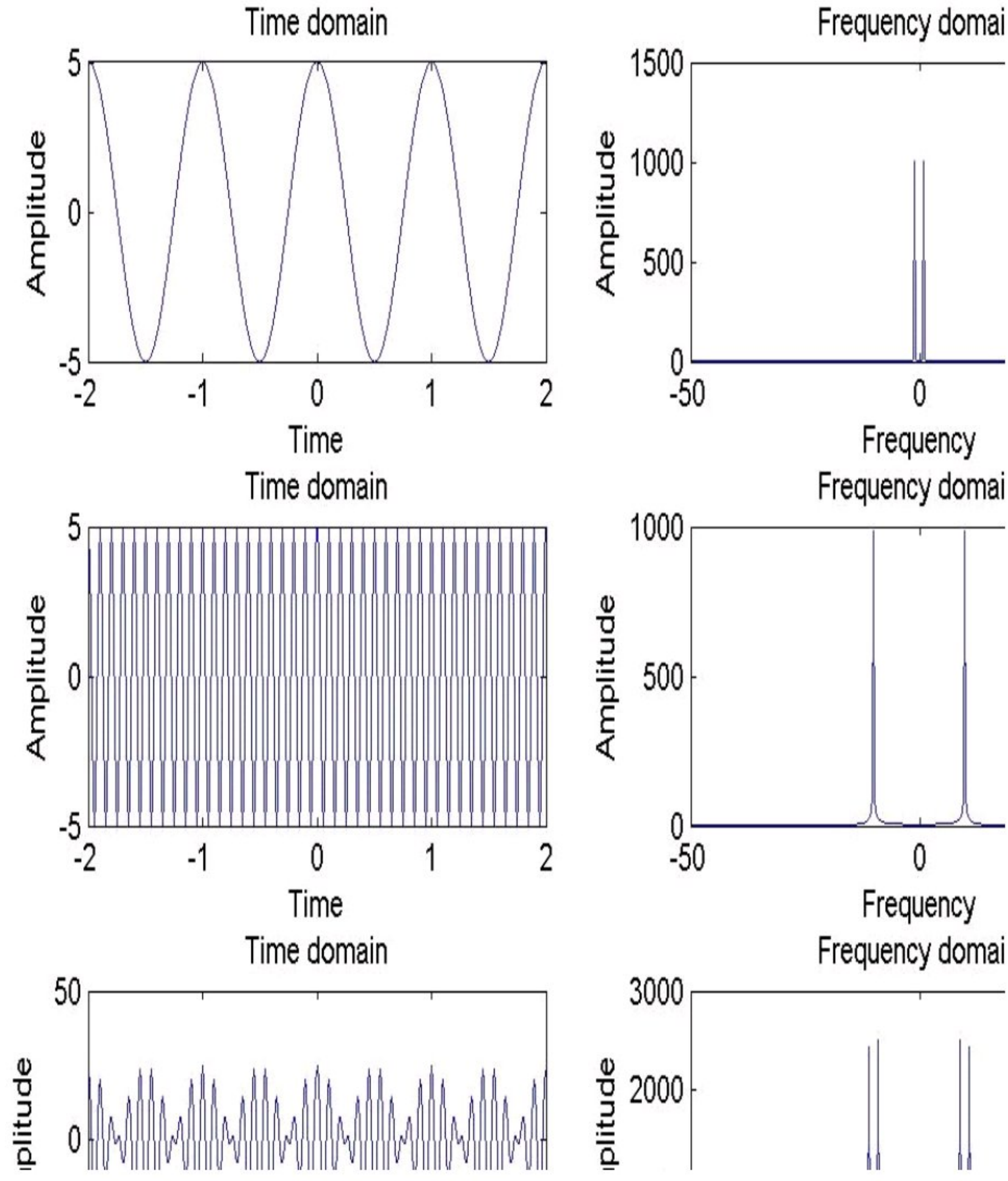
```
%plotting in frequency domain
s1=fftshift(fft(s));
m1=fftshift(fft(m));
c1=fftshift(fft(c));
subplot(3,2,2):plot(f,abs(m1));
xlabel('Frequency');
ylabel('Amplitude');
title('Frequency domain');

subplot(3,2,4):plot(f,abs(c1));
xlabel('Frequency');
ylabel('Amplitude');
title('Frequency domain');

subplot(3,2,6):plot(f,abs(s1));
xlabel('Frequency');
ylabel('Amplitude');
title('Frequency domain');

%Demodulation
dem=s.*c;
[B A]=butter(3,(fm/fs),'low');
y=filter(B,A,dem);        %Low pass filter
figure:subplot(2,2,1):plot(t,dem);          %Plotting
xlabel('Time');
ylabel('Amplitude');
```

```
title('Time domain');

subplot(2,2,2):plot(f,abs(fftshift(fft(dem))));
xlabel('Frequency');
ylabel('Amplitude');
title('Frequency domain');

subplot(2,2,3);
plot(t,y);
xlabel('Time');
ylabel('Amplitude');
title('Time domain');

subplot(2,2,4);
plot(f,abs(fftshift(fft(y))));
xlabel('Frequency');
ylabel('Amplitude');
title('Frequency domain');
%Comparing message signal and Demodulated signal
figure:plot(t,m,t,-0.75*y);
xlabel('Time');
ylabel('Amplitude');
legend('Message signal','Demodulated signal');
title('Time domain Modulated and Demodulated');
```

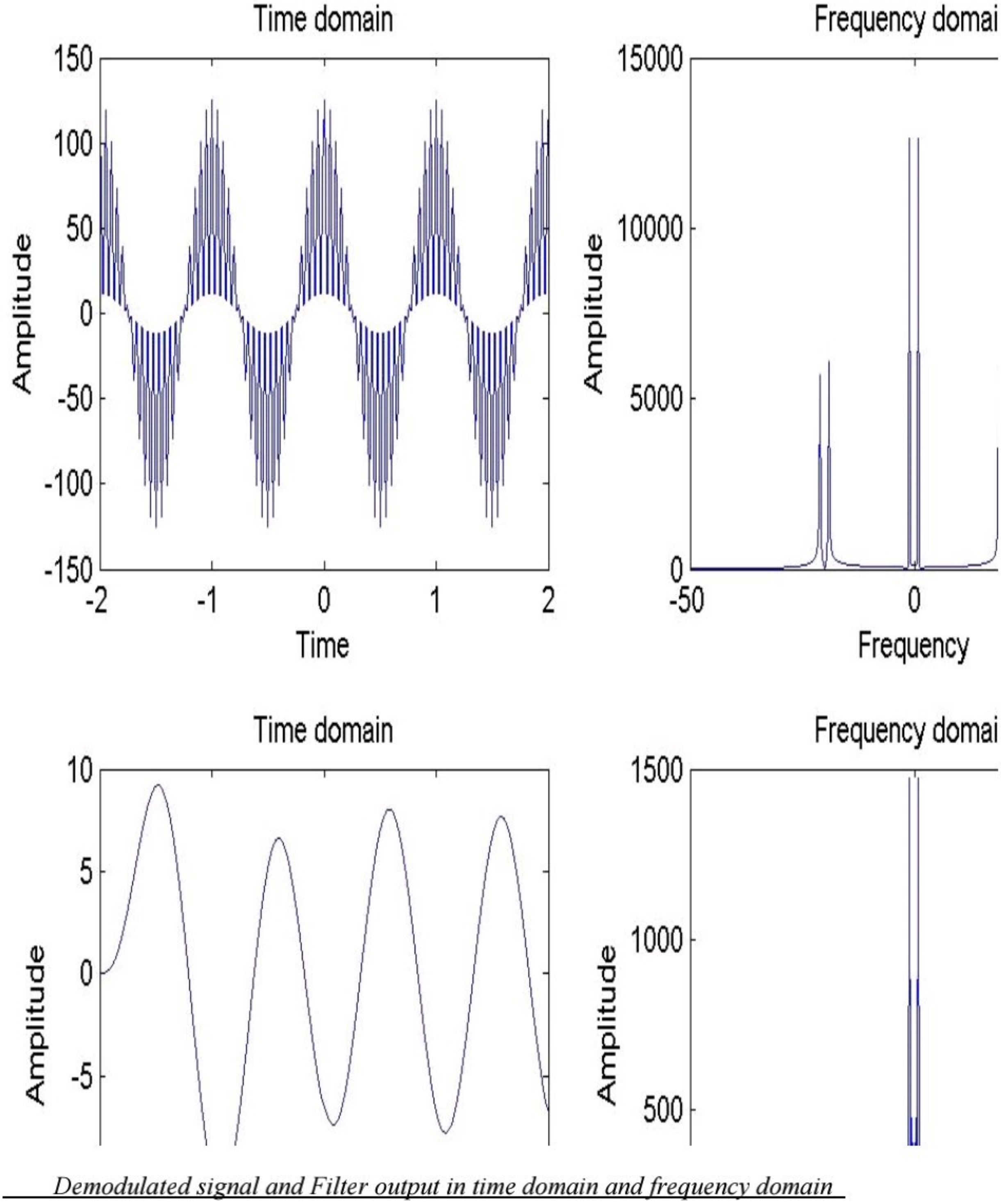
Observations:

We observed the waveforms of modulated signal, carrier signal, message signal , filter output , demodulated signal in both frequency and time domain. We even observed both message signal and demodulated signal in one plot and compared them with proper scaling.

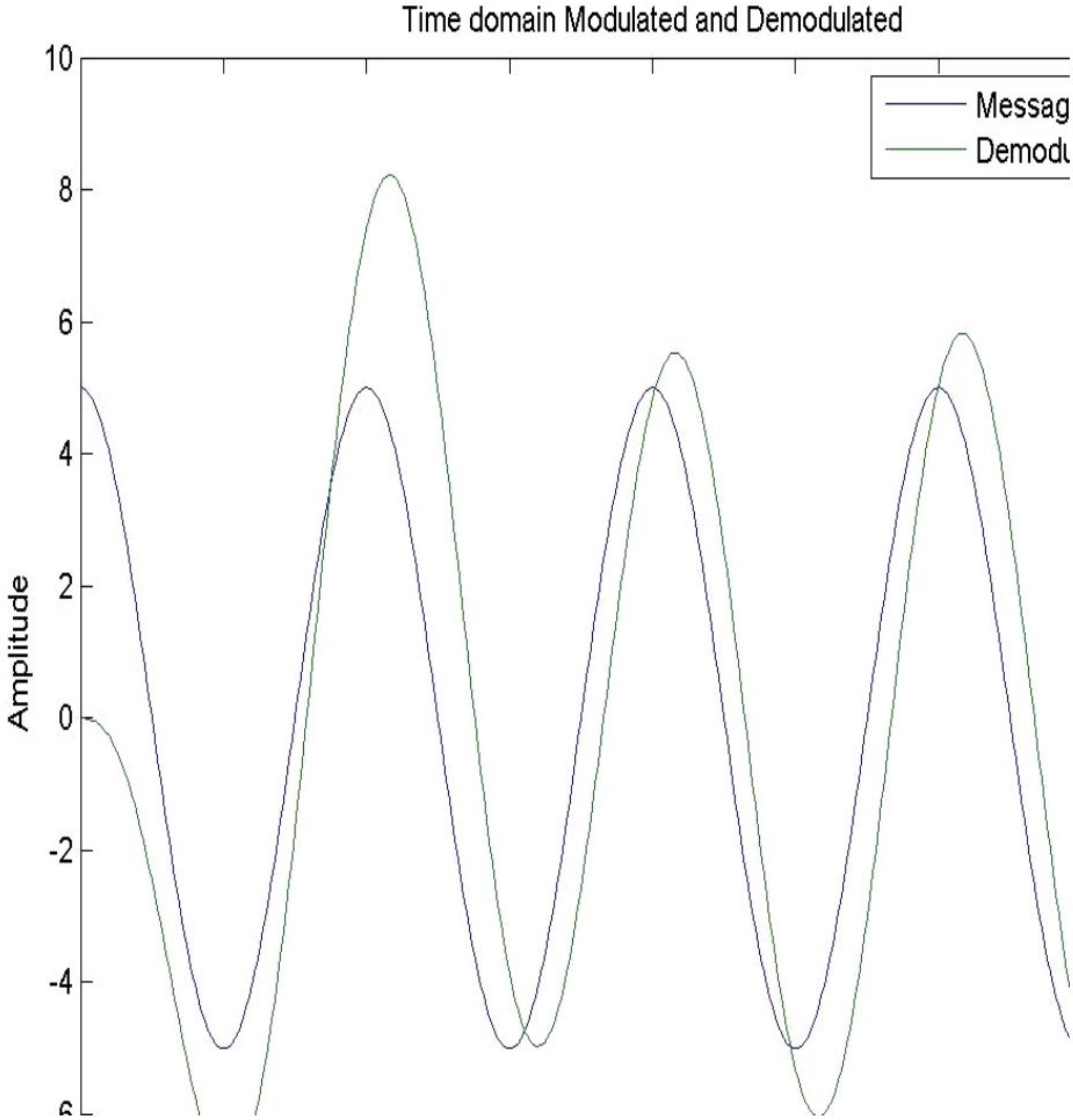
Output Waveforms:



Message signal, Carrier signal, Modulated signal in time domain and frequency domain



Demodulated signal and Filter output in time domain and frequency domain



Comparing message signal and demodulated signal

Conclusion: In this experiment we implemented Amplitude Modulation and Demodulation and plotted them in both time and frequency domain using ***MATLAB***.

Remarks: **Signature:**

DELTA MODULATION DEMODULATION

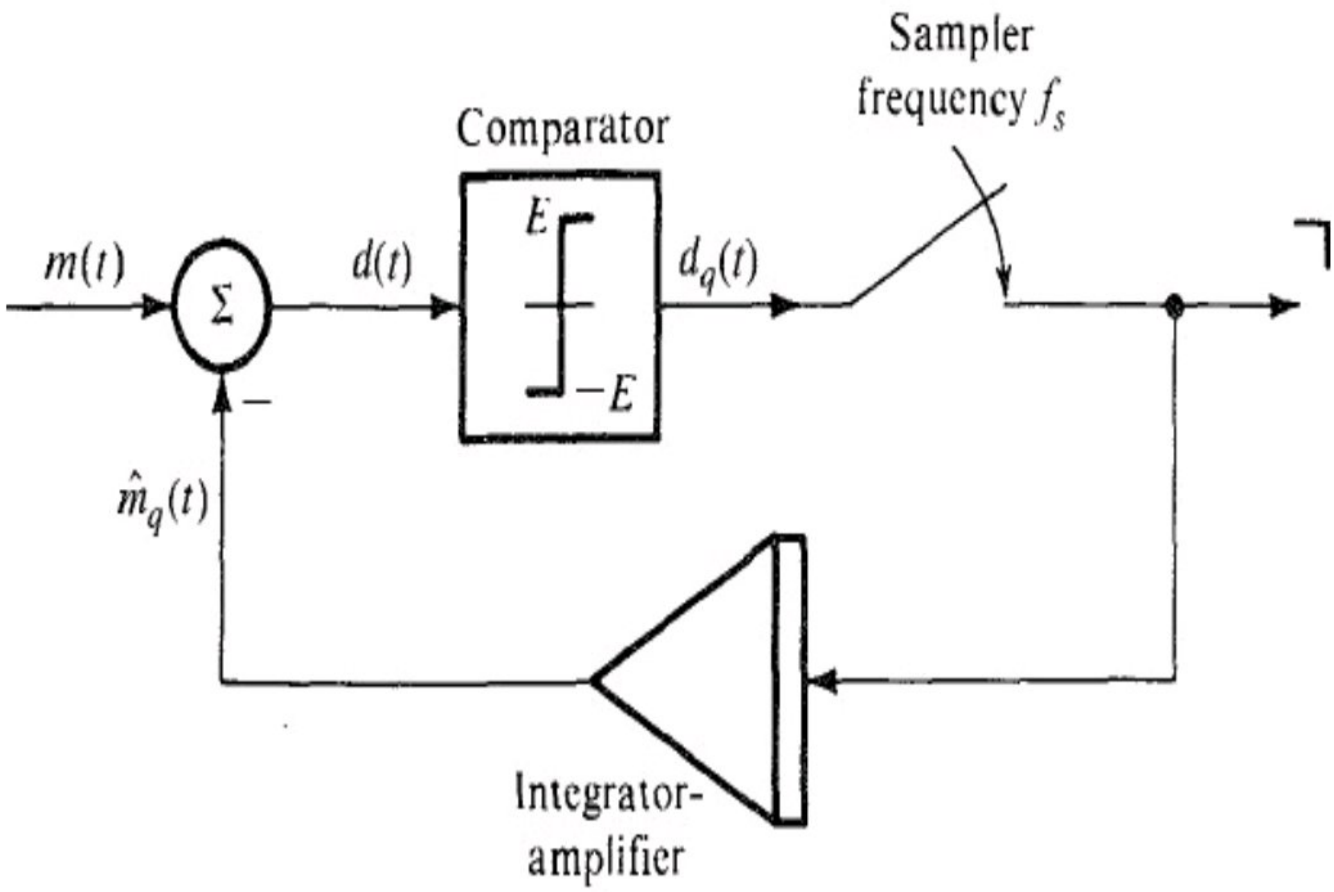
Experiment No: **09** **Date:**

Aim: **Write a MATLAB code to modulate and demodulate the given signal by Delta Modulation Technique.**

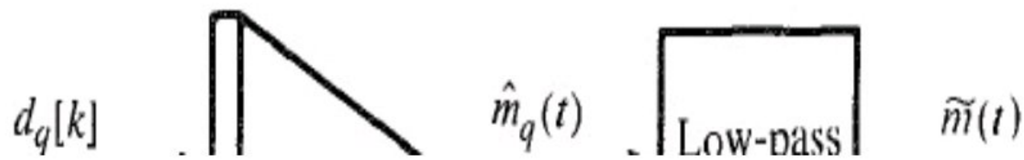
Brief Theory/Equations:

A delta modulation (DM or Δ-modulation) is an analog-to-digital and digital-to-analog signal conversion technique used for transmission of voice information where quality is not of primary importance. DM is the simplest form of differential pulse-code modulation (DPCM) where the difference between successive samples is encoded into n-bit data streams. In delta modulation, the transmitted data are reduced to a 1-bit data stream. Its main features are:

- The analog signal is approximated with a series of segments.
- Each segment of the approximated signal is compared to the preceding bits and the successive bits are determined by this comparison.
- Only the change of information is sent, that is, only an increase or decrease of the signal amplitude from the previous sample is sent whereas a no-change condition causes the modulated signal to remain at the same 0 or 1 state of the previous sample.
- To achieve high signal-to-noise ratio, delta modulation must use oversampling techniques, that is, the analog signal is sampled at a rate several times higher than the Nyquist rate.

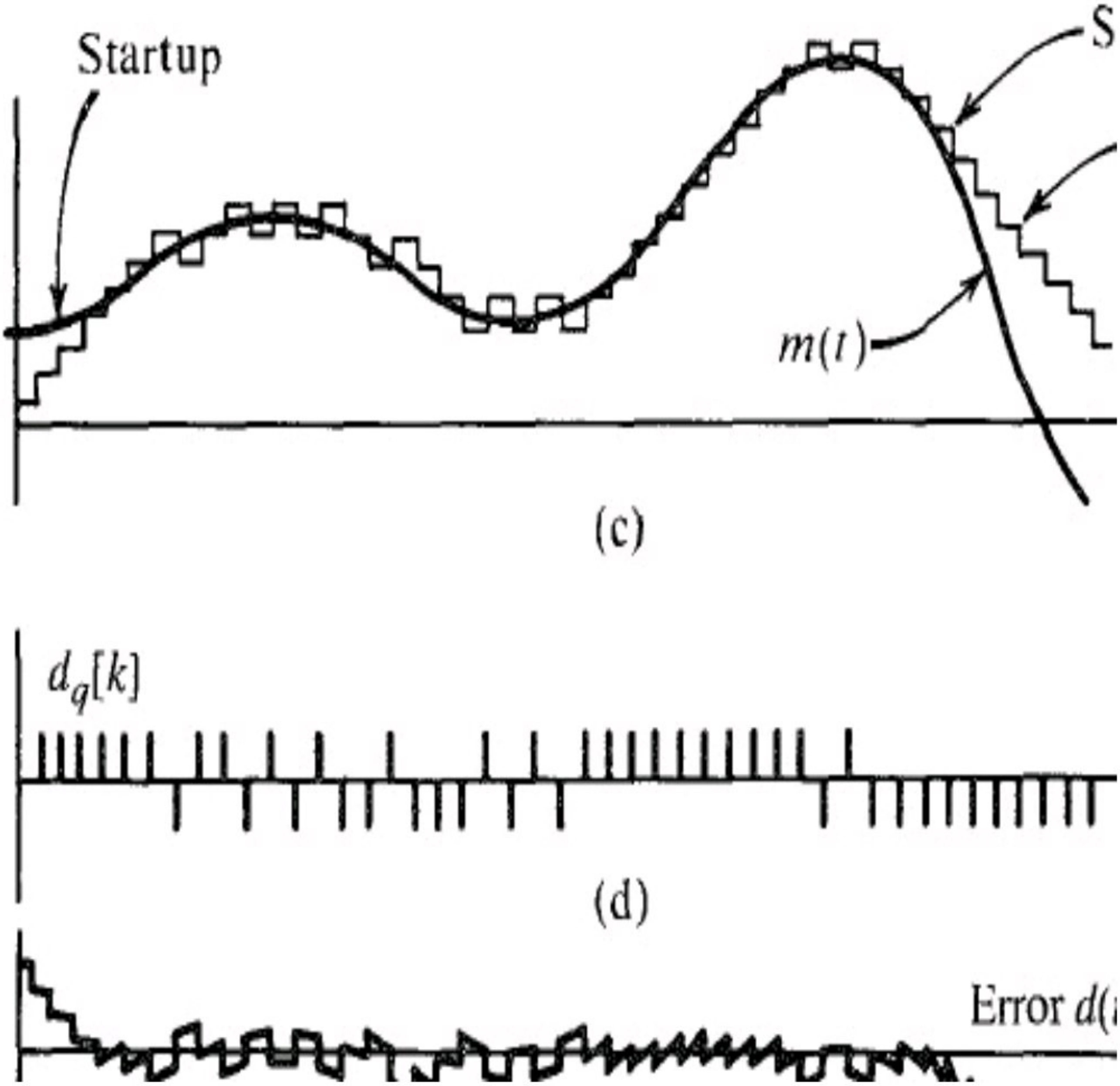


(a) Delta modulator



The modulator consists of comparator and sampler in the direct path and an integrator amplifier in the feedback path. The analog signal is compared with the feedback signal The error signal is applied to the comparator. If is positive, the comparator output is of constant amplitude E and if is negative, the comparator output is of constant value –E. Thus, the difference is the binary signal that is needed to generate 1 bit DPCM (Differential Pulse Code Modulation). The comparator output is sampled by a sampler at sampling frequency , samples per second where is much higher than the nyquist sampling rate. The sampler thus produces train of narrow pulses (to simulate impulses) with a positive pulse when and negative pulse when . Each sample is coded by a single binary pulse. The pulse train is delta modulated pulse train. The modulated signal is amplified and integrated in the feedback path to generate which tries to follow

Each pulse in at the input of the integrator gives rise to a step function (positive or negative depending on the polarity of pulse) in If for example, a positive pulse is generated in which gives rise to a positive step in trying to equalize to in small steps at every sampling instant as shown in figure below. It can be seen that is kind of staircase approximation of When is passed through a low pass filter, the coarseness of staircase in is eliminated and we get smoother and better approximation of The demodulator at the receiver consists of an amplifier-integrator (identical to that of feedback path of modulator) followed by a low pass filter.



In DM, the modulated signal carries information not about the signal samples but about the difference between successive samples. If the difference is positive or negative, a positive or negative pulse is generated in the modulated signal Basically, therefore, DM carries the information about the derivative of hence the name delta modulation. This can also be seen from the fact that integration of delta modulated signal yields which is an approximation of The information of difference between successive samples is transmitted by a 1 bit code word.

Threshold of Coding and Overloading

Threshold and overloading effects can be seen in the figure c. Variations in smaller than the step value (threshold of coding) are lost in DM. Moreover, If is too fast, derivative of it is too high, cannot follow and overloading occurs. This is known as slope overloading, which gives rise to the slope overload noise. This noise is one of the basic limiting factors in the performance of DM. We should expect slope overload rather than amplitude overload in DM, because DM basically carries the information about . The granular nature of the output signal gives rise to the granular noise similar to the quantization noise. The slope overload noise can be reduced by increasing E (the step size). This unfortunately increases the granular noise. There is an optimum value of E, which yields the best compromise giving the minimum overall noise. This optimum value of E depends on the sampling frequency and the nature of the signal.

The slope overload occurs when cannot follow . During the sampling interval , is capable of changing by , where is the height of the step (amplitude). Hence, the maximum slope that can follow is , or , where , is the sampling frequency. Hence, no overload occurs if

Consider the case of tone modulation (meaning a sinusoidal message):

The condition for no overload is

Hence, the maximum amplitude of this signal that can be tolerated without overload is given by

The overload amplitude of the modulating signal is inversely proportional to the frequency . For higher modulating frequencies, the overload occurs for smaller amplitudes. For voice signals, which contain all frequency components up to (say) 4 kHz, calculating by using in above equation will give an overly conservative value. It has been shown that for voice signals can be calculated by using

Thus, the maximum voice signal amplitude that can be used without causing slope overload in DM is the same as the maximum amplitude of a sinusoidal signal of reference frequency that can be used without causing slope overload in the same system

Fortunately, the voice spectrum (as well as the television video signal) also decays with frequency and closely follows the overload characteristics. For this reason, DM is well suited for voice (and television) signals. Actually, the voice signal spectrum (curve b) decreases as up to 2000 Hz, and beyond this frequency, it decreases as . If we had used a double integration in the feedback circuit instead of a single integration, would be proportional to . Hence, a better match between the voice spectrum and the overload characteristics is achieved by using a single integration up to 2000 Hz and a double integration beyond 2000 Hz. Such a circuit (the double integration) responds fast but has a tendency to instability, which can be reduced by using some low-order prediction along with double integration. A double integrator can be built by placing in cascade two low-pass RC integrators with time constants R1 C1 = 1 /200pi and R2C2 = 1/4000pi, respectively. This results in single integration from 100 to 2000 Hz and double integration beyond 2000 Hz.

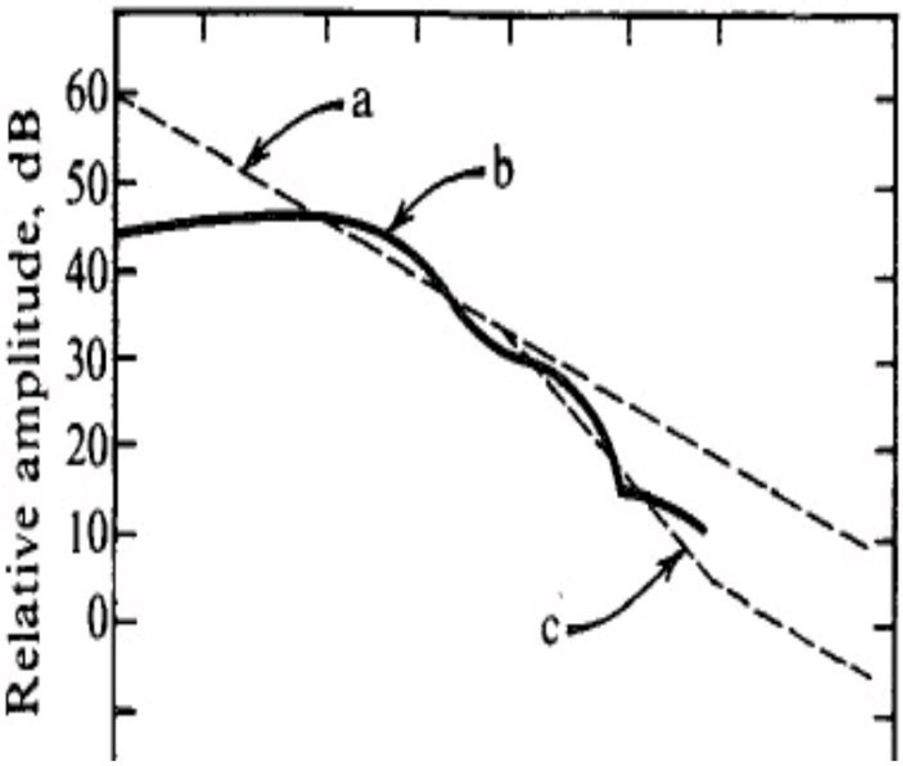


Figure 6.21 Voice

<u>**Delta Modulation : A special case of DPCM**</u>

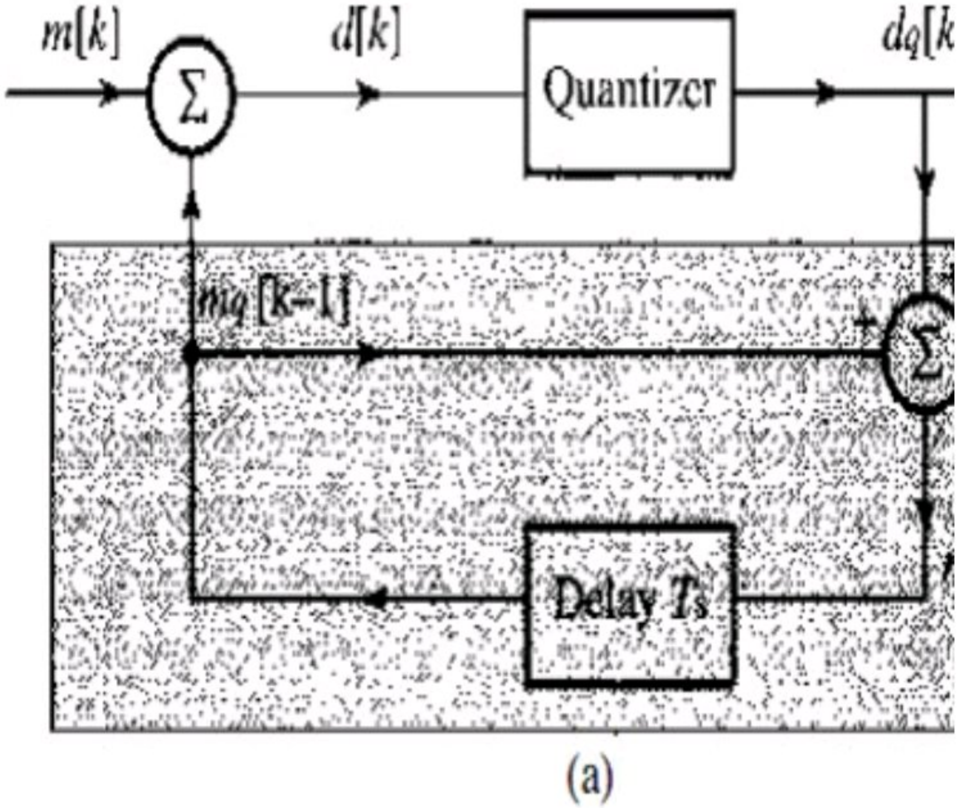
Sample correlation used in DPCM is further exploited in ***delta modulation (DM)*** by oversampling (typically four times the Nyquist rate) the baseband signal. This increases the correlation between adjacent samples, which results in a small prediction error that can be encoded using only one bit (L = 2). Thus, DM is basically a 1-bit DPCM, that is, a DPCM that uses only two levels (L = 2) for quantization of . In comparison to PCM (and DPCM), it is a very simple and inexpensive method of *A/D* conversion. A 1-bit codeword in DM makes word framing unnecessary at the transmitter and the receiver. This strategy allows us to use fewer bits per sample for encoding a baseband signal.

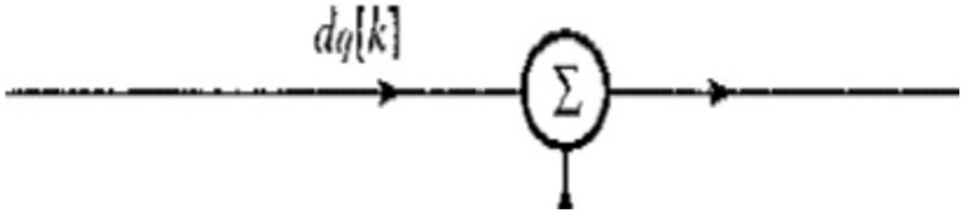
In DM, we use a first-order predictor, which, as seen earlier, is just a time delay of , (the sampling interval). Thus, the DM transmitter (modulator) and receiver (demodulator) are identical to those of the DPCM in Fig. below, with a time delay for the predictor, as shown in

Fig, from which we can write

Hence,

Figure 6.30 Delta modulation is a special case of DPCM.





Or we can write,

Proceeding in this manner, assuming zero initial condition, i.e. =0, we write

This shows that the receiver (demodulator) is just an accumulator (adder). If the output is represented by impulses, then the accumulator (receiver) may be realized by an integrator because its output is the sum of the strengths of the input impulses (sum of the areas underthe impulses). We may also replace with an integrator the feedback portion of the modulator (which is identical to the demodulator). The demodulator output is , which when passed through a low-pass filter yields the desired signal reconstructed from the quantized samples.

Algorithm:

- Impement the block diagram of DM as a special case of DPCM (Figure 6.30, page 295, "Modern Analog and Digital Communication" by B.P. Lathi 4th edition)

- Consider the input/message signal as sinusoidal `m=Am*cos(2*pi*fm*t)`, with parameters, Am=1V, fm=1Hz.

- Define the time range with sampling frequency fs=20*fm (oversampling), hence, t can be defined as `t=-3:1/fs:3;`

- Define the step size del for the delta modulator which should satisfy the condition

  Hence, `del=(2*pi*fm*Am)/fs;`
- Choose the index i=1:length(t) ; length(t)=max no. of columns in t.
  - If i=1, then mq=0.
    - So, the difference signal `d(i)=m(i);`
    - Use the sign function of MATLAB to determine whether d is +ve or –ve
    - Determine the approximate difference value dq by applying hard limiting operation i.e. by multiplying sign(d) with del.
    - Approximated message signal mq=dq, for i=1.
  - Else
    - The difference signal, `d(i)=m(i)-mq(i-1);`
    - The approximated difference operation will be same as in case for i=1.
    - Approximated message signal (staircase approximation), `mq(i)=dq(i)+mq(i-1);`
- Figure1: plot the message signal and staircase approximation signal the same window. Use the command hold on. And use command stairs(t,mq) for approximated message signal.
- Figure2: plot the delta modulated signal, consider the modulated output x to be +1 if dq>0 else x will be -1.
  - Use stem command for plotting as it is a discrete time signal

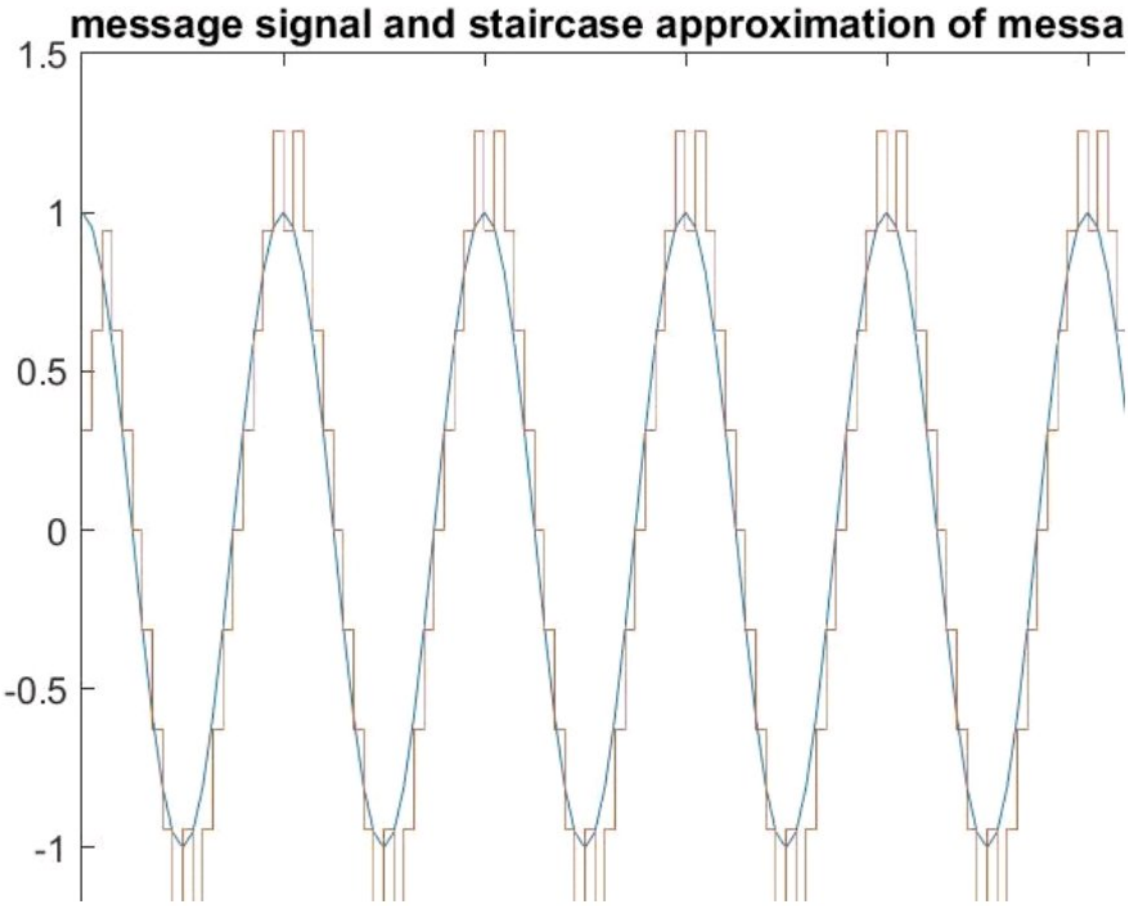
- For demodulation, pass the approximated message signal via low pass filter. (See FM demodulation algorithm for filtering logic).
- Figure 3: plot filtered output signal and original input signal in the same window.
    - `plot(t,2*y,t,m);`

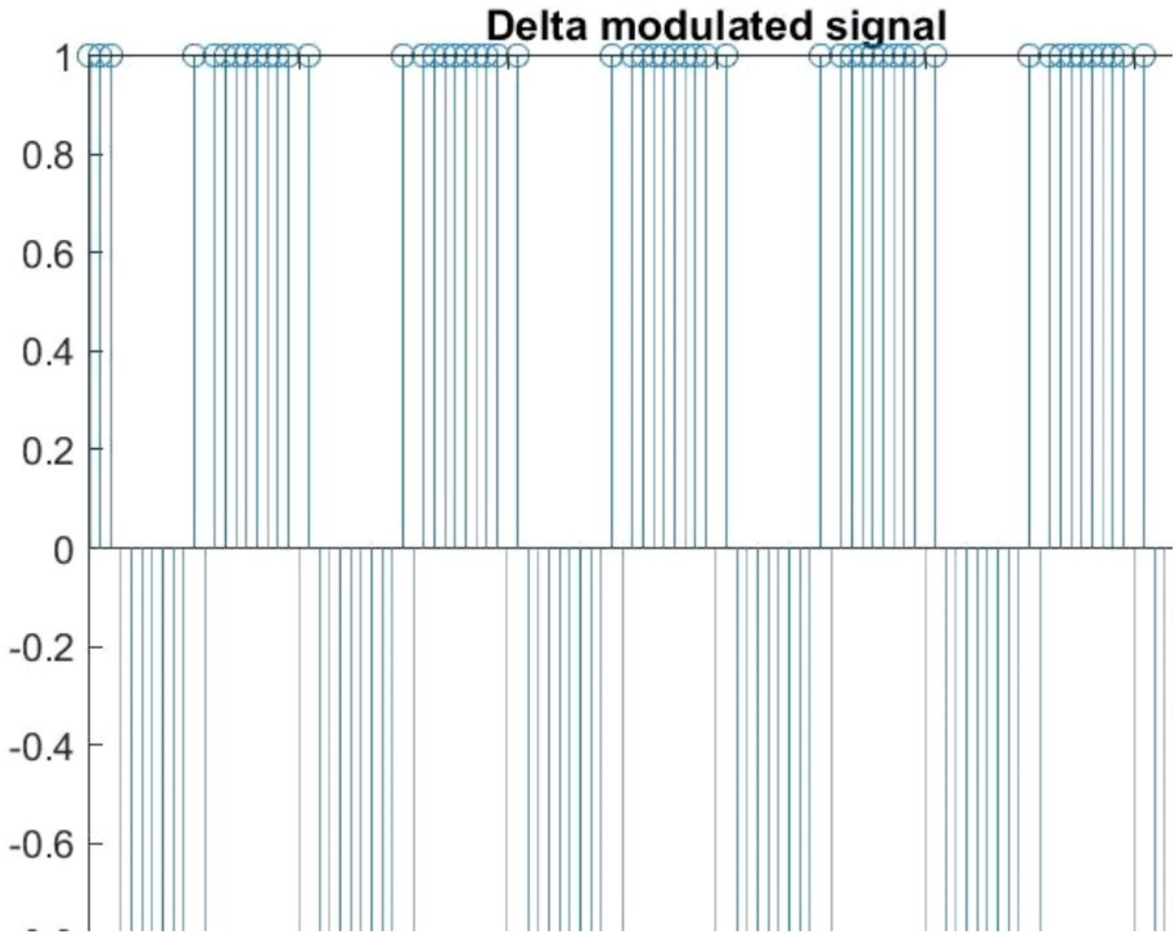
Code:

```
clc
clear all
fm=1;%defining fm
fs=20*fm;%defining fs
t=-3:(1/fs):3;%defining t
am=1;%defining am
m=am*cos(2*pi*fm*t);%the message signal
del=(2*pi*fm*am)/fs;%defining del
%Running a for loop
for i=1:length(t)
  if(i==1)
    mq=0;
    d(i)=m(i);
    dq=(sign(d))*del;
    mq=dq;
  else
    d(i)=m(i)-mq(i-1);
    dq=(sign(d))*del;
    mq(i)=dq(i)+mq(i-1);
  end
end
%plotting
plot(t,m);
%holding
hold on;
stairs(t,mq);
title('message signal and staircase approximation of message signal');%title of plot
%running for loop
for n=1:length(t)
if(d(n)>0)
  dm(n)=1;
```

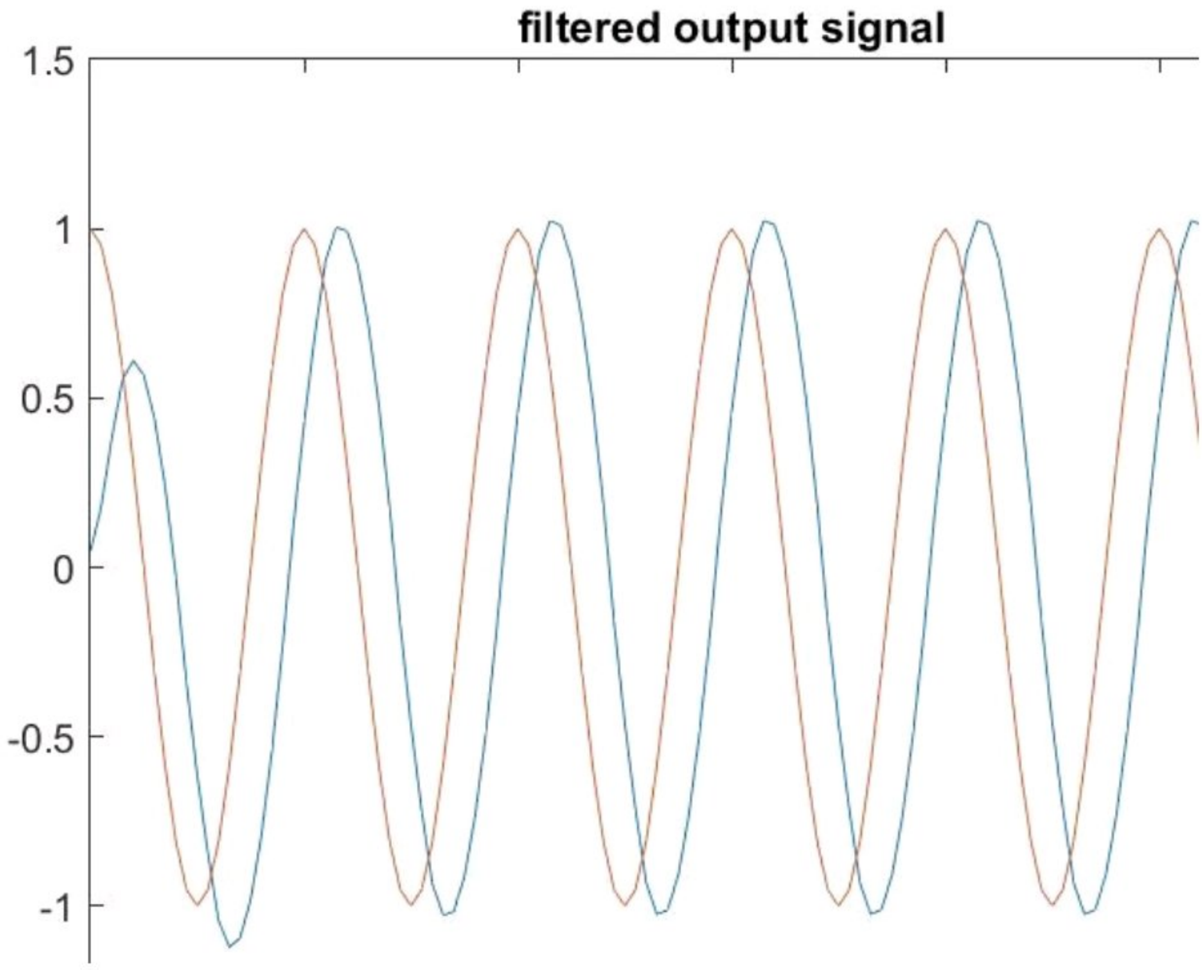
```
else
   dm(n)=-1;
end
end
%another figure
figure
stem(t,dm);
title('Delta modulated signal');%title of plot
n=1;
wn=fm/fs;%defining wn
[b a]=butter(n,wn,'low');
y=filter(b,a,mq);%y is output of filter
%plotting again
figure
plot(t,2*y,t,m);
title('filtered output signal');%title of the plot
```

Output Waveforms:


message signal and staircase approximation of messa
1.5
1
0.5
0
-0.5
-1



Delta modulated signal
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6



Conclusion: In this experiment we performed Delta modulation and demodulation of message signal using ***MATLAB***.

Remarks: **Signature:**

FREQUENCY MODULATION IN MATLAB

Experiment No:10 **Date:**

Aim: To implement frequency modulation and demodulation using MATLAB.

Brief Theory/Equations:

When the angle of the carrier wave is varied in some manner with respect to modulating signal , the technique of modulation is known as angle modulation or exponential modulation.

General form, , where is the modulated signal, is the carrier frequency in radians and is the phase function which is time varying and captures the information, which you want to convey.

So, modulating signal somehow modifies this

Let be the instantaneous phase of the carrier signal, then instantaneous frequency can be obtained by taking the derivative of instantaneous phase w.r.t. time, t.

is the instantaneous frequency of the carrier wave or modulated signal

Where, is the instantaneous phase deviation and is the instantaneous frequency deviation.

Phase Modulation: (PM)

The instantaneous phase deviation carries the information or is varied linearly with

Where, is the phase modulation constant/ phase sensitivity of the modulator measured in radians/volt.

For PM, instantaneous phase

And instantaneous frequency

Modulated signal for PM,

Frequency Modulation: (FM)

The instantaneous frequency carries the information or message signal.

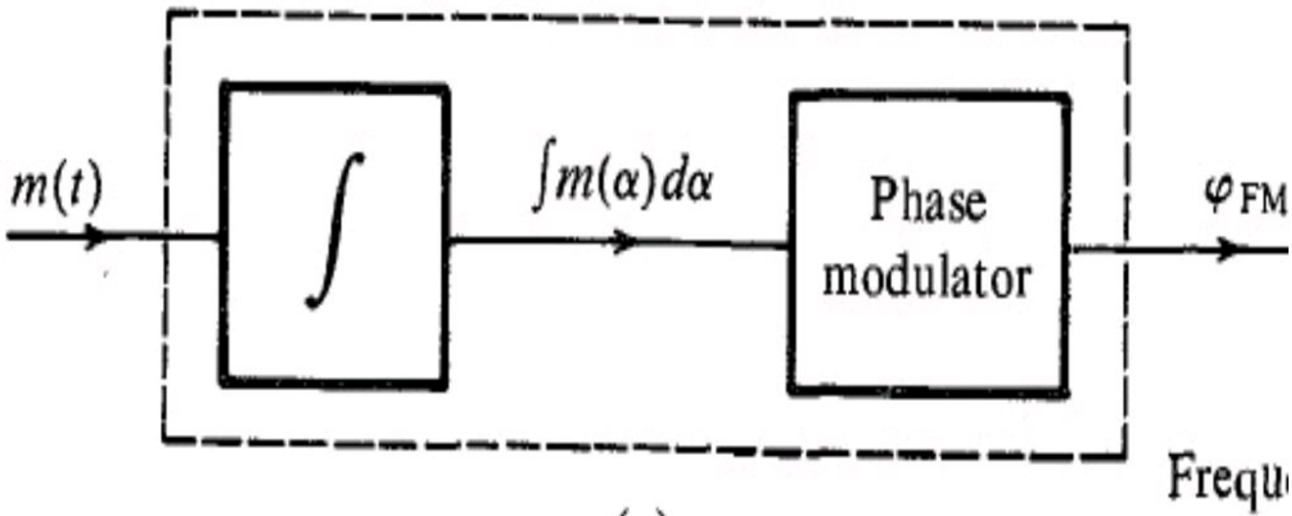
Where, is the frequency modulation constant/ frequency sensitivity of the modulator measured in radians/(seconds*volt) or Hz/volt.

For FM, instantaneous phase deviation,

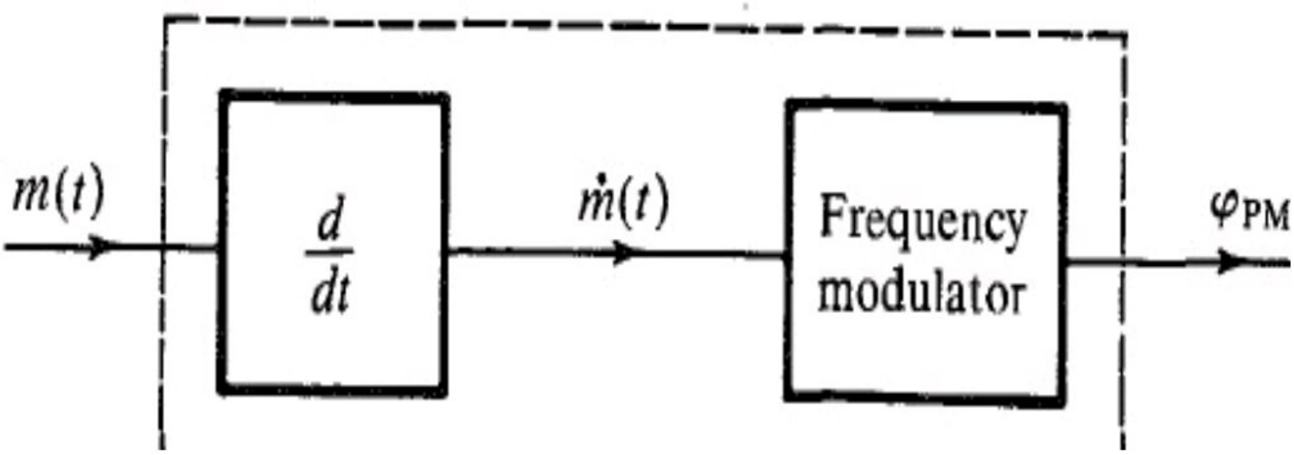
Hence, instantaneous phase,

And instantaneous frequency,

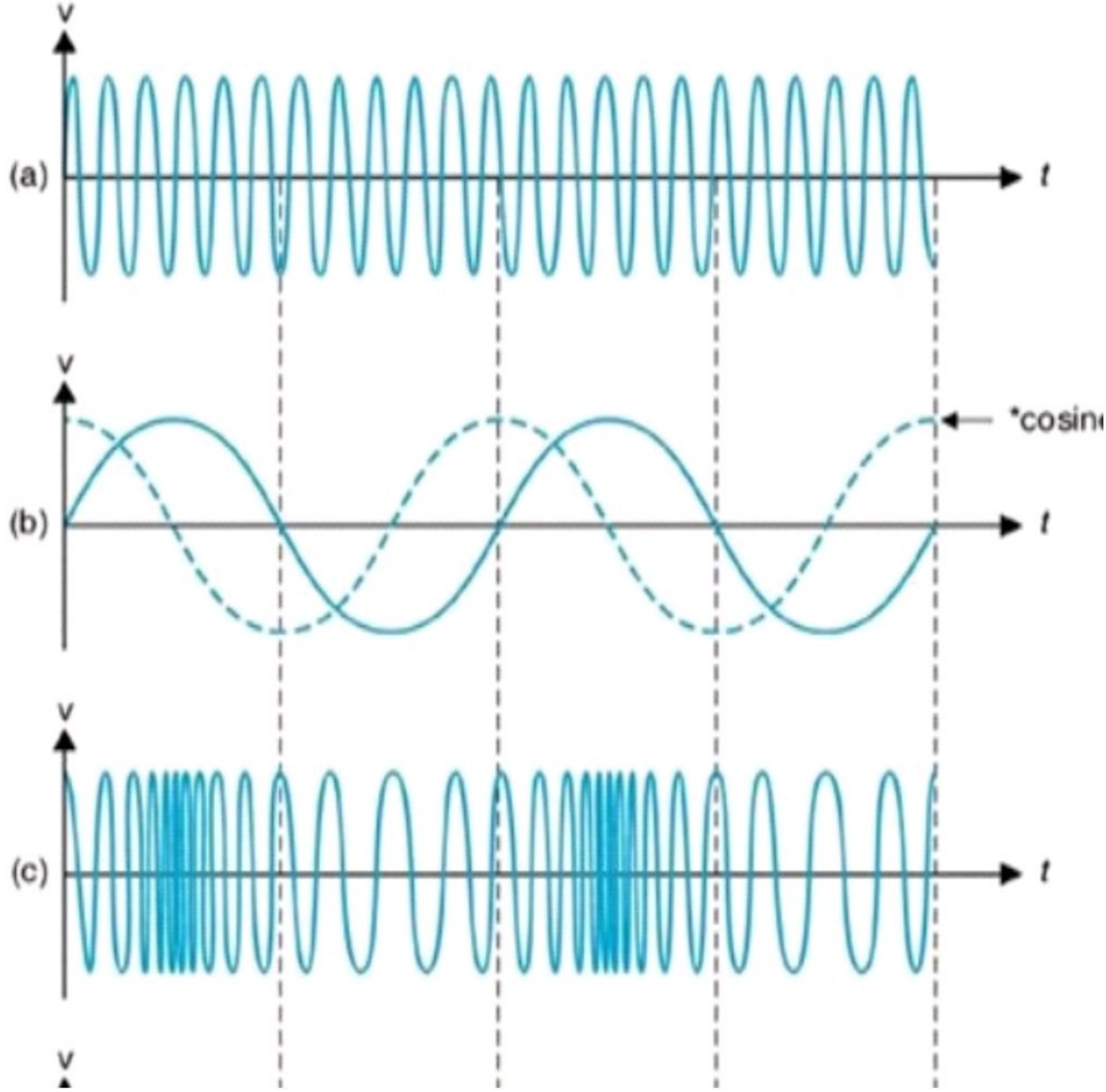
Modulated Signal for FM,



(a)



Considering, message signal as sine wave, the derivative of it will be cosine and respective FM and PM wave are shown in the next figure.



or else if we consider , the instantaneous phase deviation for PM and FM will be;

PM,

FM,

Hence, the modulated signal for PM and FM become

Where, is the modulation index for PM

Where, is the modulation index for FM

is the peak frequency deviation

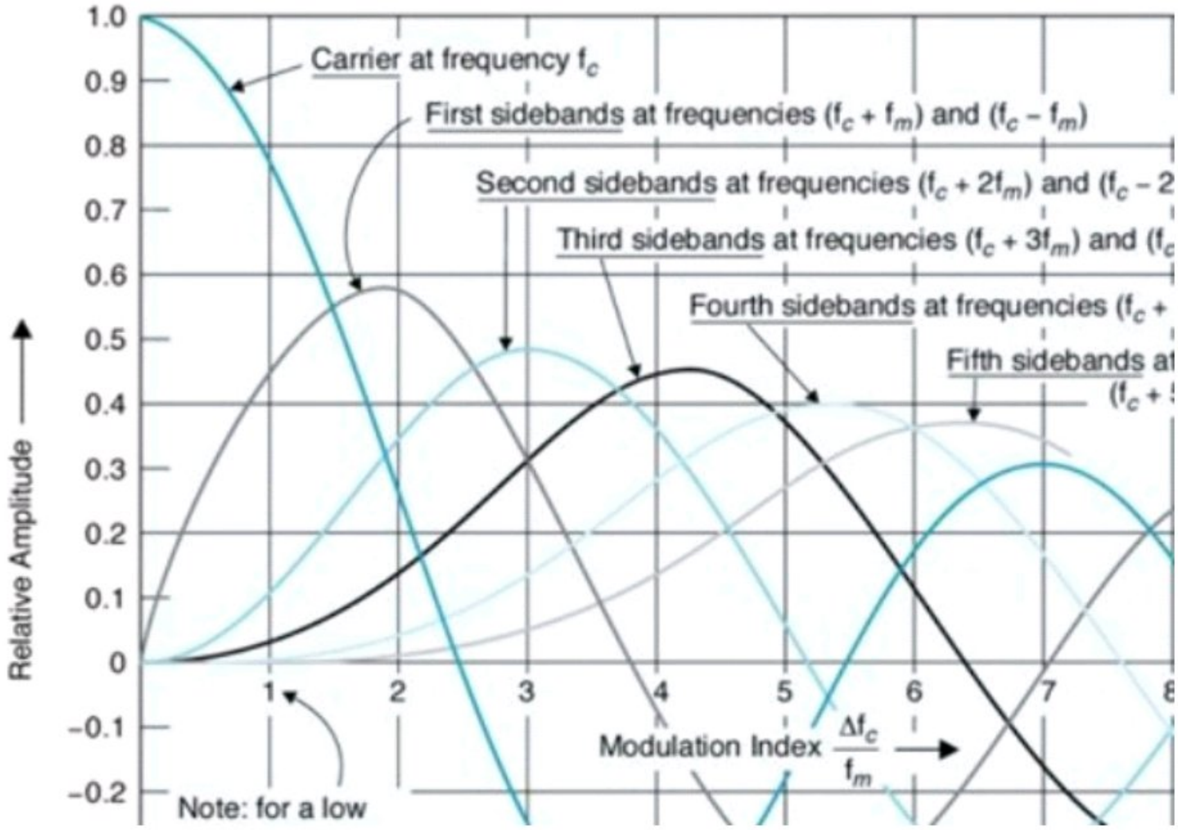
Frequency Deviation:

- The amount of change in the carrier frequency produced, by the amplitude of the input modulating signal, is called **frequency deviation**.
- The Carrier frequency swings between fmax and fmin as the input varies in its amplitude.
- The difference between fmax and fc is known as frequency deviation. = fmax – fc
- Similarly, the difference between fc and fmin also is known as frequency deviation. = fc –fmin

The frequency modulated signal can also be expressed in terms of Bessel function by taking the fourier series expansion.

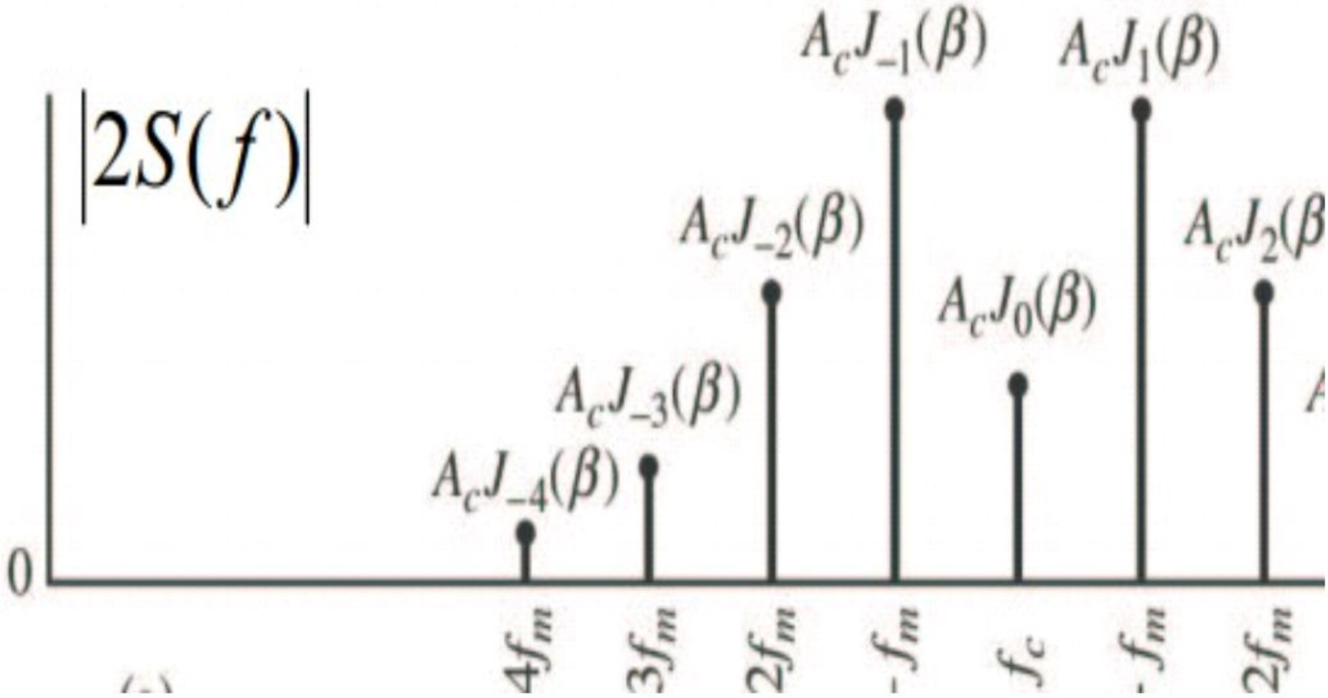
Reference: Modern Digital and Analog Communication by B.P.Lathi

Where, is Bessel function of n order with argument



By taking the fourier transform of the above equation, we get

Theoretically, single tone FM has carrier component with infinite number of side-bands at and its bandwidth is infinite.



Bandwidth of FM:

As the value of n increase, the significant power level in sideband component decreases

i.e.

So we consider only k sidebands on either sides of and look at these 2k+1 components i.e. from to and if we consider the power ratio,

Power ratio =

(is the carrier related component and is an even function)

The bandwidth is defined as band of frequencies over which signal contains 98% of its power that means the power ratio is 0.98.

In general,

In empirical sense, if we choose then power ratio turns to 0.98. (depends on the Bessel function values)

Hence,   [

Frequency Demodulation:

We know that the equation of FM wave is

$$s(t) = A_c \cos\left(2\pi f_c t + 2\pi k_f \int m\right.$$

Differentiate the above equation with respect to '**t**'.

$$\frac{ds(t)}{dt} = -A_c \left(2\pi f_c + 2\pi k_f m(t)\right) \sin\left(2\pi f_c t + 2\pi k_f\right.$$

We can write,

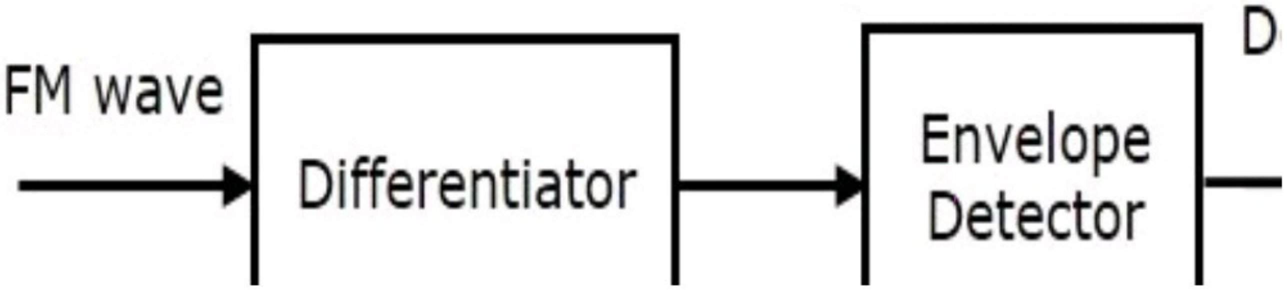
$$-\sin\theta \quad \text{as} \quad \sin\left(\theta - 18\right.$$

$$\Rightarrow \frac{ds(t)}{dt} = A_c \left(2\pi f_c + 2\pi k_f m(t)\right) \sin\left(2\pi f_c t + 2\pi k_f \int m\right.$$

$$\Rightarrow \frac{ds(t)}{dt} = A_c (2\pi f_c) \left[1 + \left(\frac{k_f}{k_c}\right) m(t)\right] \sin\left(2\pi f_c t + 2\pi k_f \int m\right.$$

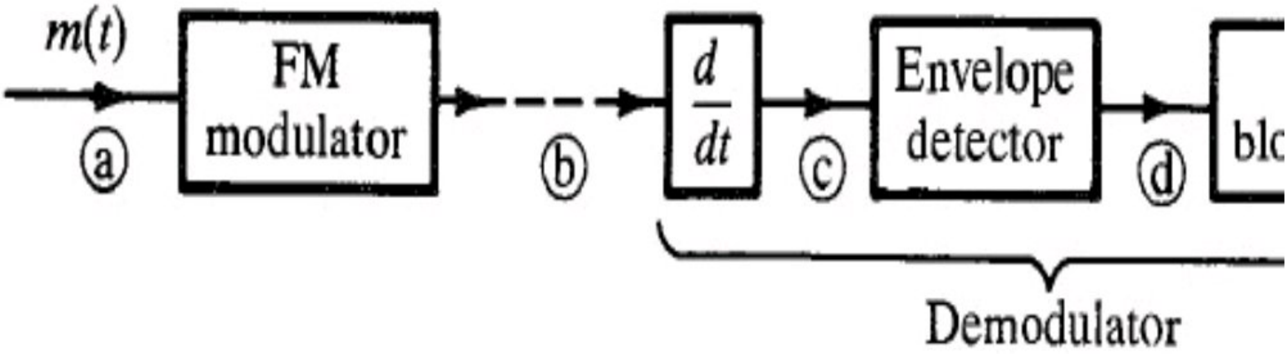
In the above equation, the amplitude term resembles the envelope of AM wave and the angle term resembles the angle of FM wave. Here, our requirement is the modulating signal m(t)m(t). Hence, we can recover it from the envelope of AM wave.

The following figure shows the block diagram of FM demodulator using frequency discrimination method.



This block diagram consists of the differentiator and the envelope detector. Differentiator is used to convert the FM wave into a combination of AM wave and FM wave. This means, it converts the frequency variations of FM wave into the corresponding voltage (amplitude) variations of AM wave. We know the operation of the envelope detector. It produces the demodulated output of AM wave, which is nothing but the modulating signal.

The FM communication link is shown below:



Algorithm:

- Define the sampling frequency say,
- Define the time range using the sampling frequency t= -10 : 1/: 10
- Consider, message signal, , where and carrier signal where . Keep the amplitude of message and carrier signal same.
- Assume and
- For PM,
- For FM,

  Use integral function in MATLAB in general so it can be done for any input signal.
- Figure1: Plot input signal, carrier signal, FM signal, derivative of input signal and PM signal using subplot(511) to subplot(515);

  Use diff function in matlab for the derivative of the signal

  For example, y=diff(x), pad the last value of the signal so as to ensure the same size because difference values will be one less than the given values.

y=[diff(x) diff(end)]

- Figure 2: plot the frequency spectrum of FM and PM signals using the commands fft and fftshift.
  - Define n=length(t) %gives no. of columns in t
  - Define the step size for frequency axis fp which should be of same size as that of t. (matrix dimensions must match for plotting).
    - df=fs/n; where fs is the sampling frequency
  - Define frequency axis fp = -fs/2:df:fs/2-df (df for getting same size matrices)
  - Take the fourier transform of FM and PM using fft command
  - Y = fft(x) returns the discrete Fourier transform (DFT) of vector x, computed with a fast Fourier transform (FFT) algorithm.
  - Y = fftshift(X) rearranges the outputs of fft by moving the zero-frequency component to the center of the array. It is useful for visualizing a Fourier transform with the zero-frequency component in the middle of the spectrum.
  - Can write in a single syntax as y=fftshift(fft(x));
  - Plot the frequency spectrum of FM and PM using the command plot(fp,y).Use subplot(211) to (212).

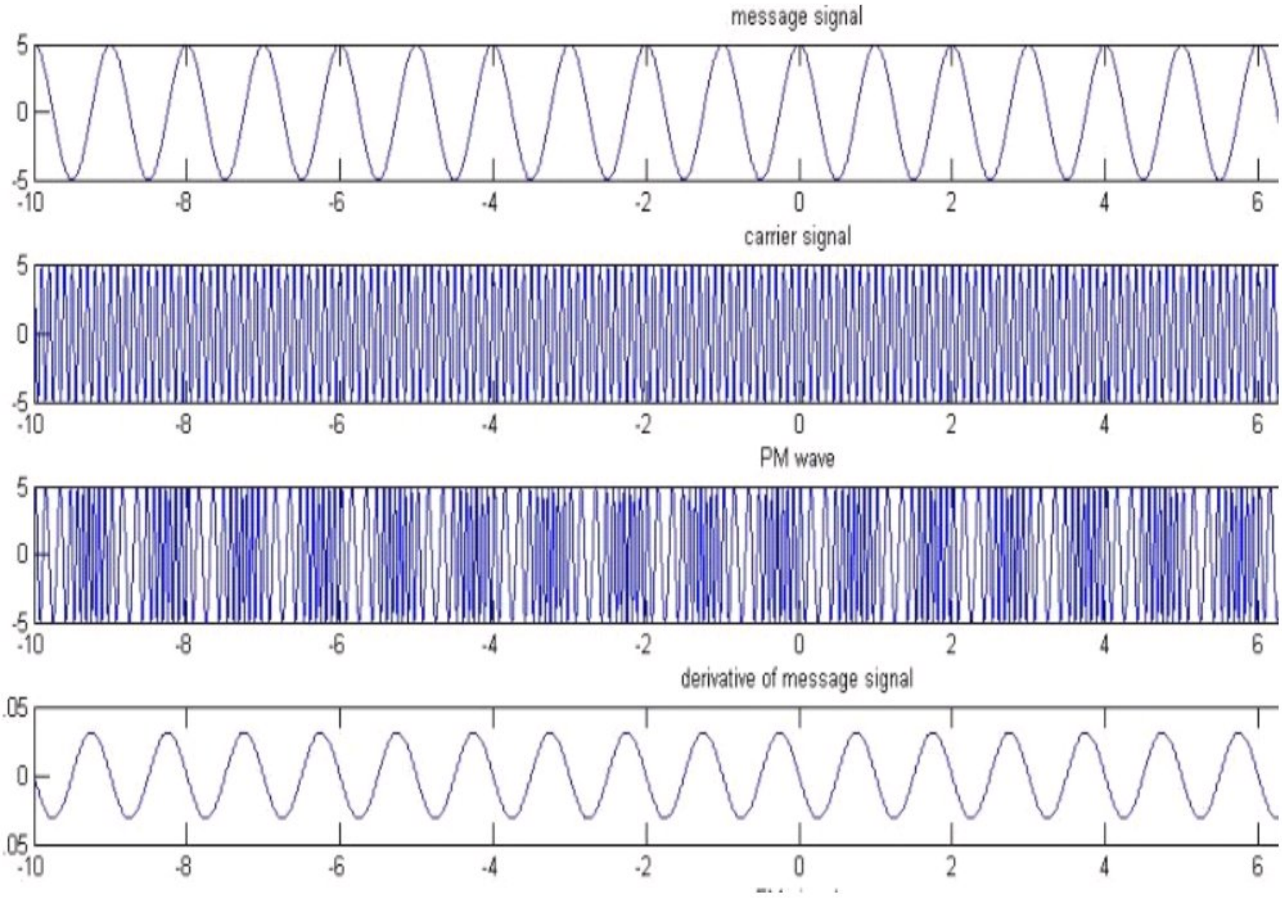
- **Demodulate the FM signal**
  - Take the derivative of FM signal using diff function of MATLAB as mentioned in one of the previous step.
  - Multiply the above signal with carrier signal, this will look like AM wave.
    - Do .* element wise multiplication
- Do the low pass filtering of the above signal using butter and filter command.
  - [b,a] = butter(n,Wn,'ftype'). **This creates a filter**
  - [b,a] = butter(n,Wn) designs an order n lowpass digital Butterworth filter with normalized cutoff frequency Wn. It returns the filter coefficients in length n+1 row vectors b and a, with coefficients in descending powers of z.

$$H(z) = \frac{b(1) + b(2)z^{-1} + \ldots + b(n +}{1 + a(2)z^{-1} + \ldots + a(n +}$$

- where the string 'ftype' is one of the following:
  - 'high' for a highpass digital filter with normalized cutoff frequency Wn
  - 'low' for a lowpass digital filter with normalized cutoff frequency Wn
  - 'stop' for an order 2*n bandstop digital filter if Wn is a two-element vector, Wn = [w1 w2]. The stopband is w1 < ω < w2.
  - 'bandpass' for an order 2*n bandpass filter if Wn is a two-element vector, Wn = [w1 w2]. The passband is w1 < ω < w2. Specifying a two-element vector, Wn, without an explicit 'ftype' defaults to a bandpass filter.
  - Cutoff frequency, Wn is that frequency where the magnitude response of the filter is . For butter, the normalized cutoff frequency Wn must be a number between 0 and 1, where 1 corresponds to the Nyquist frequency, π radians per sample.
  - Choose order, n=4 ; Wn=fm/fs; ftype='low'
- Use the filter command to apply the filter
  - y = filter(b,a,X), where b and a are the coefficients obtained in the previous step and X will the multiplied signal output (derivative of FM signal*Carrier Signal). y is the filtered output. Adjust order so that you get approximate input signal.
- Figure 3: Plots for FM demodulation
  - Include three figures using subplot(311) to subplot(313)
    - 1st – derivative of FM signal
    - 2nd - multiplied signal output (derivative of FM signal*Carrier Signal)
    - 3rd – filtered output y. Include message signal as well in this
      - Example: plot(t,m,t,y)

Expected Output waveforms:

Figure 1:



message signal
carrier signal
PM wave
derivative of message signal

Figure 2:

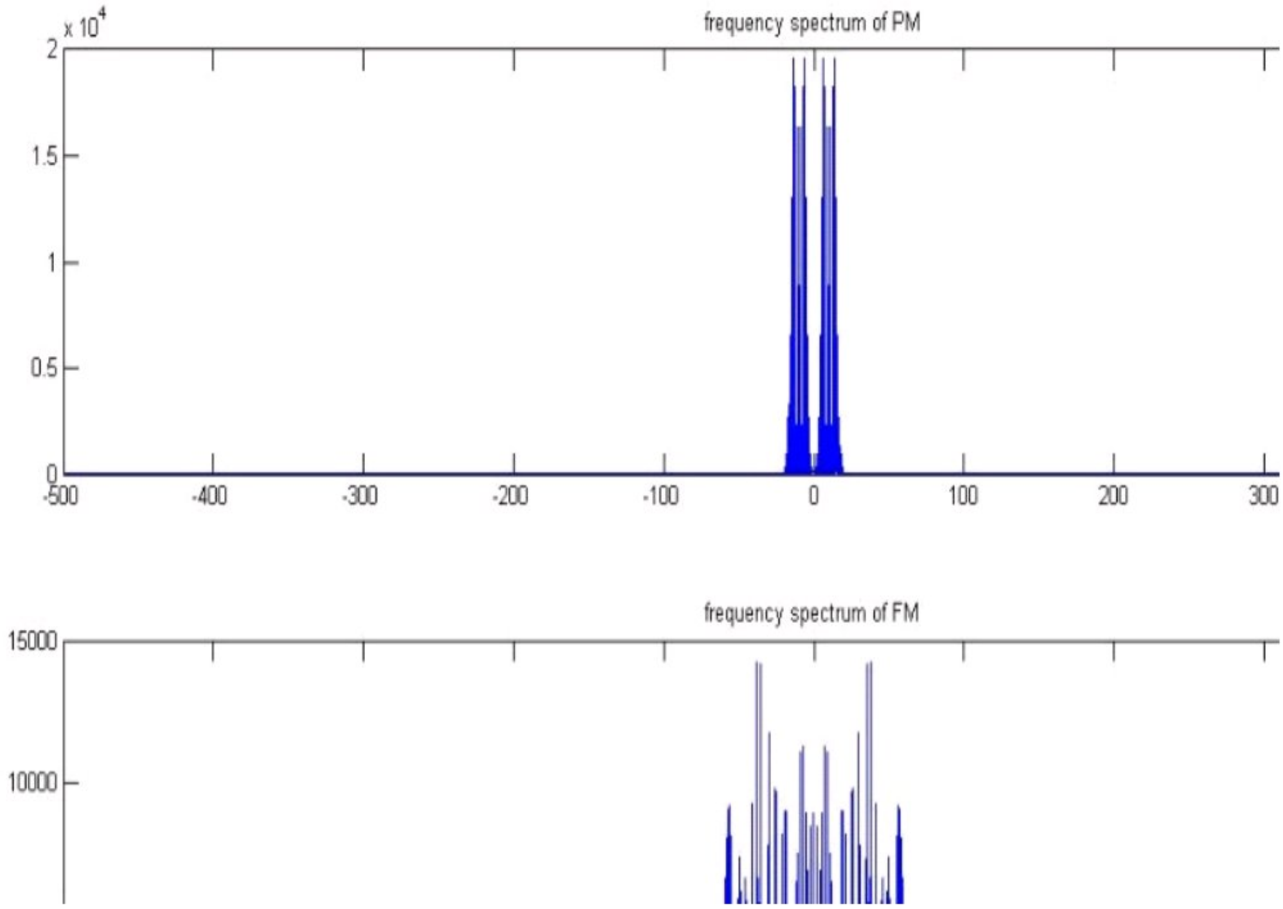
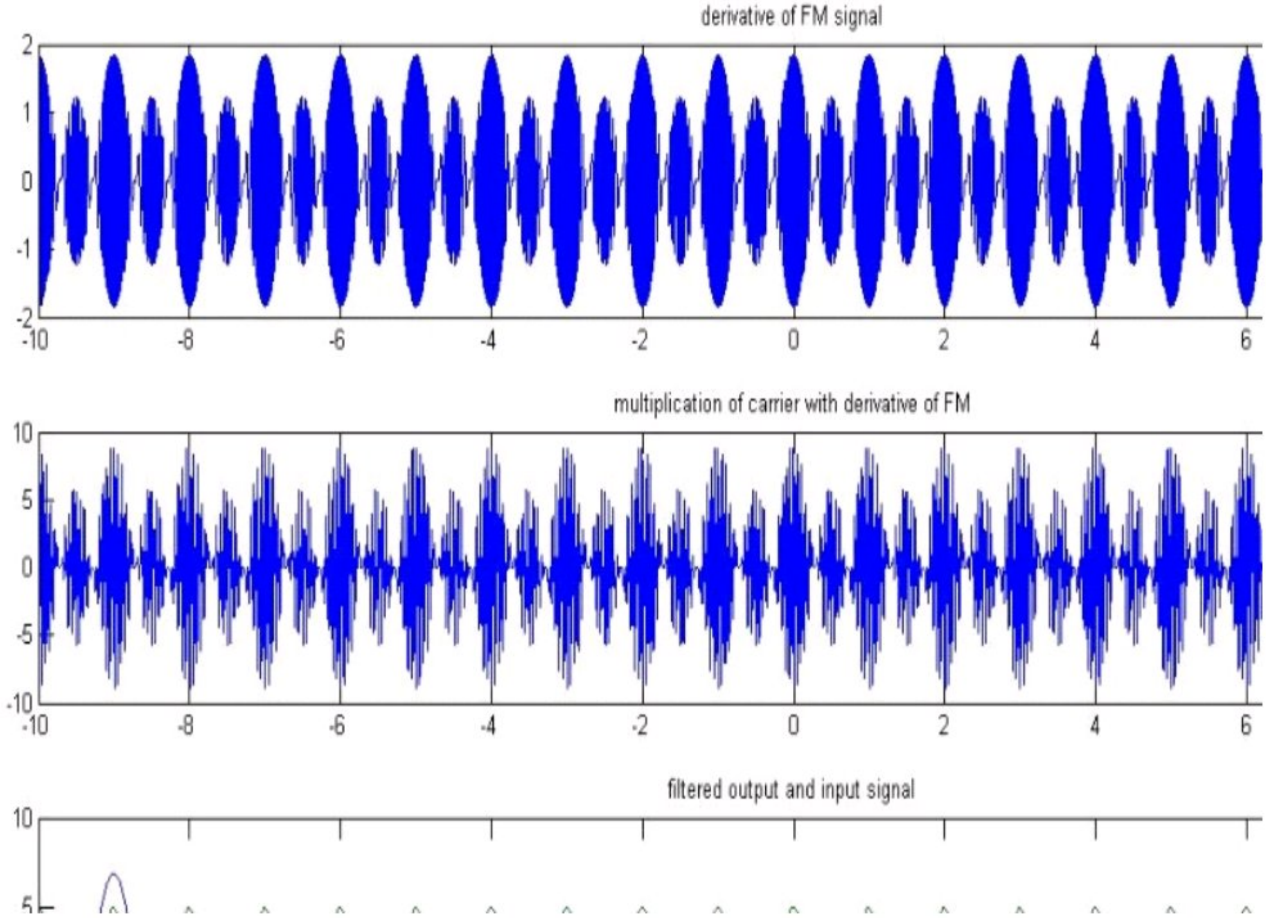

x 10^4
frequency spectrum of PM
2
1.5
1
0.5
0
-500
-400
-300
-200
-100
0
100
200
300
frequency spectrum of FM
15000
10000

Figure 3:



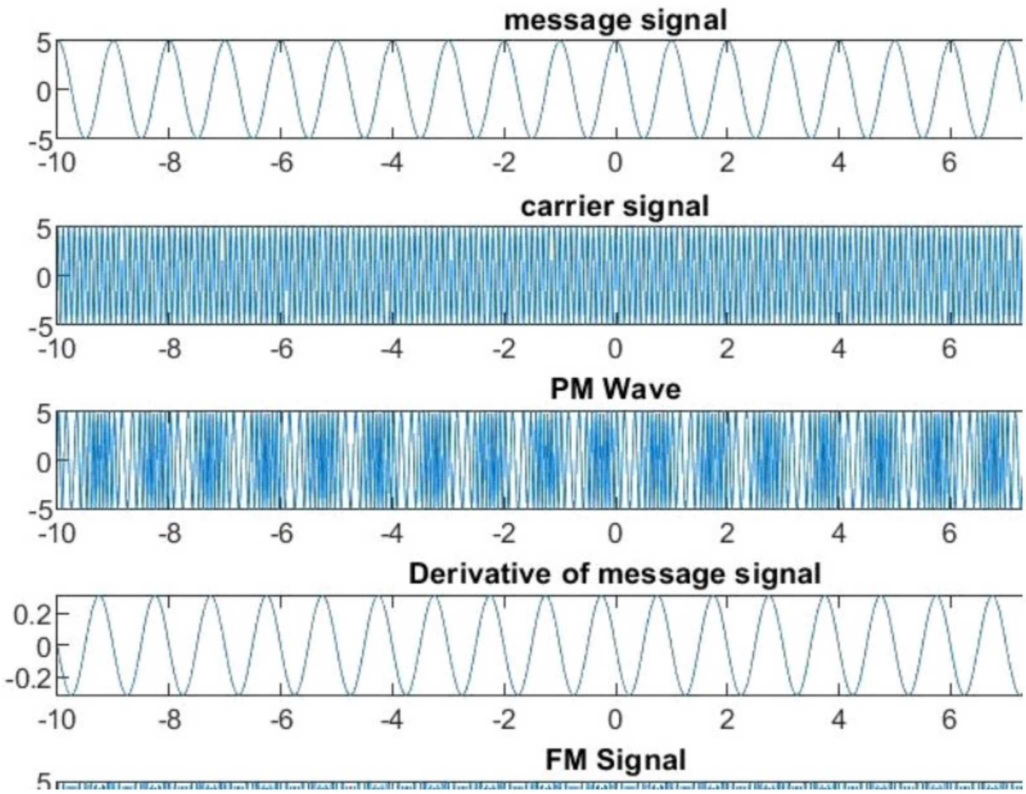
Code:

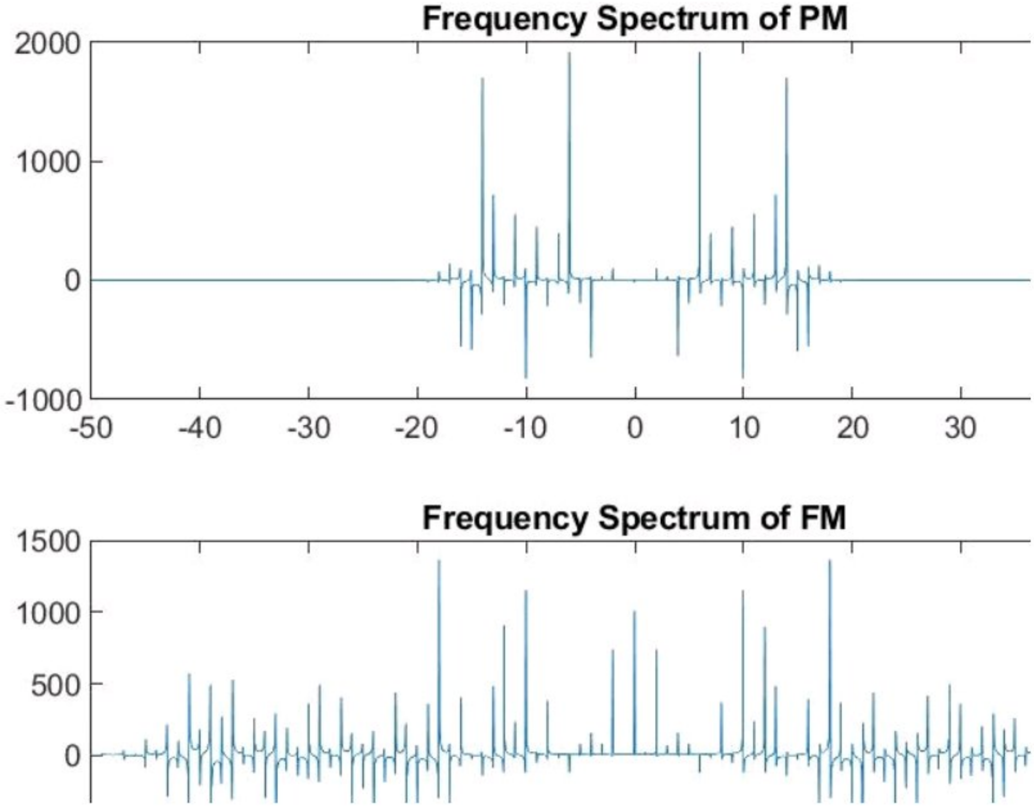
```
clc
clear all
fs=100;%declaring fs
t=-10:(1/fs):10;%declaring t
fm=1;%declaring fm
am=5;%declaring am
ac=5;%declaring ac
wm=2*pi*fm;%wm
fc=10;%declaring fc
wc=2*pi*fc;
m=am*cos(wm*t);%message signal
c=ac*cos(wc*t);%carrier signal
kp=1;%declaring kp
kf=2*pi*fm;%kf
```

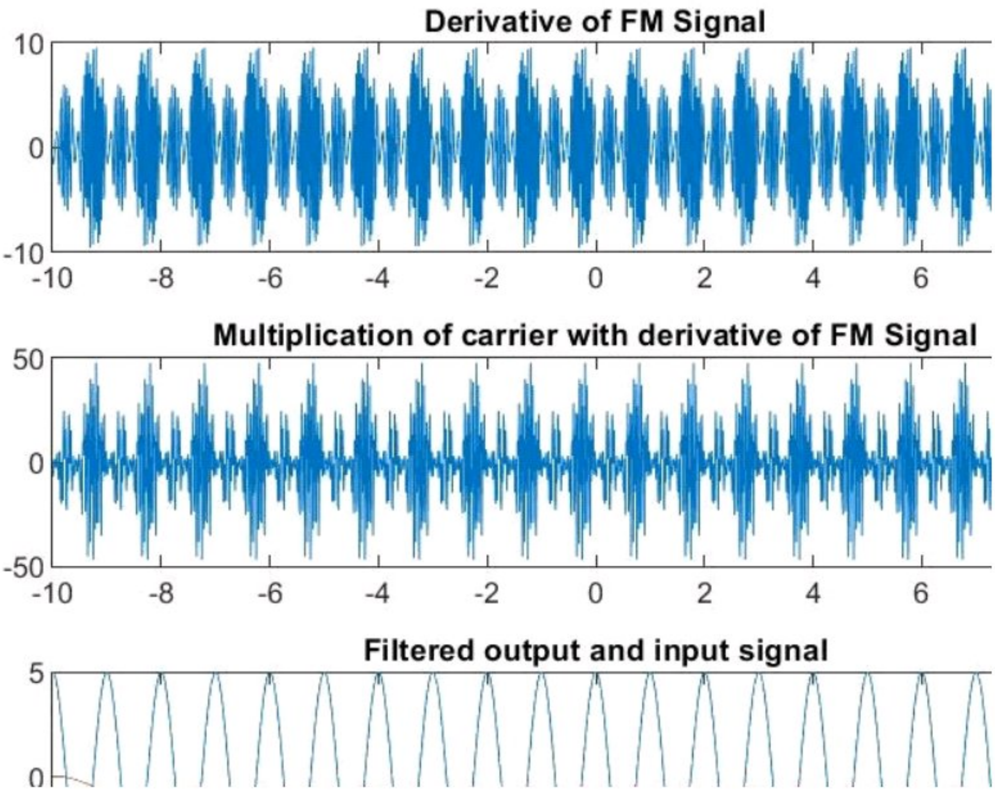
```
F = griddedInterpolant(t,m);
fun = @(l) F(l);

s1=ac*cos(wc*t+(kp*m));%PM
s2=ac*cos((wc*t)+(kf*fun(t)));%FM
s3=[diff(m) 0];%Derivative

%Plotting
subplot(5,1,1);
plot(t,m);
title('message signal');%title
subplot(5,1,2);
plot(t,c);
title('carrier signal');
subplot(5,1,3);
plot(t,s1);
title('PM Wave');
subplot(5,1,4);
plot(t,s3);
title('Derivative of message signal');
subplot(5,1,5);
plot(t,s2);
title('FM Signal');

n=length(t);
disp(n);
df=fs/n;
fp=(-fs/2):df:(fs/2)-df;
y1=fftshift(fft(s1));%In frequency domain
y2=fftshift(fft(s2));
figure;
subplot(2,1,1);%plotting
plot(fp,y1);
title('Frequency Spectrum of PM');
subplot(2,1,2);
plot(fp,y2);
title('Frequency Spectrum of FM');
%demodulation
dfm=[diff(s2) 0];

s4=dfm.*c;
[b,a] = butter(4,fm/fs,'low');
y=filter(b, a, s4);

figure;
subplot(3,1,1);%plotting
```

```
plot(t,dfm);
title('Derivative of FM Signal');
subplot(3,1,2);
plot(t,s4);
title('Multiplication of carrier with derivative of FM Signal');
subplot(3,1,3);
plot(t,m, t, y);
title('Filtered output and input signal');
```

<u>Output Waveforms:</u>




Frequency Spectrum of PM
2000
1000
0
-1000
-50
-40
-30
-20
-10
0
10
20
30
Frequency Spectrum of FM
1500
1000
500
0



Conclusion: In this experiment we have implemented frequency modulation and demodulation in *Matlab*. We also analysed the spectrum of modulated FM and PM signals.

To implement frequency modulation and demodulation using MATLAB.

Remarks: **Signature:**

Experiment no.: 11 **Date:**

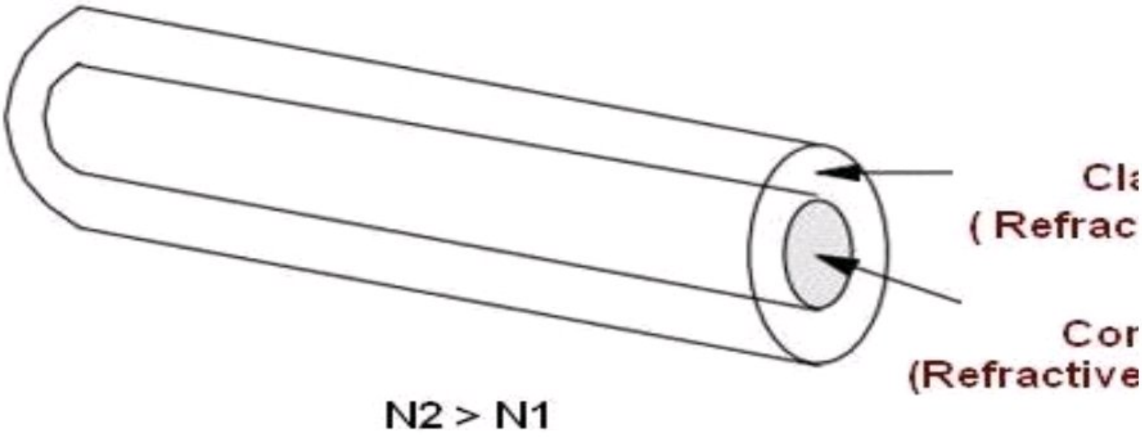
AIM: To find the Numerical Aperture of given optical fiber.

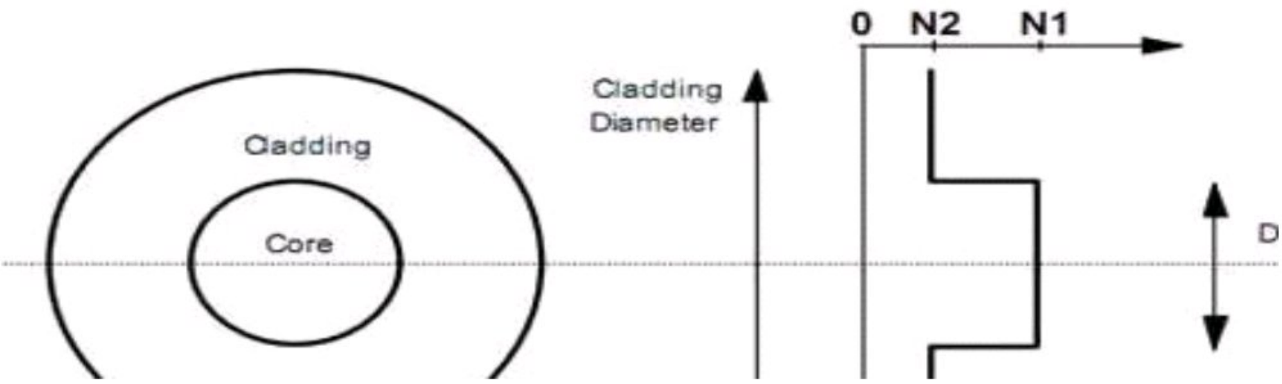
APPARATUS: Emitter, Fiber cable, Fiber stand, Detector

THEORY:

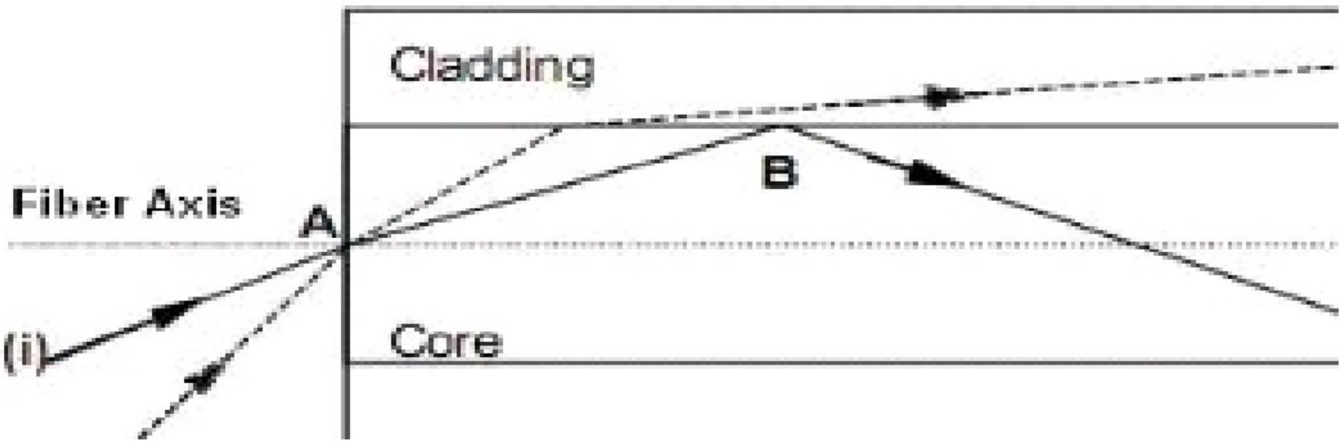
What is optic fiber?

Optical fibers are fine transparent glass or plastic fibers which can propagate light. They work under the principle of total internal reflection from diametrically opposite walls. In this way light can be taken anywhere because fibers have enough flexibility. This property makes them suitable for data communication, design of fine endoscopes, micro sized microscopes etc. An optic fiber consists of a core that is surrounded by a cladding which is normally made of silica glass or plastic. The core transmits an optical signal while the cladding guides the light within the core. Since light is guided through the fiber it is sometimes called an optical wave guide. The basic construction of an optic fiber is shown in figure (1).

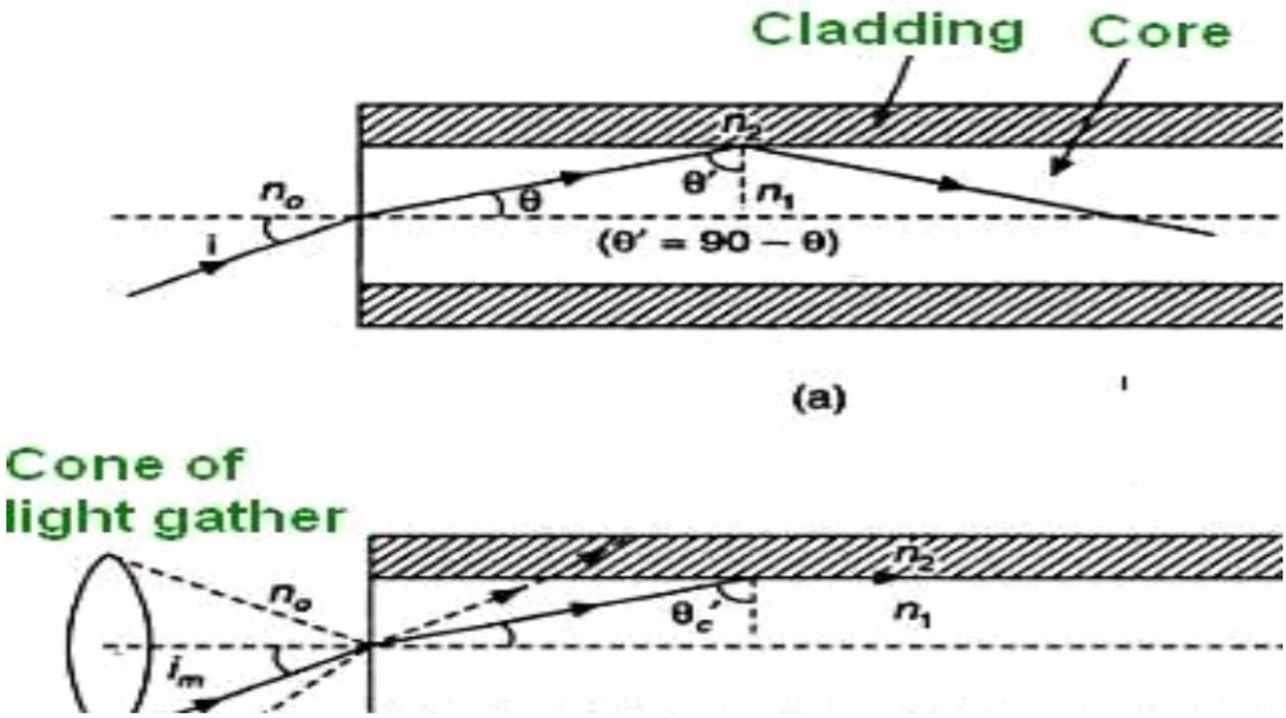




In order to understand the propagation of light through an optical fiber, consider the figure (2). Consider a light ray (i) entering the core at a point A, travelling through the core until it reaches the core cladding boundary at point B. As long as the light ray intersects the core-cladding boundary at small angles, the ray will be reflected back in to the core to travel on to point C where the process of reflection is repeated .i.e., total internal reflection takes place. Total internal reflection occurs only when the angle of incidence is greater than the critical angle. If a ray enters an optic fiber at a steep angle (ii), when this ray intersects the core-cladding boundary, the angle of intersection is too large. So, reflection back in to the core does not take place and the light ray is lost in the cladding. This means that to be guided through an optic fiber, a light ray must enter the core with an angle less than a particular angle called the acceptance angle of the fiber. A ray which enters the fiber with an angle greater than the acceptance angle will be lost in the cladding.



Consider an optical fibre having a core of refractive index n_1 and cladding of refractive index n_2. let the incident light makes an angle i with the core axis as shown in figure (3).



Then the light gets refracted at an angle θ and fall on the core-cladding interface at an angle where,

$$\theta' = (90 - \theta) \quad \text{---------------------- (1)}$$

By Snell's law at the point of entrance of light in to the optical fiber we get,

$$n_0 \sin i = n_1 \sin\theta \quad \text{-------------------- (2)}$$

Where n_0 is refractive index of medium outside the fiber. For air $n_0 = 1$.

When light travels from core to cladding it moves from denser to rarer medium and so it may be totally reflected back to the core medium if θ' exceeds the critical angle θ'_c. The critical angle is that angle of incidence in denser medium (n1) for which angle of refraction become 90°. Using Snell's laws at core cladding interface,

$$n_1 \sin\theta'_c = n_2 \sin 90$$

or

$$Sin\,\theta'_c =$$ ------------------------ (3)

Therefore, for light to be propagated within the core of optical fiber as guided wave, the angle of incidence at core-cladding interface should be greater than θ'_c. As i increases, θ increases and so θ' decreases. Therefore, there is maximum value of angle of incidence beyond which, it does not propagate rather it is refracted in to cladding medium (fig: 3(b)). This maximum value of i say i_m is called maximum angle of acceptance and $n_0 \sin i_m$ is termed as the numerical aperture (NA). From equation(2),

$$NA = n_0 \sin i_m = n_1 \sin\theta$$

$$= n_1 \sin(90 - \theta_c)$$

$$Or\ NA = n_1\, Cos\,\theta'_c$$

$$= n_1 \sqrt{1 - Sin^2\theta'_c}$$

$$\sin\theta'_c =$$

From equation (2)

$$NA = n1 \sqrt{1 -}$$

Therefore,

$$NA = \sqrt{n_1^2 -}$$

The significance of NA is that light entering in the cone of semi vertical angle i_m only propagate through the fibre. The higher the value of i_m or NA more is the light collected for propagation in the fibre. Numerical aperture is thus considered as a light gathering capacity of an optical fibre. Numerical Aperture is defined as the Sine of half of the angle of fibre's light acceptance cone. i.e. NA= Sin θ_a where θ_a, is called acceptance cone angle.
Let the spot size of the beam at a distance d (distance between the fiber end and detector) as the radius of the spot(r). Then,

$$\sin\theta = \frac{r}{\sqrt{r^2 + d^2}} \quad \text{----------------------}$$

PROCEDURE:

- Select all tools i.e. emitter, fiber, fiber stand, output screen by clicking on that.
- Now press start button.
- Vary the distance of screen (L) by scrolling the button. Diameter (D) will also vary. Note down that value.
- Repeat that process and get the value of L and D.

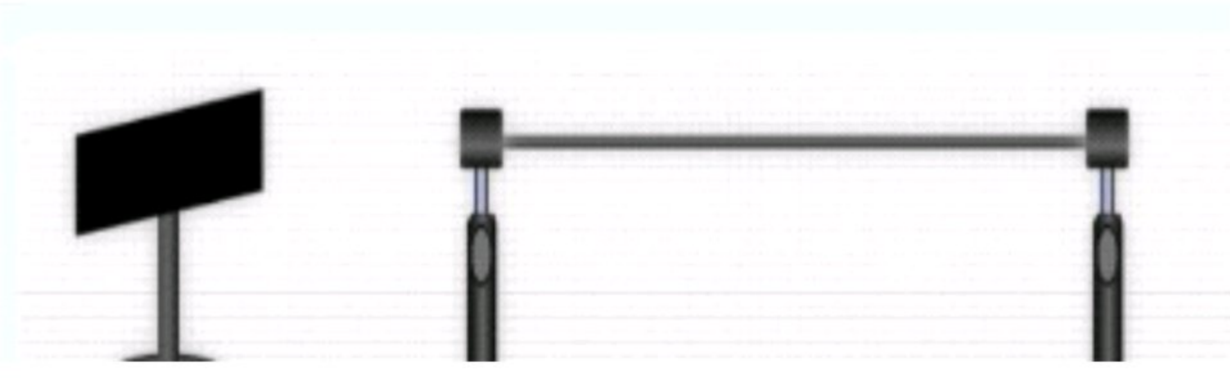


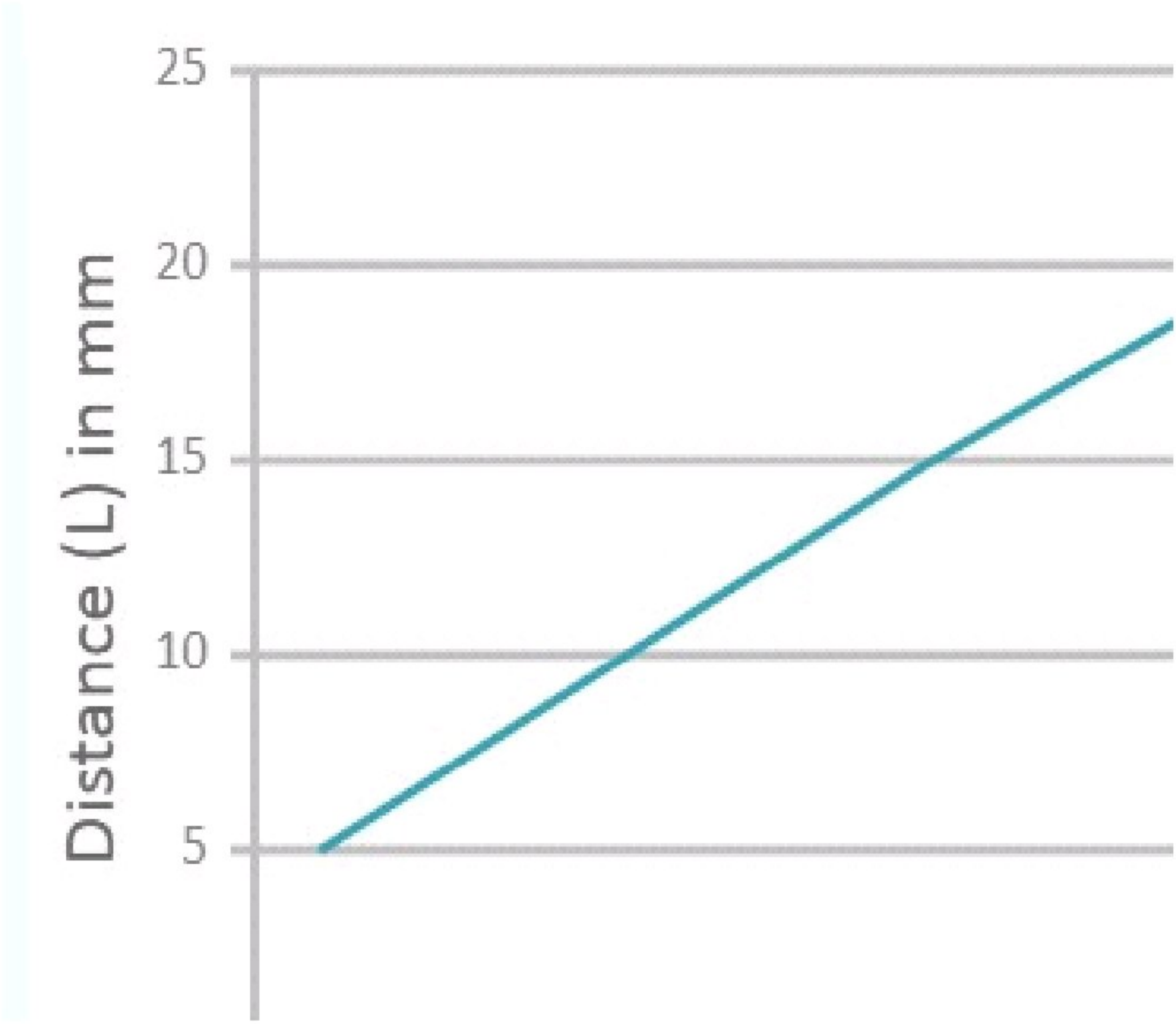
Fig.4 Tools arrangement

OBSERVAITION:

DATA TABLE

SR. NO	Distance of screen (L) in mm	Diameter(D) in mm
1	5	5.554
2	14	15.554
3	18	19.998
4	20	22.22

DRAW GRAPH:



CONCLUSION:

In this experiment, we found the numerical aperture of the given optical fiber by using virtual lab.

Remarks: **Signature:**