SARDAR VALLABHBHAI NATIONAL INSTITUTE OF TECHNOLOGY

ICHCHANATH, PIPLOD, SURAT-395007

LABORATORY MANUAL

ELECTRONICS ENGINEERING DEPARTMENT

EC: 206: PRINCIPLE OF COMMUNICATION

B.TECH II-SEMESTER IV

NAME: Yatin Aditya Tekumalla

ROLL NO: U18EC013

BATCH: 2019-2020

SARDAR VALLABHBHAI NATIONAL INSTITUTE OF TECHNOLOGY

ICHCHANATH, PIPLOD, SURAT-395007

ELECRONICS ENGINEERING DEPARTMENT

CERTIFICATE

This is to certify that Ms./Mr._________YATIN ADITYA TEKUMALLA____________________

Roll no. U18EC013_________________has satisfactorily completed the lab exercises for

EC: 206: PRINCIPLE OF COMMUNICATION of B.TECH II- SEMESTER IV in the academic

Year 2019-2020

SUBJECT CO-ORDINATOR

DATE:

INDEX

SR.NO.	DATE	PRACTICAL NAME	PAGE NO.	SIGN
1	13/01/20	To study about introduction to MATLAB and plot different types of signals using MATLAB.	1	
2	20/01/20	To observe the spectrum of sinusoidal, square and triangular signals by using Fast Fourier Transform (FFT).	10	
3	27/01/20	Study of Sampling theorem and Reconstruction of signal. Verify Nyquist criteria.	15	
4	03/02/20	To compute the Exponential Fourier Series coefficients for square wave, saw tooth wave and exponential wave. Plot its magnitude and phase spectra.	20	
5	10/02/20	To study amplitude modulation and observe the waveforms for three different Modulation indices and reception of AM signal.	26	
6	17/02/20	Write a MATLAB code for Amplitude Modulation and Demodulation considering sinusoidal signal as an input. Plot the various signal obtained at each step of the process in the time domain and frequency domain.	32	
7	24/02/20	To study about Frequency Division Multiplexing/ Demultiplexing with sinusoidal wave. Also observe the Fourier Transform of all the signals at each stage.	37	
8	16/03/20	Write a MATLAB code for Frequency Modulation and Demodulation considering sinusoidal signal as an input. Also plot the FM signal in Frequency Domain.	43	
9	23/03/20	Write a MATLAB code (A) To plot PDF and CDF for Gaussian Distribution (B) To plot PDF and CDF for Rayleigh Distribution	45	
10	20/04/20	Design a Superhetrodyne Receiver using MATLAB simulator for Frequency Modulation. Plot the various signal obtained at each step of the process in the time domain and frequency domain.	48	
11	27/04/20	Design a Bandpass filter to allow DSB Signal to pass from the givenRing Modulator.	60	

EXPERIMENT:1

AIM : To study about Introduction to MATLAB and plot different types of signals using MATLAB.

APPARATUS: MATLAB software

THEORY:

(A) INTRODUCTION TO MATLAB

❖ **What is MATLAB?**

The name MATLAB stands for "Matrix Laboratory" and was originally designed as a tool for doing numerical computations with matrices and vectors. It has since grown into a high-performance language for technical computing. MATLAB, integrating computation, visualization, and programming in an easy to-use environment, allows easy matrix manipulation, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs in other languages. Typical uses include:

- Math and Computation
- Modeling and Simulation
- Data Analysis and Visualization
- Application Development
- Graphical User Interface development

❖ **Getting Started**

Window Layout The first time you start MATLAB, the desktop appears with the default layout The following tools are managed by the MATLAB desktop:

- Command Window: Run MATLAB statements.
- Current Directory Browser: To search for, view, open, find, and make changes to MATLAB related directories and files use the MATLAB Current Directory browser.
- Command History: Displays a log of the functions you entered in the Command Window, copy them, execute them, and more.
- Workspace Browser: Shows the name of each variable, its value, and the Min and Max calculations, which MATLAB computes using the min and max functions.

❖ MATLAB Help

MATLAB has an extensive help system built into it, containing detailed documentation and help information on all of the commands and functions of MATLAB. There are two different ways to obtain help for MATLAB:

Command Line

• HELP: HELP FUN displays a description of and syntax for the function FUN. (e.g. help plot)

• DOC: DOC FUN displays the HTML documentation for the MATLAB function FUN. (e.g. doc help)

Help Browser Another source of help is the MATLAB help browser. You can invoke the MATLAB help browser by typing, help browser, at the MATLAB command prompt, clicking on the help button, or by selecting Start → MATLAB → Help from the MATLAB desktop.

❖ Variables

The simplest way to use MATLAB is for arithmetic operations. The basic arithmetic operators are $+$, $-$,/, $*$ and $\wedge$ (power). These operators can be used in conjunction with brackets (). As with all programming languages special care should be given on how a mathematic expression is written. For example, the result of the expression $5 + 10/2 * 3$ is 20 and corresponds in the expression $5 + (10/2) * 3$ and not in the expression $5 + 10/(2 * 3)$. Generally, Matlab works according to the priorities:

(1) quantities in brackets

(2) power

(3) $*$,/ working left to right

(4) $+$, $-$ working left to right

❖ M – file functions

You add new functions to the MATLAB vocabulary by expressing them in terms of existing functions. The existing commands and functions that compose the new function reside in an M-file. A line at the top of a function M-file contains the syntax definition. The name of a function, as defined in the first line of the M-file, should be the same as the name of the file without the .m extension. The main steps to follow when defining a Matlab function are:

(1) Decide on a name for the function, making sure that it does not conflict with a name that is already used by Matlab.

(2) The first line of the file must have the format:

function [list of outputs] = Function Name(list of inputs)

(3) Include the code that defines the function.

- **Loops**

There are occasions that we want to repeat a segment of code a number of different times. In such cases it is convenient to use loop structures. In MATLAB there are three loop structures:

• For Loop

The general syntax is:

```
for variable = expression
        statement
          ...
        Statement
end
```

The columns of the expression are stored one at a time in the variable while the following statements, up to the end, are executed. In practice, the expression is almost always of the form scalar, in which case its columns are simply scalars. The scope of the for statement is always terminated with a matching end.

• While Loop

The general format is

```
while expression
     statement
       ...
     statement
end
```

While repeats statements an indefinite number of times. The statements are executed while the real part of expression has all nonzero elements. Expression is usually of the form expression relation-operator expression.

• If...then...else

MATLAB evaluates the expression and, if the evaluation yields logical 1 (true) or a nonzero result, executes one or more MATLAB commands denoted here as statements. When you are nesting ifs, each if must be paired with a matching end. When using else if and/or else within an if statement, the general form of the statement is

```
if expression1
    statements1
elseif expression2
    statements2
else
    statements3
end
```

(B)TO PLOT DIFFERENT TYPES OF SIGNALS BY USING MATLAB

MATLAB CODE:

(i) Periodic Signals – Sin, Cos, Tan, Square, Sawtooth and Triangle waves

```
clc
clearall
closeall
t = -10:0.1:10
x = sin(t)
subplot(3,2,1)
plot(t,x)
xlabel('time')
ylabel('amplitude')
title('sine wave')
y = cos(t)
subplot(3,2,2)
plot(t,y)
xlabel('time')
ylabel('amplitude')
title('cos wave')
z = tan(t)
```

```
subplot(3,2,3)
plot(t,z)
xlabel('time')
ylabel('amplitude')
title('tan wave')
a = square(t)
subplot(3,2,4)
plot(t,a)
xlabel('time')
ylabel('amplitude')
title('square wave')
b = sawtooth(t)
subplot(3,2,5)
plot(t,b)
xlabel('time')
ylabel('amplitude')
title('sawtooth wave')
c = sawtooth(t,0.5)
subplot(3,2,6)
plot(t,c)
xlabel('time')
ylabel('amplitude')
title('triangle wave')
```

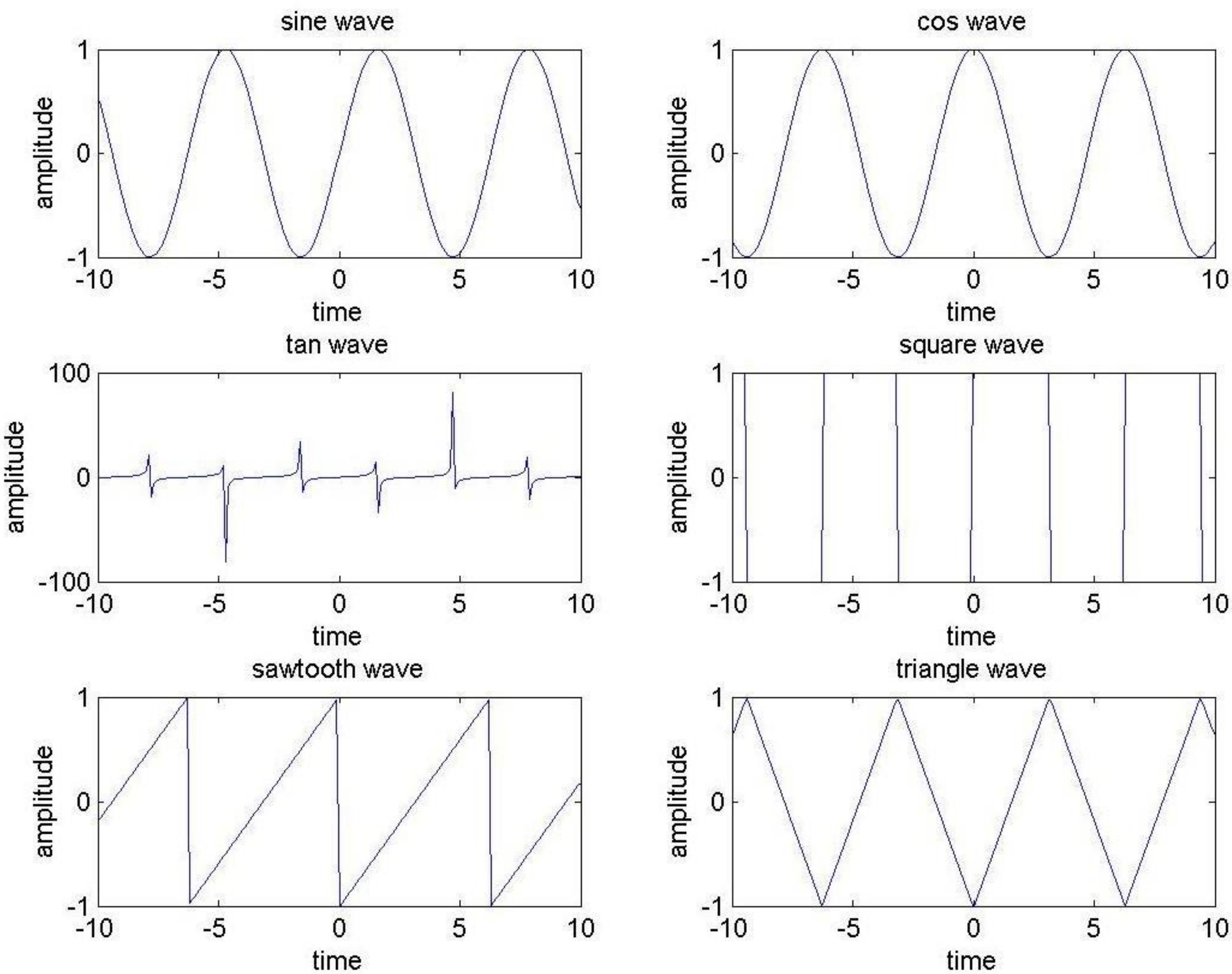
(ii) Aperiodic Signals – Unit Impulse, Unit Step, Unit Ramp, Exponential, Aperiodic triangle and Aperiodic Square

```
clc
clearall
closeall
t = -10:0.1:10
d = zeros(1,length(t))
d(t==0)=1
subplot(3,2,1)
plot(t,d)
xlabel('time')
ylabel('amplitude')
title('unit impulse')
e = zeros(1,length(t))
e(t>=0)= 1
subplot(3,2,2)
plot(t,e)
xlabel('time')
ylabel('amplitude')
title('unit step')
f = zeros(1,length(t))
f(t>=0)=t(t>=0)
subplot(3,2,3)
plot(t,f)
xlabel('time')
ylabel('amplitude')
title('unit ramp')
a=exp(t)
subplot(3,2,4)
plot(t,a)
```

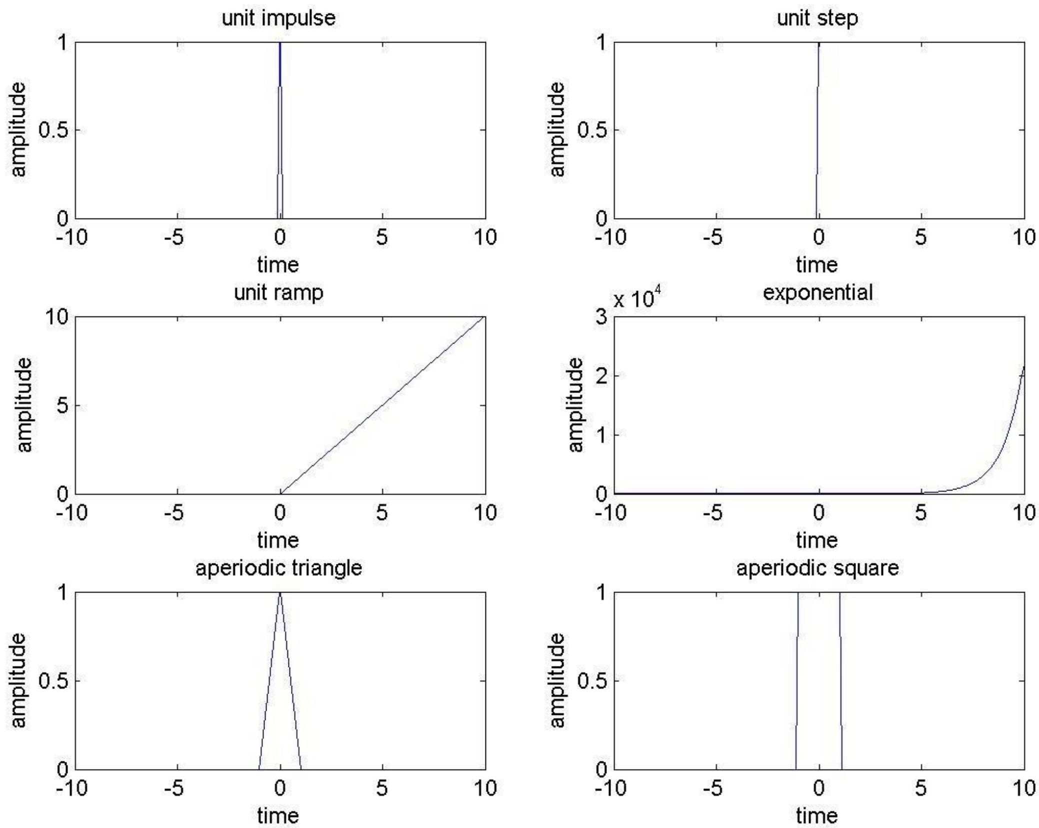
```
xlabel('time')
ylabel('amplitude')
title('exponential')
b = zeros(1,length(t))
b(t<=0&t>=-1)=1+t(t<=0&t>=-1)
b(t>=0&t<=1)=1-t(t>=0&t<=1)
subplot(3,2,5)
plot(t,b)
xlabel('time')
ylabel('amplitude')
title('aperiodic triangle')
c = zeros(1,length(t))
c(t<=1&t>=-1)=1
subplot(3,2,6)
plot(t,c)
xlabel('time')
ylabel('amplitude')
title('aperiodic square')
```

OBSERVATIONS:

(i) Perodic Signals



(ii) Aperiodic Signals



CONCLUSION:

We studied and learnt, how to write a code for various Periodic and Aperiodic Signals and plot them using MATLAB.

SIGNATURE

EXPERIMENT:2

AIM: To observe the spectrum of sinusoidal, square and triangular signals by using Fast Fourier Transform (FFT)

APPARATUS: Digital Storage Oscilloscope(DSO), Function Generator ,Connecting Wires

THEORY:

Digital Storage Oscilloscope (DSO) with FFT function and spectrum analyzer as being the same thing. They will get a signal from the Time Domain and convert it into the Frequency Domain and we can check all the harmonics and frequency component of the signal.

The frequency spectrum of an electrical signal is the distribution of the amplitude and phase of each frequency component against frequency.

There are varieties of uses that can benefit from viewing the frequency spectrum of a signal. Using the FFT math function on a time domain signal provides the user with frequency domain information and can provide the user a different view of the signal quality.

Advantages of FFT spectrum analyzer technology

- **Fast capture of waveform**: In view of the fact that the waveform is analyzed digitally. The waveform can be captured in a relatively short time and then the subsequently analyzed. This short capture time can have many advantages. It can allow for the capture of transient or short lived waveforms.
- **Able to capture non-repetitive events**: The short capture time means that the FFT analyzer can capture non repetitive waveforms, giving them a capability not possible with other spectrum analyzers.
- **Able to analyze signal phase**: As part of the signal capture process, data is gained which can be processed to reveal the phase of the signals.
- **Waveforms can be stored**: Using the FFT technology, it is possible to capture the waveform and analyze it later should this be required.

PROCEDURE:

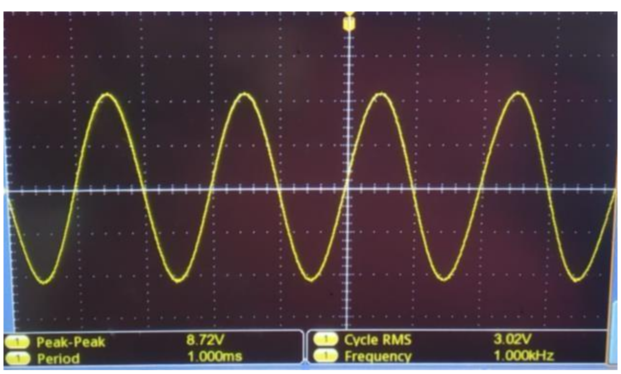
In modern digital oscilloscopes we can see the FFT mode button

- Give the input Sine wave for which spectrum is to be observed by frequency generator
- Then press FFT mode button available on DSO.
- Then follow the steps to observe frequency of spectra
    - ✓ Press 'Measure' button.
    - ✓ Select 'Frequency mode' on display
    - ✓ Set the cursor on tip of first maxima in spectra
    - ✓ Desired frequency is displayed on screen
- Then follow the steps to observe magnitude in spectra
    - ✓ Press 'Measure' button
    - ✓ Select 'Magnitude mode' on display
    - ✓ Set the cursor on tip of first maxima in spectra
    - ✓ Desired frequency is displayed on screen

Now repeat the experiment with triangle and square wave. We can also zoom the spectra by selecting FFT zoom available on display.

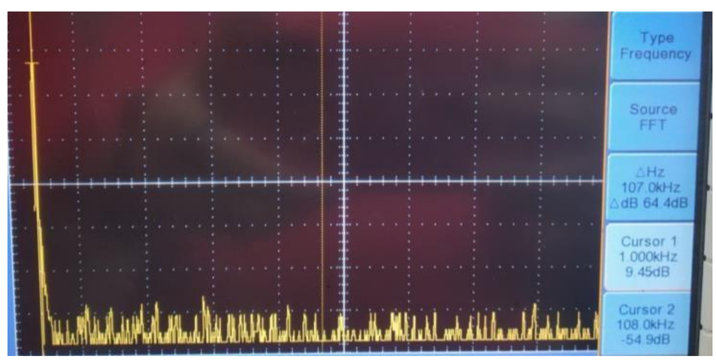
OBSERVATION:

Signal

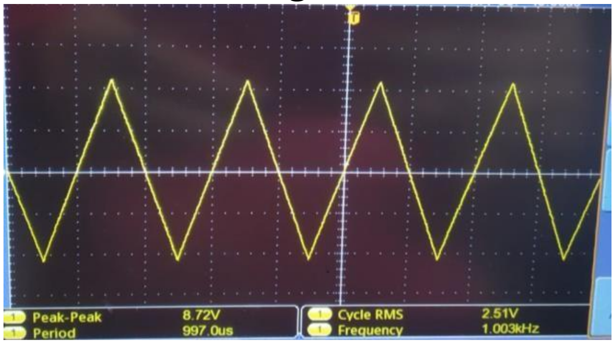
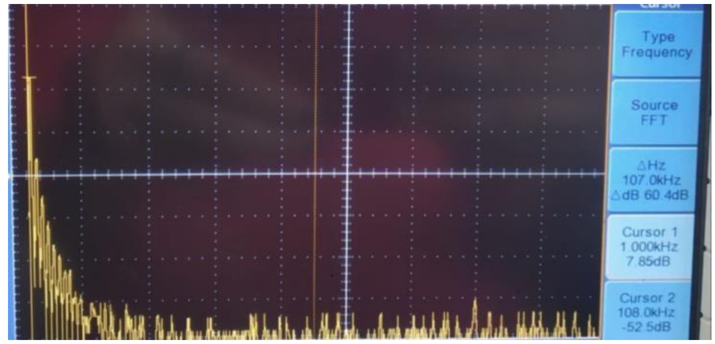
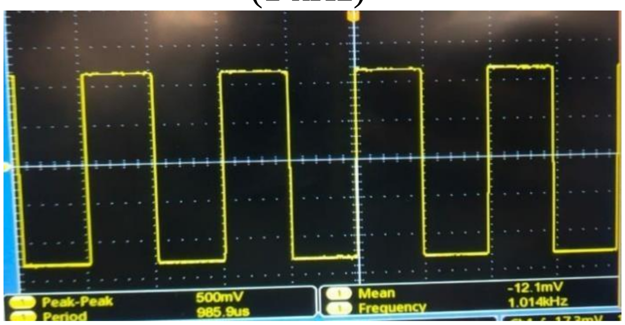
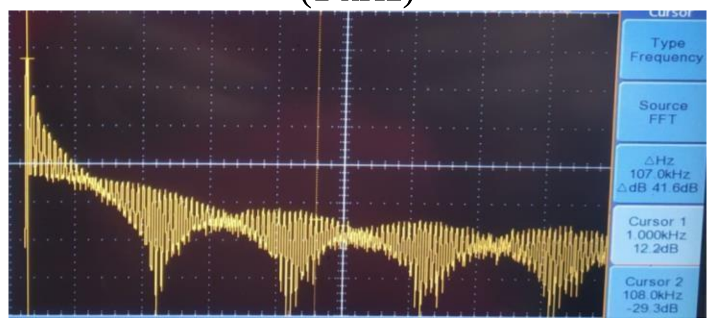
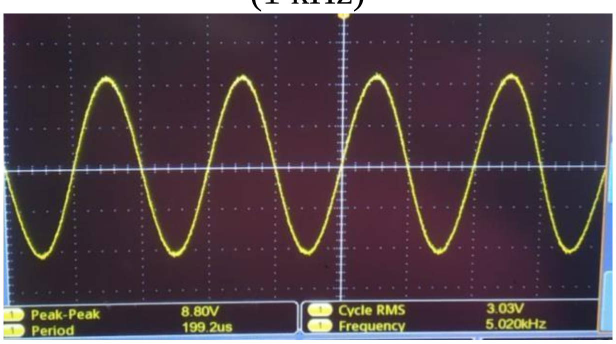
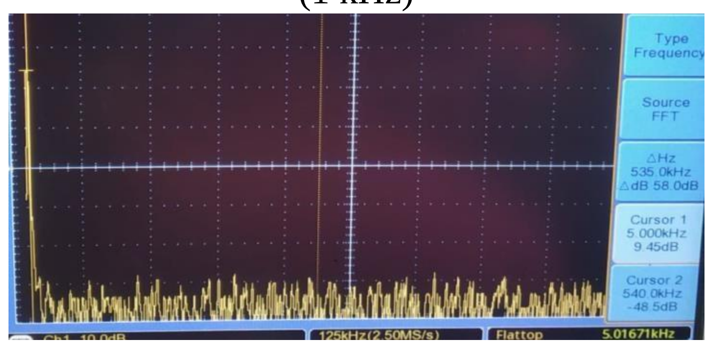
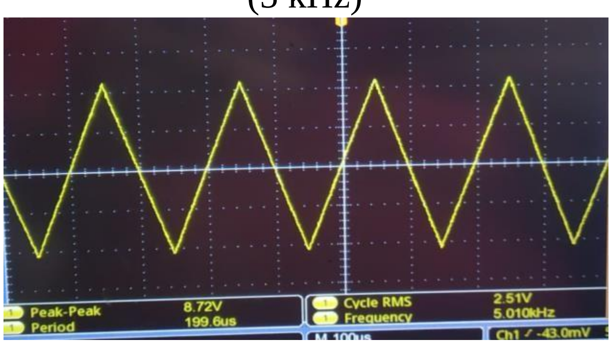
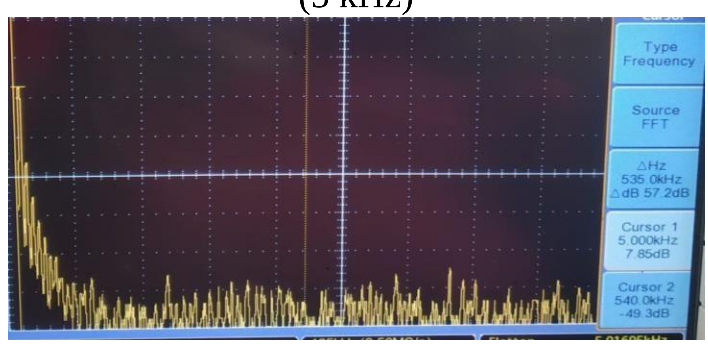


Sine Wave
(1 kHz)

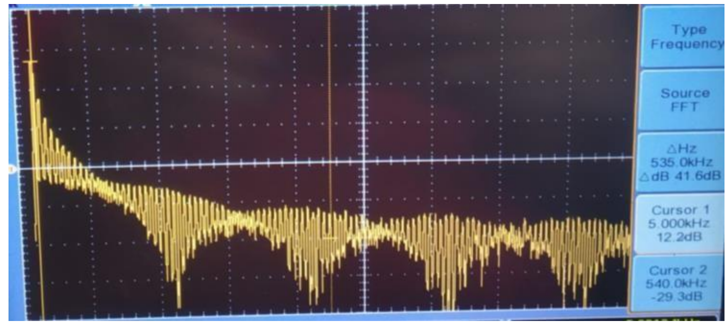
FFT



Sine Wave FFT
(1 kHz)

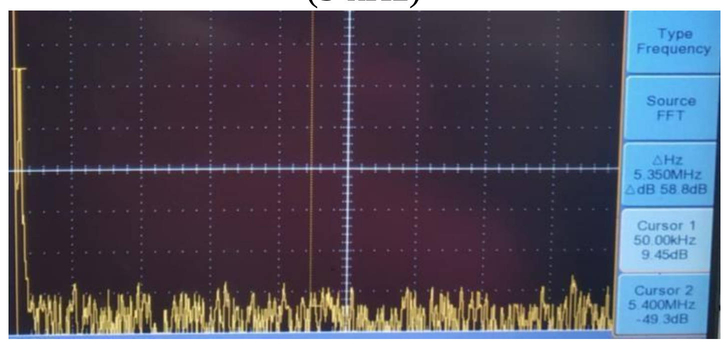
Signal	FFT
 Triangle Wave (1 kHz)	 Triangle Wave FFT (1 kHz)
 Square Wave (1 kHz)	 Square Wave FFT (1 kHz)
 Sine Wave (5 kHz)	 Sine Wave FFT (5 kHz)
 Triangle Wave (5 kHz)	 Triangle Wave FFT (5kHz)

Square Wave
(5 kHz)

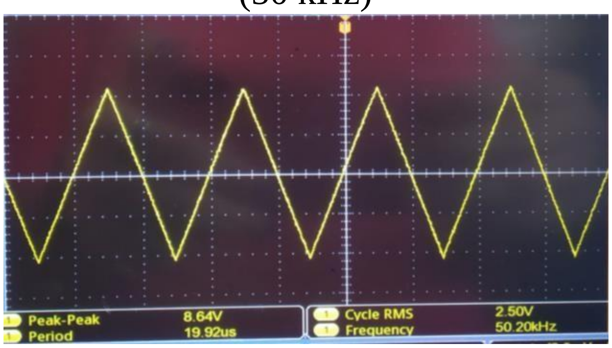


Square Wave FFT
(5 kHz)

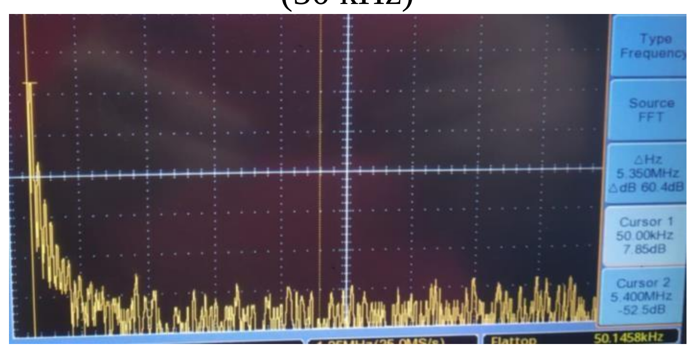
Sine Wave
(50 kHz)



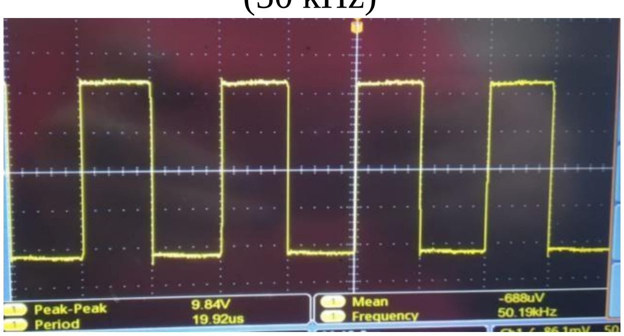
Sine Wave FFT
(50 kHz)



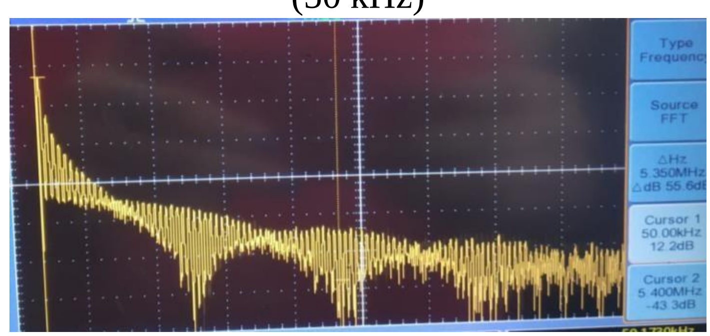
Triangle Wave
(50 kHz)



Triangle Wave FFT
(50 kHz)



Square Wave
(50 kHz)



Square Wave FFT
(50 kHz)

CONCLUSION:

From this experiment, we learnt, how the Fast Fourier Transforms(FFTs) of various signals look like at various frequencies.

SIGNATURE

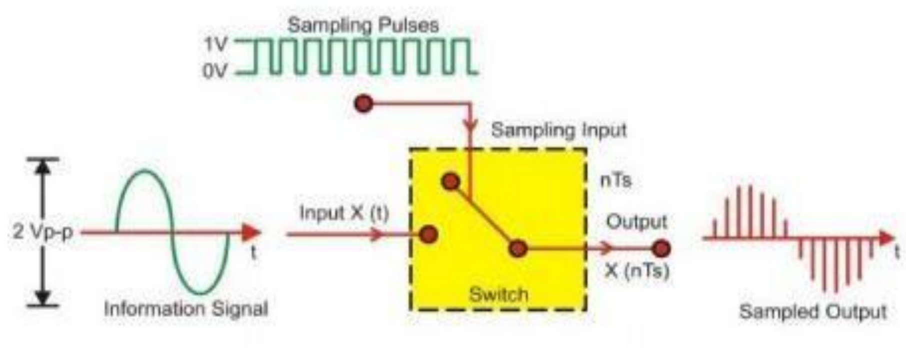
EXPERIMENT:3

AIM: Study of Sampling theorem and Reconstruction of signal. Verify Nyquist criteria

APPARATUS: Model ST 2151 trainer kit, connection wires, DSO, Power supply.

THEORY:

The signals we use in the real world, such as our voice, are called "analog" signals. To process these signals for digital communication, we need to convert analog signals to "digital" form. While an analog signal is continuous in both time and amplitude, a digital signal is discrete in both time and amplitude. To convert continuous time signal to discrete time signal, a process is used called as sampling. The value of the signal is measured at certain intervals in time. Each measurement is referred to as a sample.



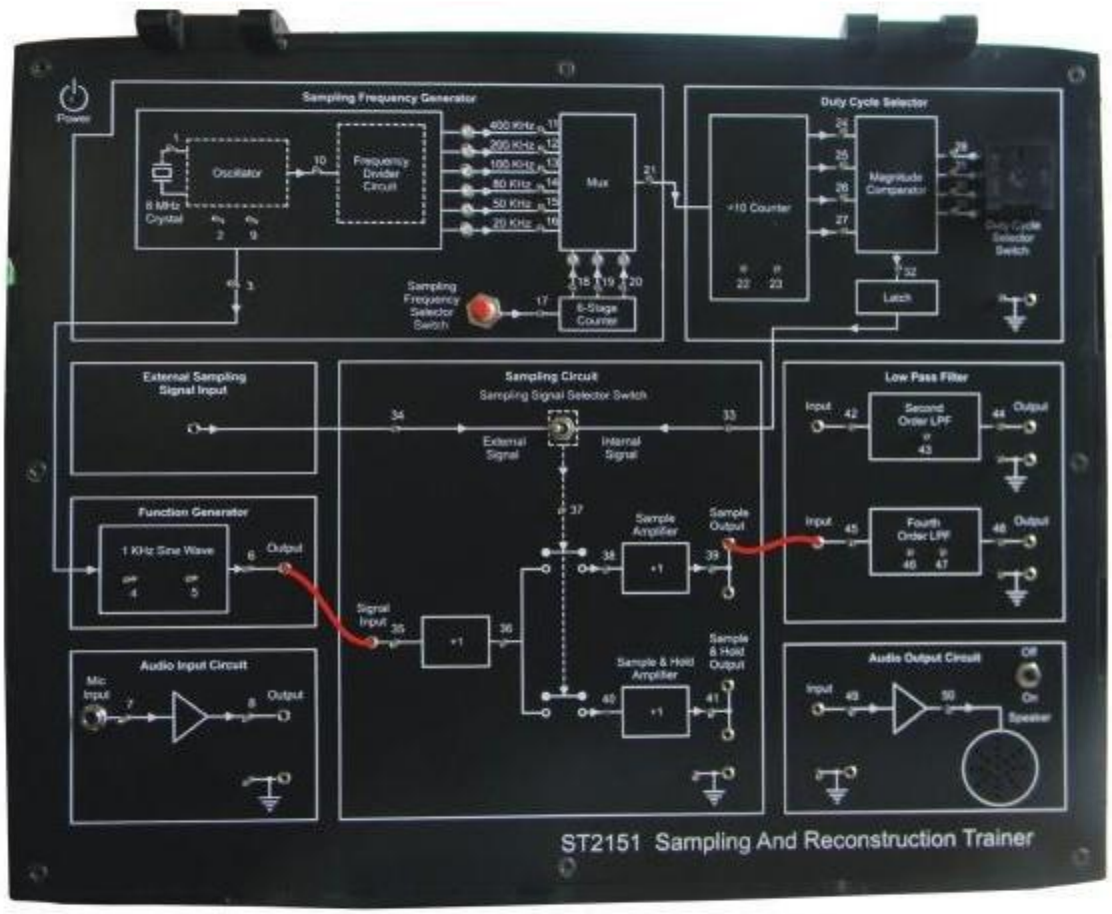
In electronics, a sample and hold circuit is used to interface real world, changing analogue signals to a subsequent system such as an analog-to-digital converter. The purpose of this circuit is to hold the analogue value steady for a short time while the converter or other following system performs some operation that takes a little time. In most circuits, a capacitor is used to store the analogue voltage and an electronic switch or gate is used to alternately connect and disconnect the capacitor from the analogue input. The rate at which this switch is operated is the sampling rate of the system.

NYQUIST CRITERION (SAMPLING THEOREM):

The Nyquist Criterion states that a continuous signal band limited to f_m Hz can be completely represented by and reconstructed from the samples taken at a rate greater than or equal to 2fm samples/second. The minimum sampling frequency is calls as NYQUIST RATE i.e. for faithful reproduction.

SAMPLE AND HOLD:

One way to maintain reasonable pulse energy is to hold the sample value until the next sample is taken. This technique is formed as Sample and Hold technique. A buffered Sample and Hold circuit consists of unity gain buffers preceding and succeeding the charging capacitor. The high input impedance of the proceeding buffer prevents the loading of the message source and also ensures that the capacitor charges by a constant rate irrespective of the source impedance.

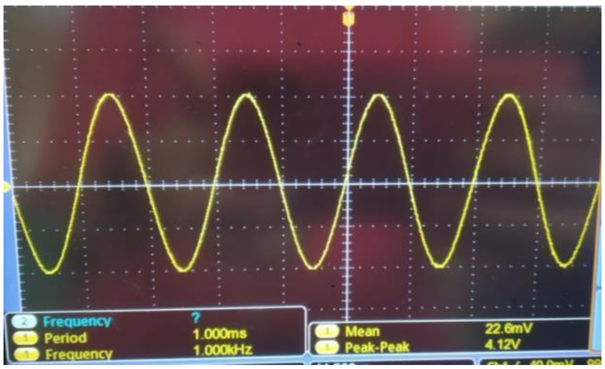


PROCEDURE:

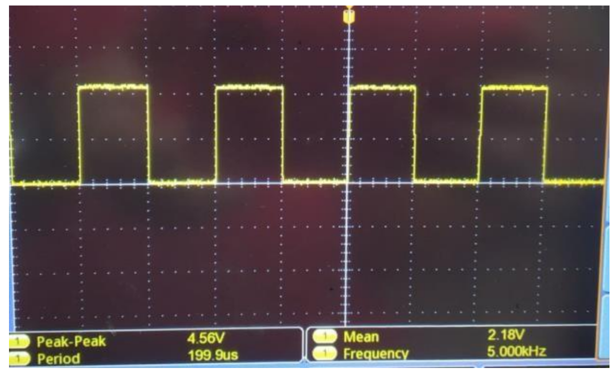
1. Connect the power cord to the trainer. Keep the power switch in ‘Off’ position.
2. Connect 1 KHz Sine wave to signal Input.
3. Connect BNC connector to the DSO and to the trainer’s output port.
4. Connect Sample Output to fourth order low pass filter Input and Sample and hold Output to second order low pass filter Input. Observe the output wave form.
5. Switch ‘On’ the trainer's power supply & Oscilloscope.

6. By pressing Sampling Frequency Selector Switch, change the sampling frequency from 2 KHz, 5 KHz, 10 KHz, 20 KHz up to 40 KHz.
7. Observe how Sample output and Sample and Hold Output changes in each case.
8. Also observe output of second order low pass filter and fourth order low pass filter

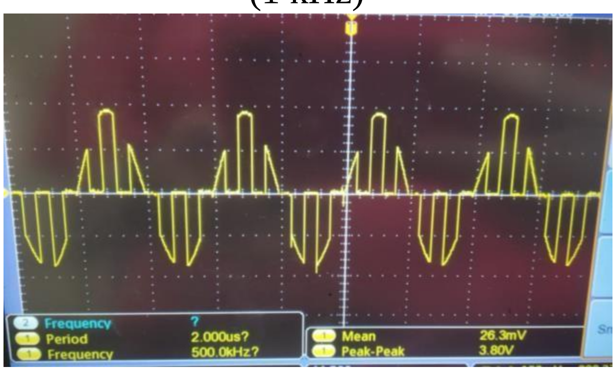
OBSERVATION:



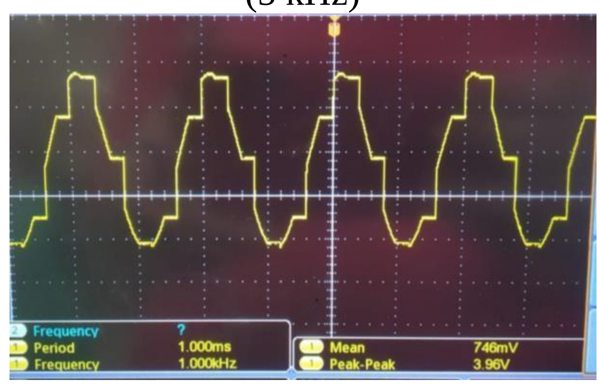
Input Message Signal
(1 kHz)



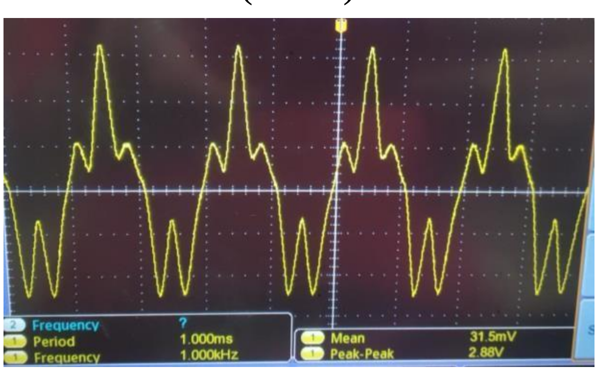
Carrier Wave
(5 kHz)



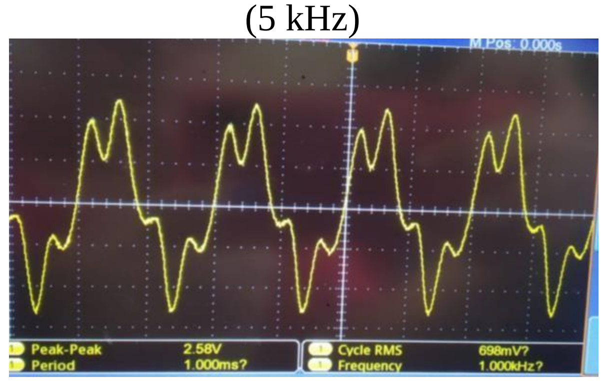
Sampled Signal
(5 kHz)



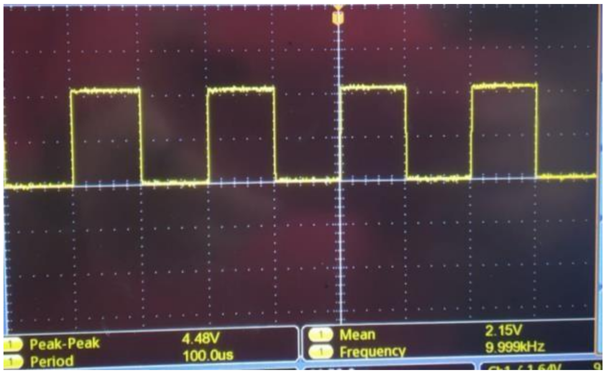
Sample and Hold Signal
(5 kHz)



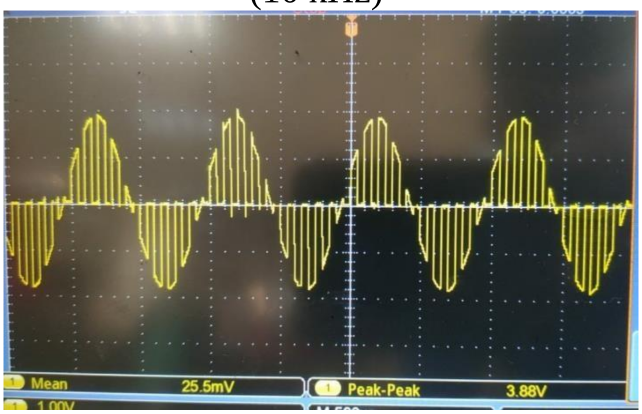
2nd Order Reconstructed Signal
(5 kHz)



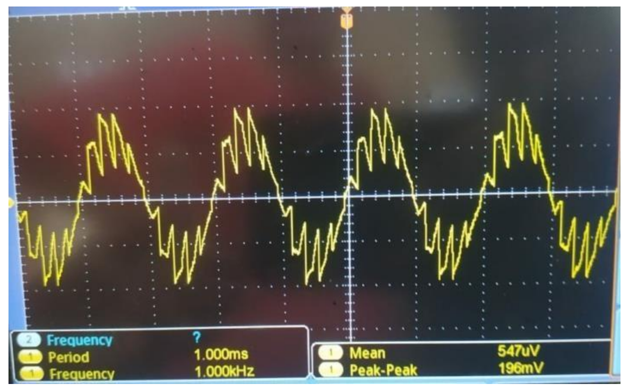
4th Order Reconstructed Signal
(5 kHz)



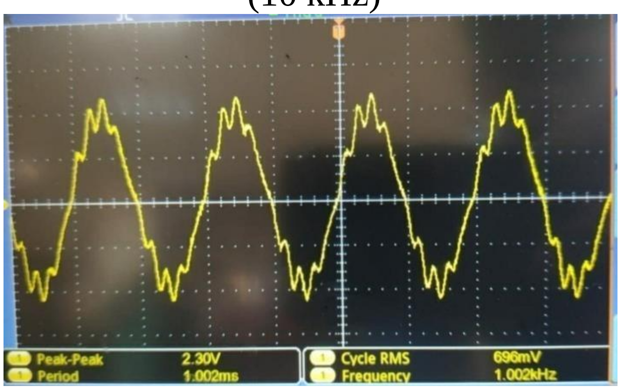
Carrier Wave
(10 kHz)



Sampled Signal
(10 kHz)



Sample and Hold Signal
(10 kHz)



2nd Order Reconstructed Signal
(10 kHz)

4th Order Reconstructed Signal
(10 kHz)

CONCLUSION:

From this experiment, we observed, as the frequency of carrier wave increases the distortion in the 2nd and 4th order reconstructed signals decreases. Also, the distortion in 2nd order reconstructed signal is more than the distortion in 4th order reconstructed signal.

SIGNATURE

EXPERIMENT:4

AIM: **To compute the Exponential Fourier Series coefficients for square wave, saw tooth wave and exponential wave. Plot its magnitude and phase spectra.**

APPARATUS: MATLAB Software

THEORY:

Fourier series is a way to represent the periodic signals in the frequency domain. It must be noted that the frequency plot achieved this way will be discrete because of which periodic signals are supposed to have a discrete frequency plot. Fourier series exist only for those signals which satisfy the Dirichlet's conditions stated as follow:

- The signal should be absolutely integrable on the finite time interval.
- The signal should have finite number of discontinuities in the interval
- The signal should also have minimum number of maxima and minima in the given interval.

Mathematically the Exponential Fourier series can be represented as

$$g(t) = \sum_{n=\infty}^{\infty} Dn\, e^{jn\omega_\circ t}$$

$$D_n = \frac{1}{T0} \int_{T0} g(t)\, e^{-jn\omega_\circ t}\, dt \quad , \omega_0 = \frac{2\pi}{T0}$$

In that g(t) is the signal and D_n is Fourier series coefficient.

MATLAB CODE:

EXPONENTIAL FUNCTION

```
clc
clear all
close all
T=pi
n=11          n=100
w=[-n:n]*2*pi/T
D=zeros(1,length(w))
fori=1:length(w)
   fun=@(t) exp(-t/2).*exp(-1j*w(i)*t)
   D(i)=(1/T)*integral(fun,0,pi)
end
subplot(3,1,1)
stem(w,abs(D))
xlabel('frequency')
ylabel('Absolute of D(n)')
subplot(3,1,2)
stem(w,angle(D))
```

```
xlabel('frequency')
ylabel('Angle of D(n)')

t=-4*pi:0.01:4*pi
g=zeros(1,length(t))
fori=1:length(t)
for j=1:length(w)
        g(i)=g(i)+(D(j).*exp(1j*w(j).*t(i)));
end
end
subplot(3,1,3)
plot(t,g)
xlabel('time')
ylabel('function')
```

SINE WAVE

```
clc
clear all
close all
T=2*pi
n=10            n=100
w=[-n:n]*2*pi/T
D=zeros(1,length(w))
fori=1:length(w)
    fun=@(t) sin(t).*exp(-1j*w(i)*t);
    D(i)=(1/T)*integral(fun,0,2*pi)
end
subplot(3,1,1)
stem(w,abs(D))
xlabel('frequency')
ylabel('Absolute of D(n)')
subplot(3,1,2)
stem(w,angle(D))
xlabel('frequency')
ylabel('Angle of D(n)')

t=-4*pi:0.001:4*pi
g=zeros(1,length(t))
fori=1:length(t)
for j=1:length(w)
        g(i)=g(i)+(D(j).*exp(1j*w(j).*t(i)));
end
end
subplot(3,1,3)
plot(t,g)
xlabel('time')
ylabel('function')
```

SQUAREWAVE

```
clc
clear all
close all
```

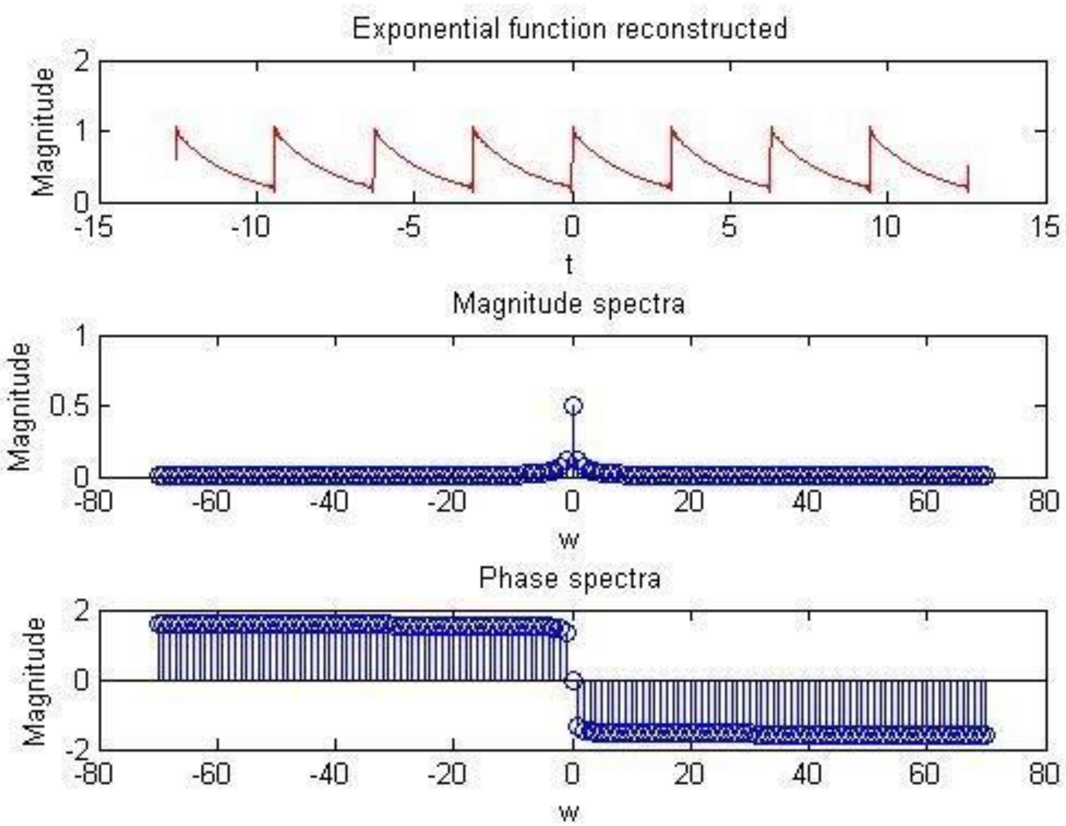
```
T=2*pi
n=100              n=200
w=[-n:n]*2*pi/T
D=zeros(1,length(w))
fori=1:length(w)
    fun=@(t) square(t).*exp(-1j*w(i)*t);
    D(i)=(1/T)*integral(fun,0,2*pi)
end
subplot(3,1,1)
stem(w,abs(D))
xlabel('frequency')
ylabel('Absolute of D(n)')
subplot(3,1,2)
stem(w,angle(D))
xlabel('frequency')
ylabel('Angle of D(n)')

t=-4*pi:0.001:4*pi
g=zeros(1,length(t))
fori=1:length(t)
for j=1:length(w)
        g(i)=g(i)+(D(j).*exp(1j*w(j).*t(i)));
end
end
subplot(3,1,3)
plot(t,g)
xlabel('time')
ylabel('function')
```

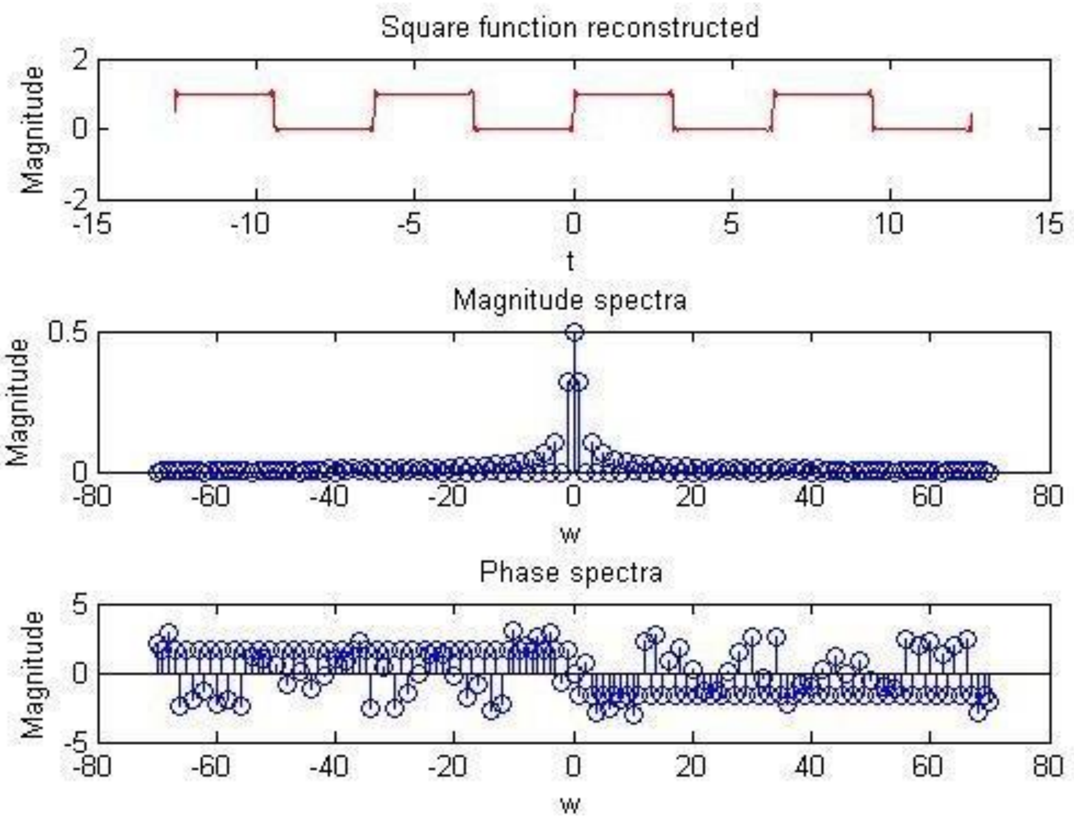
TRIANGLE WAVE

```
clc
clear all
close all
T=2*pi
n=10              n=200
w=[-n:n]*2*pi/T
D=zeros(1,length(w))
fori=1:length(w)
    fun=@(t) sawtooth(t,0.5).*exp(-1j*w(i)*t);
    D(i)=(1/T)*integral(fun,0,2*pi)
end
subplot(3,1,1)
stem(w,abs(D))
xlabel('frequency')
ylabel('Absolute of D(n)')
subplot(3,1,2)
stem(w,angle(D))
xlabel('frequency')
ylabel('Angle of D(n)')

t=-4*pi:0.001:4*pi
g=zeros(1,length(t))
fori=1:length(t)
for j=1:length(w)
```

```
        g(i)=g(i)+(D(j).*exp(1j*w(j).*t(i)));
end
end
subplot(3,1,3)
plot(t,g)
xlabel('time')
ylabel('function')
```

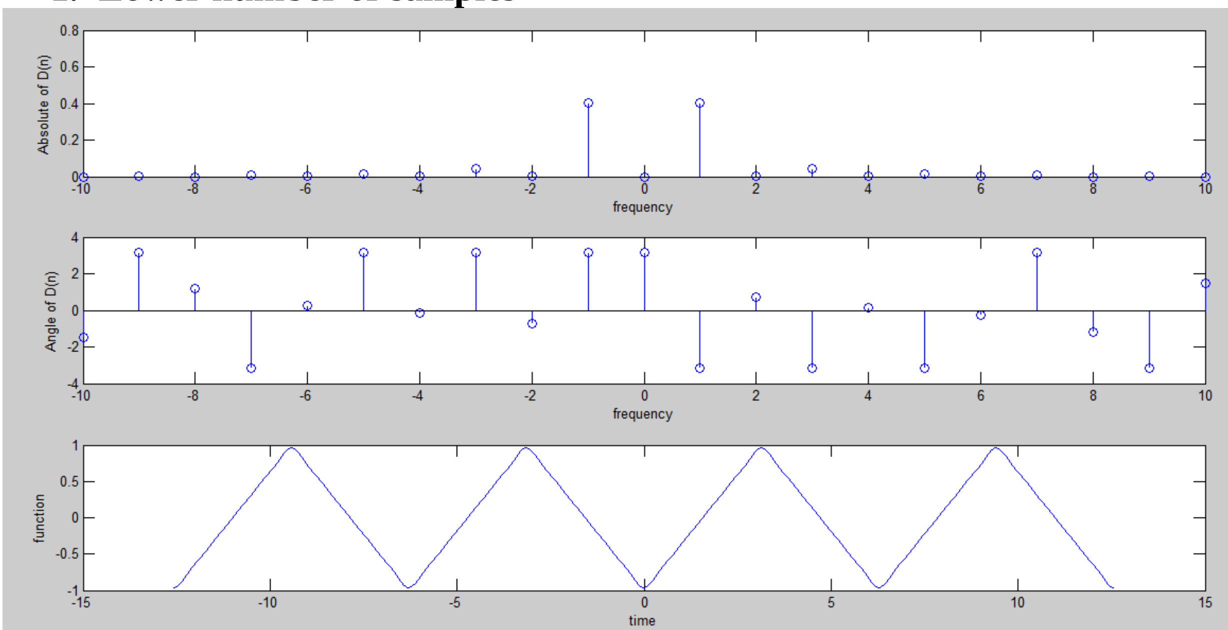
OBSERVATION:



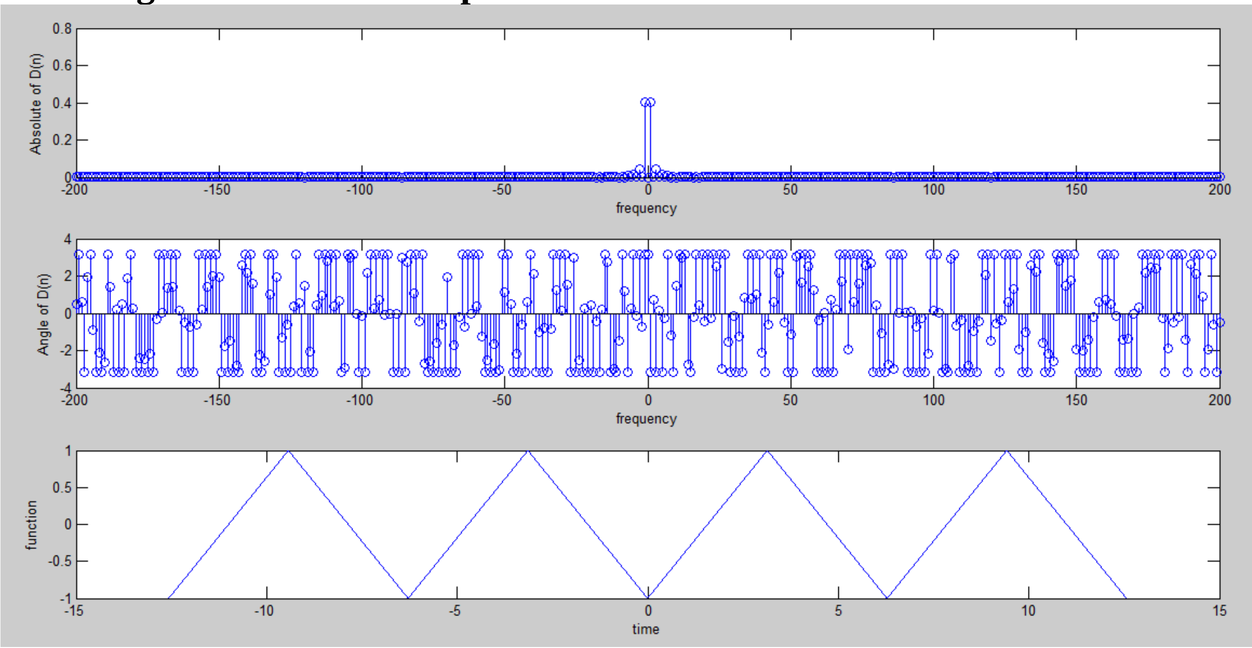


TRIANGULARWAVE:

1. Lower number of samples



2. Higher number of samples



CONCLUSION:

We studied and learnt, how to write a code to compute the Exponential Fourier Series coefficients. And did the same for square wave, saw toothwave and exponential wave. We also plottedtheir magnitude and phase spectrausing MATLAB and observed that we achieved better reconstruction if we increased the number of samples taken.

SIGNATURE

EXPERIMENT:5

AIM: To study amplitude modulation and observe the waveforms for three different Modulation indices and reception of AM signal.

APPARATUS: Trainer board ST 2201 & 2202, power supply, connecting wires, DSO

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:

- amplitude
- frequency
- 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

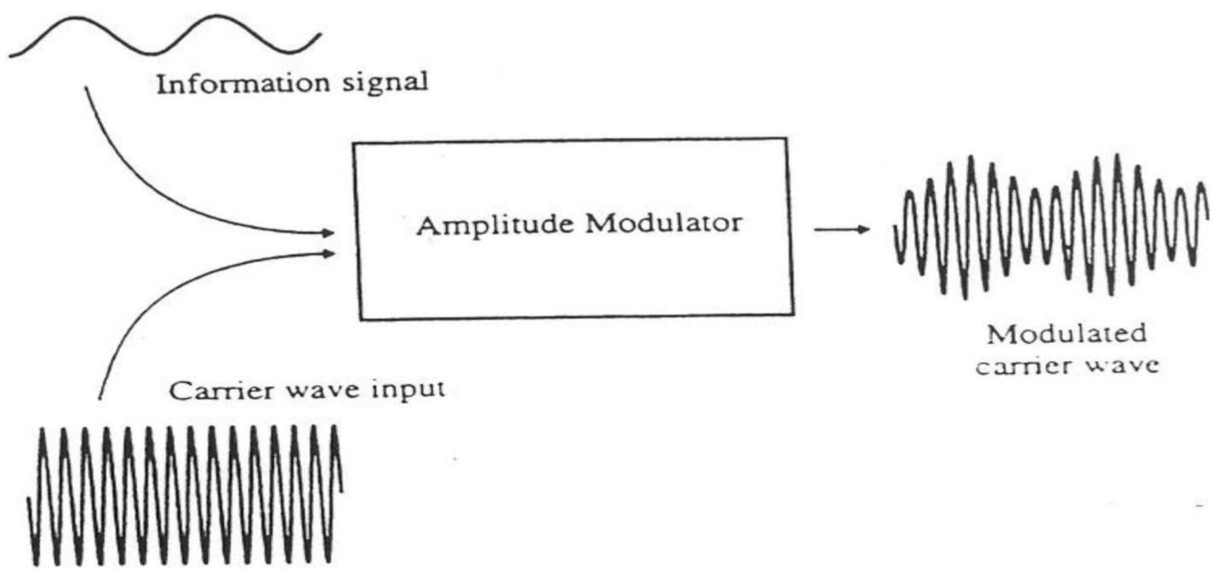


Fig. 1.1 Concept of AM wave

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.

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

AM TRANSMITTER:

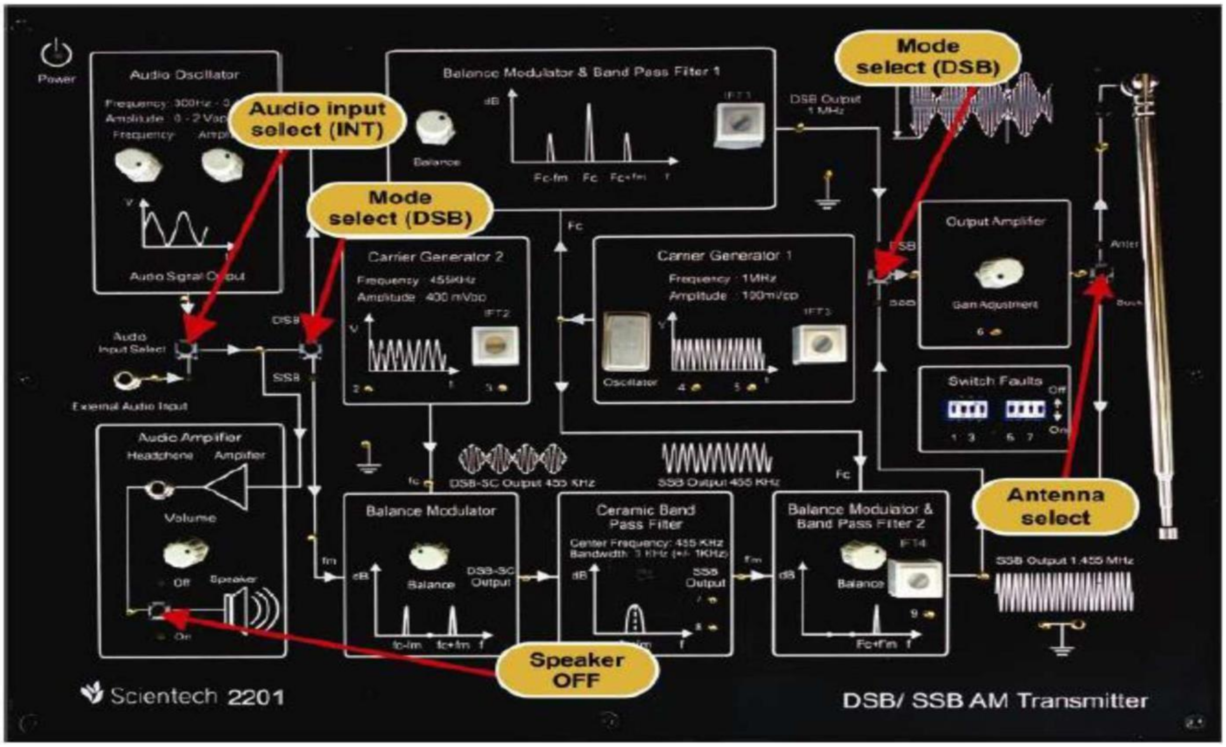


Fig. 1.2 AM Transmitter

AM RECEIVER:

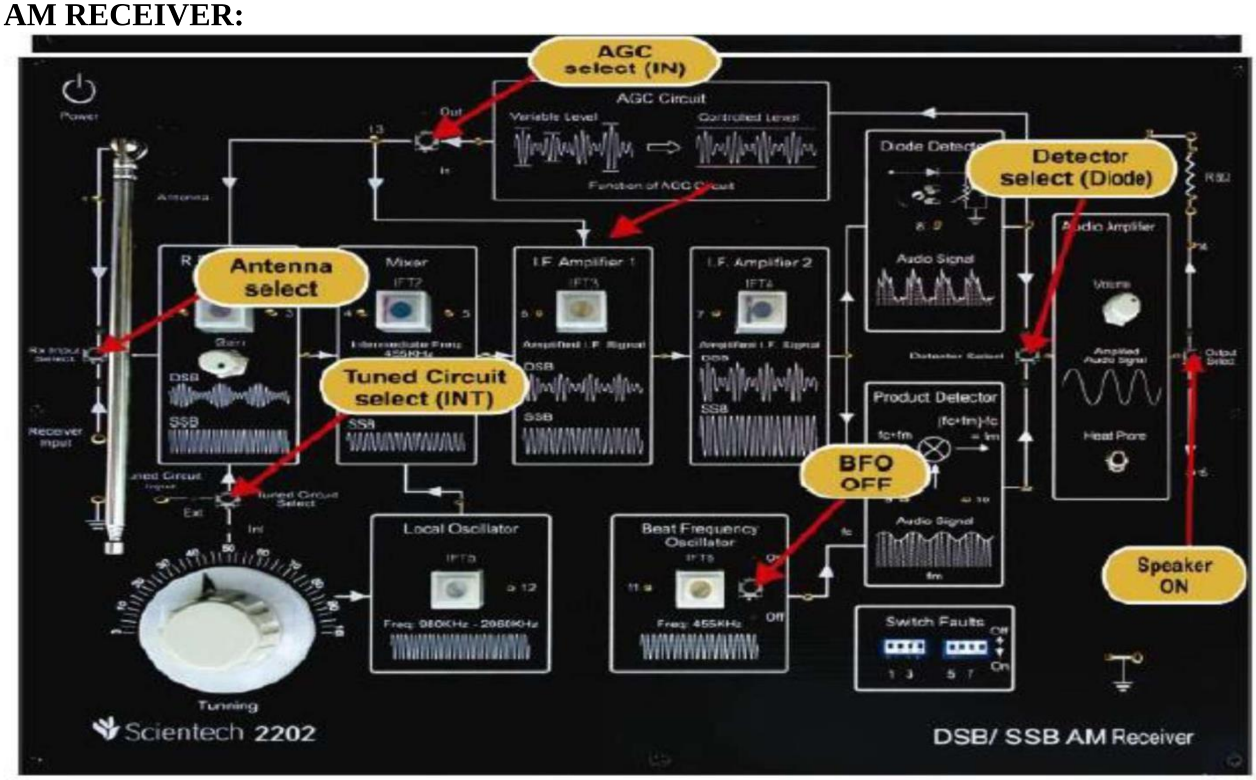


Fig. 1.3 AM Receiver

PROCEDURE:

Modulation:

- Fig. 1.2 shows the AM transmitter panel. The following initial conditions exist on the board.
  1. Audio input select switch in INT position:
  2. Mode switch in DSB position.
  3. Output amplifier's gain pot in full clockwise position.
  4. Speakers switch in OFF position.
- Turn on power to the **ST2201** board.
- 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
- Monitor, in turn, the two inputs to the balanced modulator & band pass filter circuits 1 block, at t.p.1 and t.p.9
- 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.
- 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:

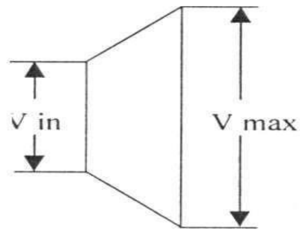
$$\text{Percentage modulation} = \frac{V\max - V\min}{V\max + V\min} \times 100\%$$

Where Vmax and Vmin are the maximum and minimum amplitudes.

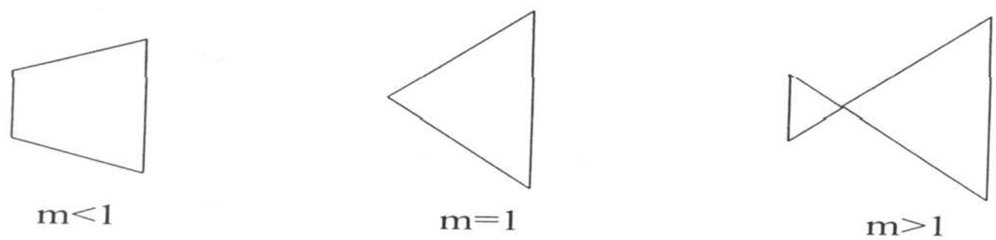
- Now vary the amplitude and frequency of the audio-frequency sine wave, by adjusting the amplitude and frequency present in the audio oscillator block. Note the effect that varying each pot has on the amplitude modulated waveform. The amplitude and frequency amplitudes of the two sidebands can be reduced 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.

To calculate modulation index of DSB wave by trapezoidal pattern.

- Repeat from step no. 1 to step no. 6
- Now apply the modulated waveform to the Y input of the oscilloscope and the modulating signal to the X input.
- Press the XY switch; you will observe the waveform similar to the one given below and then calculate the modulation index.



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



Demodulation:

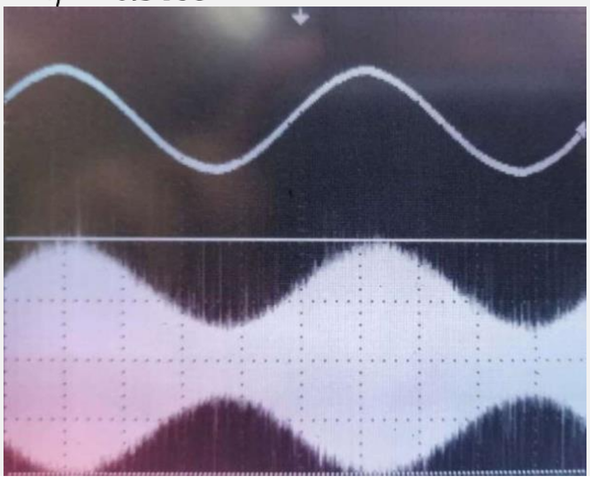
- Fig. 1.3 shows the AM receiver panel. Position the **ST2201** & **ST2202** modules, with the **ST2201** board the left, and a gap of about three inches between them.
- Ensure that proper initial conditions exist on the **ST2201 & ST2202** board
- Turn on power to the modules. We will now transmit the 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).
- 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.
- 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.

OBSERVATION:

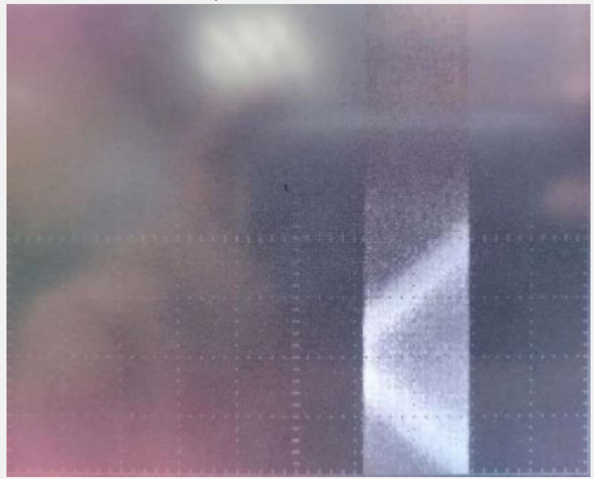
Y-T Domain | **X-Y Domain**

1 Under Modulation

$\eta = 0.5468$

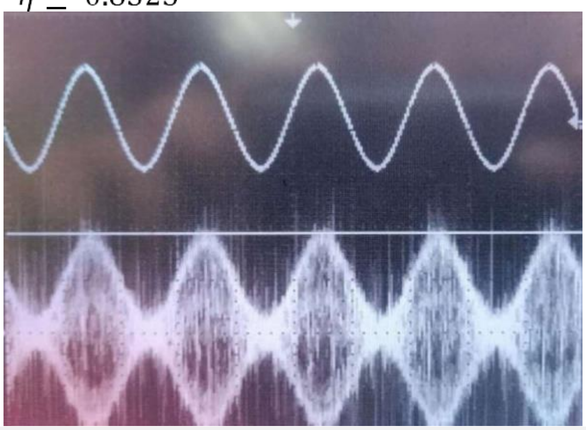


V_{Max} = 7.92 volt ; V_{Min} = 2.32 volt



2 Critical Modulation

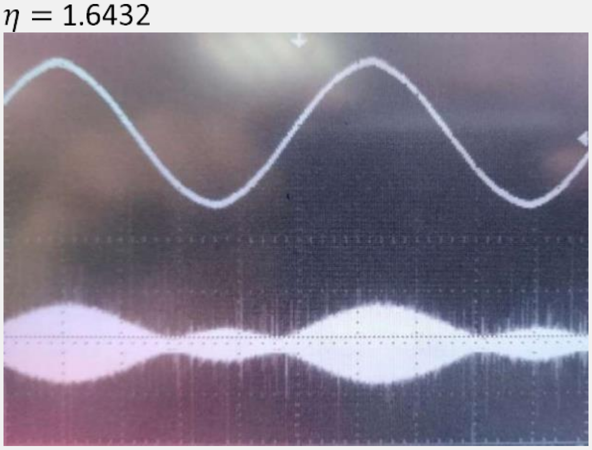
η _ 0.8325



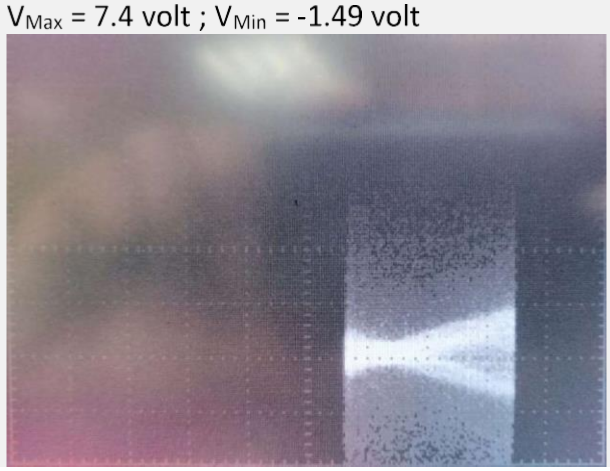
V_{Max} = 7.44 volt ; V_{Min} = 0.68 volt

3 Over Modulation

$\eta = 1.6432$



V_{Max} = 7.4 volt ; V_{Min} = -1.49 volt



CONCLUSION:

By performing this experiment, we learnt about AM kit and basic theories and implementation of both Transmitter and Receiver side.

SIGNATURE

EXPERIMENT:6

AIM: Write a MATLAB code for Amplitude Modulation and Demodulation considering sinusoidal signal as an input. Plot the various signal obtained at each step of the process in the time domain and frequency domain.

APPARATUS: MATLAB Software

MATLAB CODE:

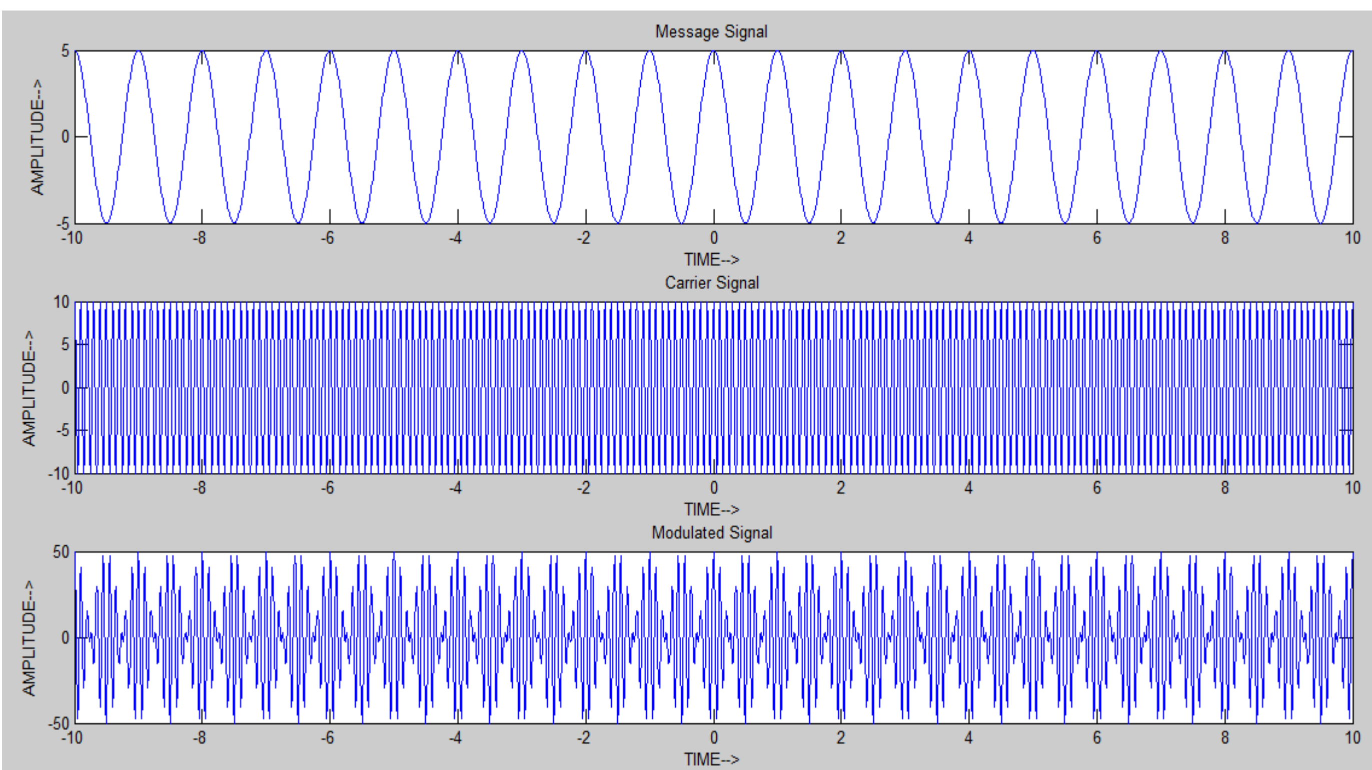
```
clc
clear all
close all
fm=1
fc=10
fs=100
t=-10:(1/fs):10
wm=2*pi*fm
wc=2*pi*fc
m=5*cos(wm*t)
c=10*cos(wc*t)
s=m.*c;


figure
subplot(3,1,1)
plot(t,m)
xlabel('TIME-->')
ylabel('AMPLITUDE-->')
title('Message Signal')
```

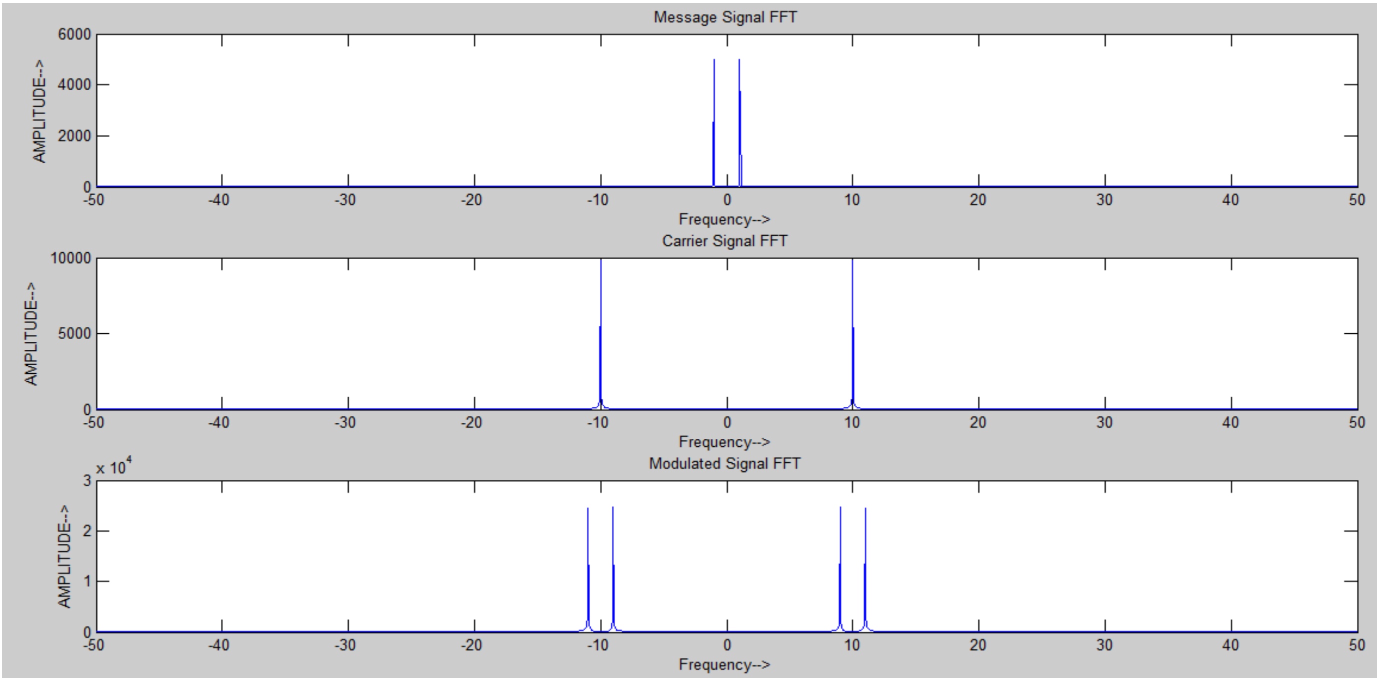
```
subplot(3,1,2)

plot(t,c)

xlabel('TIME-->')

ylabel('AMPLITUDE-->')

title('Carrier Signal')

subplot(3,1,3)

plot(t,s)

xlabel('TIME-->')

ylabel('AMPLITUDE-->')

title('Modulated Signal')


M=fftshift(fft(m));

C=fftshift(fft(c));

S=fftshift(fft(s));

n=length(t);

f=(-fs/2):(fs/(n-1)):(fs/2)

figure

subplot(3,1,1)

plot(f,abs(M))

xlabel('Frequency-->')

ylabel('AMPLITUDE-->')

title('Message Signal FFT')

subplot(3,1,2)

plot(f,abs(C))

xlabel('Frequency-->')
```

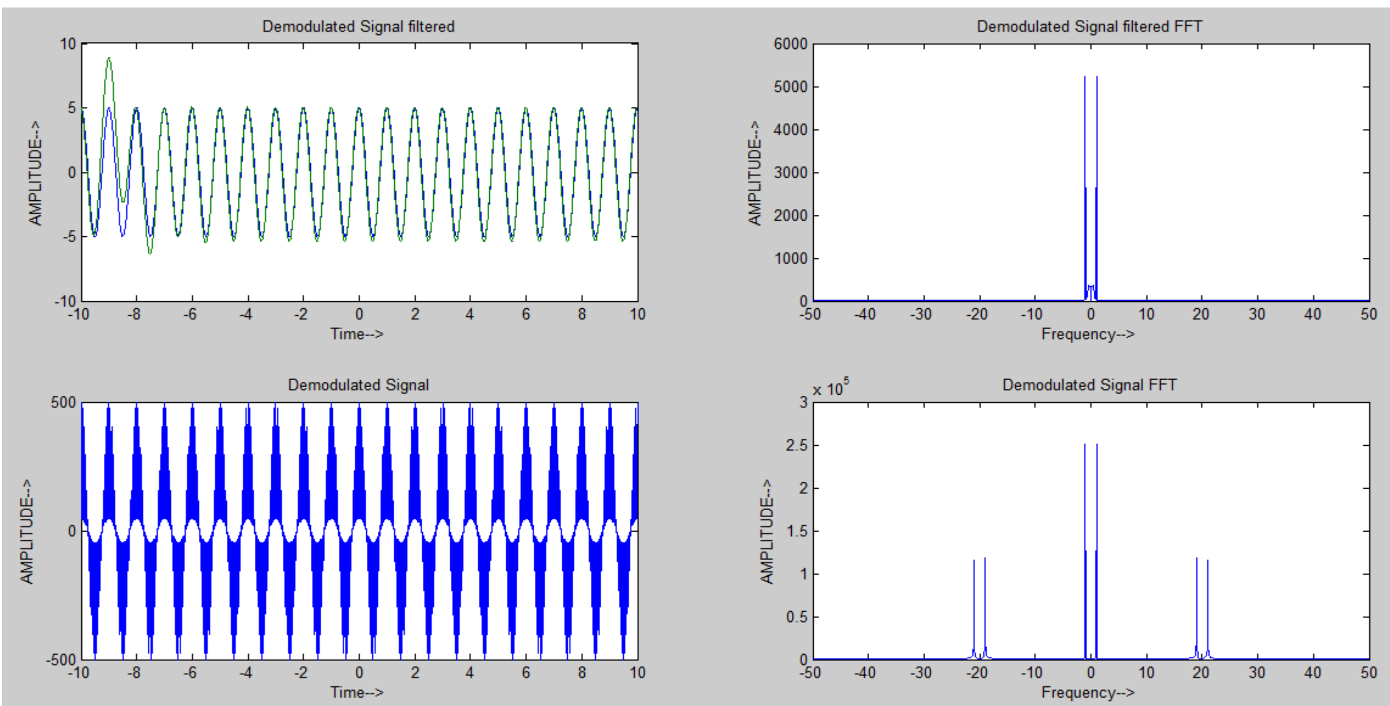
```
ylabel('AMPLITUDE-->')
title('Carrier Signal FFT')
subplot(3,1,3)
plot(f,abs(S))
xlabel('Frequency-->')
ylabel('AMPLITUDE-->')
title('Modulated Signal FFT')

dm=s.*c;
DM=fftshift(fft(dm));
[B A]=butter(5,(fm/fs),'low')
[y z]=(filter(B,A,dm));
y=(filter(B,A,dm,z))/1.5;
d=mean(y);
y=y-d;
Y=fftshift(fft(y));
figure
subplot(2,2,1)
plot(t,m,t,y)
xlabel('Time-->')
ylabel('AMPLITUDE-->')
title('Demodulated Signal filtered')
subplot(2,2,2)
plot(f,abs(Y))
xlabel('Frequency-->')
```

```
ylabel('AMPLITUDE-->')
title('Demodulated Signal filtered FFT')
subplot(2,2,3)
plot(t,dm)
xlabel('Time-->')
ylabel('AMPLITUDE-->')
title('Demodulated Signal')
subplot(2,2,4)
plot(f,abs(DM))
xlabel('Frequency-->')
ylabel('AMPLITUDE-->')
title('Demodulated Signal FFT')
```

OBSERVATION:







CONCLUSION:

In this experiment, we modulated and demodulated the given sinusoidal signal and plotted its various graphs in time and frequency domain.

SIGNATURE

EXPERIMENT:7

AIM: To study about Frequency Division Multiplexing/ Demultiplexing with sinusoidal wave. Also observe the Fourier Transform of all the signals at each stage.

APPARATUS: Trainer board ST 2211, power supply, connecting wires, DSO

THEORY:

Frequency-division multiplexing (FDM) is a scheme in which numerous signals are combined for transmission on a single communications line or channel. Each signal is assigned a different frequency (sub-channel) within the main channel. There are N signals in frequency; each is band-limited to fm Hz. In order to separate N signals in frequency, each is modulated with a carrier frequency f_{C1}, f_{C2}… f_{CN}.

Using DSB-SC, the spectral density of every modulated signal has a bandwidth of 2fm and each is 2fm and each is centered at various carrier frequencies f_{C1}, f_{C2}… f_{CN}. These carrier frequencies are chosen far enough apart such that each signal spectral density is separated from all the others. FDM transmitter is shown in following figure 1.

In Telephony, the most widely used method of modulation in FDM is single sideband modulation, which, in the case of voice signals, requires a bandwidth that is approximately equal to that of the original voice signal. Each voice input is usually assigned a bandwidth of 4 KHz. The band pass filters following the modulators are used to restrict the band of each modulated signal to its prescribed range. The resulting band pass filter outputs are combined in parallel to form the input to the common channel. At the receiving terminal, a bank of filters, with their inputs connected in parallel, is used to separate the message signals on a frequency-occupancy basis.

The original message signals are recovered by individual demodulators.FDM allows engineers to transmit multiple data streams simultaneously over the same channel, at the expense of bandwidth. To that extent, FDM provides a trade-off: faster data for less bandwidth. Also, to demultiplex an FDM signal requires a series of band pass filters to isolate each individual signal. Band pass filters are relatively complicated and expensive, therefore the receivers in an FDM system are generally expensive. As an example of an FDM system, Commercial broadcast radio (AM and FM radio) simultaneously transmits multiple signals or "stations" over the airwaves. These stations each get their own frequency band to use, and a radio can be tuned to receive each different station. Another good example is cable television, which simultaneously transmits every channel, and the TV "tunes in" to which channel it wants to watch.

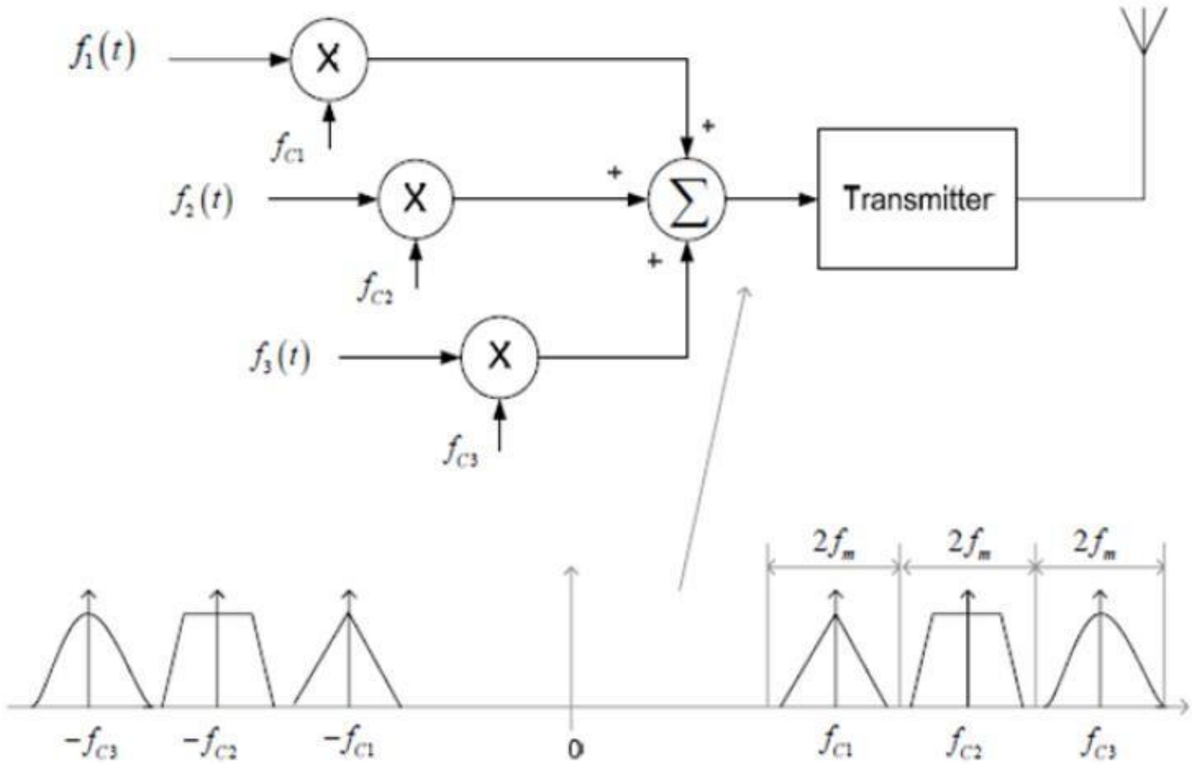


Fig.1 FDM transmitter Diagram

PROCEDURE:

- Set the modulating frequency of Ch-1 with the help of potentiometer to 2 KHz and Ch-2 to 4 KHz.
- Observe the carrier frequency 100 KHz and 200 KHz on the oscilloscope.
- Connect the Ch-1 output to the left input of the modulator Ch-1.
- Repeat step 3 for Ch-2 also.
- Connect carrier generator outputs (100 KHz and 200 KHz) to Ch-1 and Ch-2 respectively.
- Observe the modulator output on oscilloscope.
- Connect the modulator output of Ch-1 and Ch-2 to adder circuit.
- Connect the adder output to demodulator inputs in both the sections.
- Connect the respective carrier frequency to demodulator second inputs.
- Connect the outputs of demodulator of Ch-1 and Ch-2 to LPF-1 and LPF-2.

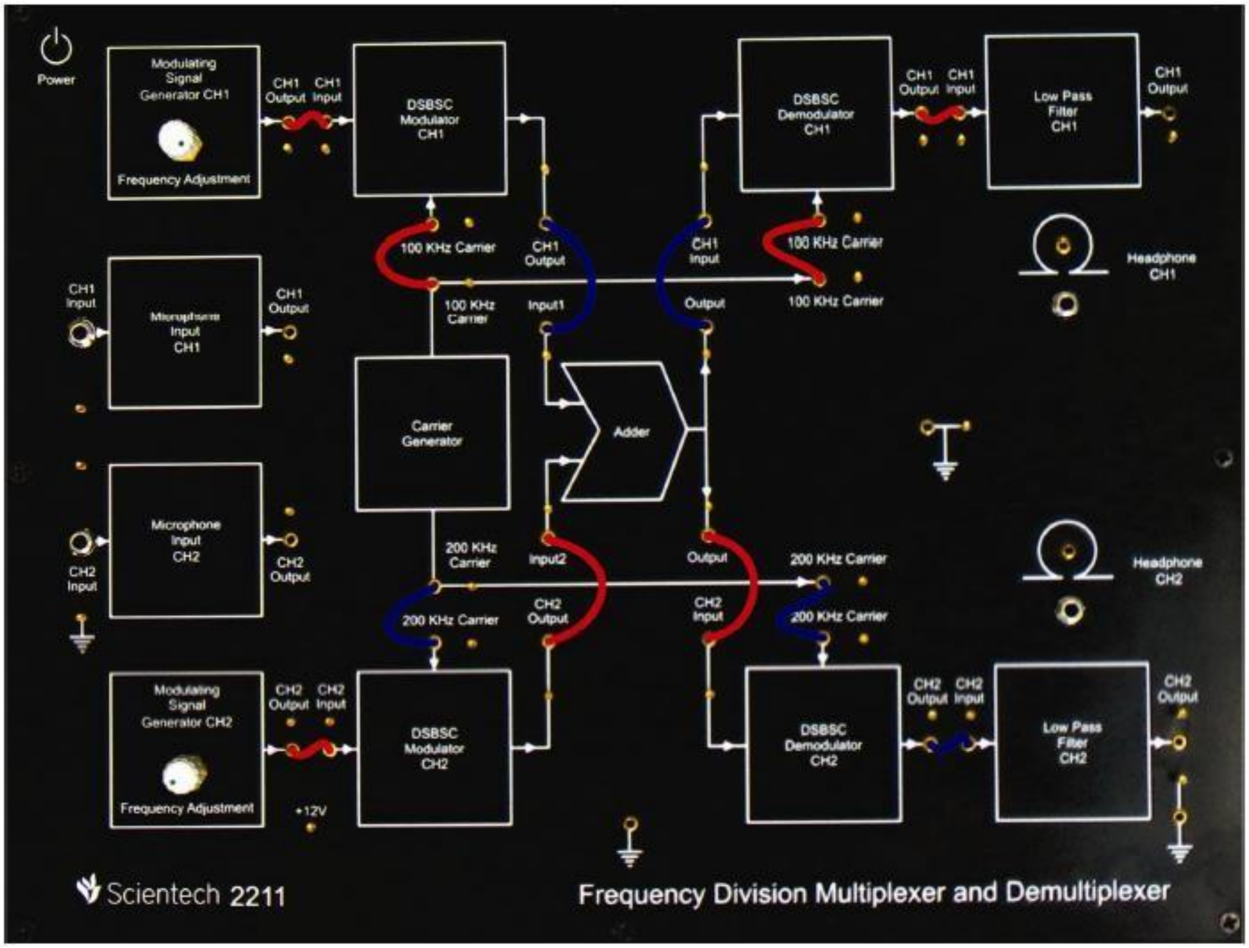
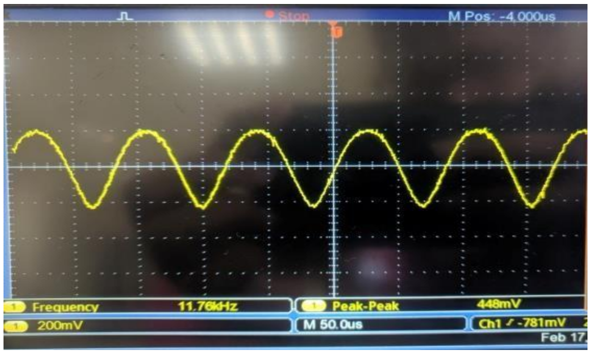


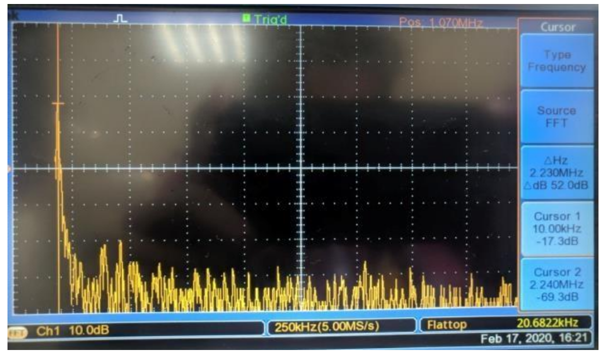
Fig.2 FDM trainer kit

OBSERVATION:

SIGNAL	FFT

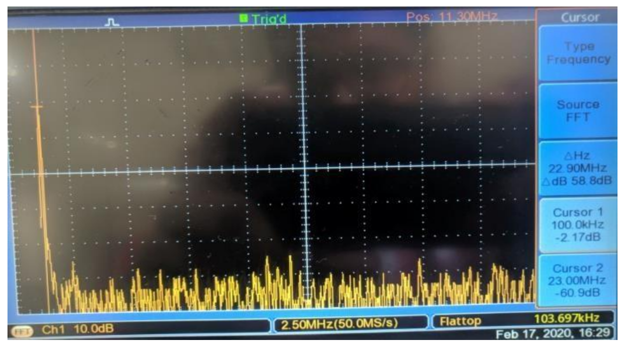


1) Message Signal (10 kHz)

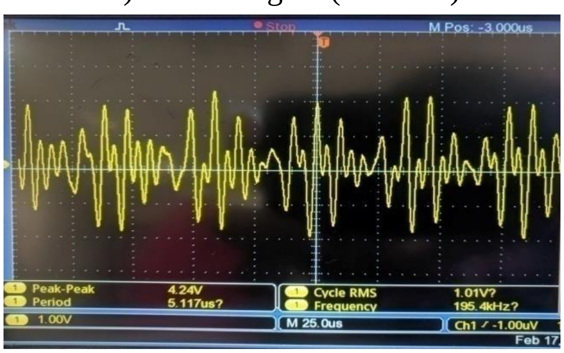


1) Message Signal FFT (10 kHz)

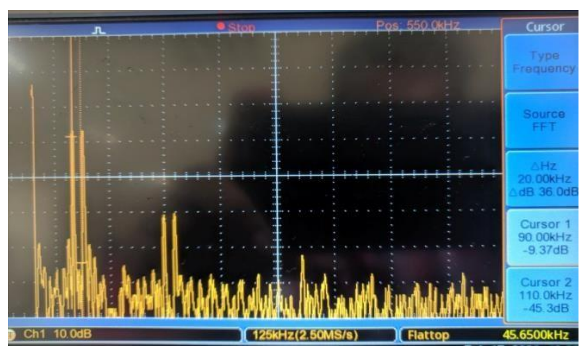
1) Carrier Signal (100 kHz)



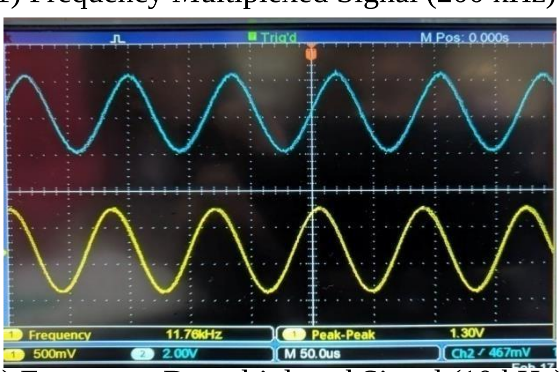
1) Carrier Signal FFT (100 kHz)



1) Frequency Multiplexed Signal (200 kHz)

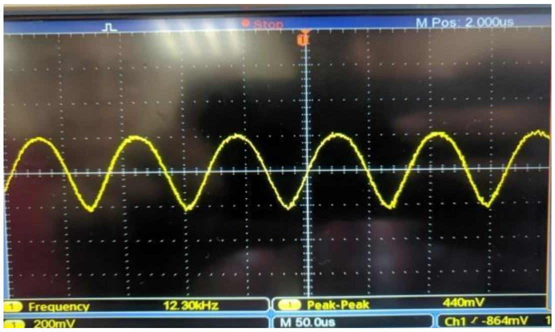


1) Double Side Band - SC (90 kHz & 110 kHz)

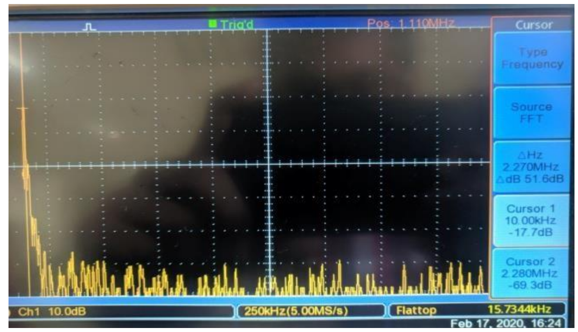


1) Frequency Demultiplexed Signal (10 kHz)

SIGNAL	**FFT**

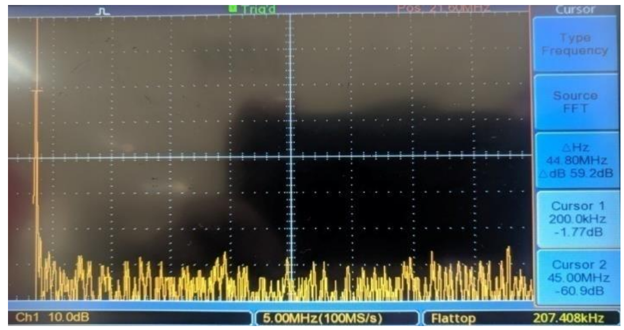


2) Message Signal (10 kHz)

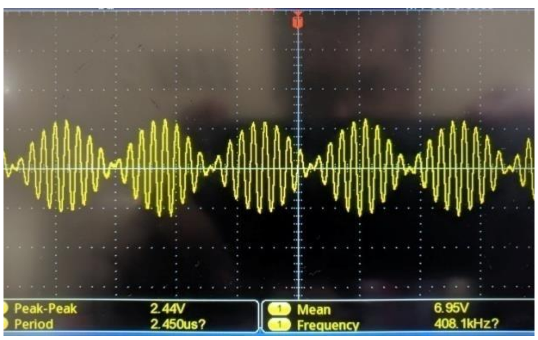


2) Message Signal FFT (10 kHz)

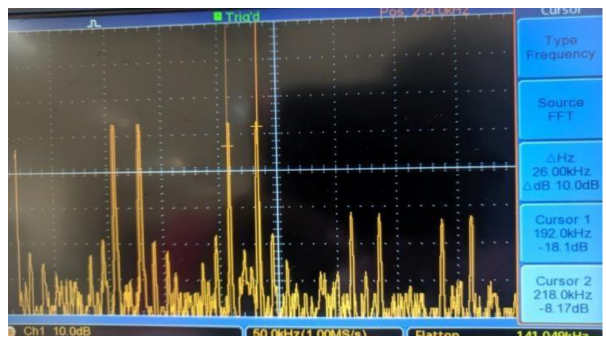
2) Carrier Signal (200 kHz)



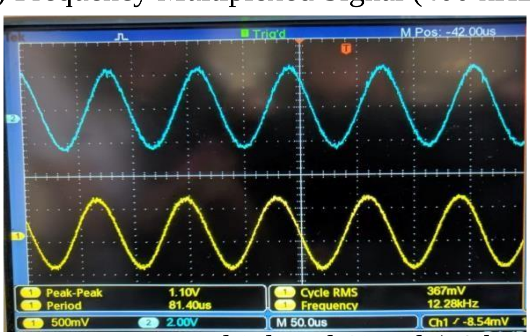
2) Carrier Signal FFT (200 kHz)



2) Frequency Multiplexed Signal (400 kHz)



2) Double Side Band-SC (190 kHz & 210 kHz)



2)Frequency Demultiplexed Signal (10 kHz)

CONCLUSION:

In this experiment, we multiplexed and de-multiplexed the message signal and the carrier signal and observed the Fast Fourier Transform(FFT) at each stage.

SIGNATURE

EXPERIMENT:8

AIM: Write a MATLAB code for Frequency Modulation and Demodulation considering sinusoidal signal as an input. Also plot the FM signal in Frequency Domain.

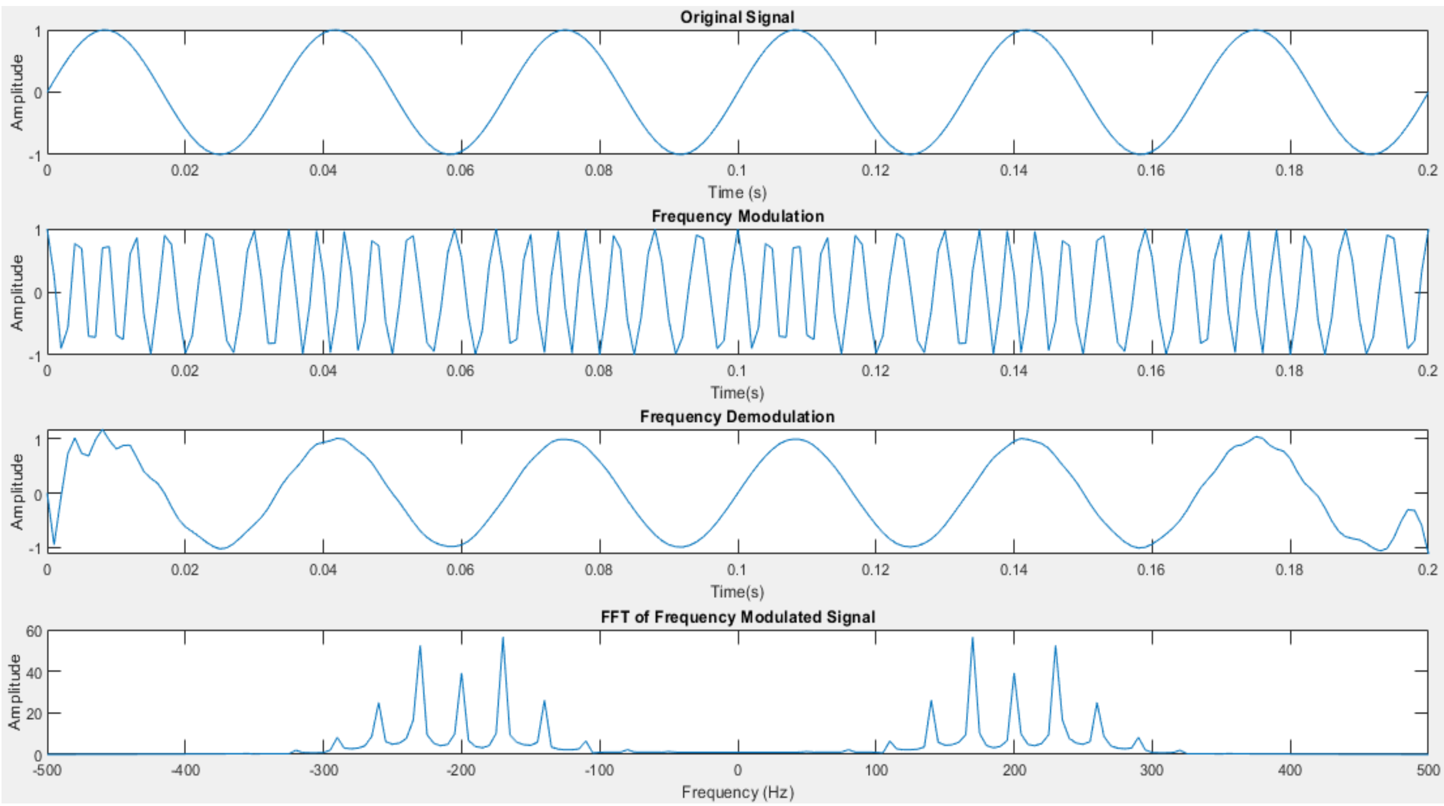
APPARATUS: MATLAB Software

MATLAB CODE:

```
clc
clear all
close all

fs = 1000;
fc = 200;
t = (0:1/fs:0.2)
fd = 50;

x = sin(2*pi*30*t);
y = fmmod(x,fc,fs,fd);

subplot(4,1,1)
plot(t,x)
title('Original Signal')
xlabel('Time (s)')
ylabel('Amplitude')
subplot(4,1,2)
plot(t,y)
title('Frequency Modulation')
xlabel('Time(s)')
ylabel('Amplitude')

z = fmdemod(y,fc,fs,fd);

subplot(4,1,3)
plot(t,z)
title('Frequency Demodulation')
xlabel('Time(s)')
ylabel('Amplitude')



Y = fftshift(fft(y))
n=length(t);
f=(-fs/2):(fs/(n-1)):(fs/2)
```

```
subplot(4,1,4)
plot(f,abs(Y))
title('FFT of Frequency Modulated Signal')
xlabel('Frequency (Hz)')
ylabel('Amplitude')
```

OBSERVATION:



CONCLUSION:

In this experiment, we frequency modulated a sinusoidal wave and then demodulated it. The output wave at each stage in time and frequency domain was observed and recorded.

SIGNATURE

EXPERIMENT:9

AIM: Write a MATLAB code

(A) To plot PDF and CDF for Gaussian Distribution

(B) To plot PDF and CDF for Rayleigh Distribution

APPARATUS: MATLAB Software

MATLAB CODE:

(A) To plot PDF and CDF for Gaussian Distribution

```
clc
clear all
close all

x=-10:0.1:10
a=sqrt(2*pi)

pdf=(exp((-x.^2)/2))/a

figure
plot(x,pdf)
title('Gaussian PDF Plot')
xlabel('Range')
ylabel('Probability Distribution')

fun=exp((-x.^2)/2)
int=cumtrapz(x,fun)
cdf=int/a

figure
plot(x,cdf)
title('Guassian CDF Plot')
xlabel('Range')
ylabel('Probability Distribution')
```

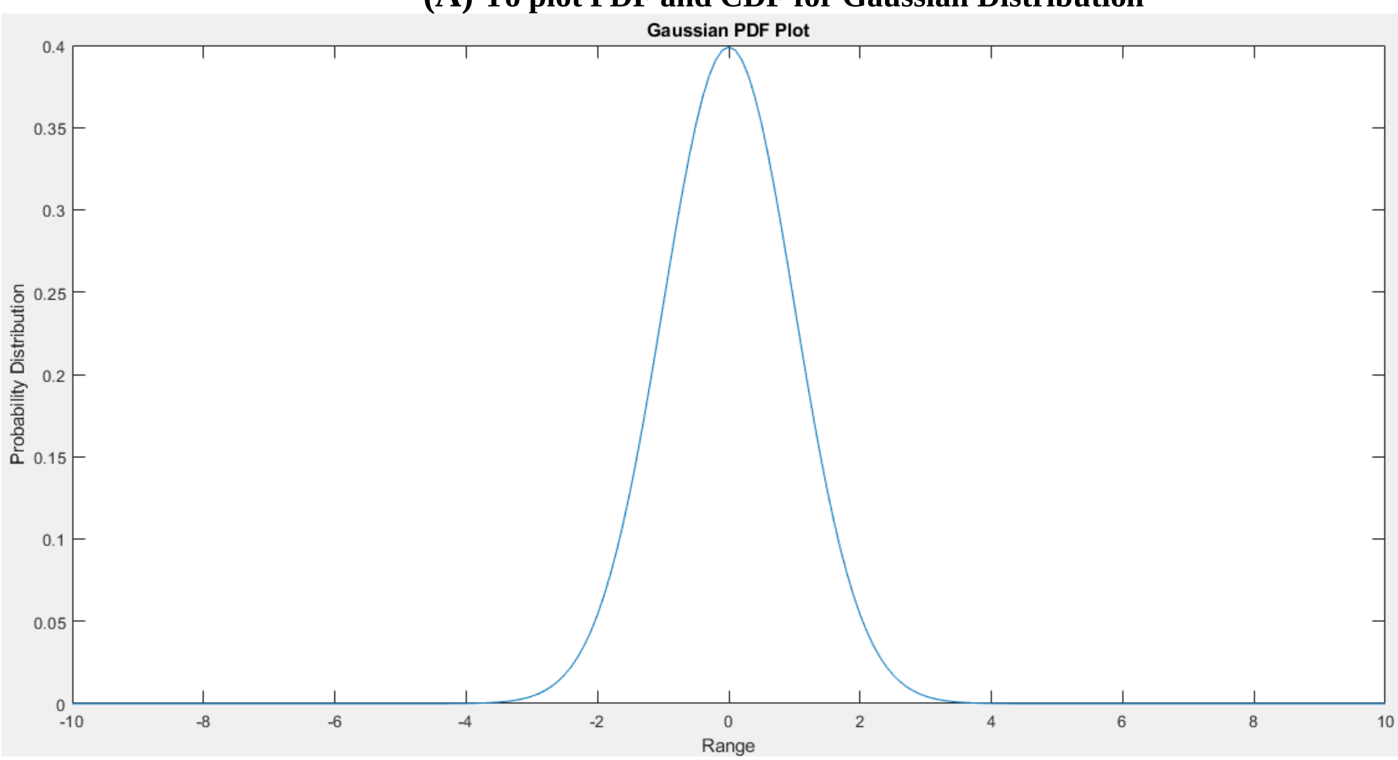
(B) To plot PDF and CDF for Rayleigh Distribution

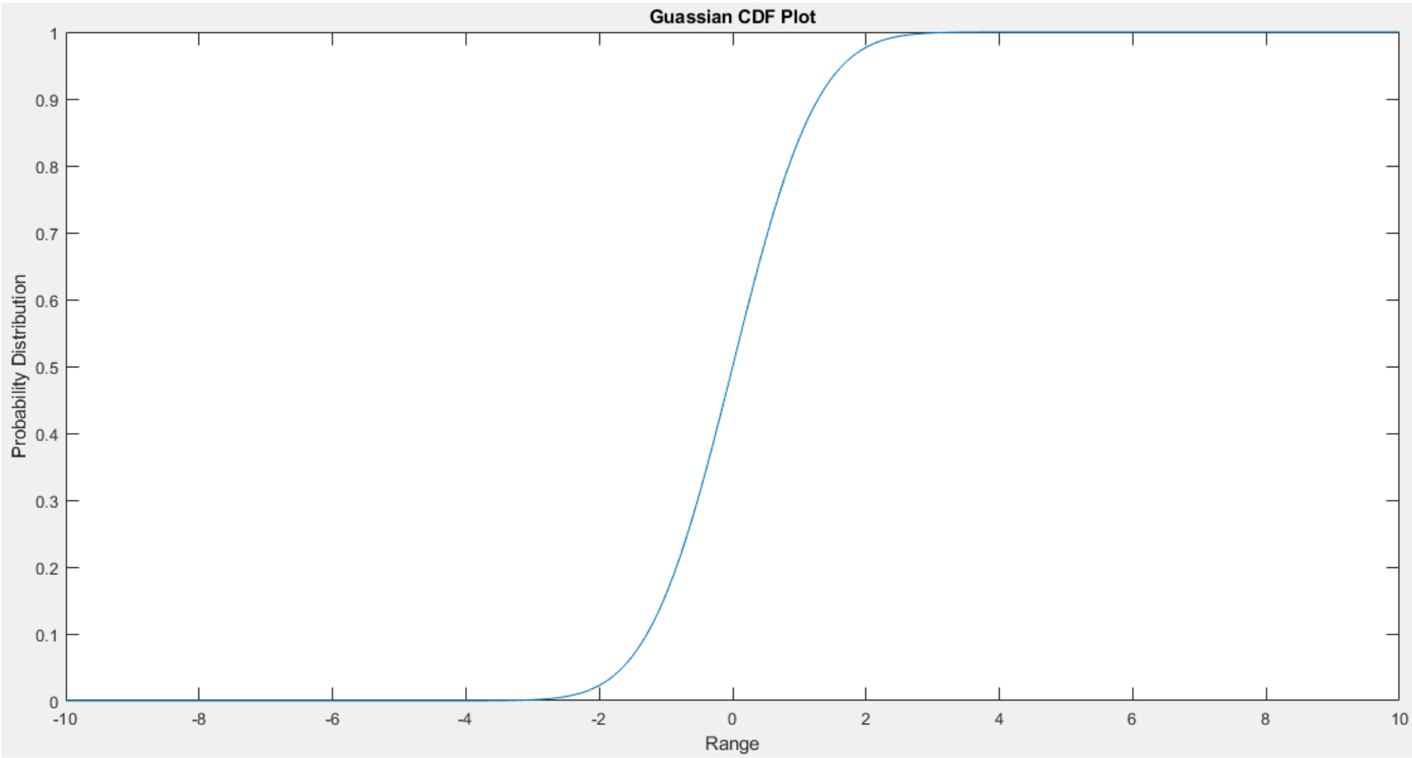
```
clc;
close all;
clear all;
```

```
sig=1;
r = 0:0.1:10;

pdf=(r/sig.^2) .* exp(-r.^2/2*(sig.^2));

figure;
plot(r,pdf)
title('Rayleigh PDF Plot');
xlabel('Range');
ylabel('Probability');

cdf=1-exp(-r.^2/(2*1));

figure;
plot(r,cdf)
title('Rayleigh CDF Plot');
xlabel('Range');
ylabel('Probability');
```

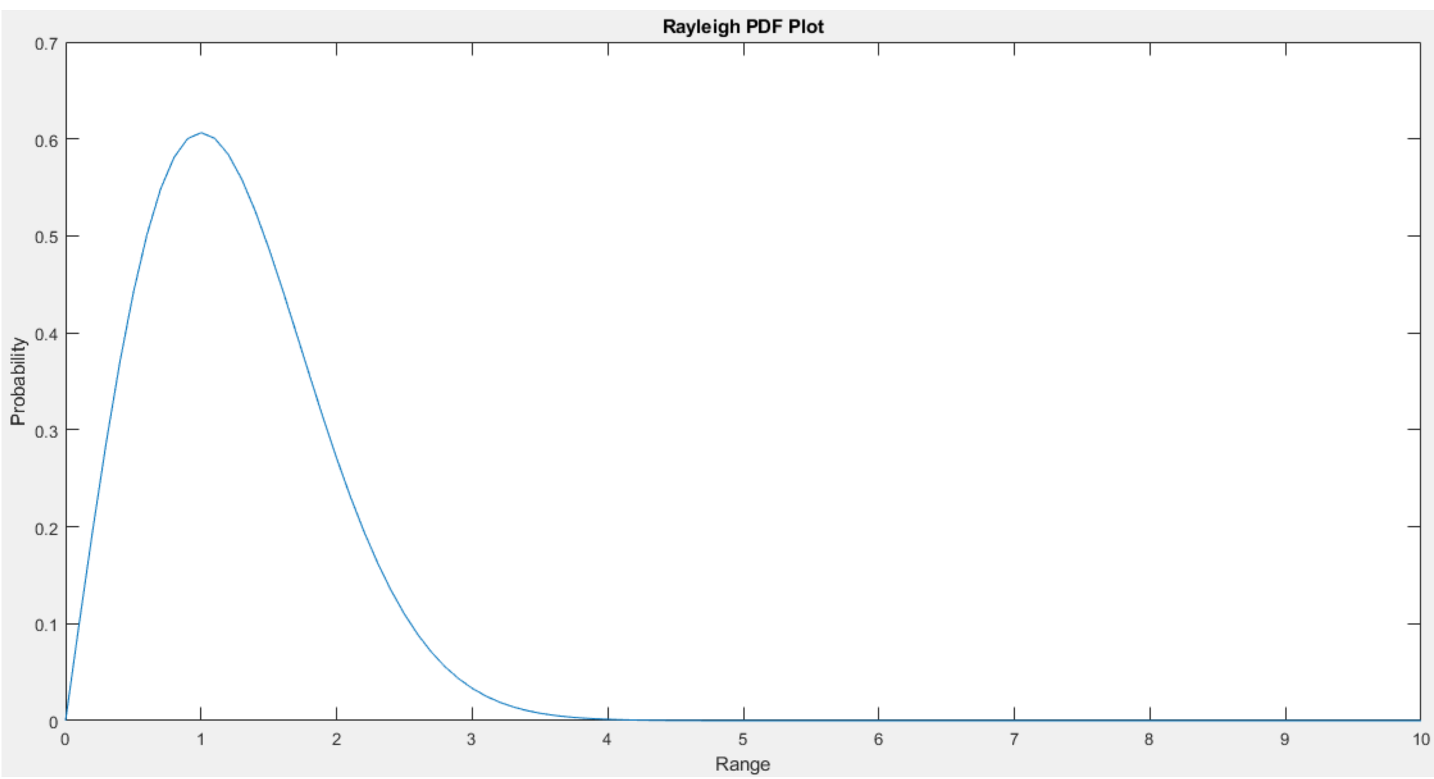
OBSERVATION:

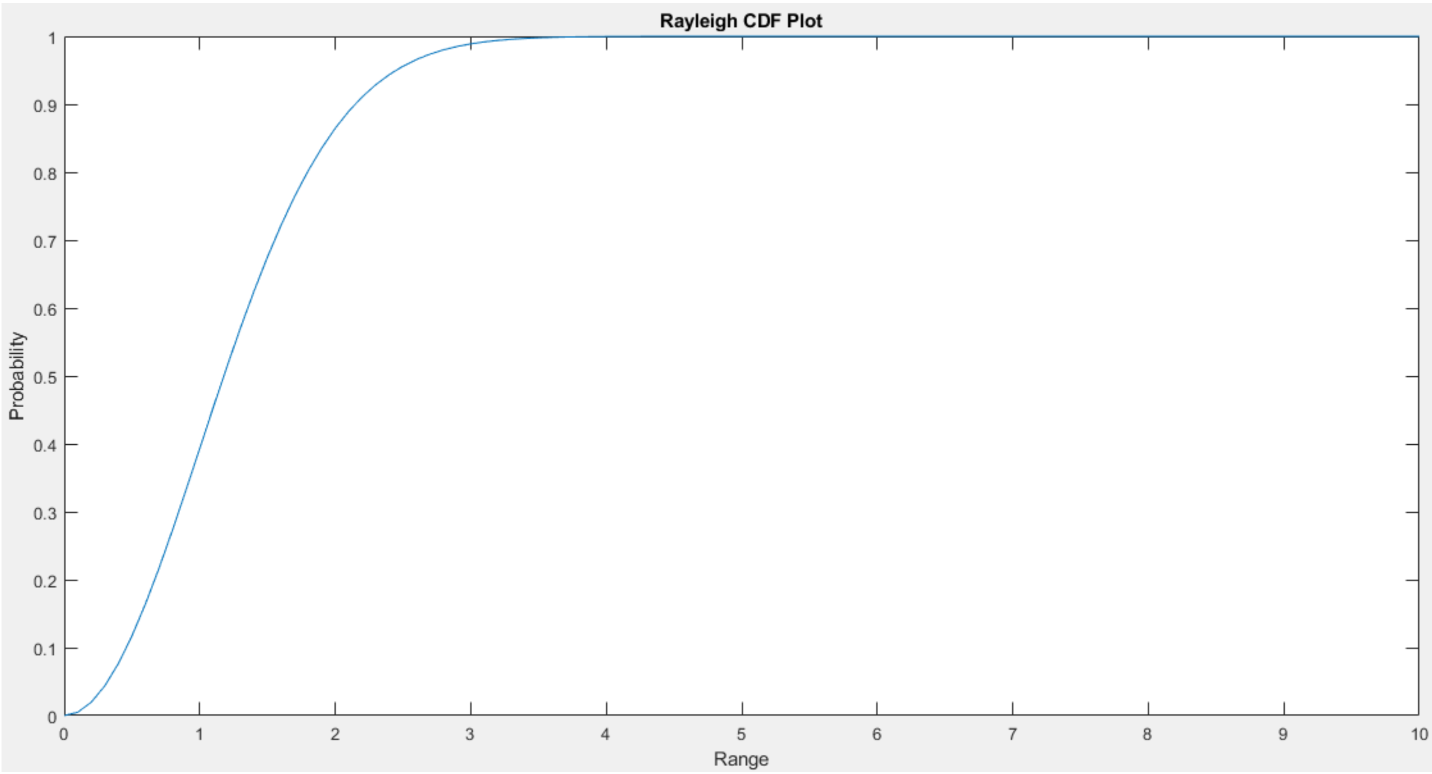
(A) To plot PDF and CDF for Gaussian Distribution





(B) To plot PDF and CDF for Rayleigh Distribution





CONCLUSION:

From this experiment, we plotted and observed the graph of Gaussian and Rayleigh Probability Distribution Function(PDF) and Cumulative Distribution Function(CDF) using their respective function formulae.

SIGNATURE

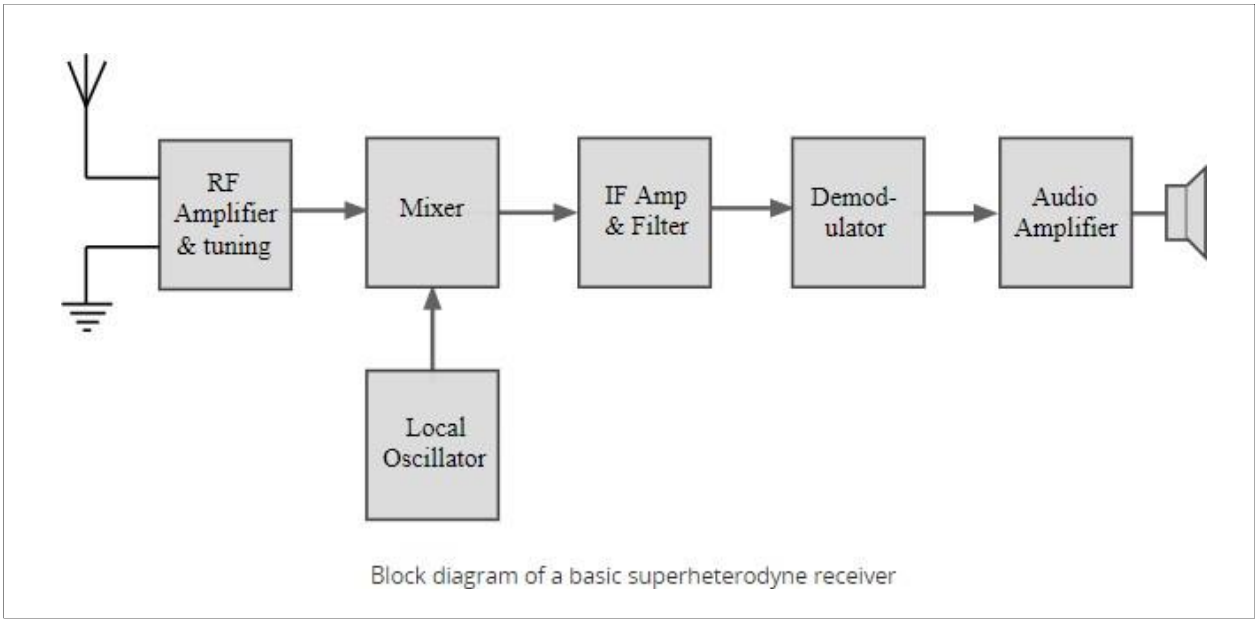
EXPERIMENT:10

AIM: Design a Superheterodyne Receiver using MATLAB simulator for Frequency Modulation. Plot the various signal obtained at each step of the process in the time domain and frequency domain.

APPARATUS: MATLAB Software

THEORY:

Signals enter the receiver from the antenna and are applied to the RF amplifier where they are tuned to remove the image signal and also reduce the general level of unwanted signals on other frequencies that are not required.



Block Diagram

The signals are then applied to the mixer along with the local oscillator where the wanted signal is converted down to the intermediate frequency. Here significant levels of amplification are applied and the signals are filtered. This filtering selects signals on one channel against those on the next. It is much larger than that employed in the front end.The advantage of the IF filter as opposed to RF filtering is that the filter can be designed for a fixed frequency. This allows for much better tuning. Variable filters are never able to provide the same level of selectivity that can be provided by fixed frequency ones.

Once filtered the next block in the superheterodyne receiver is the demodulator. This could be for amplitude modulation, single sideband, frequency modulation, or indeed any form of modulation. It is also possible to switch different demodulators in according to the mode being received.

The final element in the superheterodyne receiver block diagram is shown as an audio amplifier, although this could be any form of circuit block that is used to process or amplified the demodulated signal.

MATLAB CODE:

```
clc;


clear all;
close all;


%<------------------------Modulation ---------------------- >%
%<-----Signal Modulation---->%


freq_message = 30;
freq_carrier = 200;


freq = 1000;
time = -1:(1/freq):1;
fd=50;


w_message = 2*pi*freq_message;
w_carrier = 2*pi*freq_carrier;
```

```
message_signal = 5*cos(w_message*time);

modulated = fmmod(message_signal,freq_carrier,freq,fd);

%<-----Signal FFT --- >%
message_signal_fft = fftshift(fft(message_signal));
modulated_fft      = fftshift(fft(modulated));

freq_range = (-freq/2):(freq/(length(time)-1)):(freq/2);

%<----Ploting --->%
%<-time->%

figure
subplot(2,1,1)
plot(time,message_signal)
xlabel('TIME ->')
ylabel('AMPLITUDE  ->')
title('Message Signal 5cos(wmt)')
subplot(2,1,2)
plot(freq_range,abs(message_signal_fft))
xlabel('Frequency ->')
ylabel('AMPLITUDE ->')
title('Message Signal FFT')
```

```
figure
subplot(2,1,1)
plot(time,modulated)
xlabel('TIME ->')
ylabel('AMPLITUDE  ->')
title('Modulated Signal')
subplot(2,1,2)
plot(freq_range,abs(modulated_fft))
xlabel('Frequency ->')
ylabel('AMPLITUDE ->')
title('Modulated Signal FFT')


%<------------------------Superhetrodyne reciver----------------------->%
%<-----RF Amplifier --->%
magnitude = 10;
out_rf_amp = modulated*magnitude;

figure
subplot(2,1,1)
plot(time,out_rf_amp)
xlabel('Time ->')
ylabel('AMPLITUDE ->')
title('Output of RF Amplifier:')
```

```
subplot(2,1,2)
out_rf_amp_fft = fftshift(fft(out_rf_amp));
plot(freq_range,abs(out_rf_amp_fft))
xlabel('Frequency ->')
ylabel('AMPLITUDE ->')
title('FFT')

out_bandpass = modulated;

%<-----Freq mixer --- >%
IF = 455;
w_intermidiate = 2*pi*IF;
local_oscillator = cos((w_carrier + w_intermidiate)*time);
out_freq_mixer = local_oscillator .* out_bandpass;

out_freq_mixer_fft = fftshift(fft(out_freq_mixer));

figure
subplot(2,1,1)
plot(time,out_freq_mixer)
xlabel('Time ->')
ylabel('AMPLITUDE ->')
title('Frequency Mixer Output:')
subplot(2,1,2)
plot(freq_range,abs(out_freq_mixer_fft))
```

```
xlabel('Frequency ->')
ylabel('AMPLITUDE ->')
title('Frequency Mixer FFT')
```

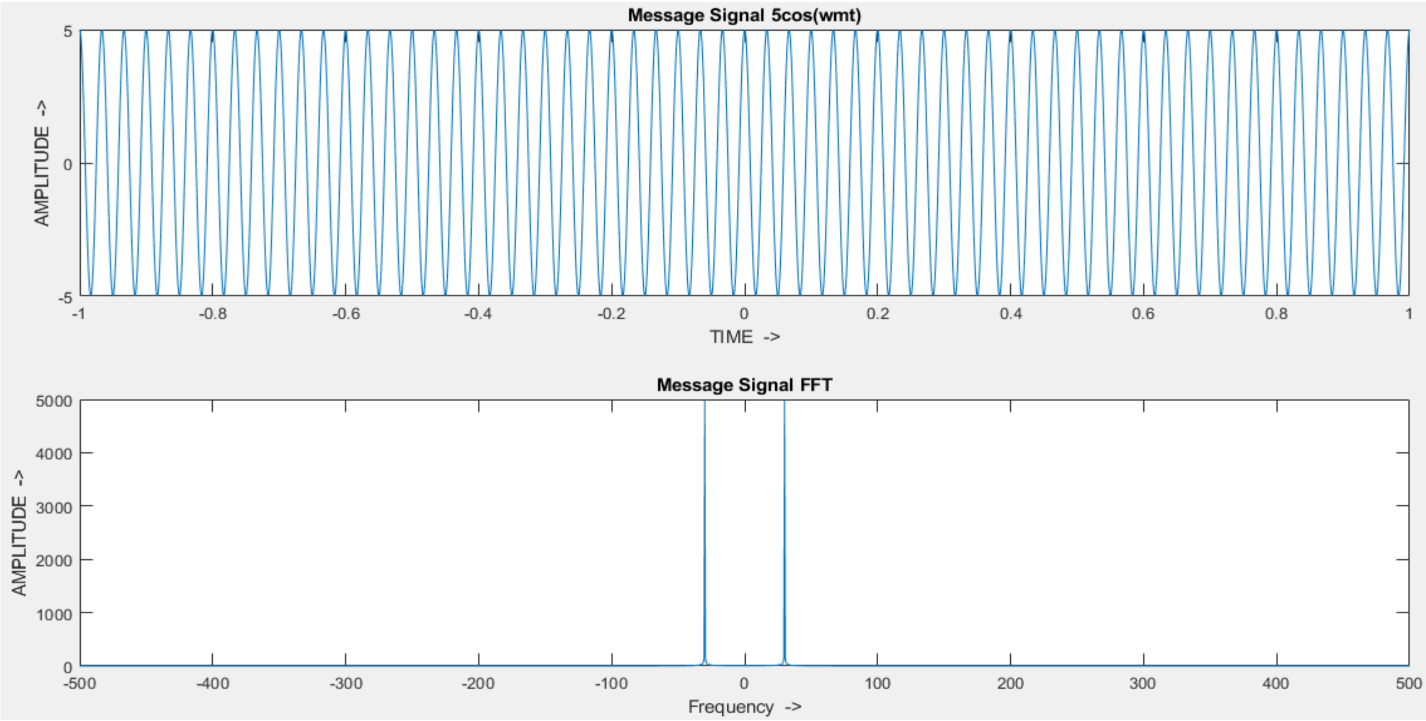
%<-----IF amplifier--- >%

```
fcut = IF-10;
[b,a] = butter(4,[450 460]/500,'bandpass');
out_band_pass     = filter(b,a,out_freq_mixer);
out_if_amp = magnitude*out_band_pass;

out_if_amp_fft = fftshift(fft(out_if_amp));

figure
subplot(2,1,1)
plot(time,out_if_amp)
xlabel('Time ->')
ylabel('AMPLITUDE  ->')
title('IF amplifier Output:')
subplot(2,1,2)
plot(freq_range,abs(out_if_amp_fft))
xlabel('Frequency ->')
ylabel('AMPLITUDE ->')
title('IF amplifier FFT')
```

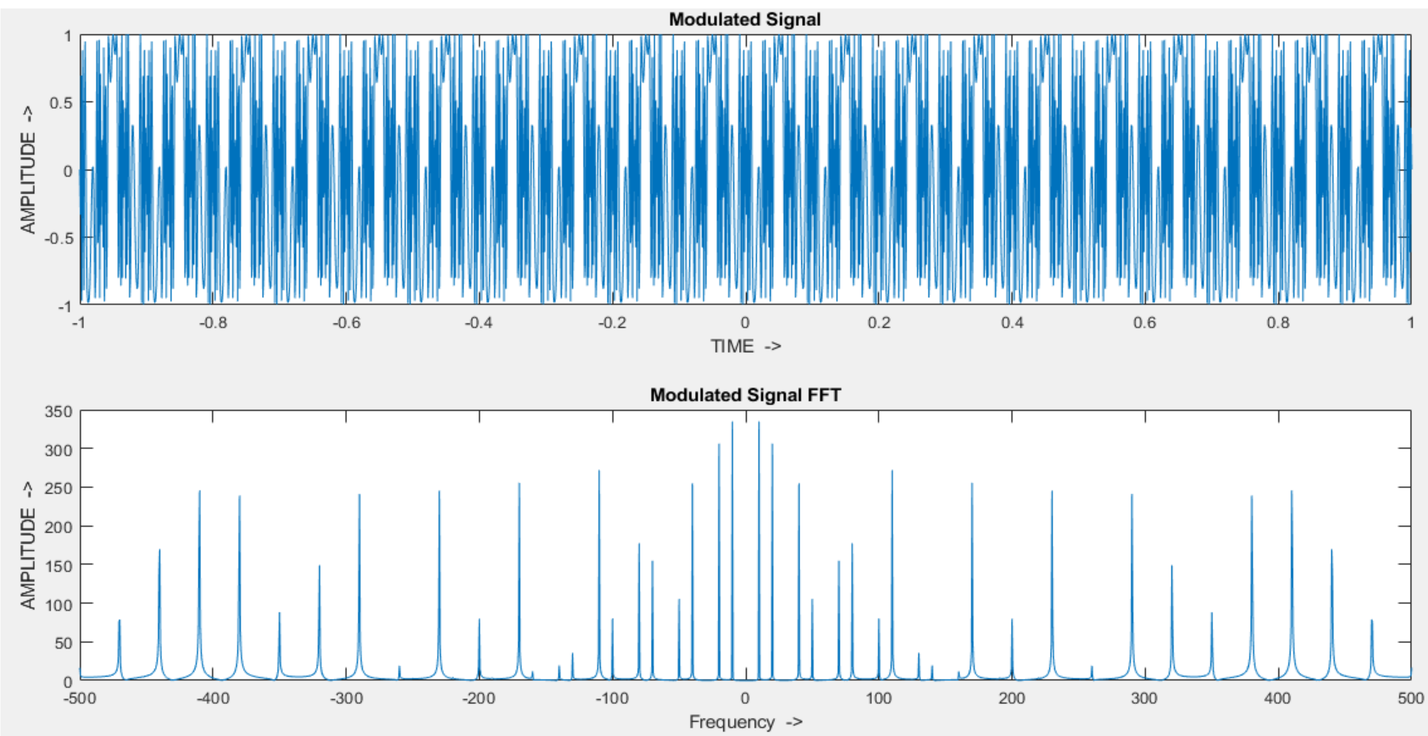
%<-----Demodulator --->%

```
[b,a] = butter(4,[28 45]/500,'bandpass');
out_band_pass     = filter(b,a,out_freq_mixer);
out_if_amp = magnitude*out_band_pass;
out_demodulator= out_if_amp*IF;


out_demodulator_fft = fftshift(fft(out_demodulator));


figure
subplot(2,1,1)
plot(time+100,out_demodulator)
xlabel('Time ->')
ylabel('AMPLITUDE ->')
title('Demodulator Output:')
subplot(2,1,2)
plot(freq_range,abs(out_demodulator_fft))
xlabel('Frequency ->')
ylabel('AMPLITUDE ->')
title('Demodulator FFT')
```

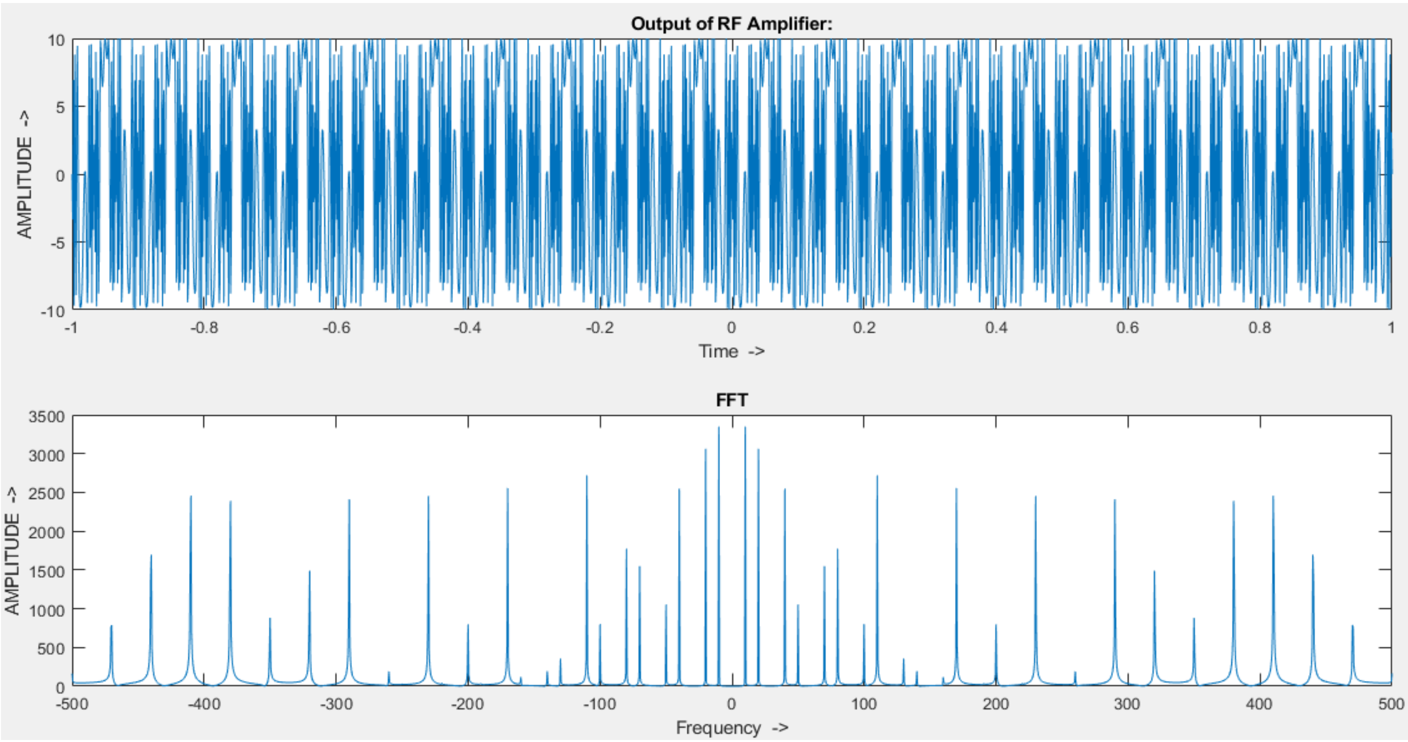
OBSERVATION:



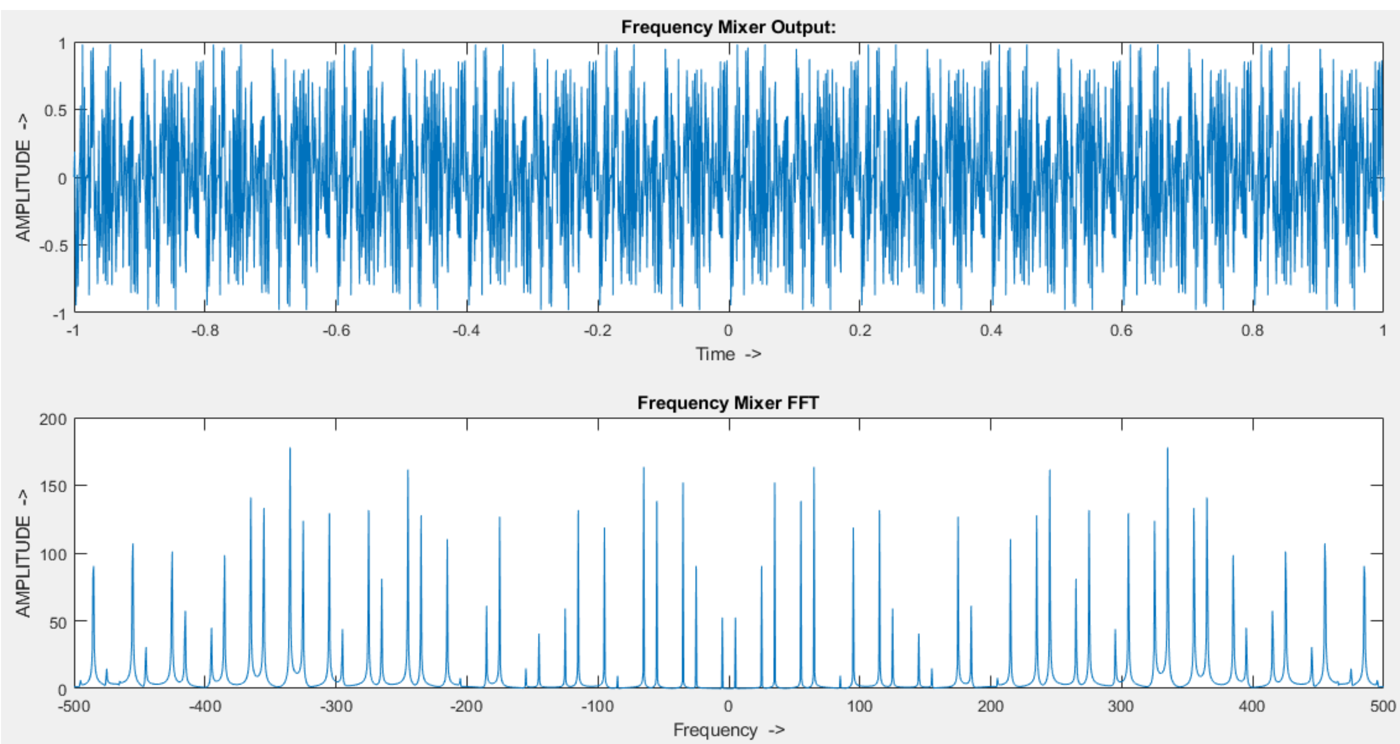
Message Signal and its FFT



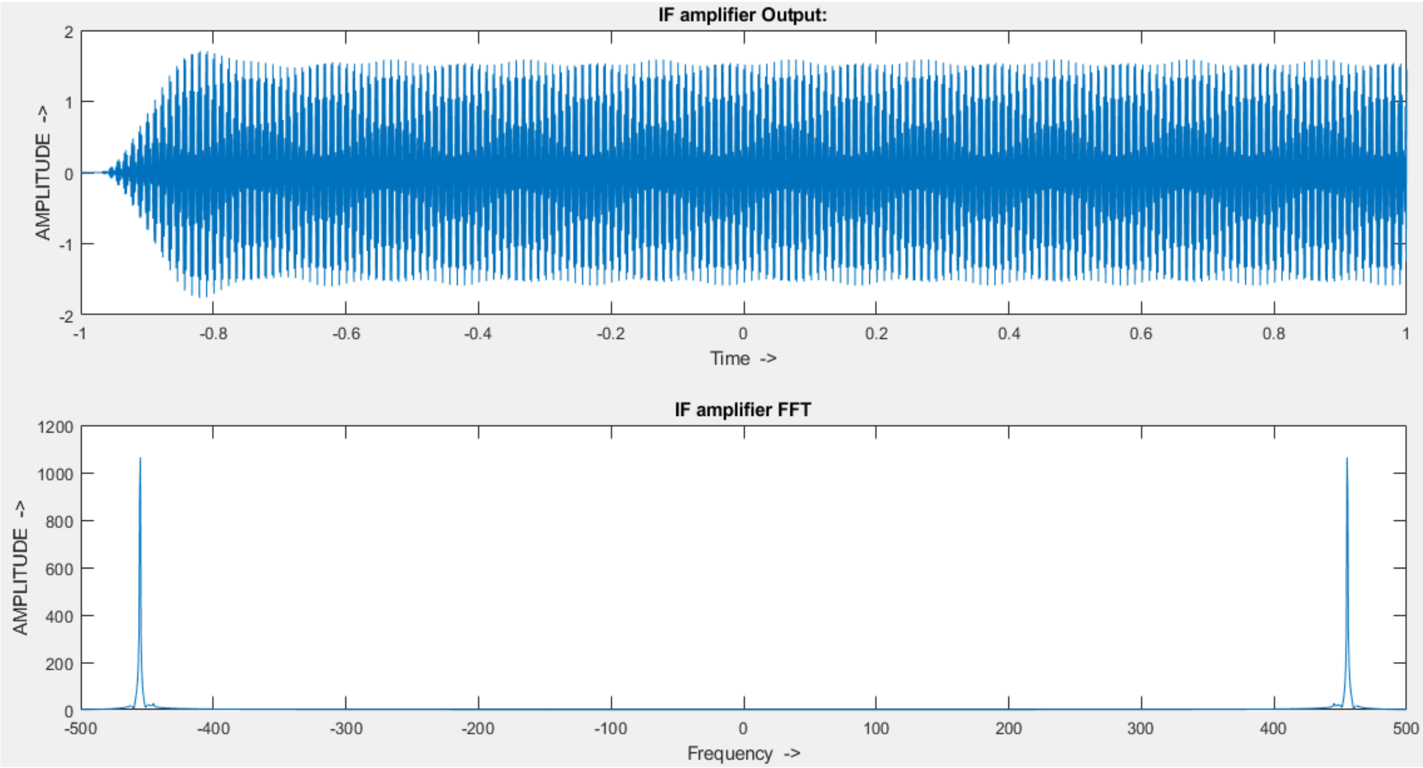
Modulated Signal and its FFT



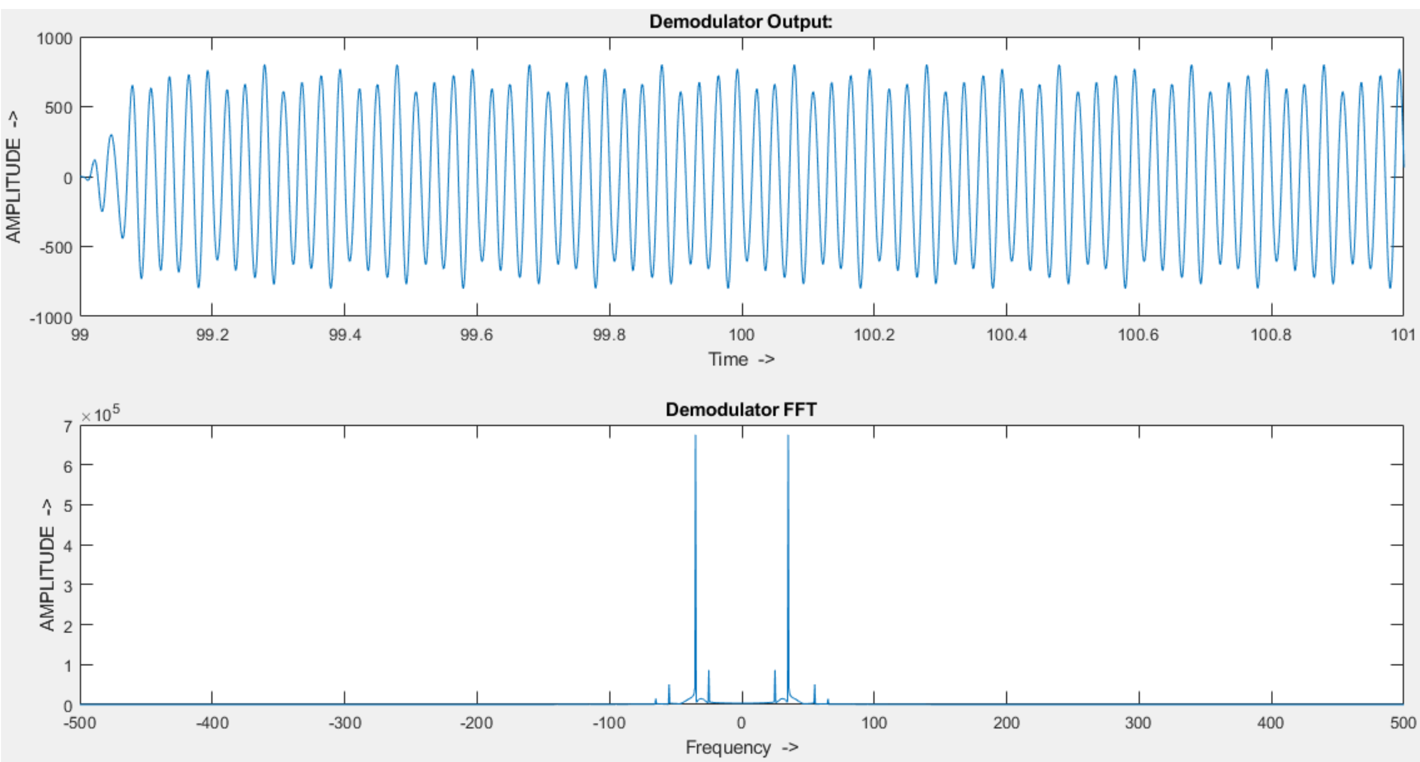
Output of RF Amplifier and its FFT



Frequency Mixer Output and its FFT



IF Amplifier Output and its FFT



Demodulated Wave and its FFT

CONCLUSION:

In this experiment, we learnt how to implement a Superheterodyne Receiver through a MATLAB code. We observed the outputs at each amplifier stages and also observed the demodulated wave is almost the same as the message signal.

SIGNATURE

classmate
Date
Page

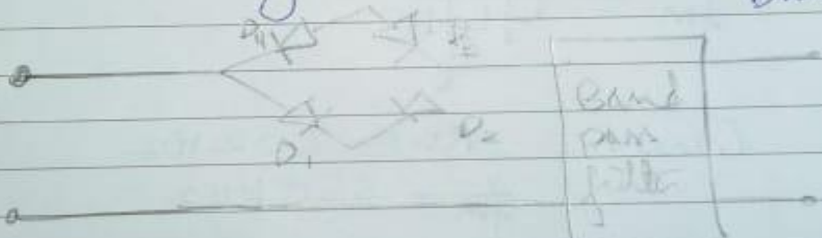
Experiment - 11

Aim :- Design a bandpass filter to allow DSB signal to pass from the given ring modulator.

Apparatus :- Ring modulator & Band pass filter.

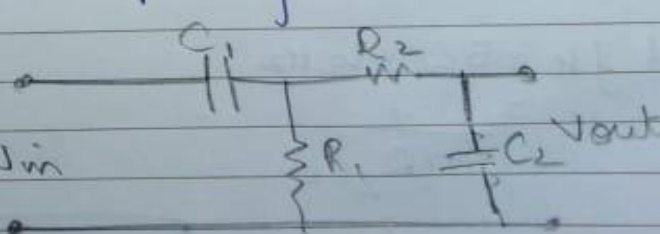
Circuit diagram :-

$f_c = 550\,kHz$
$BW = 20\,kHz$

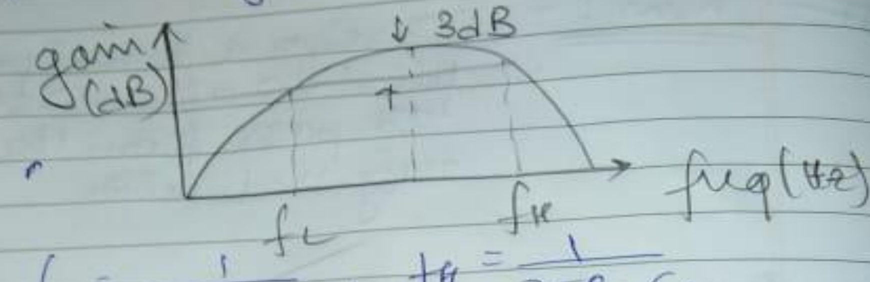


Calculations :-

Band pass filter :-



Output graph



$f_L = \frac{1}{2\pi R_1 C_1}$, $f_H = \frac{1}{2\pi R_2 C_2}$

$B.W = f_H - f_L$

$f_m = \sqrt{f_L f_H}$

Given:- $BW = 20\,kHz$

$f_m = 550\,kHz$.

$\therefore f_L \cdot f_H = 302500\,MHz$

$f_H - f_L = 20\,kHz$

at $f_H = 560\,kHz$

$= \frac{1}{2\pi R_2 C_2}$

$\therefore R_2 C_2 = \frac{1}{2\pi(560 \times 10^3)}$, $\boxed{R = 3\,k\Omega \quad C = 9.473 \times 10^{-11}\,F}$

$$f_L = 540\,kHz = \frac{1}{2\pi R_1 C_1}$$

$$\therefore R_1 C_1 = \frac{1}{2\pi(540 \times 10^3)}$$

for $\boxed{R_1 = 3k\,\Omega \quad C_P = 9.824 \times 10^{-11}\,F}$

Results :-

- $R_1 = 3k\,\Omega$
- $R_2 = 3k\,\Omega$
- $C_1 = 9.824 \times 10^{-11} F$
- $C_2 = 9.473 \times 10^{-11} F$

Conclusion :-

⇒ We can theoretically have infinte different combinations here but low values of resistance aren't possible so we choose 3k Ω as it is available.

SIGNATURE