MANIPAL INSTITUTE OF TECHNOLOGY
MANIPAL
A constituent institution of Manipal University

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGG.

CERTIFICATE

This is to certify that Ms./Mr. ……………………...…………………………………

Reg. No.: …..…………………… Section: …………… Roll No.: ………………

has satisfactorily completed the lab exercises prescribed for Communication lab of 3rd year B.Tech., Degree at MIT Manipal, in the academic year 2022-2023.

Date: ……......................................

Signature
Faculty in Charge

Signature
Head of the Department

CONTENTS

Course Outcomes

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

CO1: Measure characteristics of microwave oscillators.

CO2: Measure the parameters of passive microwave components.

CO3: Experimentally demonstrate digital modulation and demodulation schemes using MATLAB and LabView.

CO4: Design and characterize microstrip antennas and Transmission/Reception using NI-USRP 2901.

Evaluation Plan

- **Internal Assessment Weightage:** 60% (60 Marks)
  - Continuous evaluation component (for each experiment):10 marks (conduction 5 marks, Journal 3 marks and Exercise 2 marks).
  - The assessment will depend on punctuality, preparation, conduction, maintaining the observation note and answering the questions in viva voce.
- **End Semester Assessment Weightage:** 40% (40 Marks)
- **Total Weightage:** 100% (100 Marks)

INSTRUCTIONS TO THE STUDENTS

1. Students must come with Laboratory Manual, Observation book and Journal to every laboratory session.
2. Students must come with polar graph sheets for Experiment 5 and Smith chart for Experiment 3 to the laboratory.
3. Students shall enter the laboratory in time and follow Institution dress code.
4. Students must show the results to respective faculties and get it signed before leaving the lab.
5. For Hardware experiments, the circuits must be verified by the faculty before switching on the supply.
6. The students must take permission from respective faculties to go out of the lab.

Course Plan

COMMUNICATION LABORATORY

Sl. No.	Name of Experiment	Timeline
1	*Measurement of Low VSWR and Power Divider*	WEEK 1
2	*Microstrip Antennas.*	WEEK 2
3	*Microstrip Components.*	WEEK 3
4	**Generation of Rayleigh and Rician Fading Channels using MATLAB.**	WEEK 4
5	**BPSK Modulator and Demodulator. MATLAB**	WEEK 5
6	**8-PSK Mod-Demod and find the BER over AWGN channel. MATLAB**	WEEK 6
7	Introduction to **LabView** and USRP 2901: Design Examples.	WEEK 7
8	Transmission and Reception (Real Time Audio) using USRP 2901. Modulation and Detection of SISO and MIMO.	WEEK 8
9	HFSS Tool	WEEK 9
10	Final End Sem Exam	WEEK 10

EXPERIMENT -1
MEASUREMENT OF LOW VSWR and Understanding Power Divider

Aim

To measure low VSWR.

Apparatus

Klystron power supply, klystron mount, isolator, frequency meter, attenuator, slotted line moving carriage detector, matching load (horn antenna), BNC, Oscilloscope, VSWR meter.

Block Diagram:

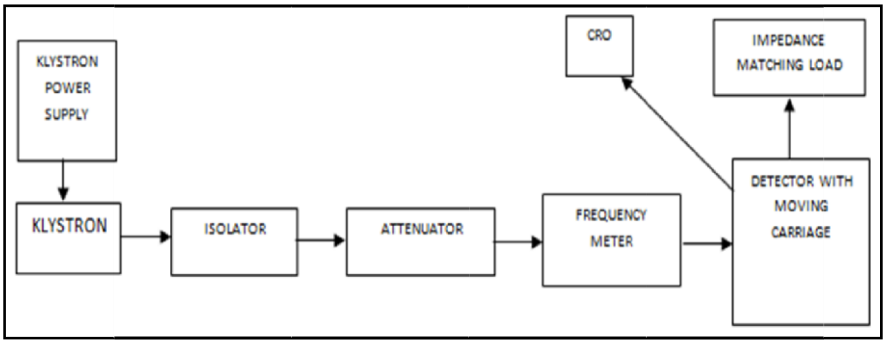


Figure 1: VSWR measurement set up **Procedure**

1. Before switching on, keep beam voltage to minimum and Repeller voltage to maximum on Klystron power supply. Switch ON the power supply and wait for the power supply unit to warm up.

Low VSWR

1. Connect the components as shown in block diagram 1.
2. Before turning the Klystron supply, ensure all knobs are at zero position.
3. Select AM knob on Klystron Supply.
4. Adjust amplitude, frequency knobs of AM and moving carriage to get undistorted square wave of maximum amplitude on Oscilloscope, if required adjust Repeller voltage.
5. Slide the moving carriage to get maximum and minimum amplitudes i.e. V_{max} and V_{min}, measure these amplitudes on Oscilloscope. Also note down their positions on moving carriage.
6. Calculate VSWR using $S= \sqrt{\dfrac{V_{max}}{V_{min}}}$

Low VSWR

Sl. No.		1	2	3
1	V_{max}			
2	V_{min}			
3	VSWR			

Result

Low VSWR =

B) Power Divider: Aim: To find the Resonance

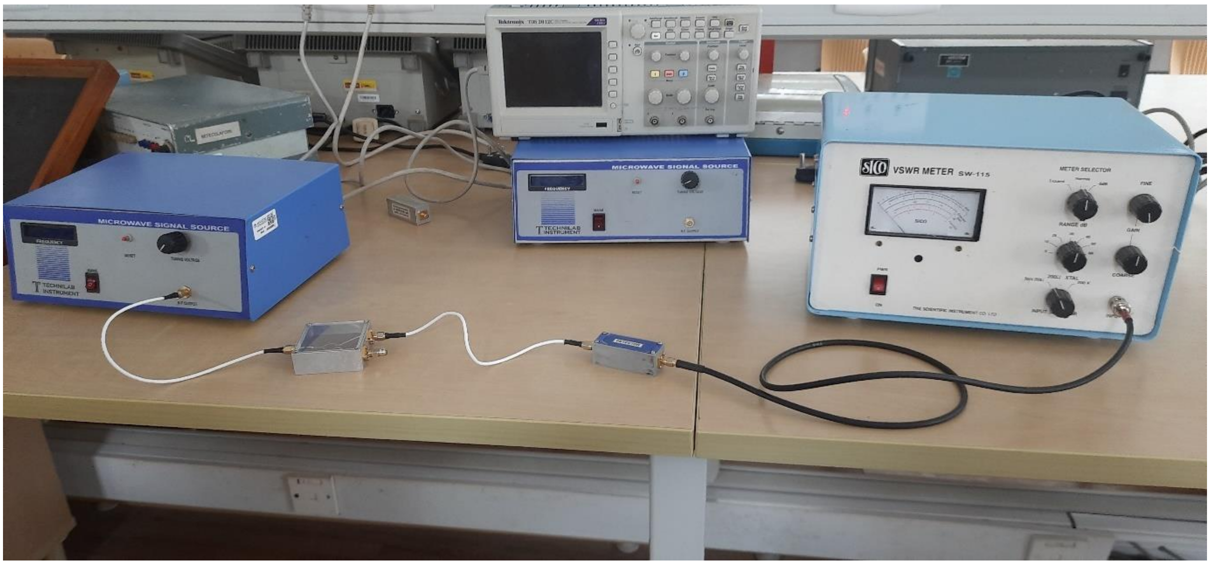


Figure 2 (A)

Figure 2 (B)

Procedure:

1. Set up the system as shown in Figure 2.
2. Measure the input power fed to port 1(P1) at different frequencies from 1.9GHz to 3.3GHz in steps of 0.1GHz by connecting VCO to CRO/VSWR meter through detector.
3. Measure the power divided output at port 2(P2) for the above set of frequencies by connecting port 2 to VSWR meter/CRO through detector and isolating the other port.
4. Similarly Measure the power divided at port 3(P3) for the above set of frequencies by connecting port 3 to VSWR meter/CRO through detector and isolating the other port. Find the resonance.

EXPERIMENT-2

MICROSTRIP ANTENNAS

Aim

To design a Microstrip antenna. Plot their radiation patterns and calculate Beam-width and Directivity.

Apparatus

Microstrip antennas, VCO, Detector, VSWR meter, CRO, Antenna stand.

Printed Dipole antenna:

Design:

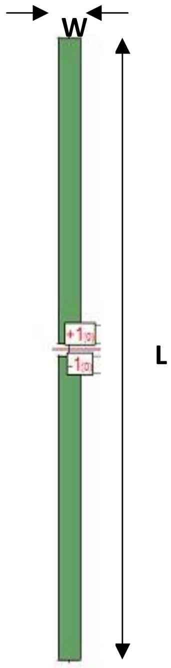


Figure 3: Dipole antenna

Calculate Length of the dipole 'L' by fixing width 'w' using the relation:

$$L + w = 0.48\left(\frac{\lambda_o}{\sqrt{\varepsilon_{eff}}}\right)$$

Measurement Setup:

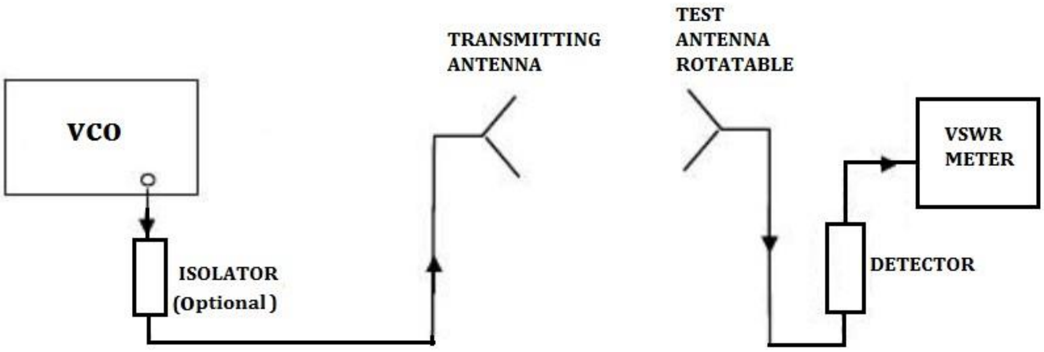


Figure 4: Radiation pattern measurement setup

Procedure:

1. Setup the system as shown in Figure 4, to obtain radiation patterns of different types of antenna.
2. Keeping the turning knob voltage at minimum and gain control knob at maximum, switch on the VCO.
3. Vary the tuning voltage and set the frequency to resonant frequency (f_o) of the antenna (around 2.5GHz). Calculate and keep minimum distance between transmitting and receiving antenna using the Fraunhofer distance formula S ___ λ where 'd' is the largest dimension of the antenna and $\lambda = c/f_o$.
4. Align the antenna in E-Plane by maintaining the Line of sight (zero-degree reference on the turn-table) between transmitting and receiving antennas and note down the readings on Oscilloscope/VSWR meter.
5. Rotate the turn-table in clockwise and anticlockwise directions up to 90° on either sides of reference in steps of 5 and note down the readings on Oscilloscope/VSWR meter.
6. Repeat the steps 4 and 5 by aligning the antenna in H-Plane.
7. Plot the radiation pattern on polar graph sheet.

Note: For E plane and H plane alignment of antenna, refer the figures 5 A and B.

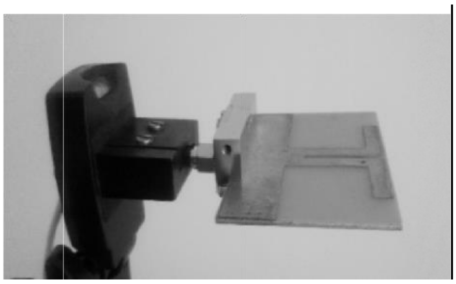


Figure A

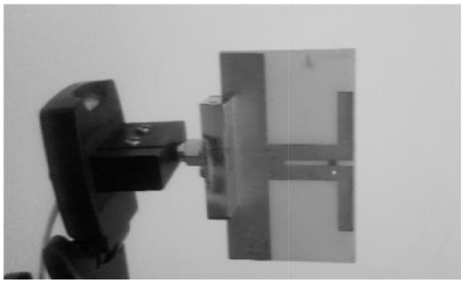


Figure B

Figure 5 A: E plane alignment of printed Dipole antenna.

Figure 5 B: H plane alignment of printed Dipole antenna

Plots:

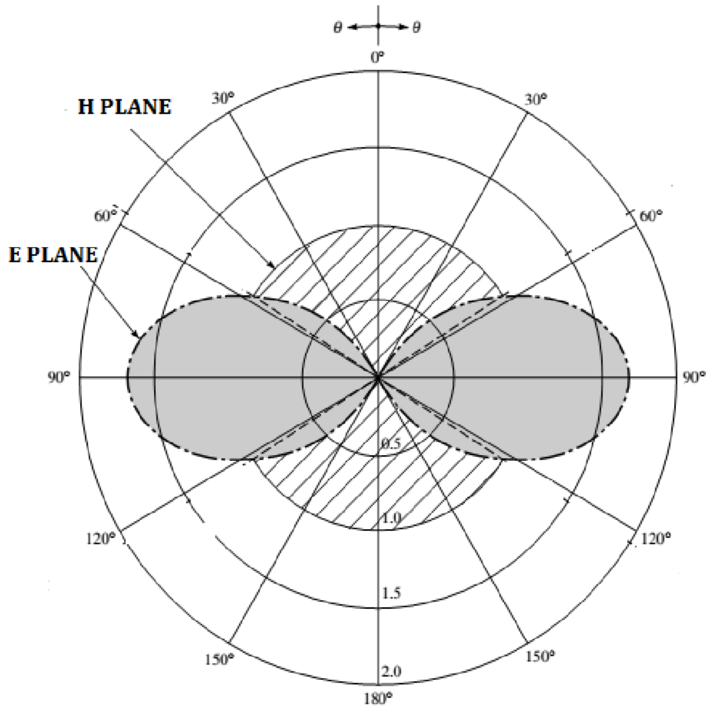


Figure 6: Radiation pattern of Dipole antenna

Calculations:

1. Calculate 3 dB Beamwidth (HPBW) for E Plane (θ_E) and H Plane (θ_H) from the plot.

2. Calculate Directivity using the relation:

$$D = \frac{41253}{\theta_E \theta_H}$$

Exercise:

1. Design a printed dipole antenna for a resonant frequency of 4 GHz.

EXPERIMENT - 3
MICROSTRIP COMPONENTS

Aim

A. To design a Branch Line Coupler (BLC) and analyze its Coupling, Isolation and Directivity.

B. To plot the transmission characteristics of Ring resonator and determine the Quality factor and effective dielectric constant of the substrate.

Apparatus

Microstrip Components, VCO, Detector, VSWR meter, CRO, SMA Cables.

Measurement setup

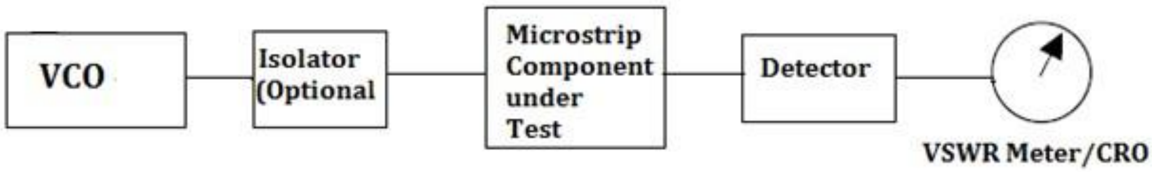


Figure 7: Measurement setup for Microstrip Components

A. Branch line coupler (BLC):

Theory:

A two-stub branch line coupler is a fundamental direct-coupled structure in which the main line is directly bridged to the secondary line by means of two-shunt branches as shown in the Figure 8.2. The length L of each branch and their spacing are all quarter wavelength $\lambda_0/4$ in the transmission medium at the center frequency f_0. Z_a and Z_b denote the characteristic impedances of the series branch and shunt branch respectively. All input and output lines have the same characteristic impedance Z_0.

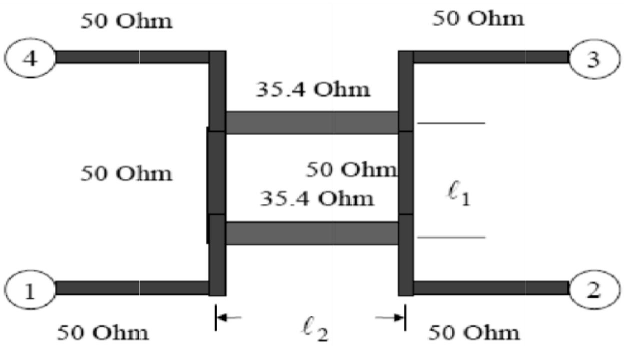


Figure 8: Branch line coupler

Design:

Center Frequency; f_o = 5GHz.
Mid band Coupling; C =3 dB.
Source and Load Impedances; $Z_o = 50\Omega$.

Since the coupling is given as 3-dB, we can write

$$-20\log S_{12} = -20\log S_{13} = 3$$

$$S_{12} = S_{13} = \frac{1}{\sqrt{2}}$$

Since $Z_o = 50\Omega$, impedance of the series and shunt branch of the coupler is:

$$Z_a = \frac{Z_O}{\sqrt{2}} = \frac{50}{\sqrt{2}} = 35.4\Omega; Z_b = Z_O = 50\Omega$$

To get the final layout in microstrip, we need to convert the impedances and electrical lengths into physical dimensions. The final dimensions of the branch line coupler for the substrate parameter ε_r=3.2, h=0.762 mm are:

For $Z_0=Z_b$=50ohmline, w=1.834 mm, ε_{eff1}=2.584, $L = \frac{\lambda}{4\sqrt{\varepsilon_{eff1}}} = 9.322mm$

For $Z_0=Z_b$=35.4ohmline, w=3.048 mm, ε_{eff2}=2.697, $L = \frac{\lambda}{4\sqrt{\varepsilon_{eff2}}} = 9.322mm$

Procedure:

Coupling (S_{31}):

1. Set up the system as shown in Figure 8.
2. Measure the input power fed to port 1(P1) at different frequencies from 2GHz to 3GHz in steps of 0.1GHz by connecting VCO to CRO/VSWR meter through detector.
3. Terminate ports 2 and 4 with 50-ohm matched loads.
4. Measure the power coupled at port 3(P3) for the above set of frequencies by connecting port 2 to VSWR meter/CRO through detector.
5. Plot Coupling: C = -10 log (P3/P1) = $P1_{db}$-$P3_{db}$ as a function of frequency.
6. Calculate the bandwidth of the coupler over which Coupling is 3±1dB.

Isolation (S_{41}) and Directivity (S_{43}):

1. Terminate ports 2 and 3 with 50-ohm matched loads.
2. Measure the power coupled at port 4(P4) for the above set of frequencies by connecting port 4 to VSWR meter/CRO through detector.
3. Plot Isolation = -10 log (P4/P1) = $P1_{db}$-$P4_{db}$ as a function of frequency.
4. Plot Directivity = Isolation – Coupling as a function of frequency.

Plot:

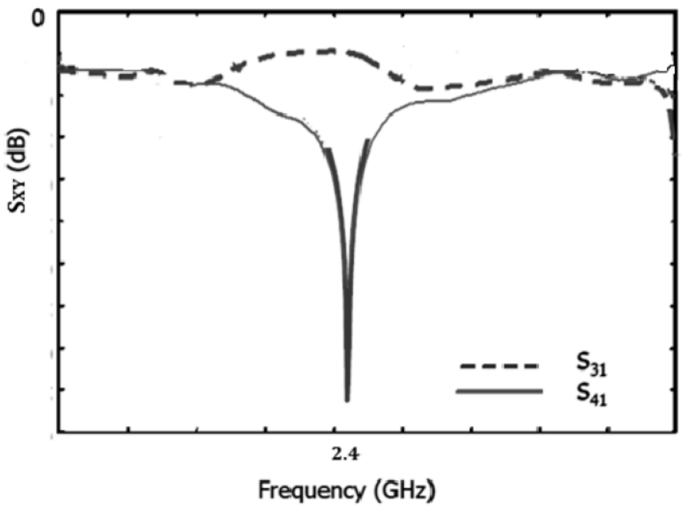


Figure 9: Characteristics of Microstrip Branch line coupler

B. Ring resonator:

Theory:

The ring resonator is known as a simple printed resonator that is useful for making approximate measurement of dielectric constant. Additionally, it is used in filters and to an extent in antennas. In principle, it is a simple structure, but accurate analysis of a ring resonator is difficult because of the input and output coupling to straight microstrip printed lines. Two structures shown in Figure 10, are Ring resonator coupled with (a) open-ended lines and (b) with coupling arcs.

Looking at a ring resonator in isolation, it may appear that the field would be in the form of a wave circulating around in either direction, but in reality, the coupling structure plays a very important role.

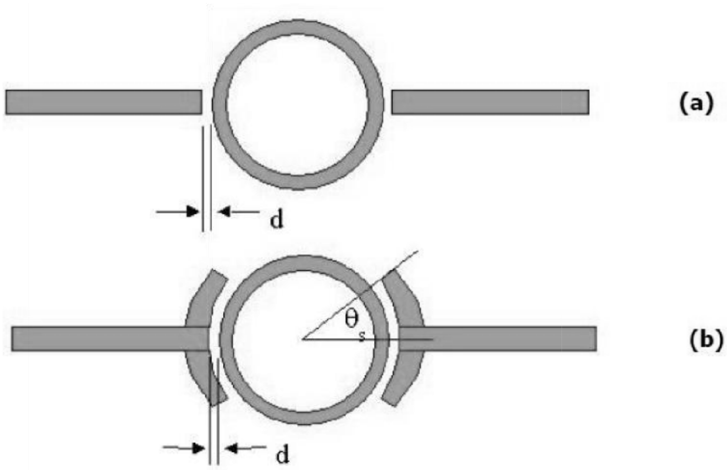


Figure 10: Ring resonator coupled with (a) open-ended lines and (b) with coupling arcs

It may be noticed that both the structures are symmetrical. It follows that whatever voltage wave is excited in the clockwise direction, an identical voltage wave will be excited in the anticlockwise direction as well. This gives rise to the standing wave pattern, common to resonators. The voltage maxima in both cases are located at the center of the coupled section. Looking at this behavior, it may be more accurate to look upon the ring resonator as two half-wave transmission line resonators (open ended) connected in parallel. The coupling structures disturb these open ends and shift the

resonant frequency slightly. It is apparent that as the coupling is reduced, the frequency selectivity of the resonator increases, but the losses also increase drastically. So one has to arrive at a compromise – sufficient coupling so that losses are not excessive, yet the Centre frequency is identifiable with sufficient precision.

Procedure:

5. Set up the system as shown in Figure 8.
6. Measure the input power fed to port 1(P1) at different frequencies from 2GHz to 3GHz in steps of 0.1GHz by connecting VCO to CRO/VSWR meter through detector.
7. Measure the power coupled at port 2(P2) for the above set of frequencies by connecting port 2 to VSWR meter/CRO through detector.
8. Plot Insertion loss = -10 log (P2/P1) = $P1_{db}$-$P2_{db}$ as a function of frequency.
9. From the plot, determine resonant frequency and bandwidth.

Plot:

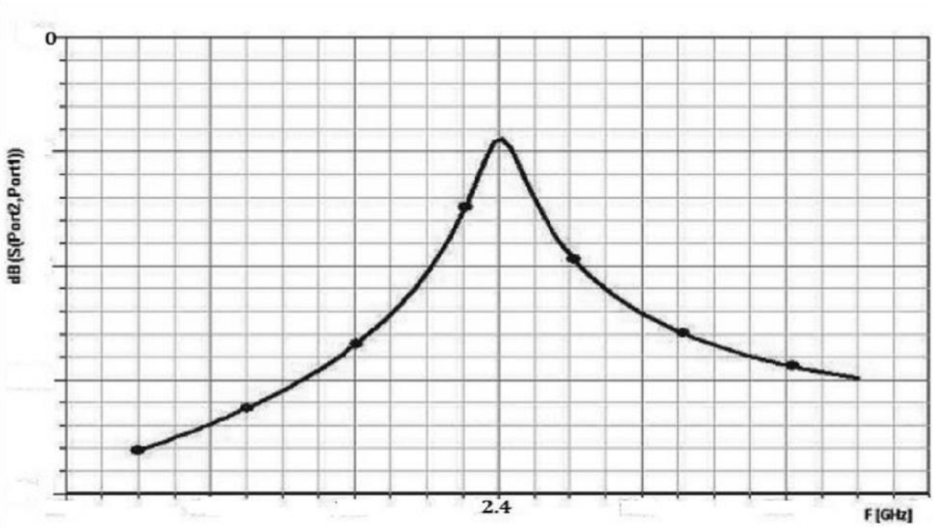


Figure 11: Transmission characteristics of a ring resonator with coupling

Calculations:

Calculate Q factor, Q = f_o/B.W

Exercise: 1. Design a ring resonator for a resonant frequency of 6 GHz.

EXPERIMENT - 4

Generation of Rayleigh and Rician Fading Channels using MATLAB.

Rayleigh Fading and Rayleigh Distribution

THEORY:

The delays associated with different signal paths in a multipath fading channel change in an unpredictable manner and can only be characterized statistically. When there are a large number of paths, the central limit theorem can be applied to model the time-variant impulse response of the channel as a complex-valued Gaussian random process. When the impulse response is modeled as a zero-mean complex-valued Gaussian process, the channel is said to be a Rayleigh fading channel. The model behind Rician fading is similar to that for Rayleigh fading, except that in Rician fading a strong dominant component is present. This dominant component can for instance be the line-of-sightwave. Here the Rayleigh Fading model is assumed to have only two multipath components X(t) and Y(t).Rayleigh Fading can be obtained from zero-mean complex Gaussian processes (X(t) and Y(t)).Simply adding the two Gaussian Random variables and taking the square root (envelope) gives a single tap Rayleigh distributed process. The phase of such random variable follows uniform

distribution.

Consider two Gaussian random variables with zero mean and same variance

$\boldsymbol{X \sim N(0,\sigma^2)}$ and $\boldsymbol{Y \sim N(0,\sigma^2)}$.

Let's define a complex Gaussian random variable (to mimic IQ channel) as

$$Z = X + jY$$

Now, the envelope of the complex random variable is given by

$$R = \sqrt{X^2 + Y^2}$$

And the phase is given by

$$\varphi = tan^{-1}\left(\frac{Y}{X}\right)$$

The envelope follows Rayleigh distribution and the phase will be uniformly distributed. The probability density function (Rayleigh distribution) of the above-mentioned amplitude response is given by :

$$f(r) = \frac{r}{\sigma^2}\exp\left(-\frac{r^2}{2\sigma^2}\right)$$

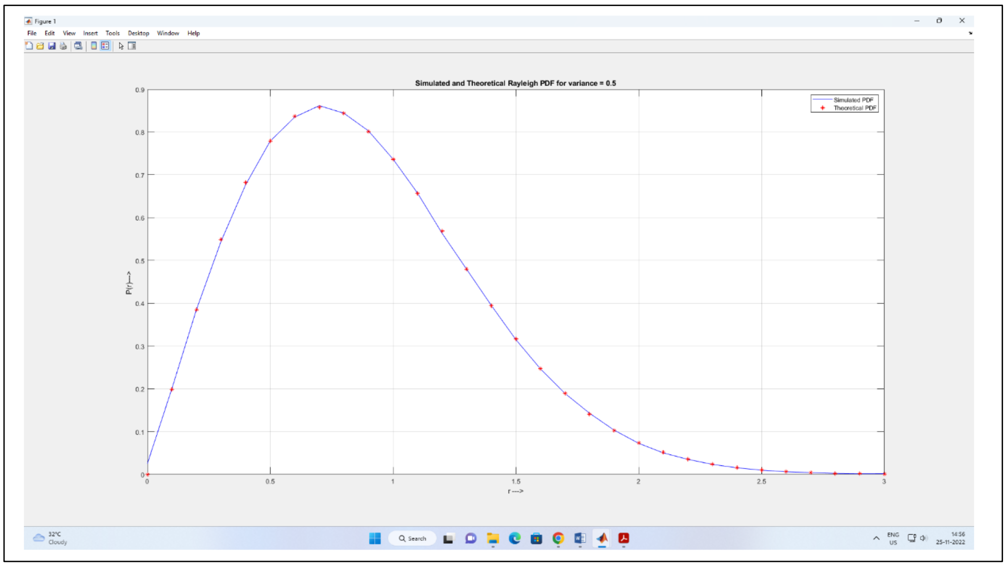
MATLAB CODE:

```
%----Rayleigh_PDF-----------------------------------------
%----------Input Section----------------
N=1000000; %Number of samples to generate
variance = 0.5; % Variance of underlying Gaussian random variables
%--------------------------------------
%Independent Gaussian random variables with zero mean and unit variance
x = randn(1, N);
y = randn(1, N);
%Rayleigh fading envelope with the desired variance
```

```
r = sqrt(variance*(x.^2 + y.^2));
%Define bin steps and range for histogram plotting
step = 0.1;
range = 0:step:3;
%Get histogram values and approximate it to get the pdf curve
h = hist(r, range);
approxPDF = h/(step*sum(h)); %Simulated PDF from the x and y samples

%Theoritical PDF from the Rayleigh Fading equation
theoretical = (range/variance).*exp(-range.^2/(2*variance));
plot(range, approxPDF,'b', range, theoretical,'r*');
title('Simulated and Theoretical Rayleigh PDF for variance = 0.5')
legend('Simulated PDF','Theoretical PDF')
xlabel('r --->');
ylabel('P(r)---> ');
grid;
%PDF of phase of the Rayleigh envelope
theta = atan(y./x);
figure(2)
hist(theta); %Plot histogram of the phase part
%Approximate the histogram of the phase part to a nice PDF curve
[counts,range] = hist(theta,100);
step=range(2)-range(1);
%Normalizing the PDF to match theoretical curve
approxPDF = counts/(step*sum(counts)); %Simulated PDF from the x and y samples
bar(range, approxPDF,'b');
hold on
plotHandle=plot(range, approxPDF,'r');
set(plotHandle,'LineWidth',3.5);
axis([-2 2 0 max(approxPDF)+0.1])
hold off
title('Simulated PDF of Phase of Rayleigh Distribution ');
xlabel('\theta --->');
ylabel('P(\theta) --->');
grid;
```

Figure:12: Theoretical PDF



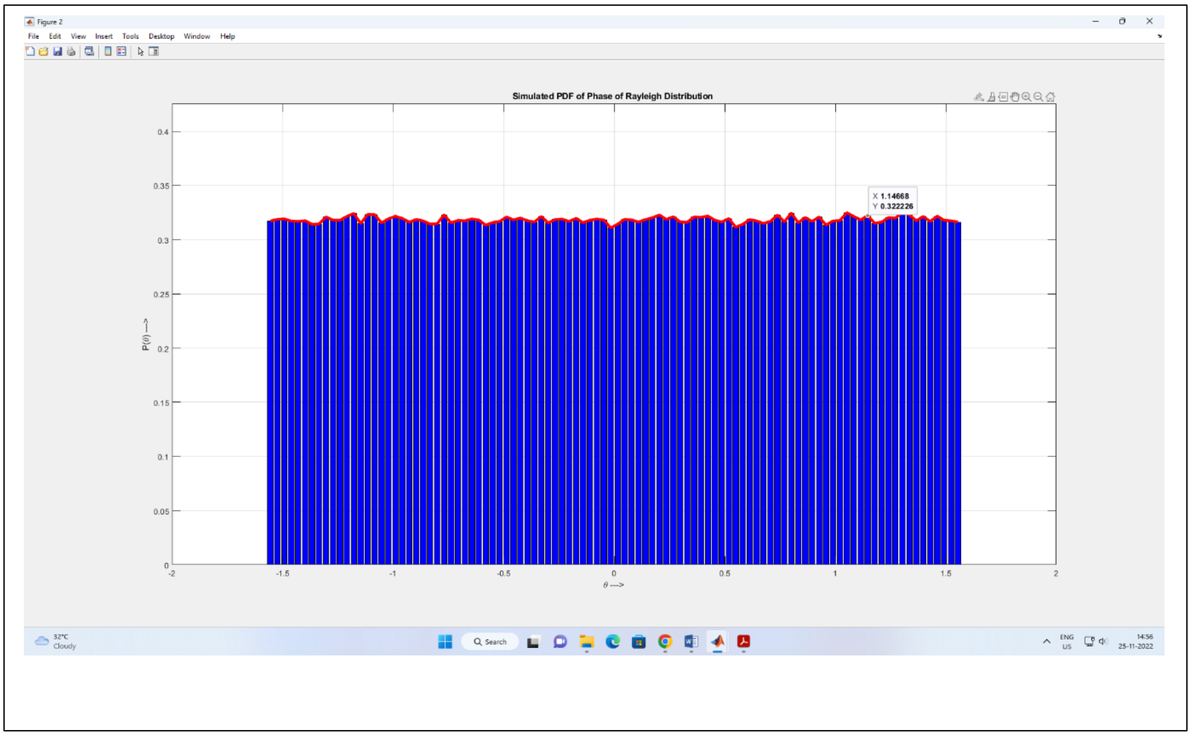


Figure:13: Phase of the Rayleigh Distribution.

Rician Fading and Rician Distribution:

The model behind Rician fading is similar to that of Rayleigh fading, except that in Rician fading a strong dominant component is present. This dominant component can for instance be the line-of-sight wave. This is modeled by using two Gaussian random variables – one with zero mean and another with non-zero mean.

Consider two Gaussian random variables with zero mean and same variance

$\boldsymbol{X \sim N(m1, \sigma^2)}$ and $\boldsymbol{Y \sim N(m2, \sigma^2)}$.

Let us define a complex Gaussian random variable (to mimic IQ channel) as

$Z = X + jY$

Now, the envelope of the complex random variable is given by:

$$R = \sqrt{X^2 + Y^2}$$

And the phase is given by

$$\varphi = tan^{-1}\left(\frac{Y}{X}\right)$$

And the phase is given by:

$$s = \sqrt{m_1^2 + m_2^2}$$

The non-centrality parameter (the imbalance in the means) is caused by the presence of dominant path in a Rician Fading environment. Due to this, the Rician K factor - representing the ratio of power of Line-Of-Sight (LOS) (or dominant multipath component) and the power of Non-Line-Of-Sight (NLOS) (or the remaining multipath components) is defined in such scenario.

$$\kappa = \frac{Power\ of\ LOS\ component}{Power\ of\ NLOS\ components}$$

Statistically, this can be represented as the power in the faded envelope that has been produced by the means of X and Y.

$$\kappa = \frac{m_1^2 + m_2^2}{2\sigma^2} = \frac{s^2}{2\sigma^2}$$

The envelope (R) follows Rician distribution, whose PDF is given by:

$$f_R(r) = \frac{r}{\sigma^2} exp\left(-\frac{r^2 + s^2}{2\sigma^2}\right) I_0\left(\frac{rs}{\sigma^2}\right)$$

Where, Io(x) is the modified zeroth order Bessel function of the first kind.

Matlab Code:

```
N=100000; %Number of Samples
sigma=1; %Variance of underlying Gaussian Variables
s=[0 1 2 4]; %Non-Centrality Parameter
plotStyle={'b-','r-','k-','g-'};
%Simulating the PDF from two Gaussian Random Variables
for i = 1: length(s)
X = s(i) + sigma.*randn(1,N); %Gaussian RV with mean=s and given sigma
Y = 0 + sigma.*randn(1,N); %Gaussian RV with mean=0 and same sigma as Y
Z=X+1i*Y;
[val,bin]=hist(abs(Z),1000); % pdf of generated Raleigh Fading samples
plot(bin,val/trapz(bin,val),plotStyle{i}); %Normalizing the PDF to match theoretical result
%Trapz function gives the total area under the PDF curve. It is used as the normalization factor
hold on;
end
%Theoretical PDF computation
for i=1:length(s)
x=s(i);
m1=sqrt(x);
m2=sqrt(x*(x-1));
r=0:0.01:9;
ss=sqrt(m1^2+m2^2);
x=r.*ss/(sigma^2);
f=r./(sigma^2).*exp(-((r.^2+ss^2)./(2*sigma^2))).*besseli(0,x);
plot(r,f,plotStyle{i},'LineWidth',2.5);
legendInfo{i} = ['s = ' num2str(s(i))];
```

```
hold on;
end
legend(legendInfo);
```

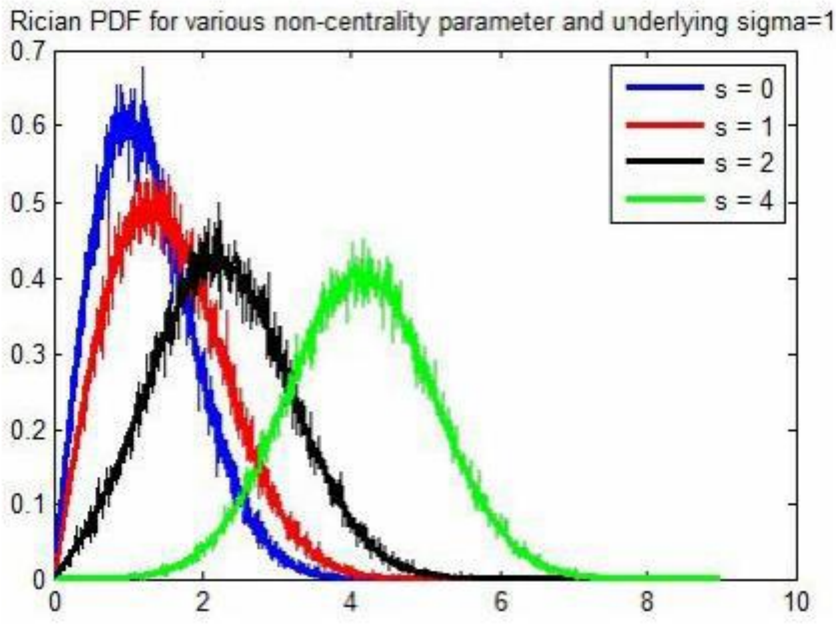


Figure:14

EXPERIMENT - 5

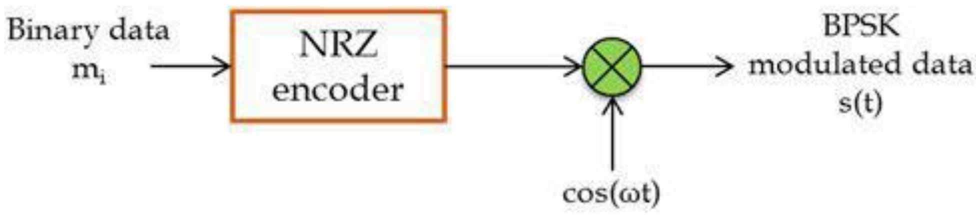
BPSK Modulator and Demodulator over AWGN Channel [1]:

BPSK MODULATOR:

In Binary Phase Shift Keying (BPSK) only one sinusoid is taken as basis function modulation. Modulation is achieved by varying the phase of the basis function depending on the message bits. The following equation outlines BPSK modulation technique.

$$S_0(t) = A\cos(\omega t) \qquad \Rightarrow represents\ '0'$$
$$S_1(t) = A\cos(\omega t + \pi) \qquad \Rightarrow represents\ '1'$$

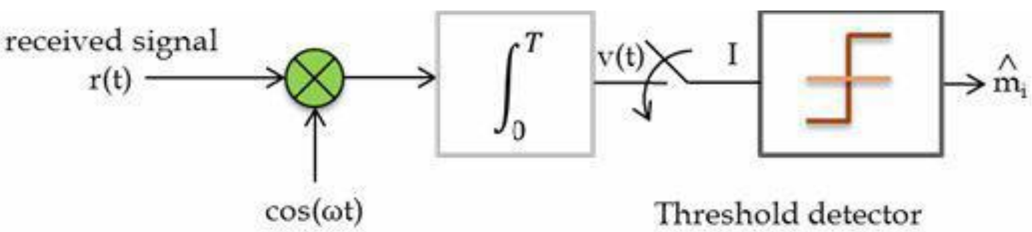
Here, A is the amplitude of the sinusoidal signal, is the carrier angular frequency measured in radians/second, t is the instantaneous time. S0(t) represents the carrier signal when '0' is transmitted and S1(t) represents the carrier signal when '1' was transmitted.
The constellation diagram of BPSK will show the constellation points lying entirely on the x axis. It has no projection on the y axis. This means that the BPSK modulated signal will have an in-phase component (I) but no quadrature component (Q). This is because it has only one basis function.



BPSK Modulator

BPSK DEMODULATOR:

For BPSK demodulator, a coherent demodulator is taken as an example. In coherent detection technique the knowledge of the carrier frequency and phase must be known to the receiver. This can be achieved by using a Costas loop or a Phase Lock Loop (PLL) at the receiver.
A PLL essentially locks to the incoming carrier frequency and tracks the variations in frequency and phase. In the demodulator the received signal is multiplied by a reference frequency generator (assuming the PLL/Costas loop is present). The multiplied output is integrated over one bit period using an integrator. A threshold detector makes a decision on each integrated bit based on a threshold. Since an NRZ signalling format is used with equal amplitudes in positive and negative direction, the threshold for this case would be '0.



BPSK Demodulator

This scheme uses a simple BPSK modulation scheme in which a random binary stream is represented in NRZ format and the resulting signal is multiplied by reference carrier frequency. AWGN channel noise generated according to the desired noise variance is added with the BPSKmodulated signal. The noise added signal is demodulated by multiplying it with the reference.

NRZ Encoder

MATLAB CODE

```
function [time,output,Fs]=NRZ_Encoder(input,Rb,amplitude,style)
Fs=16*Rb; %Sampling frequency ,
oversamplingfactor= 32;
Ts=1/Fs; % Sampling Period
Tb=1/Rb; % Bit period
output=[];
switch lower(style)
case {'manchester'}
for count=1:length(input)
for tempTime=0:Ts:Tb/2-Ts
output=[output (-1)^(input(count))*amplitude];
end
for tempTime=Tb/2:Ts:Tb-Ts
output=[output (-1)^(input(count)+1)*amplitude];
end
end
case {'unipolar'}
for count=1:length(input)
for tempTime=0:Ts:Tb-Ts
output=[output input(count)*amplitude];
end
end
case {'polar'}
for count=1:length(input)
for tempTime=0:Ts:Tb-Ts
output=[output amplitude*(-1)^(1+input(count))];
end
end
otherwise
disp('NRZ_Encoder(input,Rb,amplitude,style)-Unknown method given as "style" argument');
disp('Accepted Styles are "Manchester", "Unipolar" and "Polar"');
end
time=0:Ts:Tb*length(input)-Ts;
end
```

MATLAB CODE

```
%clear; %clear all stored variables
N=100; %number of data bits
noiseVariance = 0.5; %Noise variance of AWGN channel
data=randn(1,N)>=0; %Generate uniformly distributed random data
Rb=1e3; %bit rate
amplitude=1; % Amplitude of NRZ data
[time,nrzData,Fs]=NRZ_Encoder(data,Rb,amplitude,'Polar');
```

```
Tb=1/Rb;
subplot(4,2,1);
stem(data);
xlabel('Samples');
ylabel('Amplitude');
title('Input Binary Data');
axis([0,N,-0.5,1.5]);
subplot(4,2,3);
plotHandle=plot(time,nrzData);
xlabel('Time');
ylabel('Amplitude');
title('Polar NRZ encoded data');
set(plotHandle,'LineWidth',2.5);
maxTime=max(time);
maxAmp=max(nrzData);
minAmp=min(nrzData);
axis([0,maxTime,minAmp-1,maxAmp+1]);
grid on;
Fc=2*Rb;
osc = sin(2*pi*Fc*time);

%BPSK modulation
bpskModulated = nrzData.*osc;
subplot(4,2,5);
plot(time,bpskModulated);
xlabel('Time');
ylabel('Amplitude');
title('BPSK Modulated Data');
maxTime=max(time);
maxAmp=max(nrzData);
minAmp=min(nrzData);
axis([0,maxTime,minAmp-1,maxAmp+1]);

%plotting the PSD of BPSK modulated data
subplot(4,2,7);
h=spectrum.welch; %Welch spectrum estimator
Hpsd = psd(h,bpskModulated,'Fs',Fs);
plot(Hpsd);
title('PSD of BPSK modulated Data');
%-------------------------------------------

%Adding Channel Noise
%-------------------------------------------
noise = sqrt(noiseVariance)*randn(1,length(bpskModulated));
received = bpskModulated + noise;
subplot(4,2,2);
plot(time,received);
xlabel('Time');
ylabel('Amplitude');
title('BPSK Modulated Data with AWGN noise');
%-------------------------------------------
```

```
%BPSK Receiver
%-------------------------------------------
%Multiplying the received signal with reference Oscillator
v = received.*osc;
%Integrator
integrationBase = 0:1/Fs:Tb-1/Fs;
for i = 0:(length(v)/(Tb*Fs))-1,
y(i+1)=trapz(integrationBase,v(int32(i*Tb*Fs+1):int32((i+1)*Tb*Fs)));
end

%Threshold Comparator
estimatedBits=(y>=0);
subplot(4,2,4);
stem(estimatedBits);
xlabel('Samples');
ylabel('Amplitude');
title('Estimated Binary Data');
axis([0,N,-0.5,1.5]);
%------------------------------------------


%Bit Error rate Calculation
BER = sum(xor(data,estimatedBits))/length(data);
%Constellation Mapper at Transmitter side
subplot(4,2,6);
Q = zeros(1,length(nrzData)); %No Quadrature Component for BPSK
stem(nrzData,Q);
xlabel('Inphase Component');
ylabel('Quadrature Phase component');
title('BPSK Constellation at Transmitter');
axis([-1.5,1.5,-1,1]);

%constellation Mapper at receiver side
subplot(4,2,8);
Q = zeros(1,length(y)); %No Quadrature Component for BPSK
stem(y/max(y),Q);
xlabel('Inphase Component');
ylabel('Quadrature Phase component');
title(['BPSK Constellation at Receiver when AWGN Noise Variance
=',num2str(noiseVariance)]);
axis([-1.5,1.5,-1,1]);
```

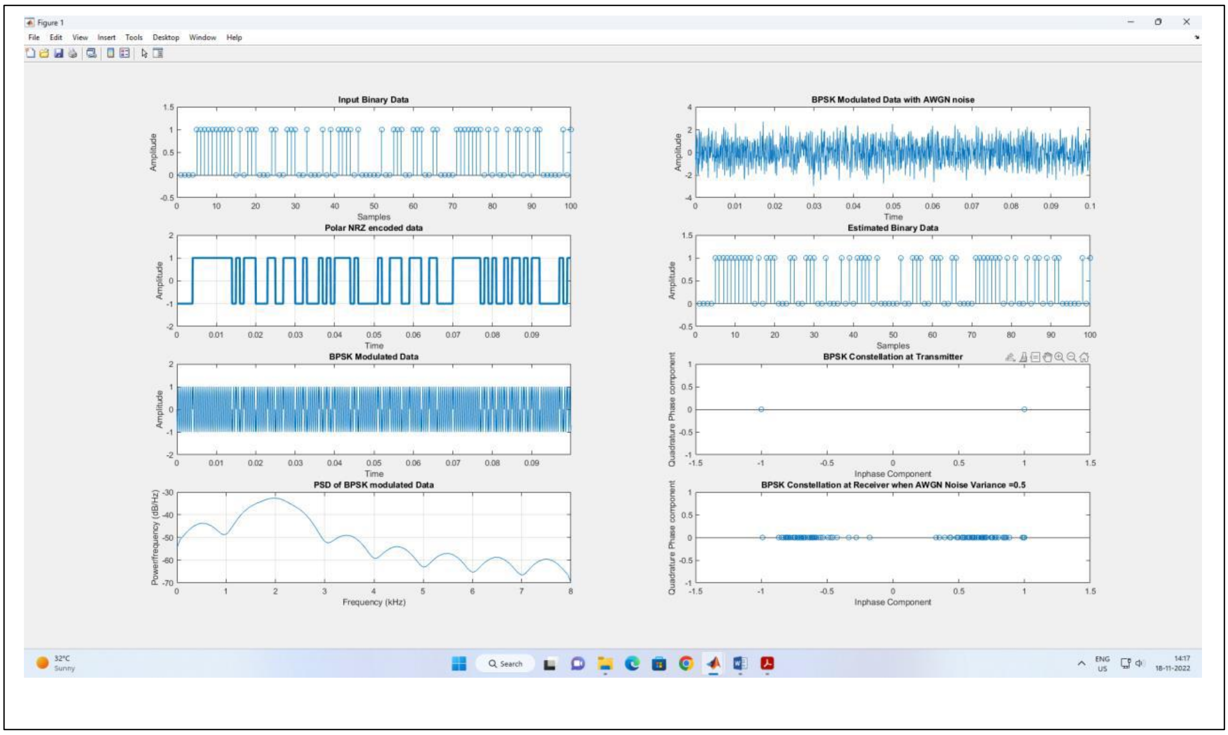


Figure:15.

Exercise:

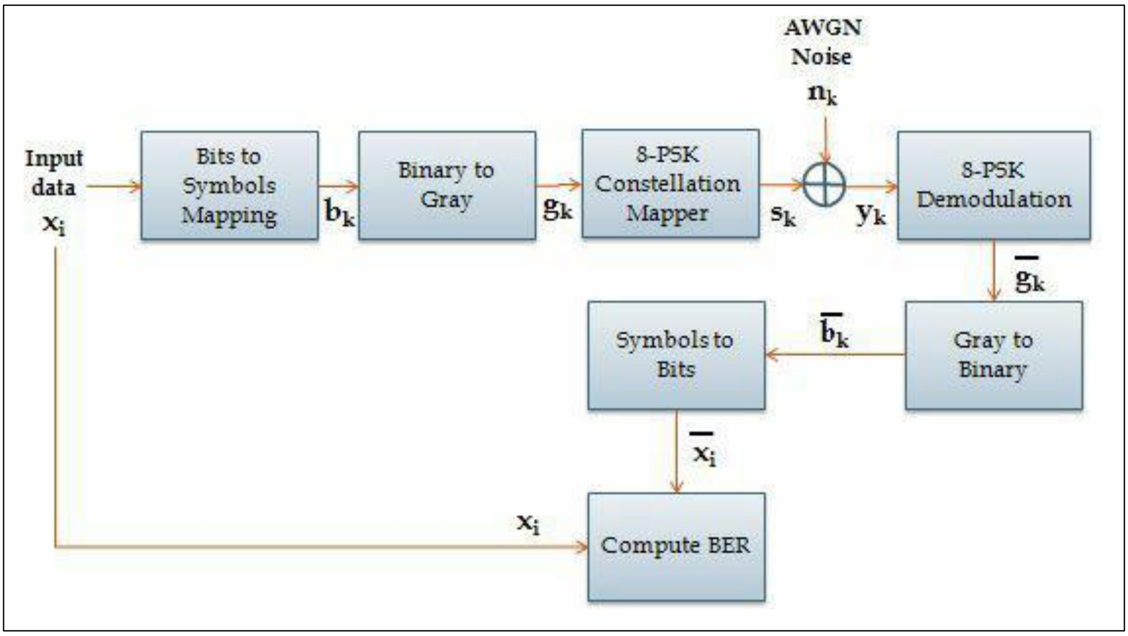
1. **Design a QPSK Modulator and Demodulator system and evaluate the BER over an AWGN Channel .**
2. **Design a FSK Modulator and Demodulator system and evaluate the BER over an AWGN Channel .**

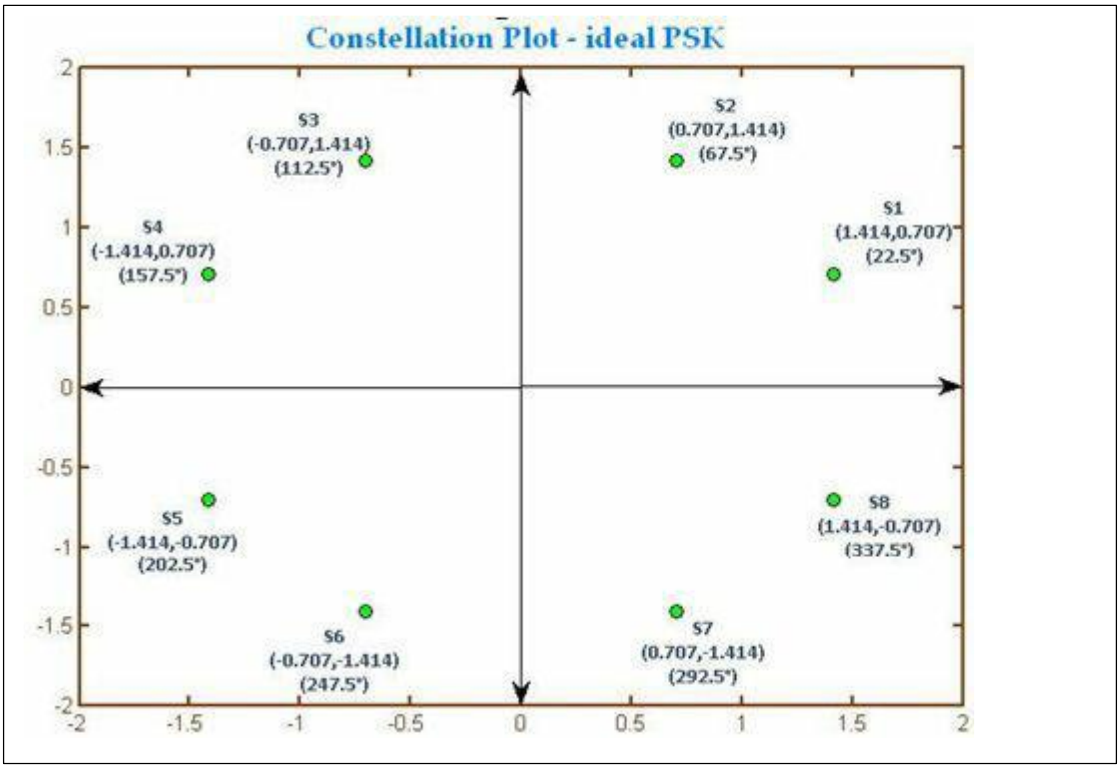
EXPERIMENT - 6

8-PSK Modulator and Demodulator over AWGN Channel [1]:

BER vs. Eb/N0 performance curves for 8-PSK is given here. A baseband simulation model for 8-PSK modulation is given below

Figure: 16 and 17





A complex baseband M-PSK signal is represented by

$$s_n(t) = \sqrt{\frac{2E_s}{T}} cos(\omega t + \theta_n) \quad ,n = 1,2, \dots ,M$$

where E_s is the symbol energy of each M-PSK Symbol, T is the bit period, θ_n is the phase shift foreach of the symbol.
For 8-PSK, M=8.
In 8-PSK modulator, every 3 input data bits are mapped to one complex-valued 8-PSK symbol (Whereas in QPSK 2 input bits are mapped to one complex valued QPSK symbol).
Each symbol is Gray coded prior to mapping them to the 8-PSK constellation (this is to allow only one bit change in the constellation, thereby avoiding abrupt phase changes in the constellation which may lead to additional errors in the receiver).
In 8PSK there are eight possible symbols that can be transmitted. We will use eight different phases $(22.5^0, 67.5^0, 112.5^0, 157.5^0, 202.5^0, 247.5^0, 292.5^0, 337.5^0)$ to represent each 8-PSK symbol. Each symbol is 45° away from adjacent symbols.

The 8PSK symbols are transmitted over an AWGN channel that adds Gaussian Noise to the transmitted symbol.

Receiver:
The detection process is performed by finding one of the eight constellation symbols that has the smallest Euclidean distance with the received symbol. Some may prefer to use a sector-based detection approach in which the received signal constellation is divided into sectors. For 8-PSK, the entire constellation can be equally divided into 8 sectors, with each angle of the sector corresponding to one of the 8 constellation symbols. The detection process can be performed by using the angle of sectors. For example, the angle sector defined by [0, π/4] corresponds to the 8-PSK symbol S1. Instead of this approach, this simulation is based on minimum Euclidean Distance. The detector computes the Euclidean Distance of each received symbol against all possible ideal symbols. Then it pinpoints an ideal symbol which gives the minimum Euclidean Distance.

Euclidean Distance is computed using the following equation
$$D = \sqrt{(y_r - s_r)^2 + (y_i - s_i)^2}$$

Where y=received signal (with noise), s=points on ideal constellation and the subscripts ‘r’ and ‘i’ are for their real and imaginary part respectively.
The detector calculates the Euclidean distance of a received symbol against all possible symbols in the constellation and picks a symbol from the ideal constellation that gives the minimum Euclidean Distance.
Finally, the demodulated symbols are converted back to binary bits and the BER (Bit Error Rate) is calculated.

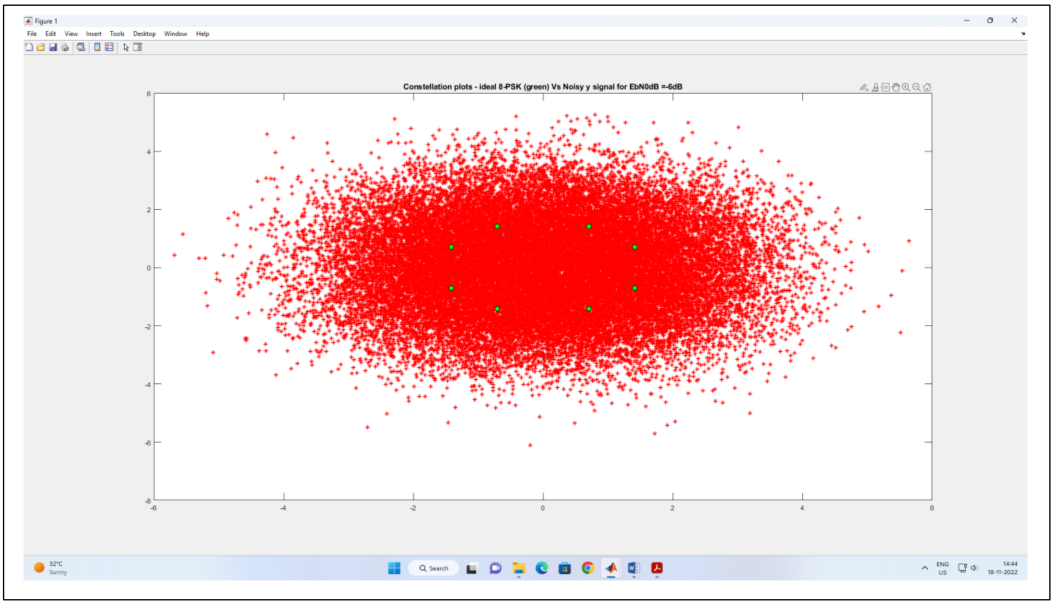
Gray to binary:
MATLAB CODE:
```
function [binaryCoded]=gray2bin(grayInput)
[rows,cols]=size(grayInput);
binaryCoded=zeros(rows,cols);
for i=1:rows
binaryCoded(i,1)=grayInput(i,1);
for j=2:cols
binaryCoded(i,j)=xor(binaryCoded(i,j-1),grayInput(i,j));
end
end
end
```

Binary to Gray
MATLAB CODE:
```
function [grayCoded]=bin2gray(binaryInput)
[rows,cols]=size(binaryInput);
grayCoded=zeros(rows,cols);
for i=1:rows
grayCoded(i,:)=[binaryInput(i,1) xor(binaryInput(i,2:cols),binaryInput(i,1:cols-1))];
end
end
```

MATLAB CODE:
```
clc; clear; close all;
N=100000; %Number of data bits to send over the channel
EbN0dB=-6:2:12;
N=N+rem((3-rem(N,3)),3); %add additional bits to the data to make the length multiple of 3
% one 8-PSK symbol contains 3 binary bits
x=rand(1,N)>=0.5;%Generate random 1's and 0's as data;
%Club 3 bits together and gray code it individually
inputSymBin=reshape(x,3,N/3)';
g=bin2gray(inputSymBin);
%Convert each Gray coded symbols to decimal value this is to ease the process of mapping based on
%arrayindex
b=bin2dec(num2str(g,'%-1d'))';
%8-PSK Constellation Mapper
%8-PSK mapping Table
map=[1.414 0.707;0.707 1.414;-0.707 1.414;-1.414 0.707;-1.414 -0.707;-0.707 -1.414;0.707 -
1.414;1.414 -0.707];
s=map(b(:)+1,1)+1i*map(b(:)+1,2);
%Simulation for each Eb/N0 value
M=8; %Number of Constellation points M=2^k for 8-PSK k=3
Rm=log2(M); %Rm=log2(M) for 8-PSK M=8
Rc=1; %Rc = code rate for a coded system. Since no coding is used Rc=1
simulatedBER = zeros(1,length(EbN0dB));
theoreticalBER = zeros(1,length(EbN0dB));
count=1;
figure;
for i=EbN0dB
%-------------------------------------------
%Channel Noise for various Eb/N0
%-------------------------------------------
```

```
%Adding noise with variance according to the required Eb/N0
EbN0 = 10.^(i/10); %Converting Eb/N0 dB value to linear scale
noiseSigma = sqrt(2)*sqrt(1./(2*Rm*Rc*EbN0)); %Standard deviation for AWGN Noise
%Creating a complex noise for adding with 8-PSK s signal
%Noise is complex since 8-PSK is in complex representation
n = noiseSigma*(randn(1,length(s))+1i*randn(1,length(s)))';
y = s + n;
plot(real(y),imag(y),'r*');hold on;
plot(real(s),imag(s),'ko','MarkerFaceColor','g','MarkerSize',7);hold off;
title(['Constellation plots - ideal 8-PSK (green) Vs Noisy y signal for EbN0dB =',num2str(i),'dB']);
pause;
%Demodulation
%Find the signal points from MAP table using minimum Euclidean distance
demodSymbols = zeros(1,length(y));
for j=1:length(y)
[minVal,minindex]=min(sqrt((real(y(j))-map(:,1)).^2+(imag(y(j))-map(:,2)).^2));
demodSymbols(j)=minindex-1;
end
demodBits=dec2bin(demodSymbols)-'0'; %Dec to binary vector
xBar=gray2bin(demodBits)'; %gray to binary
xBar=xBar(:)';
bitErrors=sum(sum(xor(x,xBar)));
simulatedBER(count) = log10(bitErrors/N);
theoreticalBER(count) = log10(1/3*erfc(sqrt(EbN0*3)*sin(pi/8)));
count=count+1;
end
figure;
plot(EbN0dB,theoreticalBER,'r-*');hold on;
plot(EbN0dB,simulatedBER,'k-o');
title('BER Vs Eb/N0 (dB) for 8-PSK');legend('Theoretical','Simulated');grid on;
xlabel('Eb/N0 dB');
ylabel('BER - Bit Error Rate');
grid on;
```

Figures 18.1 to 18.9: Shows Ideal PSK vs noise signal from -4dB to 12dB Respectively.



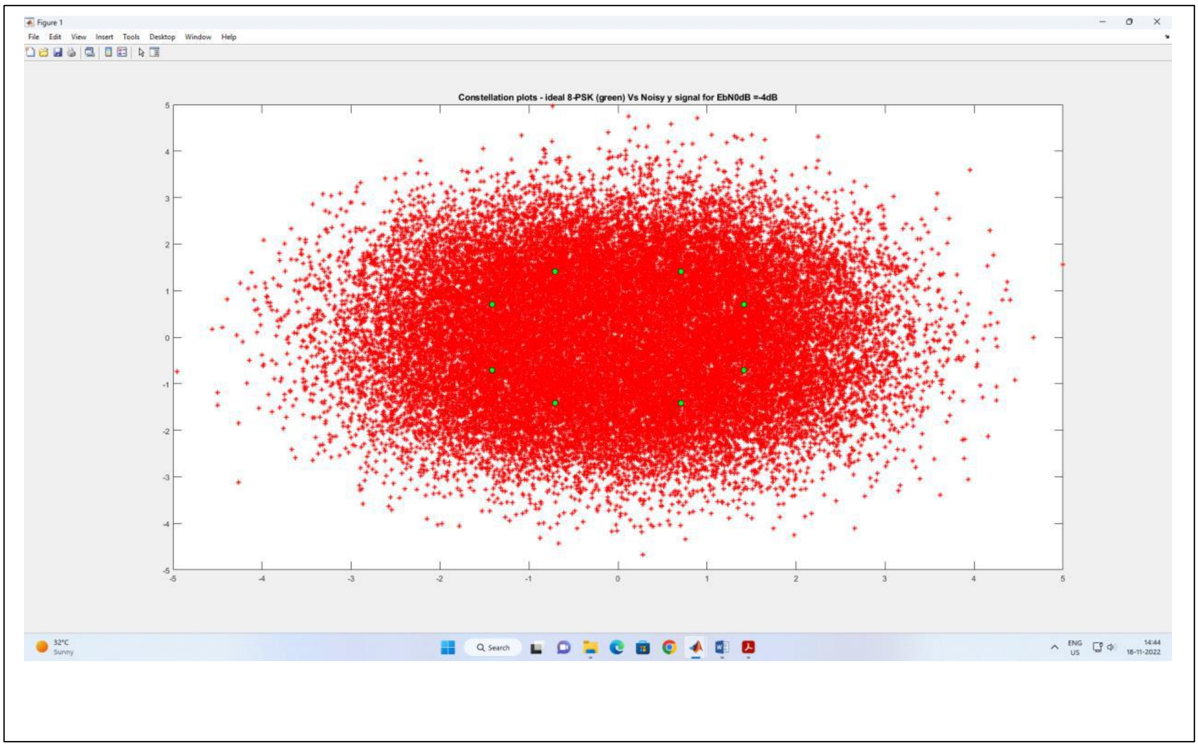


Figure 1
File Edit View Insert Tools Desktop Window Help
Constellation plots - ideal 8-PSK (green) Vs Noisy y signal for EbN0dB =-4dB
32°C
Sunny
Search
ENG
US
14:44
18-11-2022

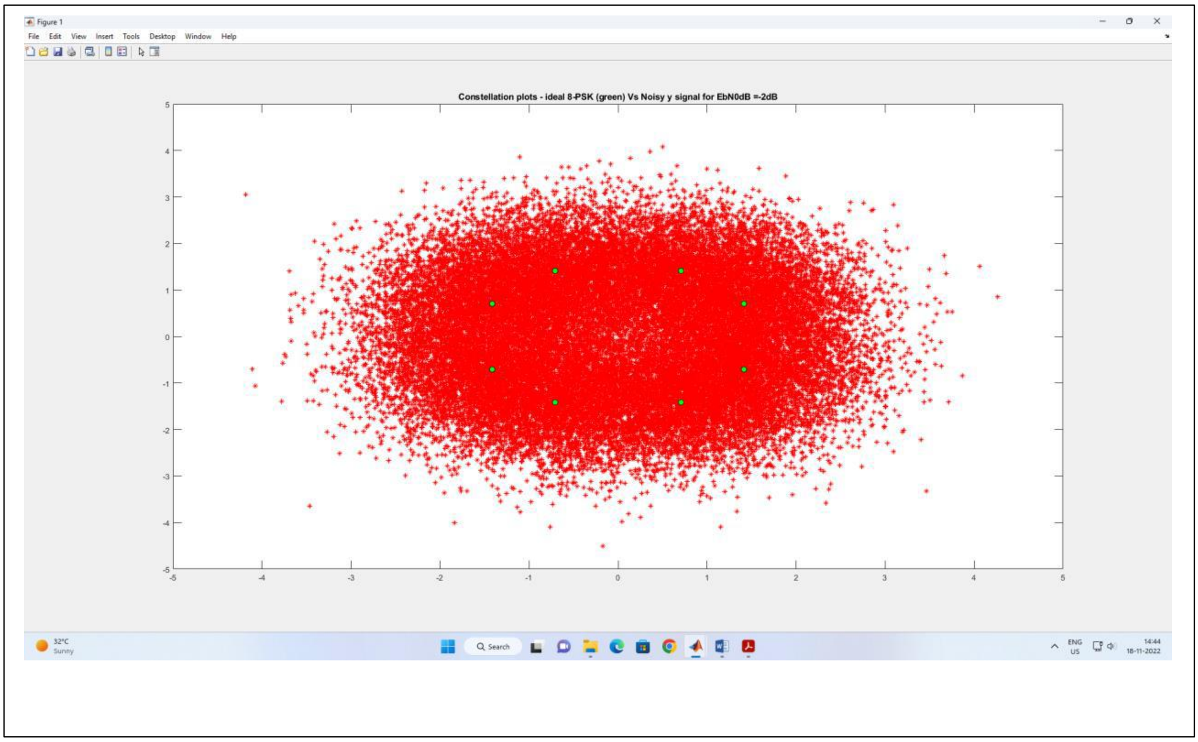


Figure 1
File Edit View Insert Tools Desktop Window Help
Constellation plots - ideal 8-PSK (green) Vs Noisy y signal for EbN0dB =-2dB
32°C
Sunny
Search
ENG
US
14:44
18-11-2022

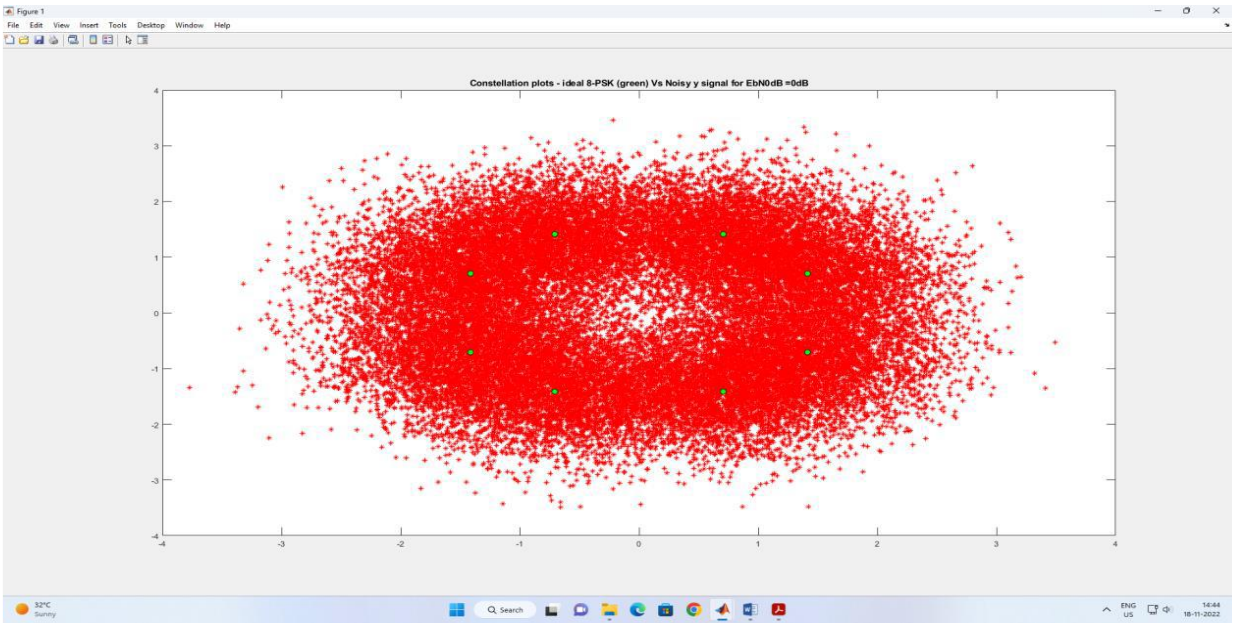


Figure 1
File Edit View Insert Tools Desktop Window Help
Constellation plots - ideal 8-PSK (green) Vs Noisy y signal for EbN0dB =0dB

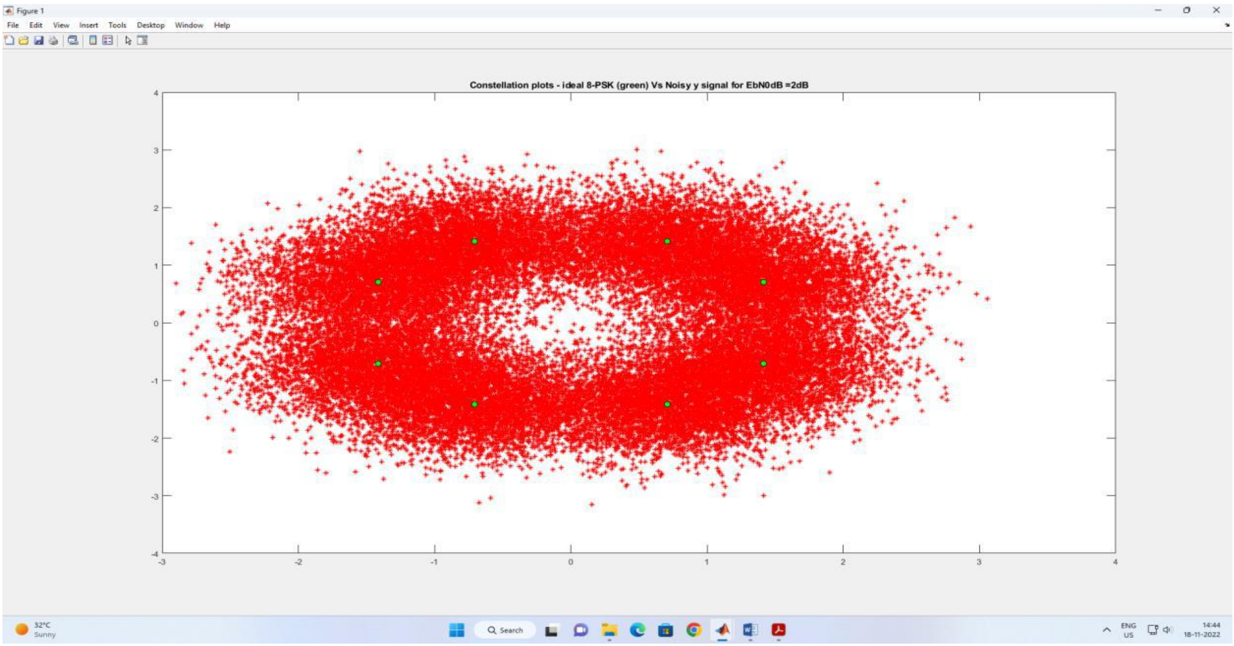


Figure 1
File Edit View Insert Tools Desktop Window Help
Constellation plots - ideal 8-PSK (green) Vs Noisy y signal for EbN0dB =2dB

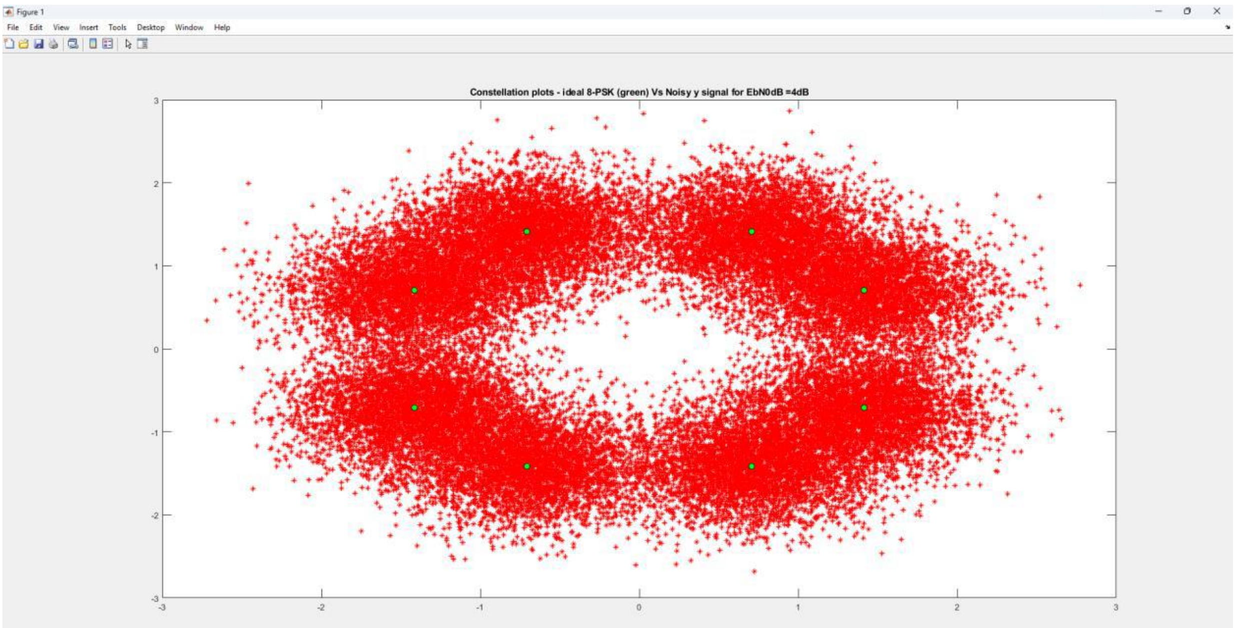


Figure 1
File Edit View Insert Tools Desktop Window Help
Constellation plots - ideal 8-PSK (green) Vs Noisy y signal for EbN0dB =4dB

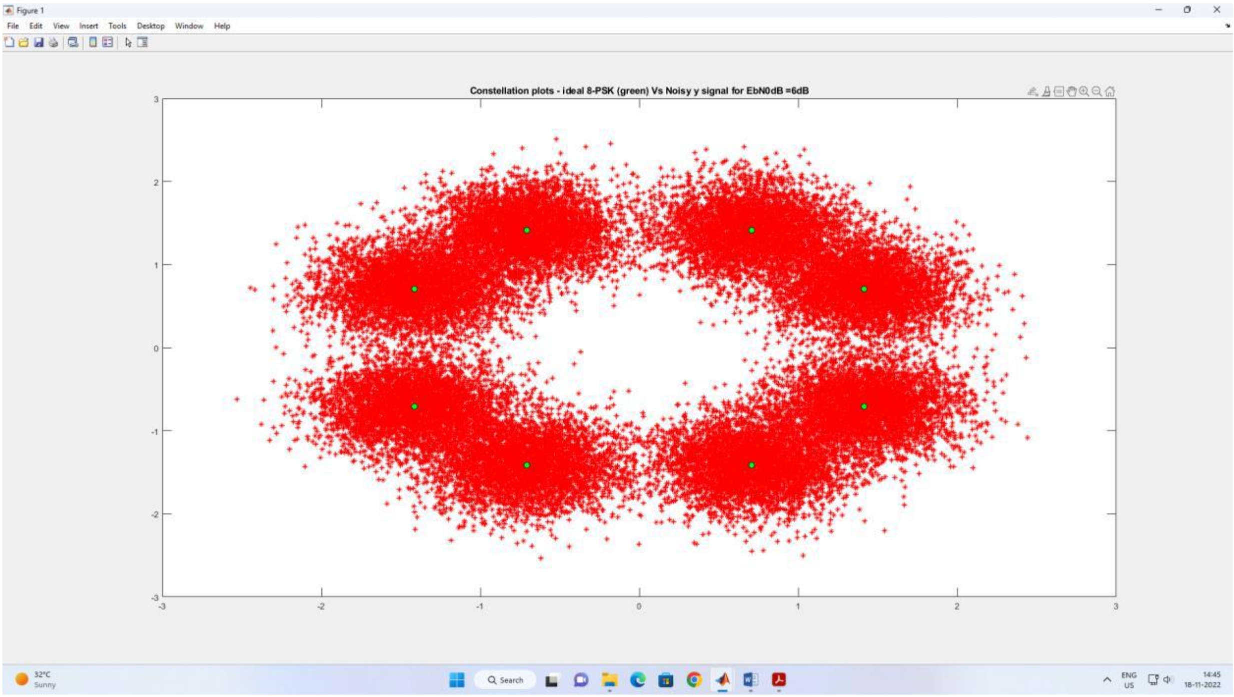


Figure 1
File Edit View Insert Tools Desktop Window Help
Constellation plots - ideal 8-PSK (green) Vs Noisy y signal for EbN0dB =6dB
32°C
Sunny
Search
ENG
US
14:45
18-11-2022

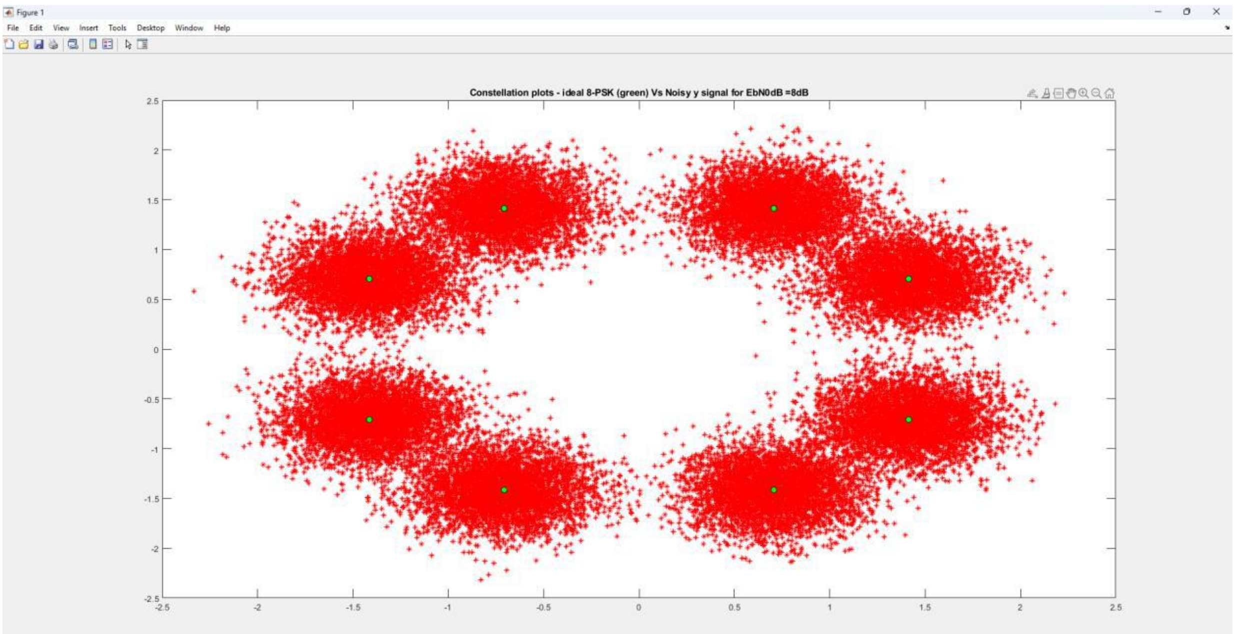
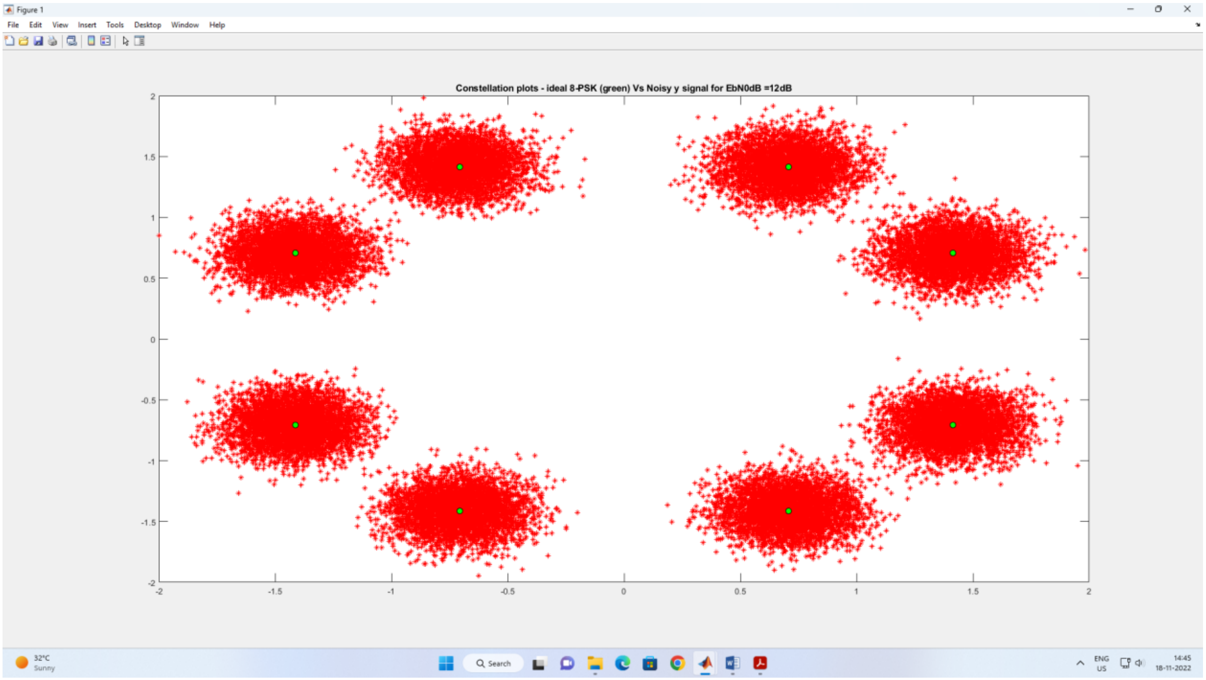


Figure 1
File Edit View Insert Tools Desktop Window Help
Constellation plots - ideal 8-PSK (green) Vs Noisy y signal for EbN0dB =8dB

At 12dB



Exercise:

1. **Design a 16QAM Modulator and Demodulator system and evaluate the BER over an AWGN Channel .**
2. **Design a 16 PSK Modulator and Demodulator system and evaluate the BER over a Rayleigh and Rician Channel .**
3. **Comment on the feasibility of utilizing 16QAM and 16PSK over a mobile broadcasting terminal.**

EXPERIMENT - 7

Introduction to **LabView** and USRP 2901: **[2]:**

FREQUENCY DIVISION MULTIPLEXING

AIM

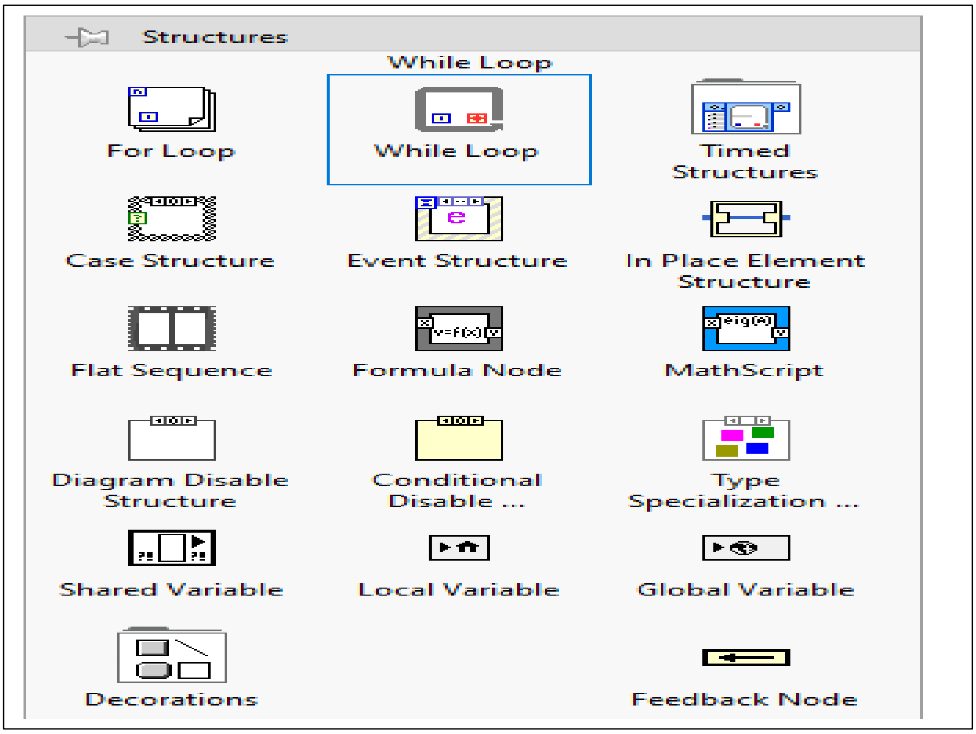
1. To estimate the frequency division multiplexing for different message and carrier signals.
2. To observe the change in output of sound at specified frequency, bits per sample, sample rate, and no. of channels.

MOTIVATION:

- To learn and use NI LABVIEW and work with its features for optimization.
- To check implementation of the circuits using LABVIEW.

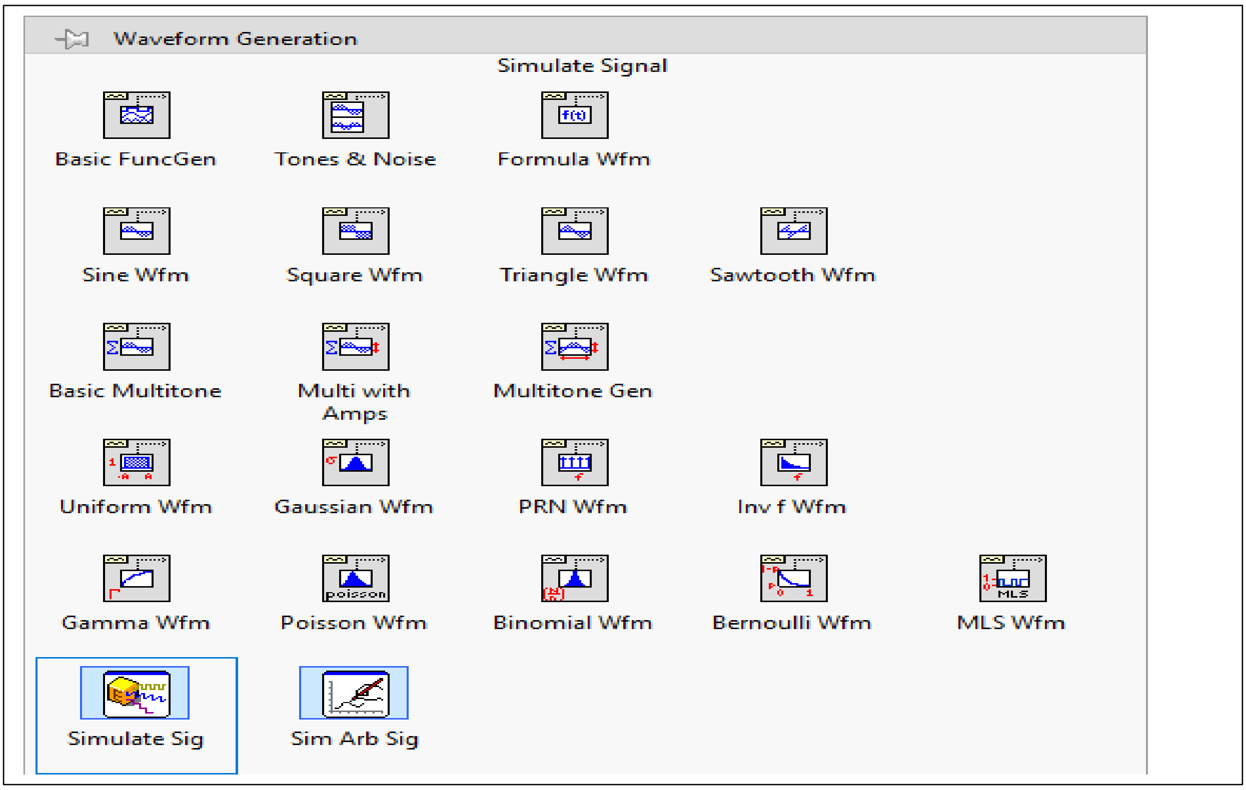
IMPLEMENTATION OF FREQUENCY DIVISION MULTIPLEXING

1. **Insert a while loop**
Right click in block diagram> structures> click while loop

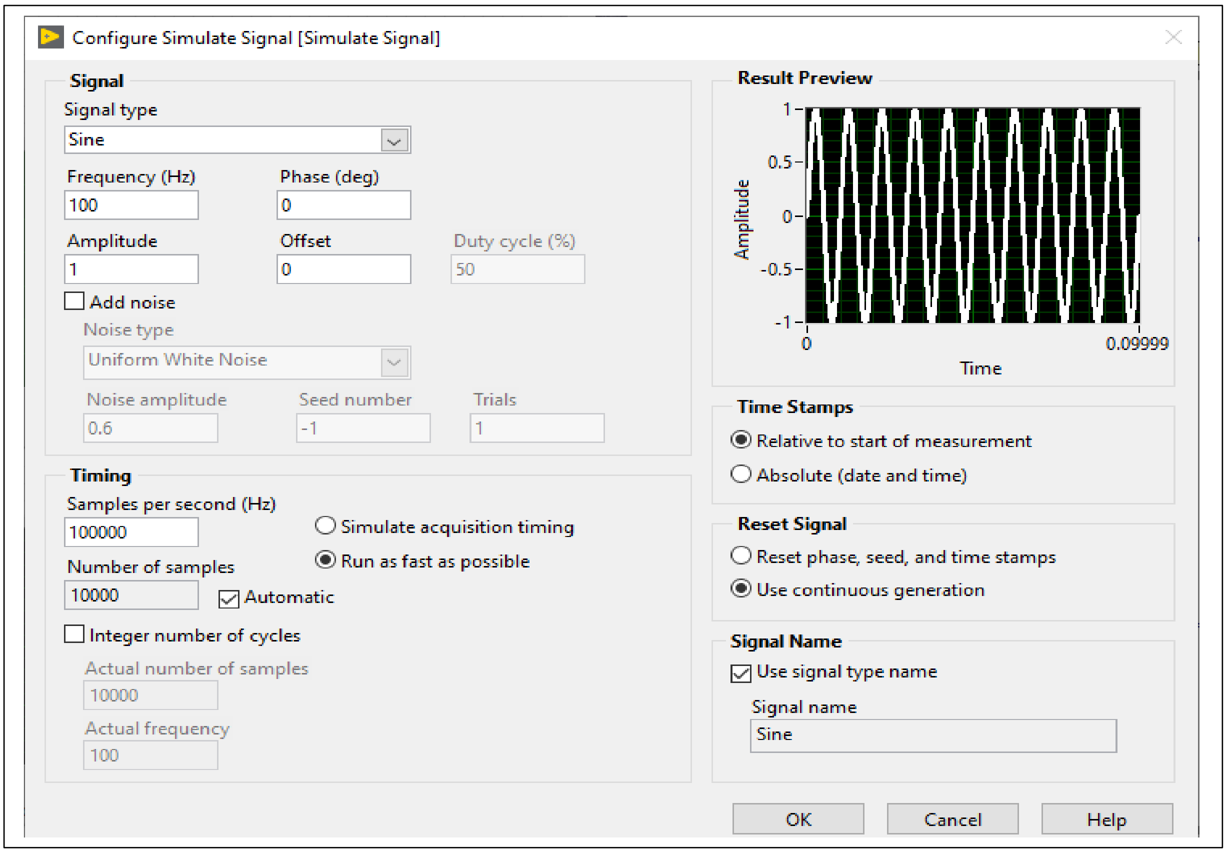


2. Insert signal generators for message and carrier signals respectively.

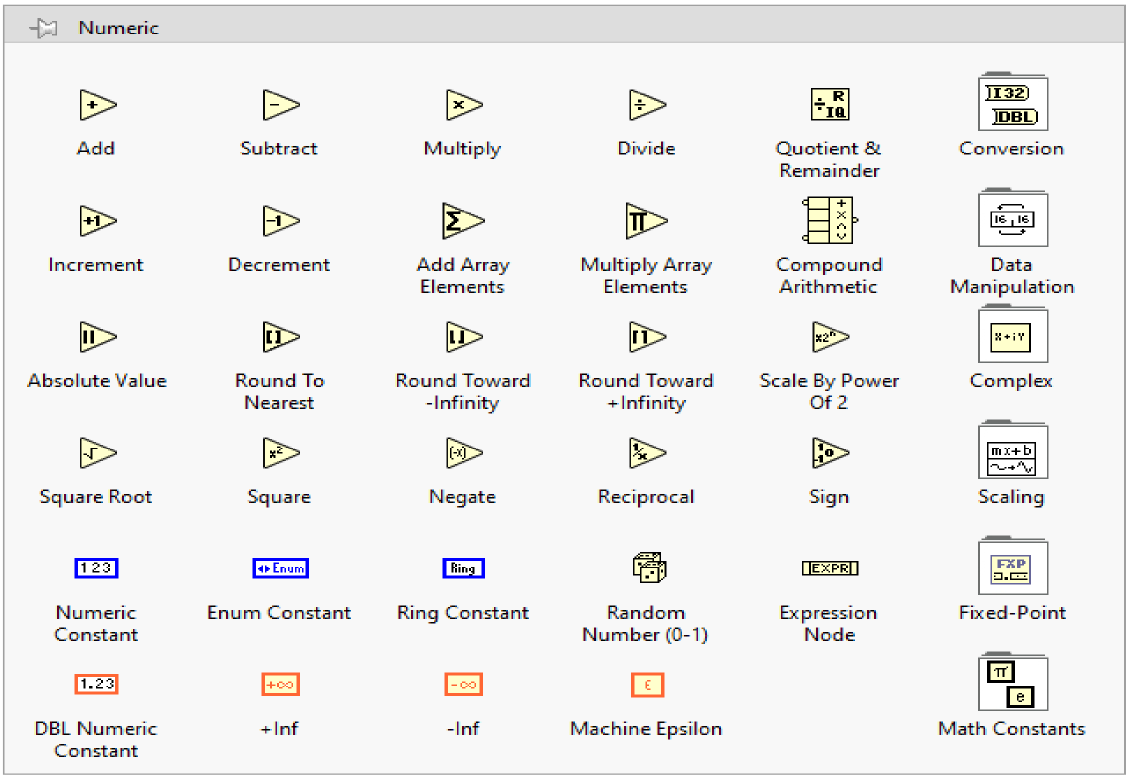
Right click in block diagram> signal processing> wfm generation> simulate sig



3 : Modify the values of frequencies according to the requirements.
Right Click on simulate sig. block> properties> change values if required



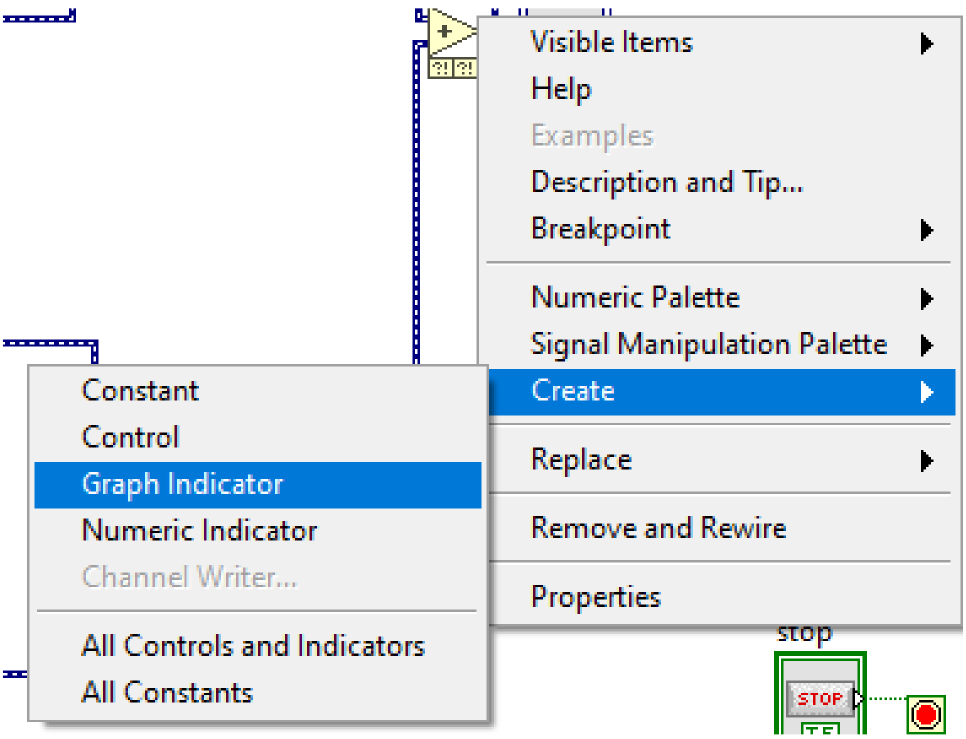
Right click in block diagram> mathematics> numeric> select adders and multipliers.
Connect the message and carrier signals respectively to the multipliers and output of multipliers to the adder.



5.

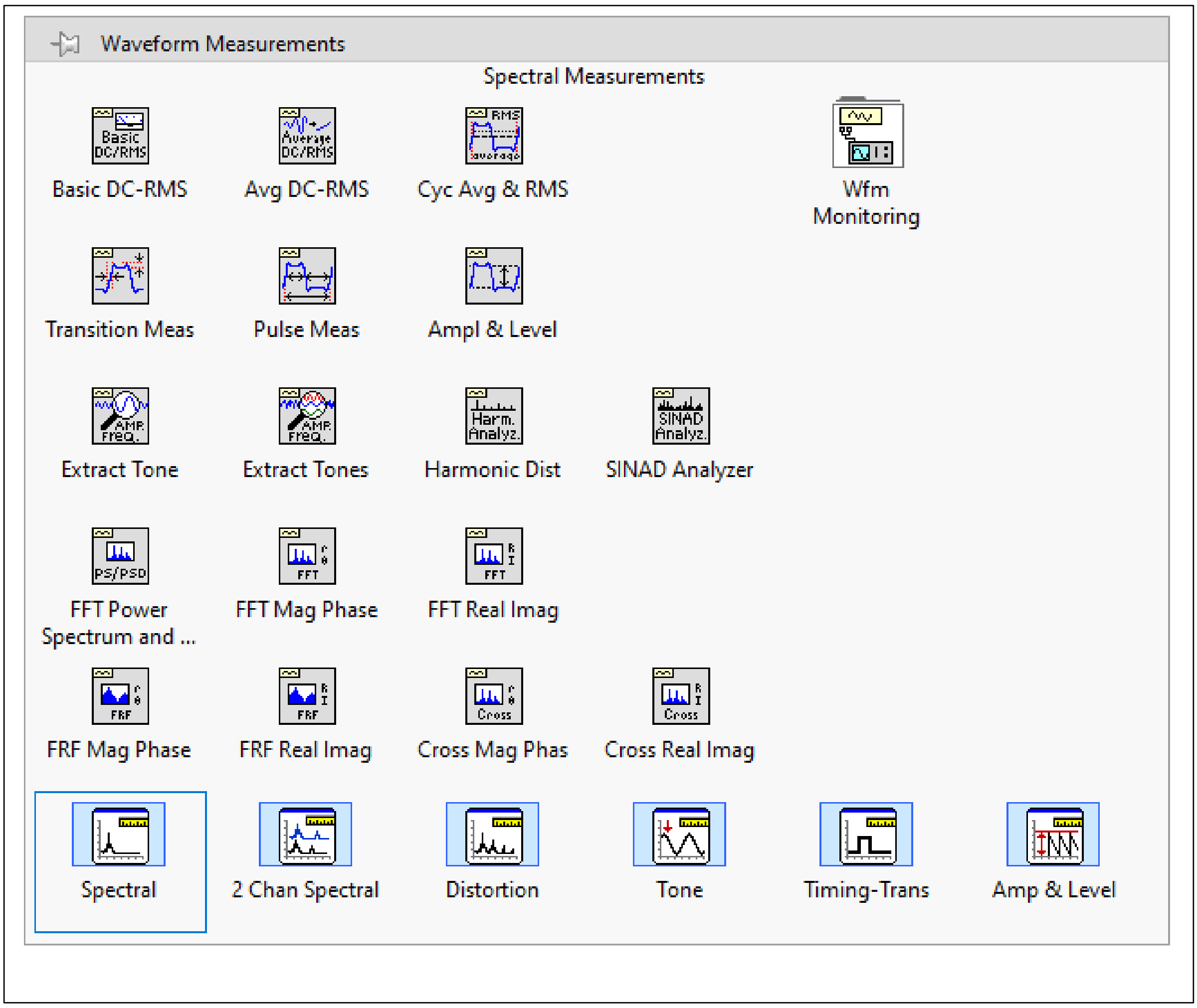
Waveform generation of the output.

Right click at adder output> create> graph indicator.



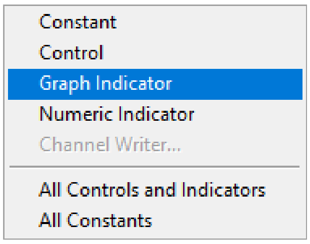
6. Adding spectral measurements block for conversion of time domain representation into the frequency domain.

Right click in block diagram> signal processing> wfm measure> spectral

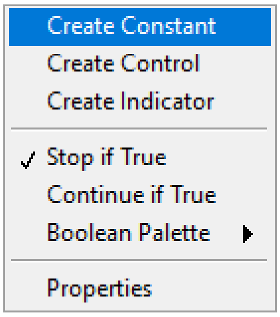


7. **Insert waveform graph block to view the output at spectral measurements block.**

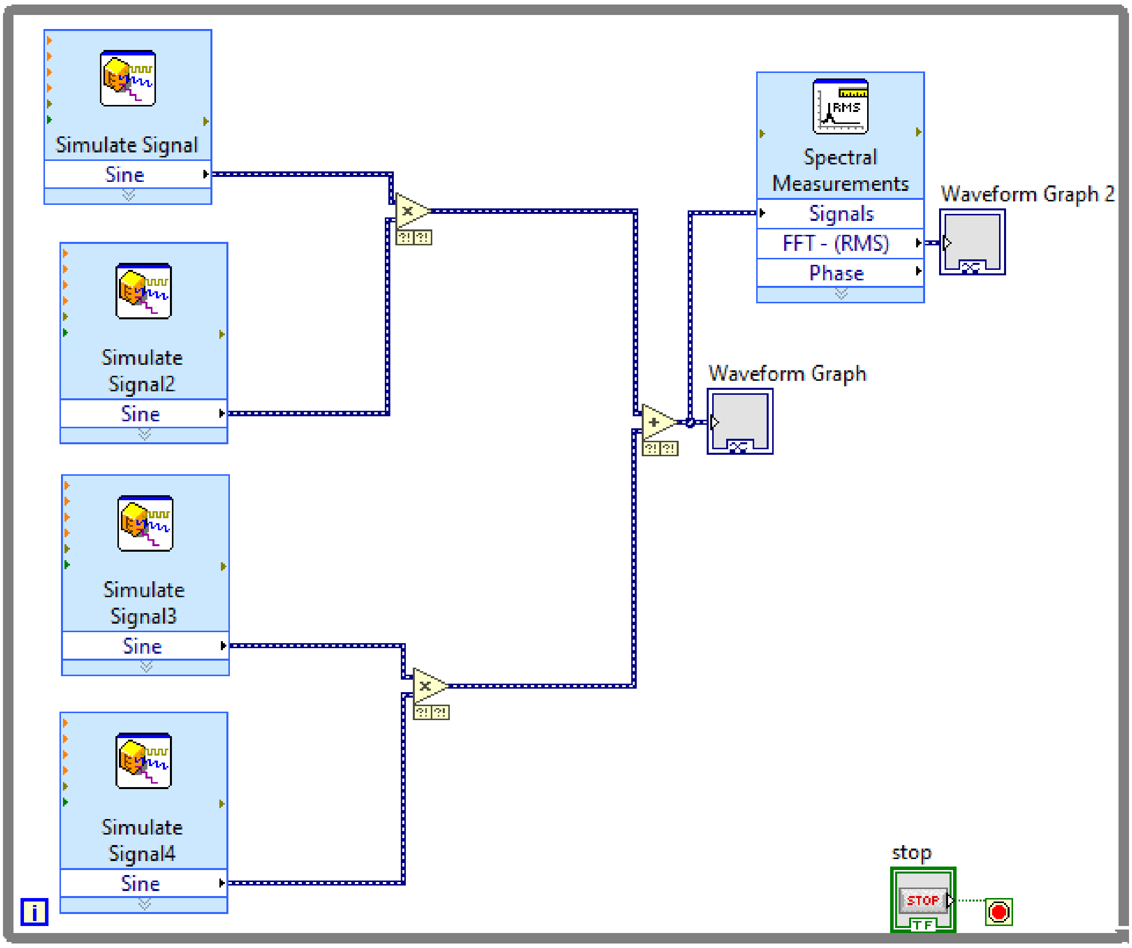
Right click > FFT – (RMS) output> create> graph indicator.



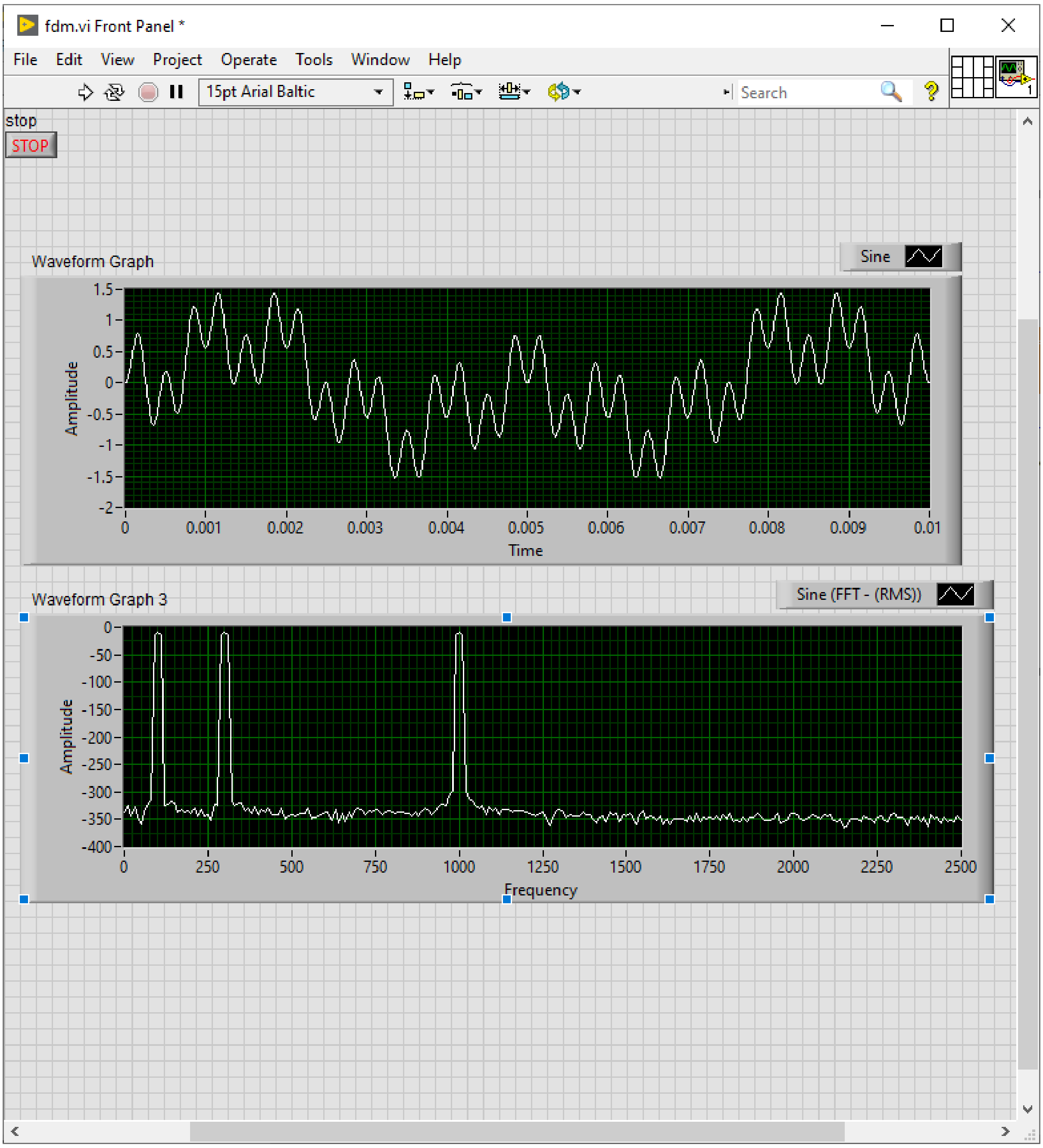
8. **End of while loop (convert loop condition to a control).**
Right click on loop condition> create control.



9. FINAL CIRCUIT FOR FREQUENCY DIVISION MULTIPLEXING (FDM)



RESULTS :



fdm.vi Front Panel *
File Edit View Project Operate Tools Window Help
15pt Arial Baltic
Search
stop
STOP
Waveform Graph
Sine
Amplitude
Time
Waveform Graph 3
Sine (FFT - (RMS))
Amplitude
Frequency

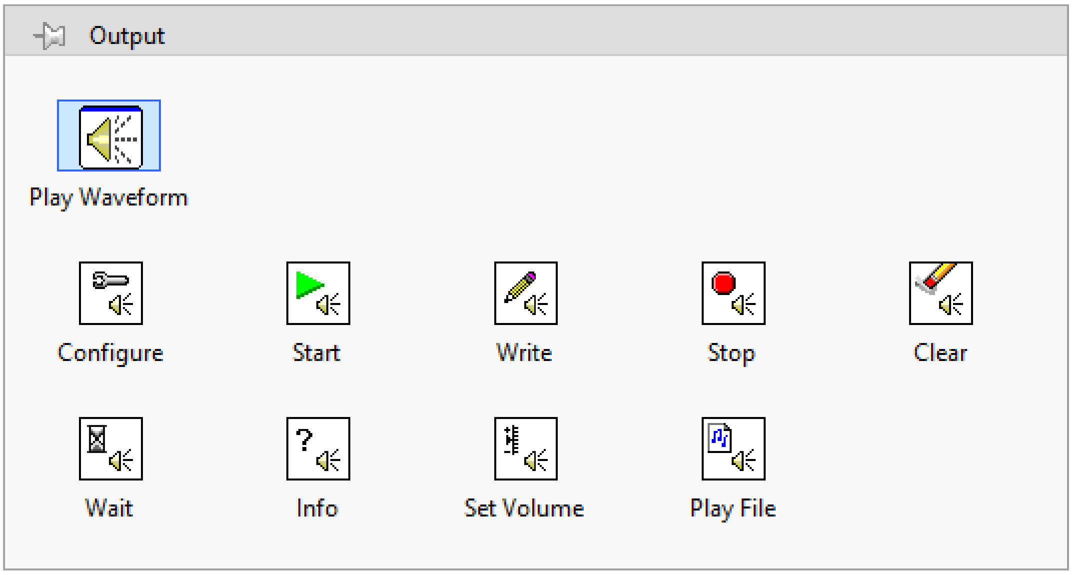
EXPERIMENT - 8

REAL TIME AUDIO OUTPUT WITH INTERACTIVE CONTROLS: Using USRP 2901.

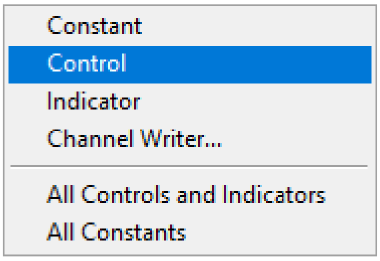
IMPLEMENTATION OF REAL TIME AUDIO OUTPUT WITH INTERACTIVE CONTROLS

1. **Insert sound write, configure, clear block**

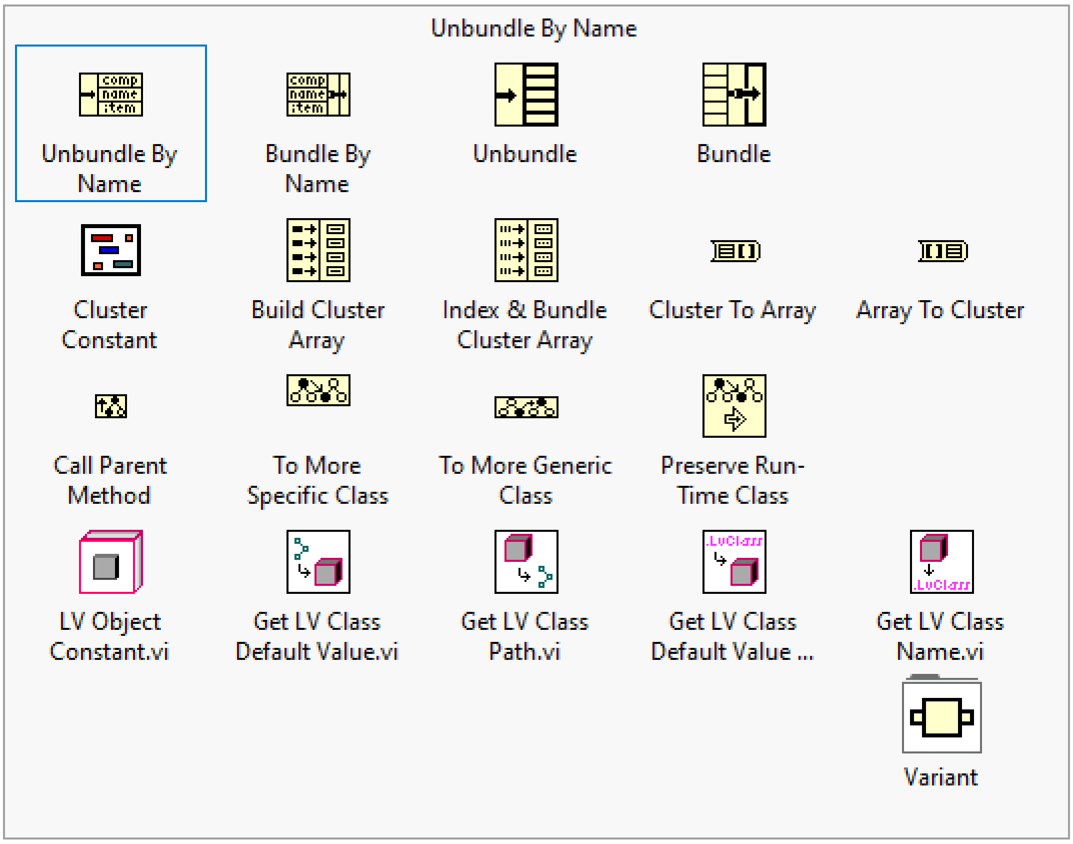
   Right click on block diagram> graphics and sound> sound> output> configure, start, clear.



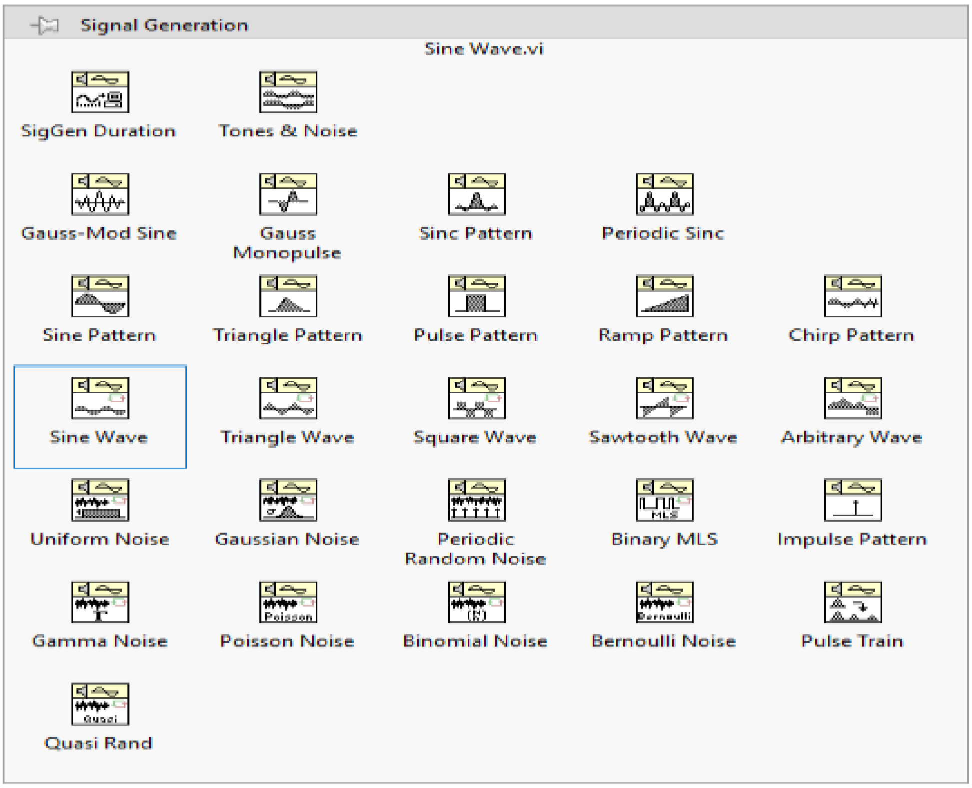
2. **Get sound format block, which helps, in specifying the frequency values. Right click at sound format at configure block> create> control**



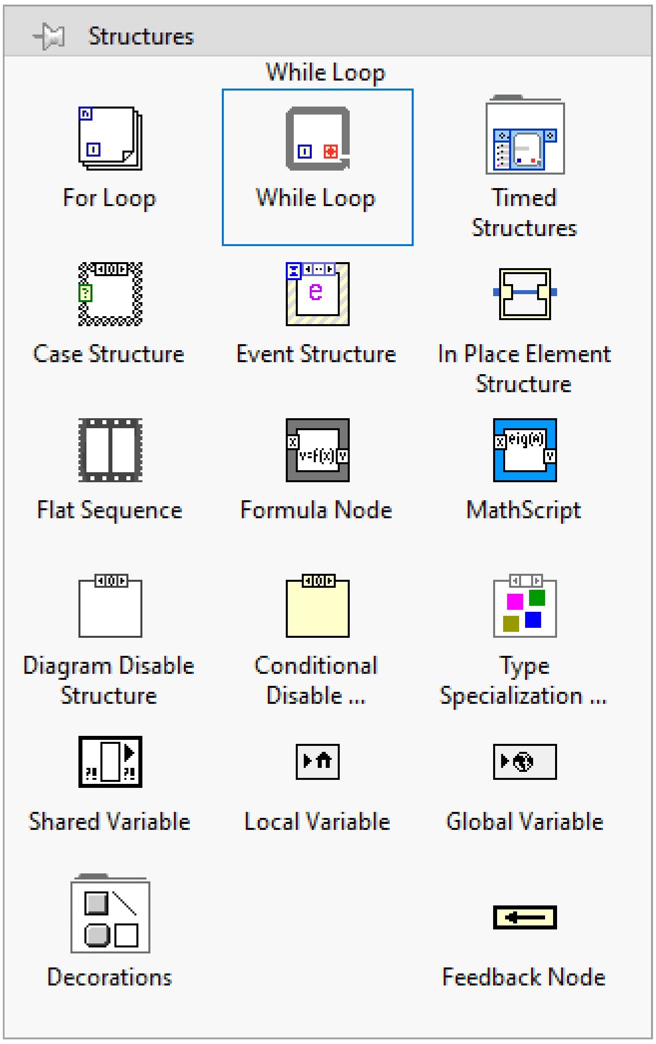
3. **Insert unbundle by name block. Right click on sound format output> cluster, class & variant palette> unbundle by name.**



4. **For sound generation we use sinusoidal signal source.**
   Right click in block diagram> signal processing> sig generation> sine wave

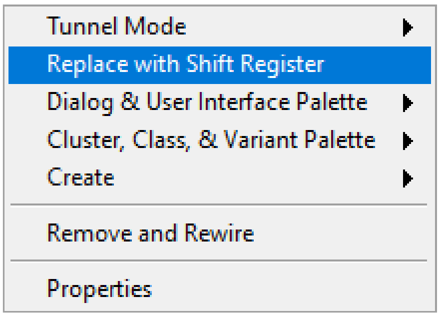


5. **Insert a while loop (re-entrant) for the sound write and sine wave block.**
Right click in block diagram> structures> click while loop



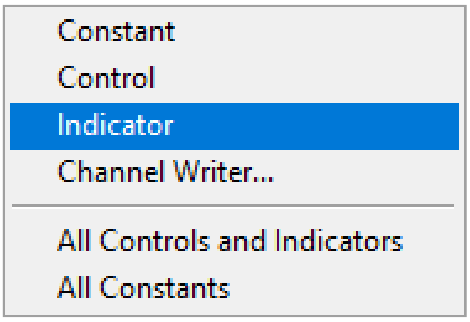
6. **Propagate the error cluster by joining error out of sound configure block to error in of sound write block and error out of sound write block to error in of sound clear block. Join all the task id similarly.**

7. **Convert the error out lines through the while loop into the shift register (at the intersection of while loop block and error out line).**
Right click at the junction> replace with shift register



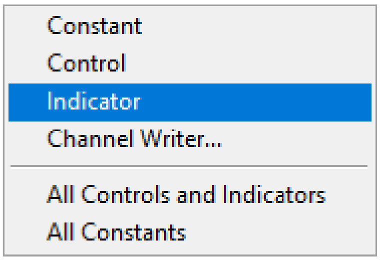
8. Create default error out block.

Right click at sound clear block (error out)> create> indicator.

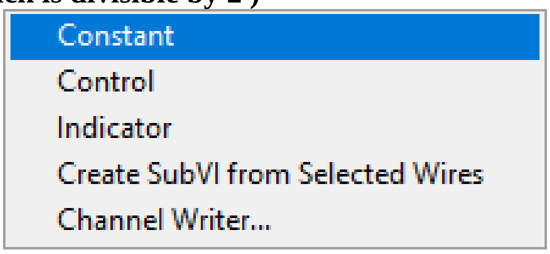


9. Create the sample mode indicator. Using continuous mode instead of finite mode for sampling.

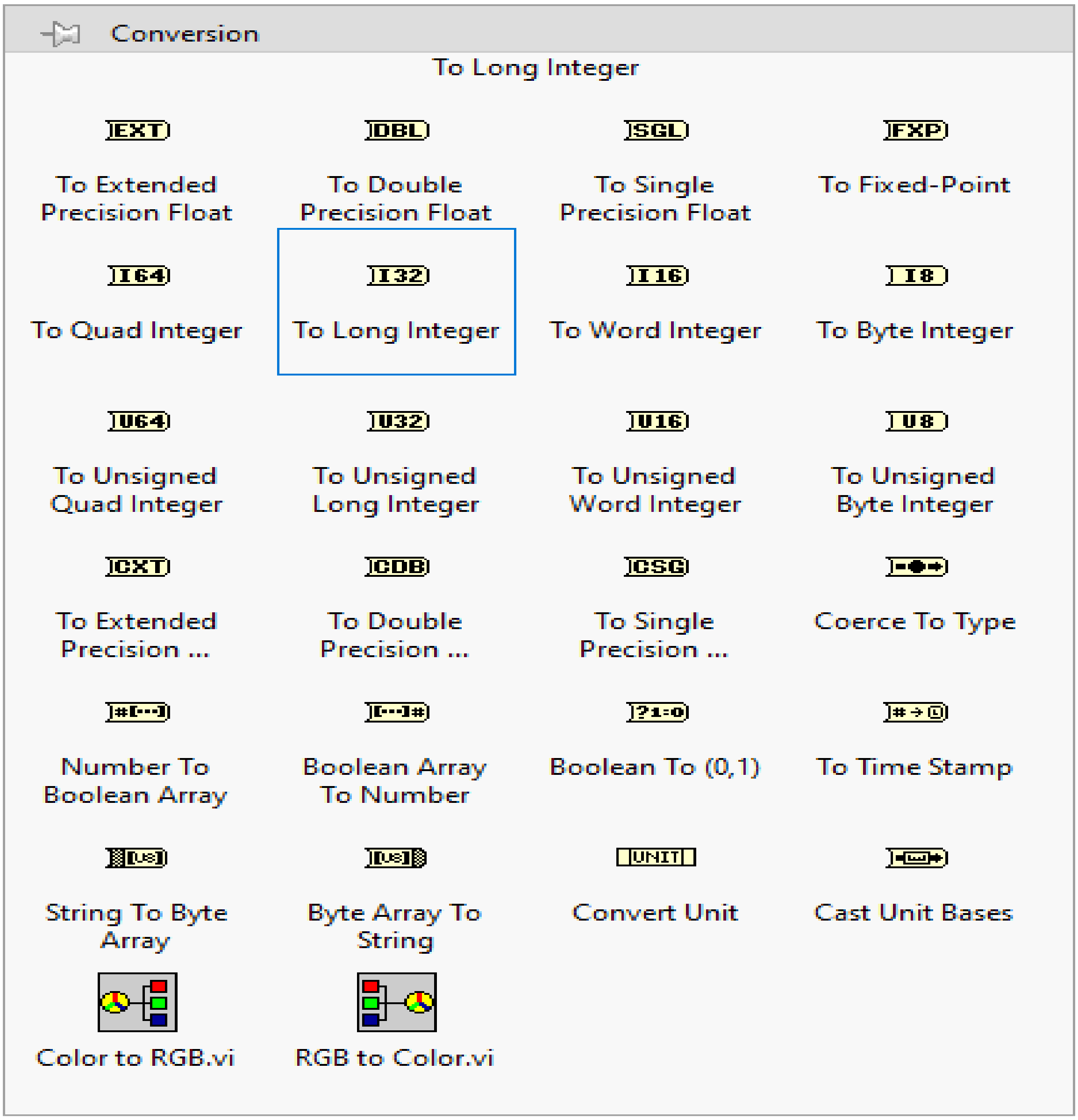
Right click on configure block (sample mode)> create> indicator.



10. Use divider and give one of the input sample rate, and other input as a constant (consider numbers which is divisible by 2)



11. **Using block size to see how many bits is used and send it to configure**.
Right click on block diagram> numeric> conversion> long integer.



12. **Insert another divider inside the while loop and give one of the input from sample rate block. Create another input of divider into a control signal.**

13. **Give the output of the divider to the frequency pin of the sine wave block.**

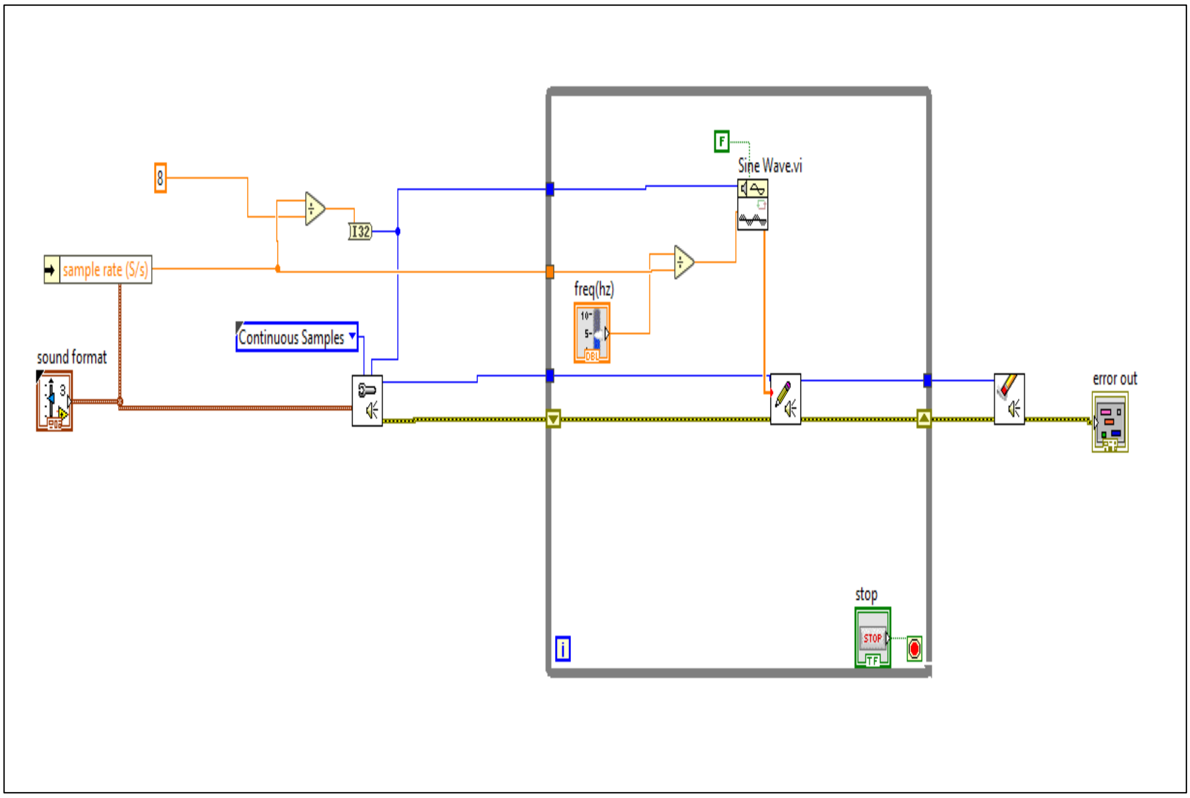
14. **Connect the sine wave output from the block to the sound write block input.**

15. **Convert condition loop block of the while loop to a control.**

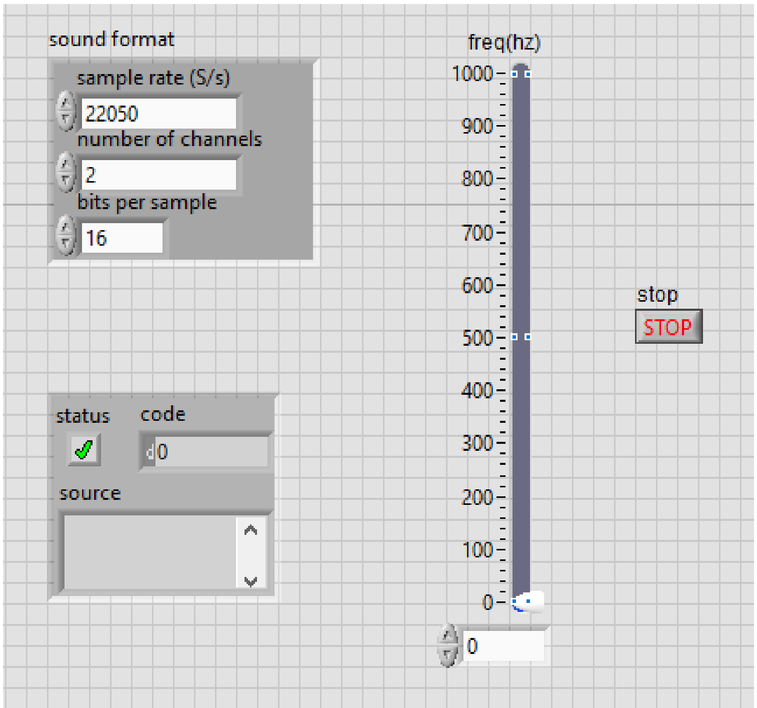
16. **Convert reset phase of sine wave block to constant, and turn it down to false.**

17. **Run the circuit. (Change values of the circuit for change in outputs).**

FINAL CIRCUIT OF REAL TIME AUDIO OUTPUT WITH INTERACTIVE CONTROLS:



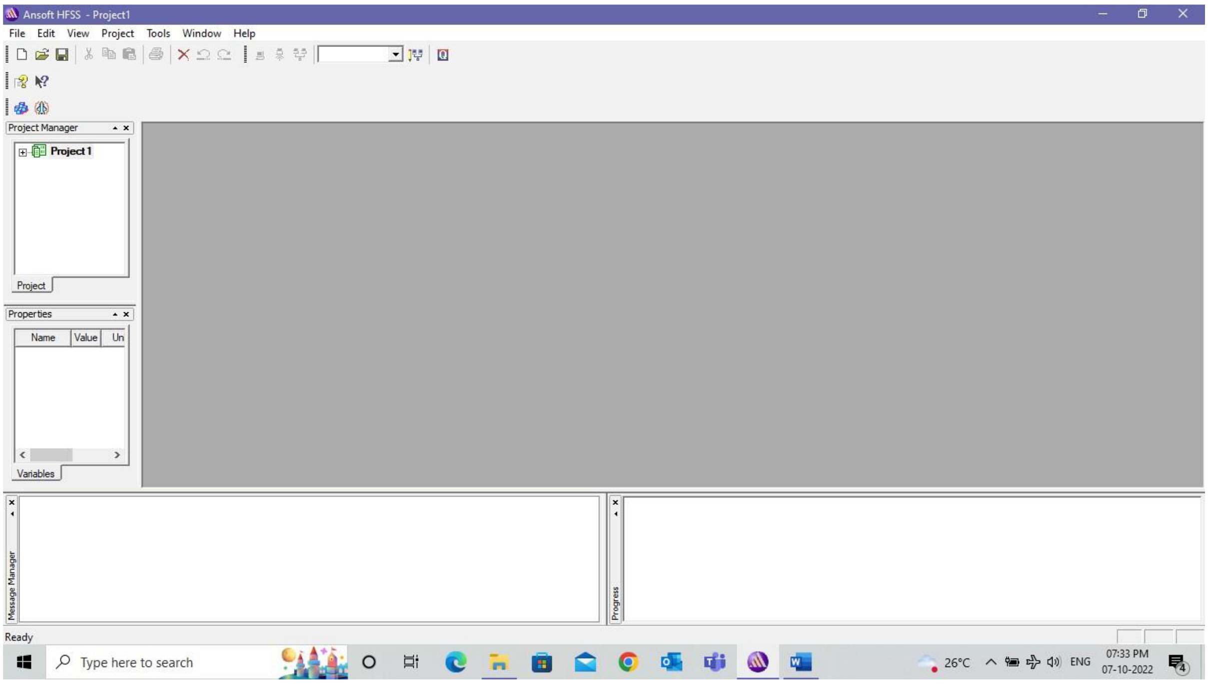
Results:



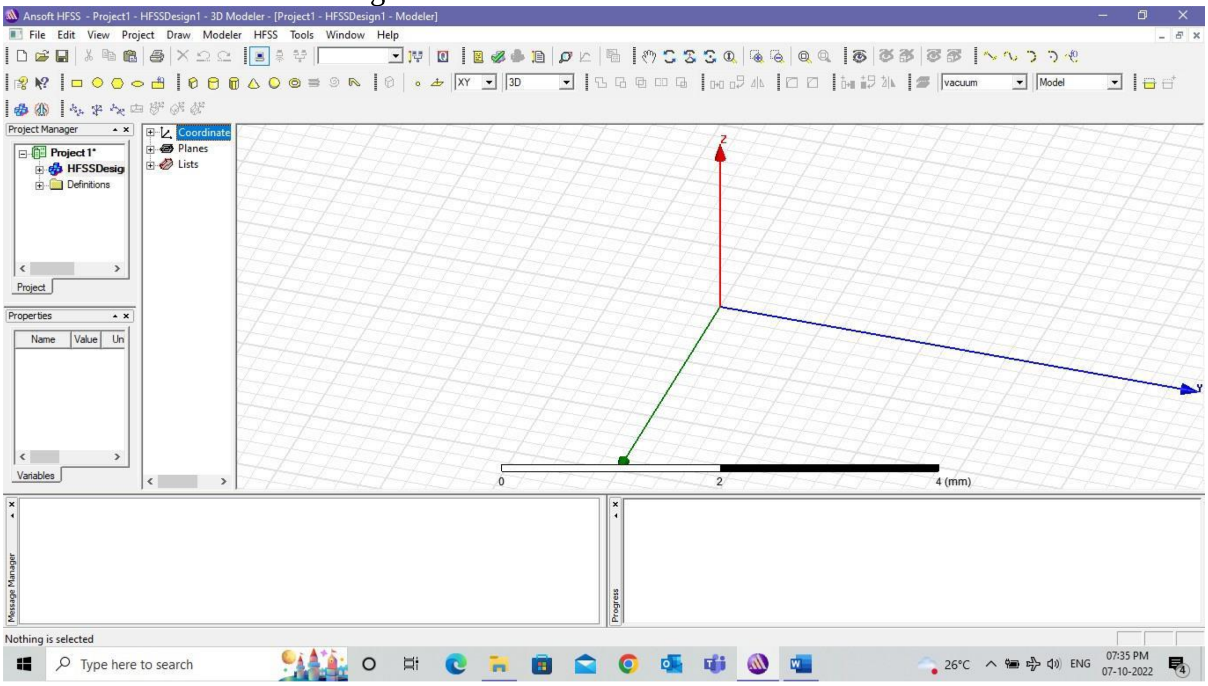
Experiment-9

Design of an Microstrip Antenna using HFSS

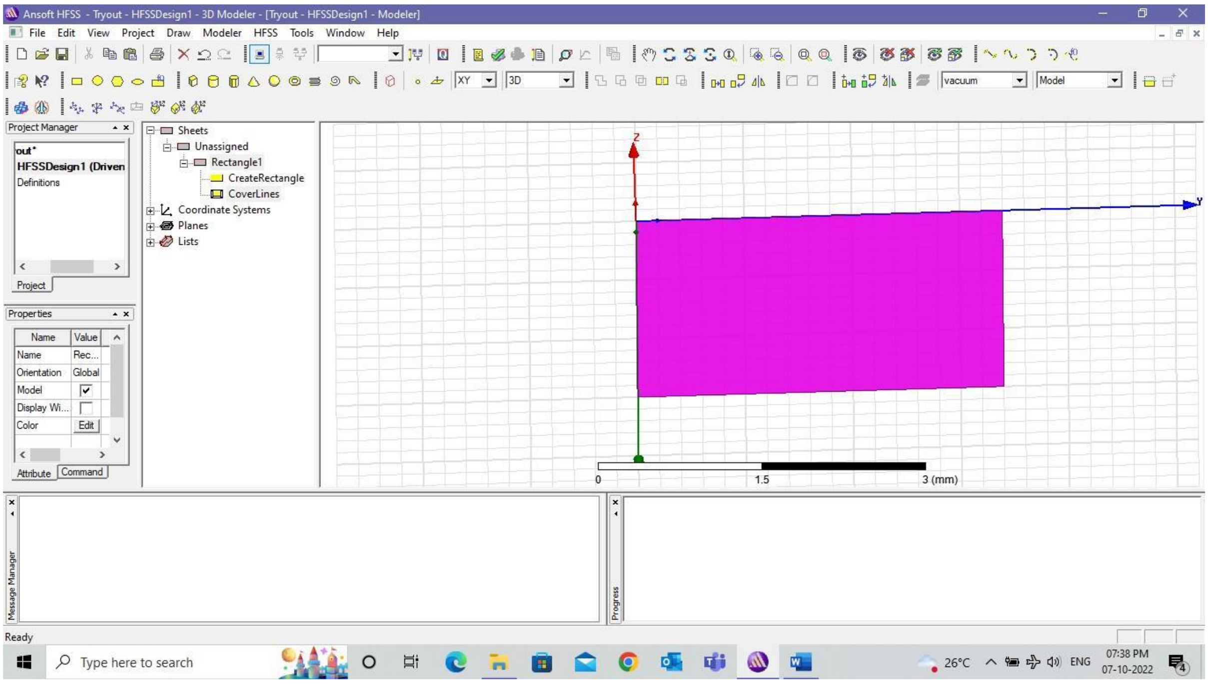
- Open the HFSS tool
- Click on new Project



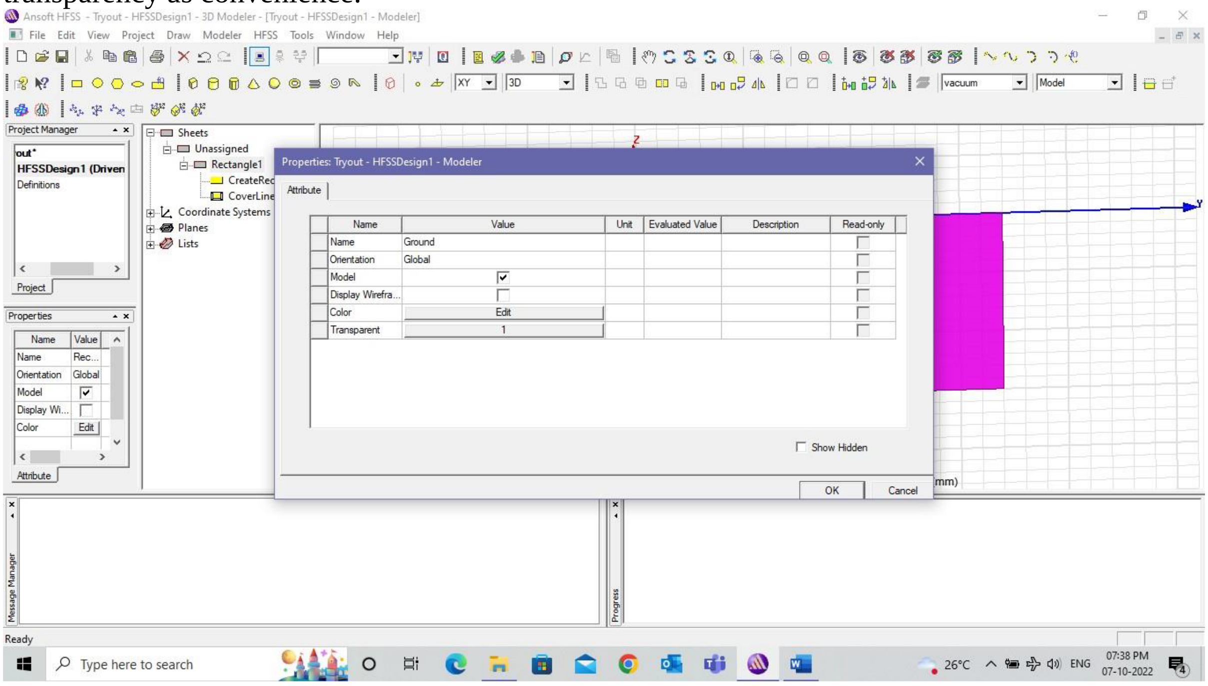
- Click on Insert HFSS Design and save it with the file name



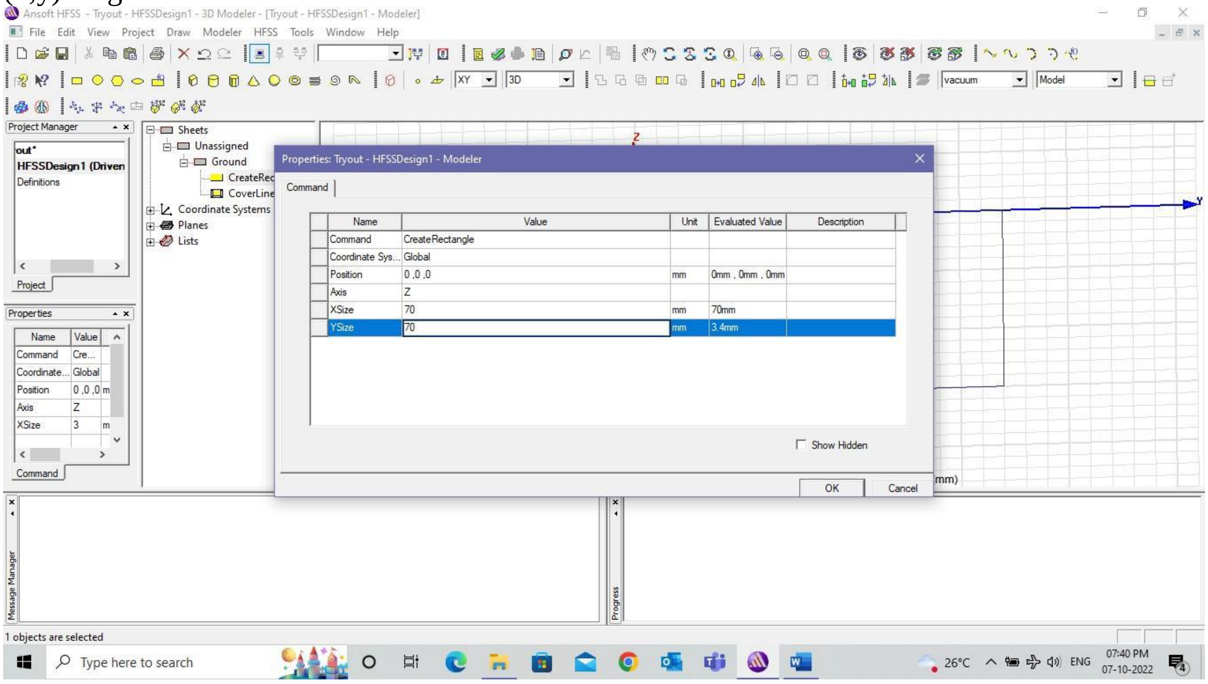
• Click on draw rectangle and place it in the design as desired.



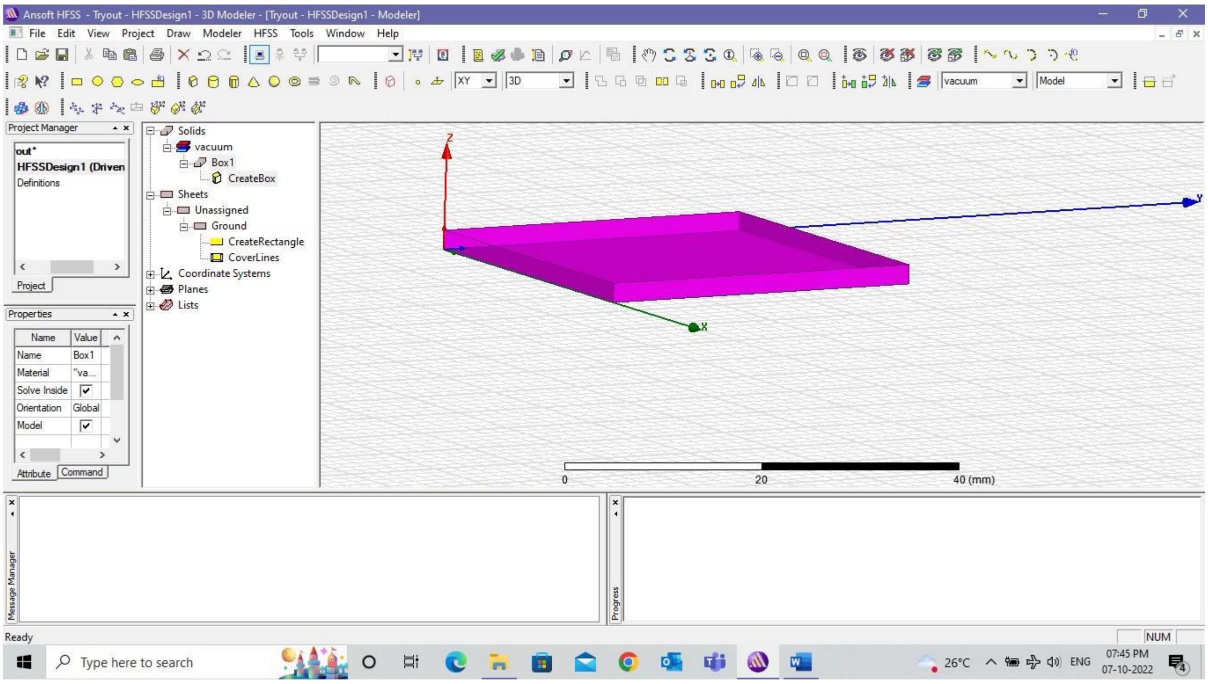
• Double click on the rectangle element and rename according to the design and set the transparency as convenience.



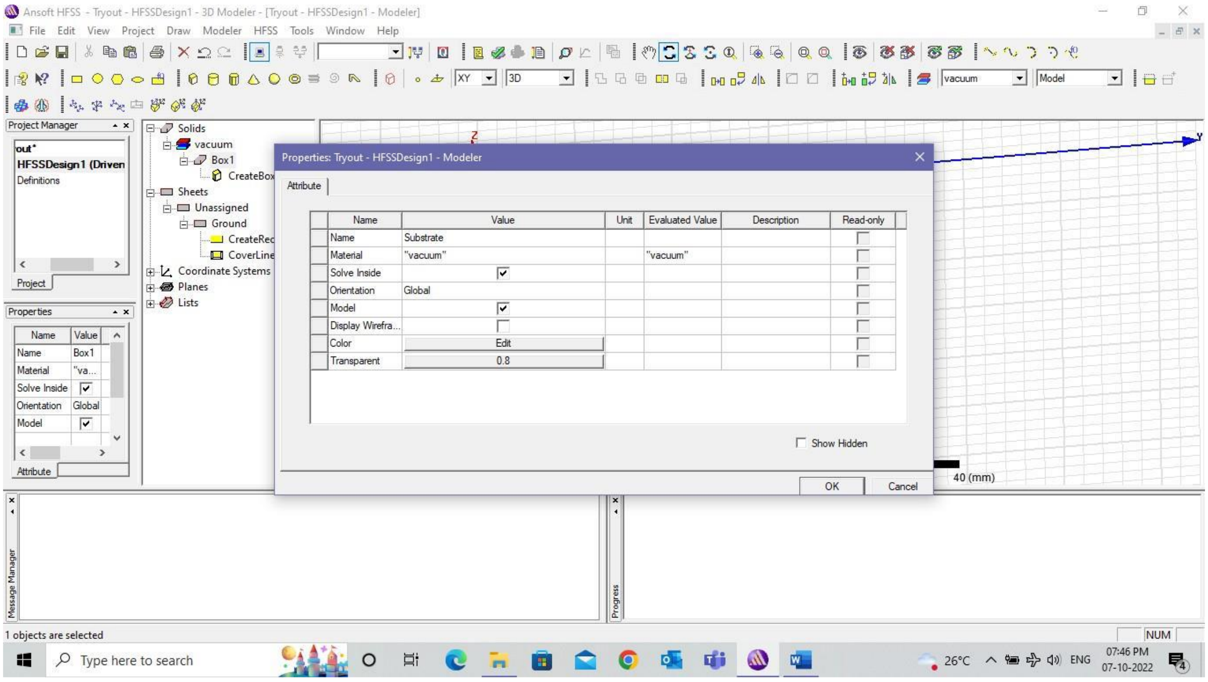
- Select Create Rectangle and change the position (x, y, z) accordingly and change the axes (x,y) length.



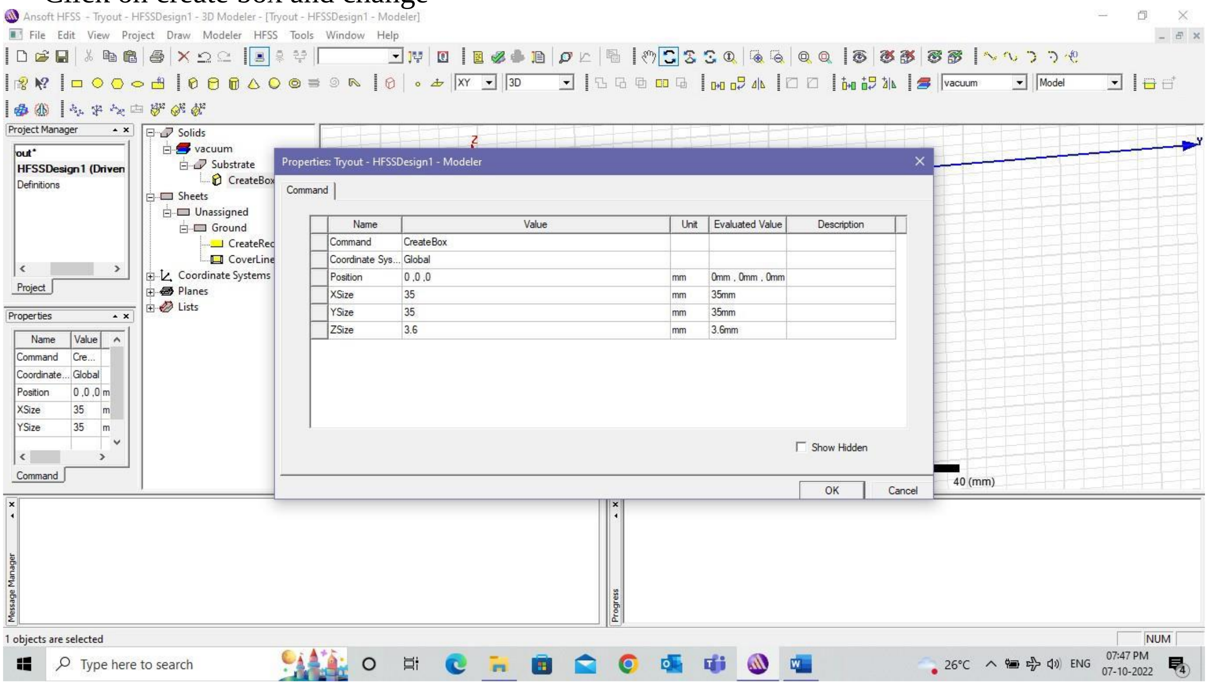
- Select the cube and place it on the drawn rectangle.



- Double click on box1 element and rename it to substrate.

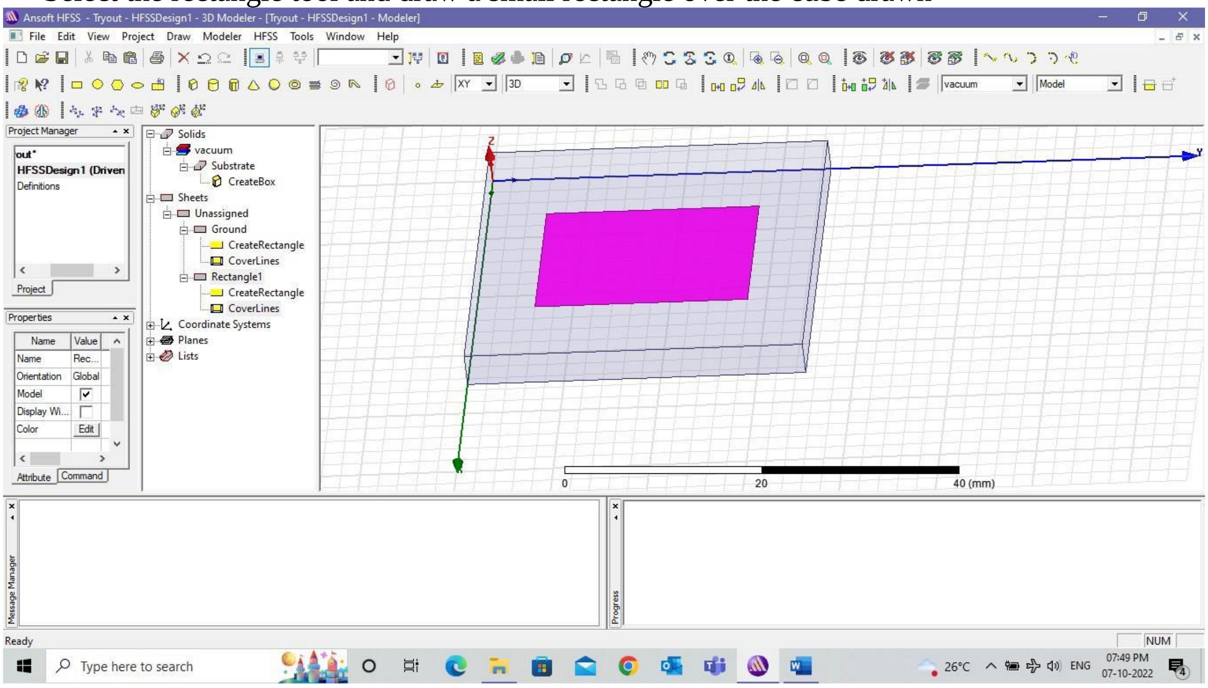


- Click on create box and change

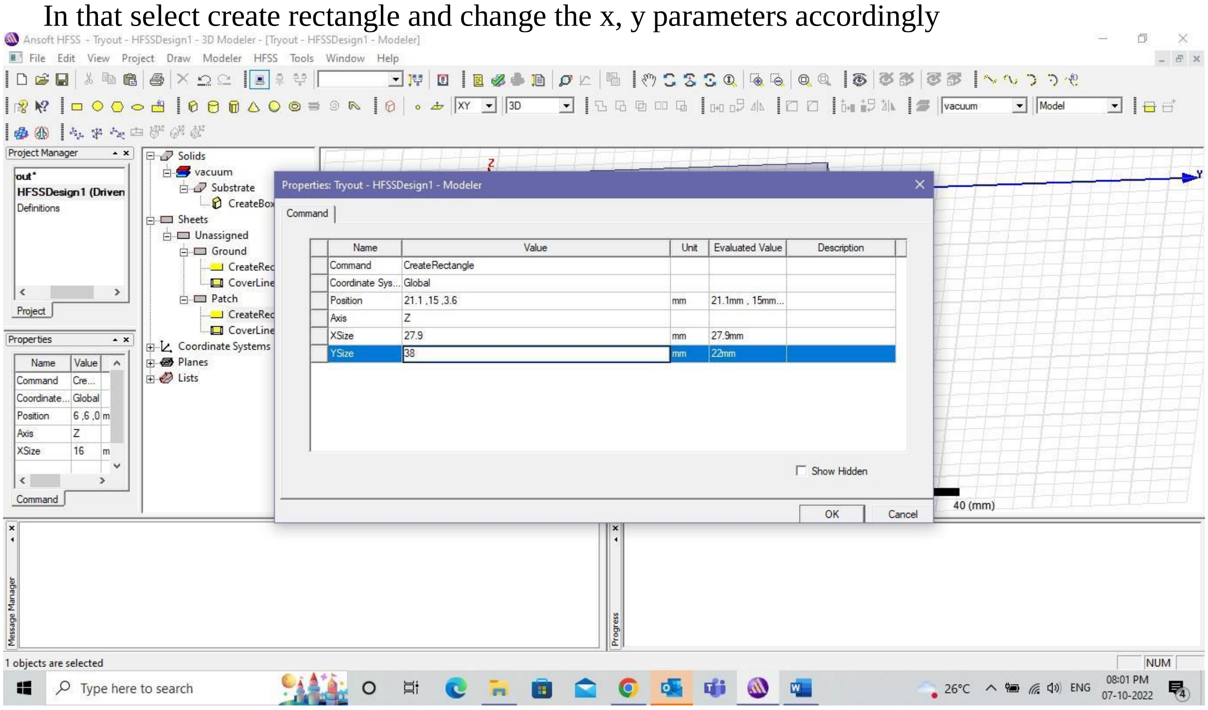


- the x, y, z sizes along with the positions.

• Select the rectangle tool and draw a small rectangle over the cube drawn

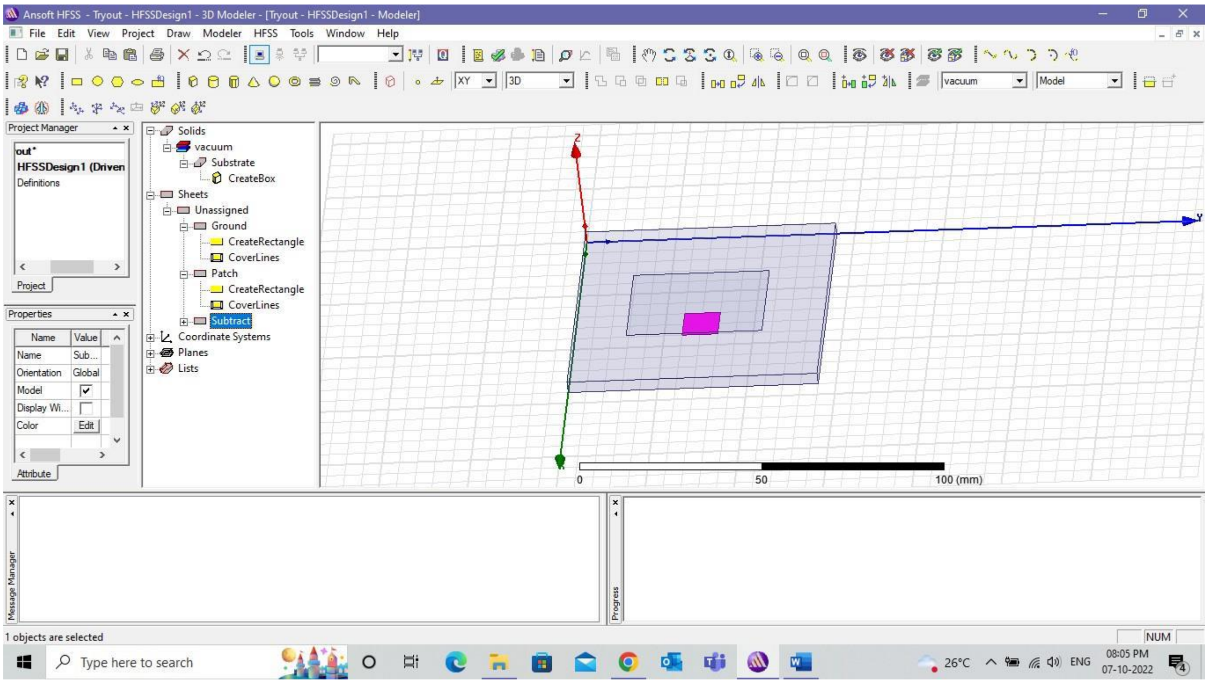


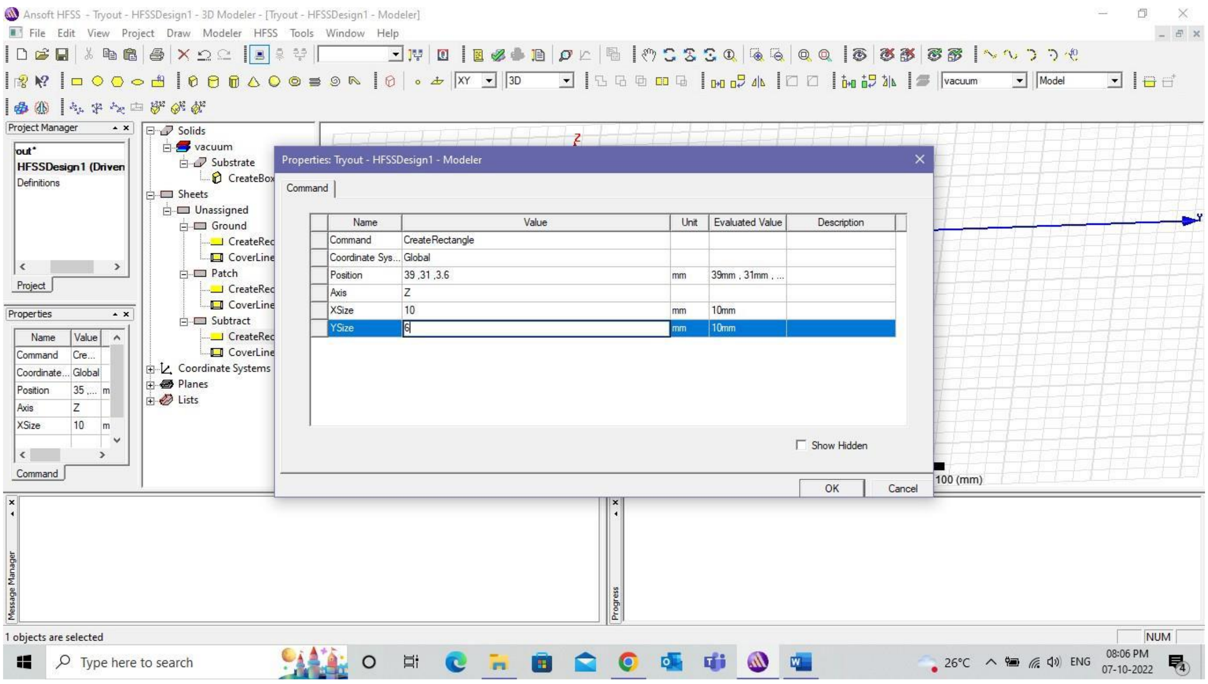
• In that select create rectangle and change the x, y parameters accordingly



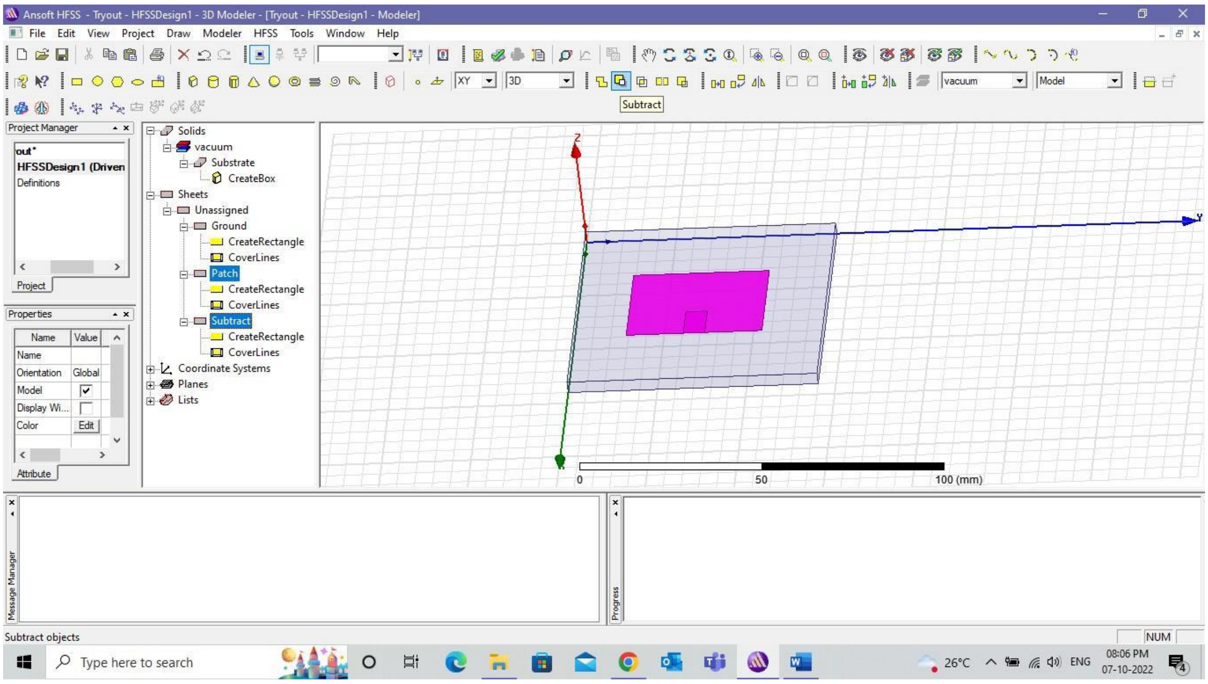
• Select the rectangle tool and draw a small rectangle over the previous drawn rectangle and name it as subtract.

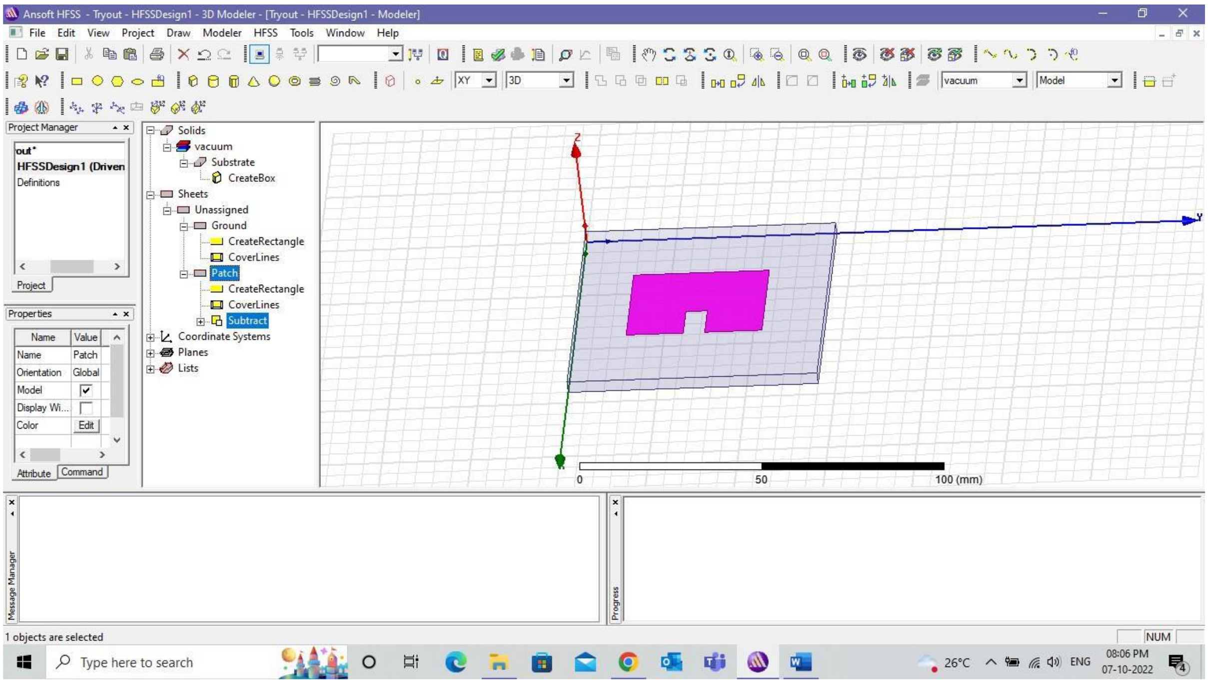
• Change the x and y sizes accordingly.



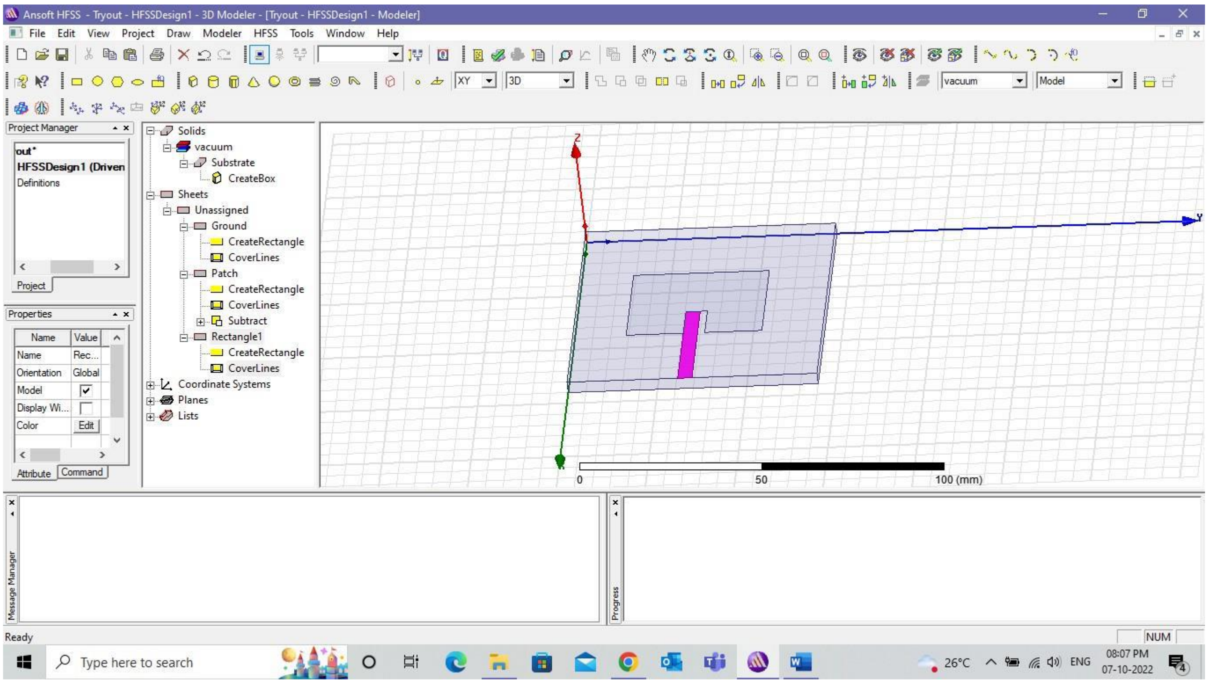


• Select the patch first and the subtract and click on subtract tool which removes the newly created portion from the patch

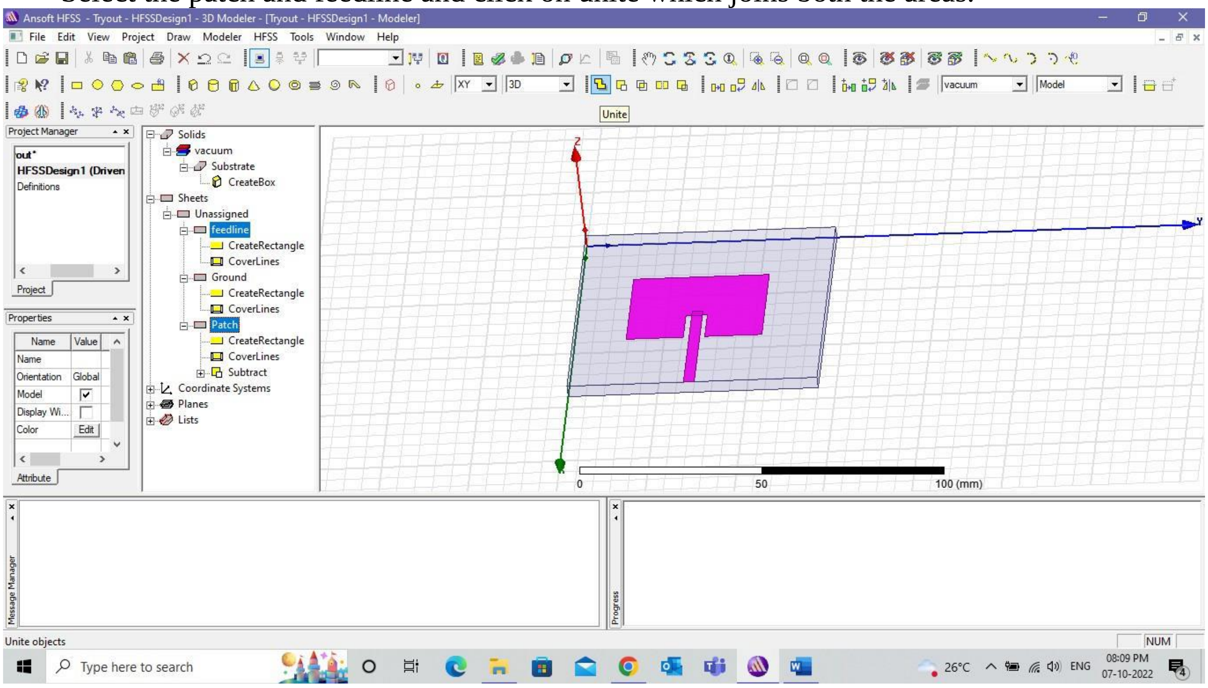




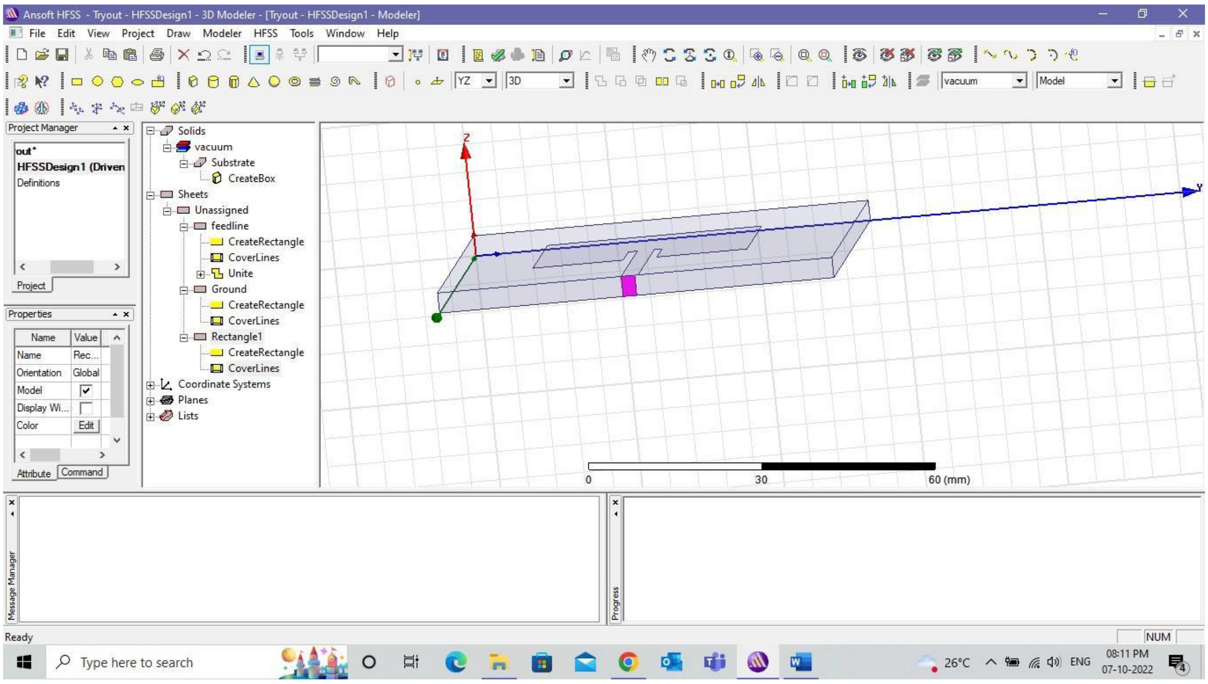
- Choose the rectangle tool and draw small lengthy rectangle joining the patch and edge



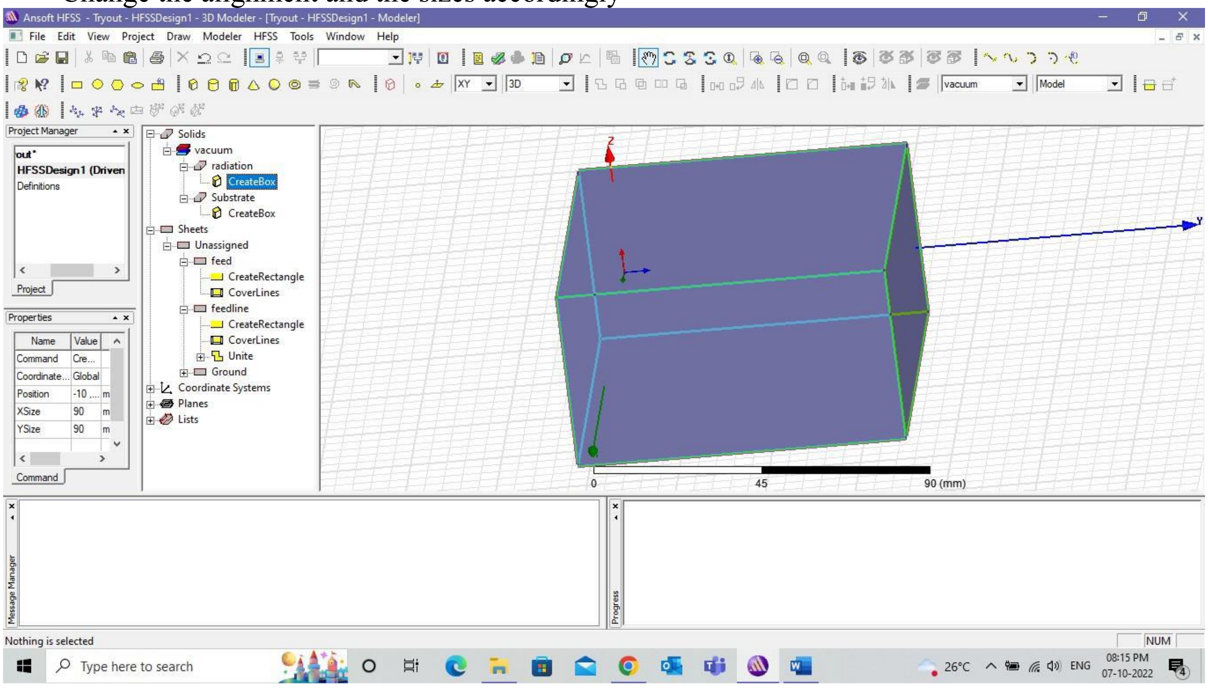
- Change the position and x, y sizes accordingly
- • Select the patch and feedline and click on unite which joins both the areas.



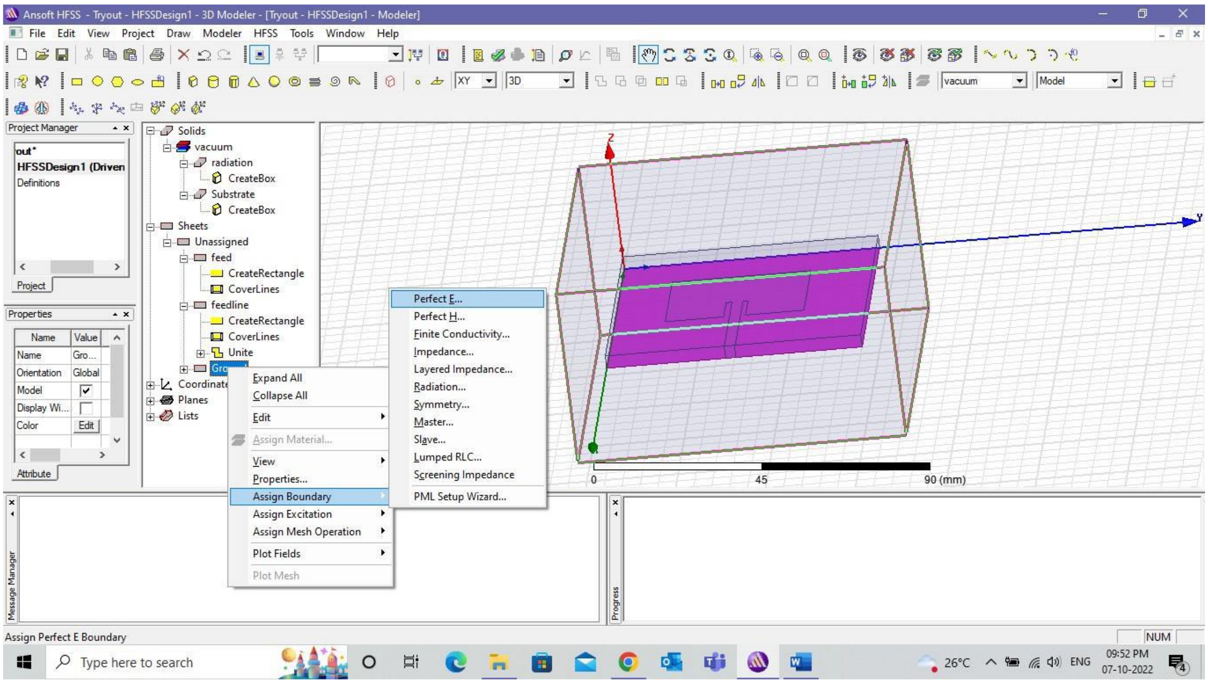
- Change the alignment to YZ plane and draw a small rectangle and join with feedline.



- Create a cube enclosing all the design elements and name it as radiation box
- • Change the alignment and the sizes accordingly



- Select the ground and chose the option assign boundary and select prefect E



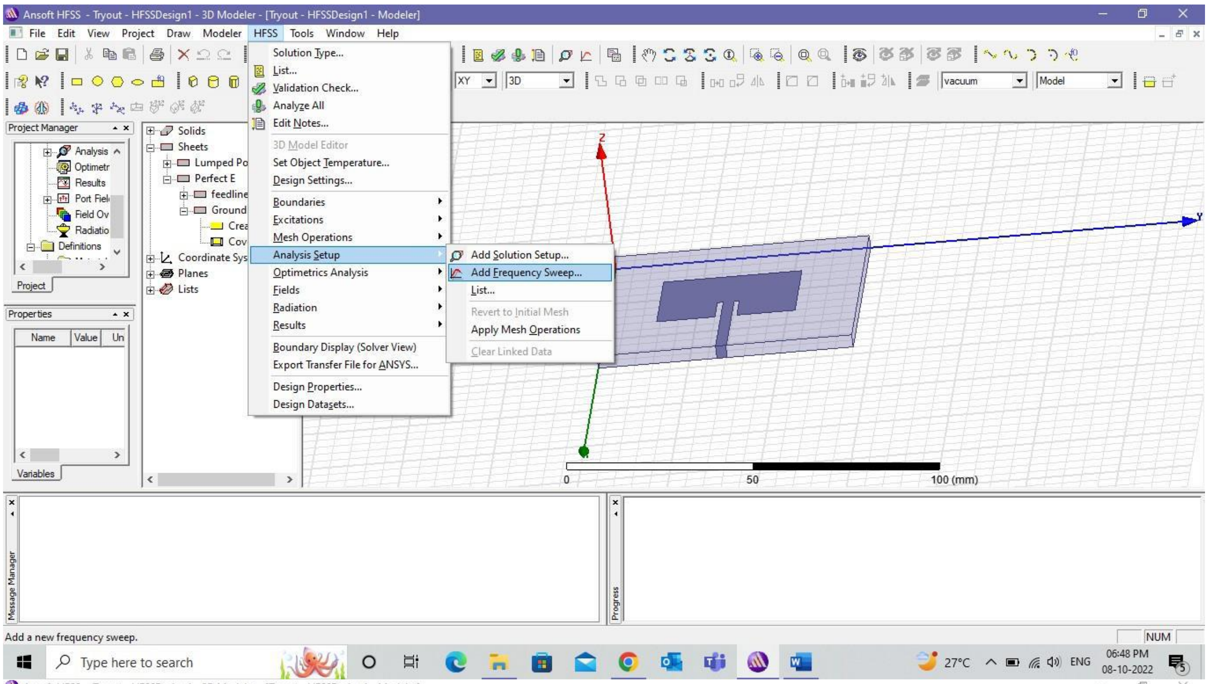
- Repeat the same steps for Patch
- • Select radiation box and go to assign boundary and select radiation
- • Select the substrate and change the material to fr4_Epoxy in its properties.
- • Go to HFSS->Solution type and change it to terminal driven
- • Go to Fed->assign excitation, select it as lumped port, chose the ground as reference
- • Go to HFSS->Analysis setup->Solution step, set solution frequency to 2.45Ghz, number of passes 20.

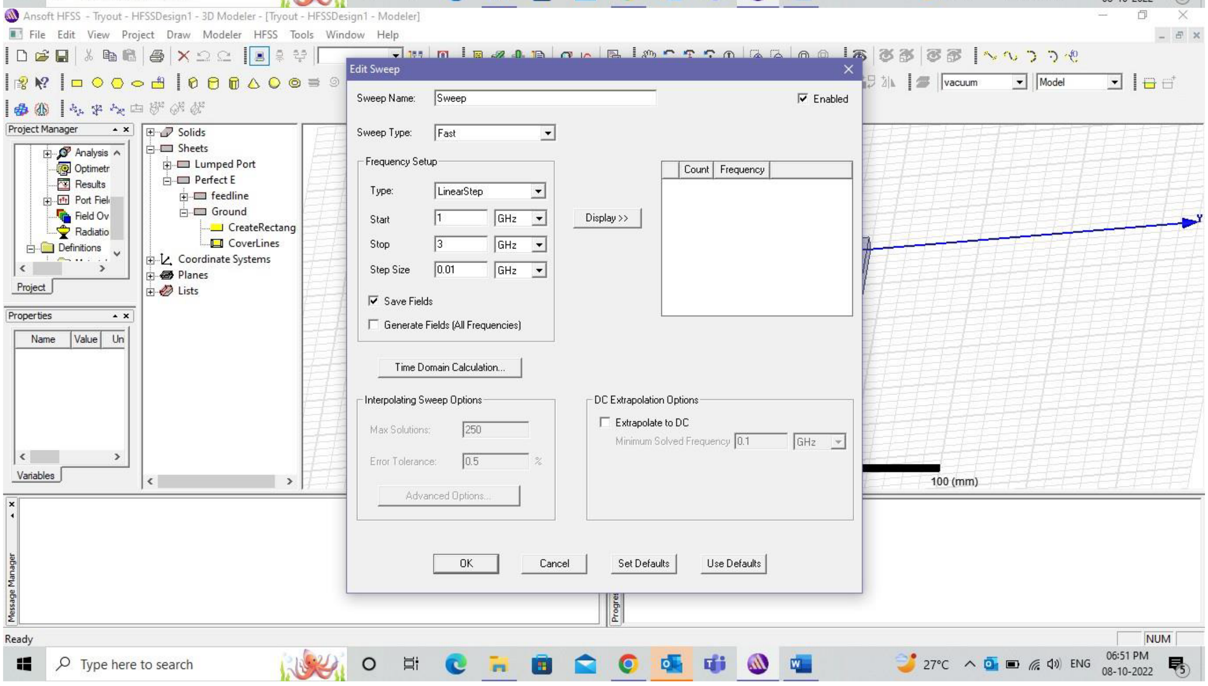
• Go to HFSS->Analysis setup->frequency sweep, • Sweep-type=fast

• Start=1GHz

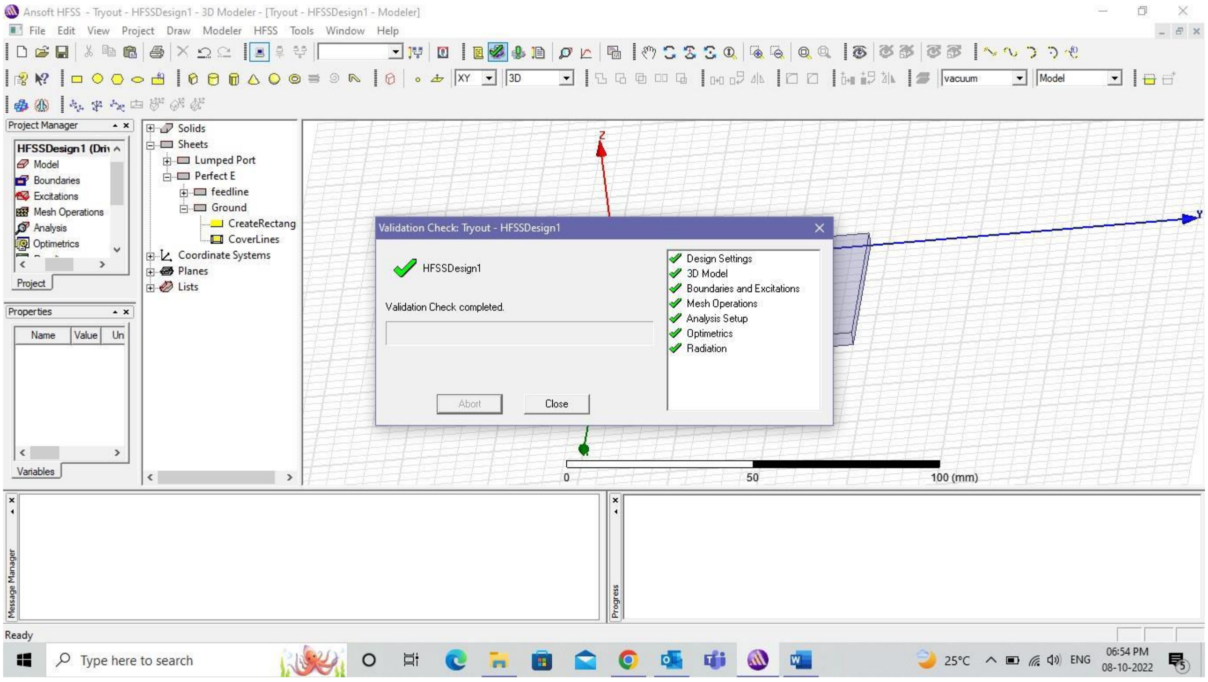
• Stop=3Ghz

• Step size = 0.01GHz

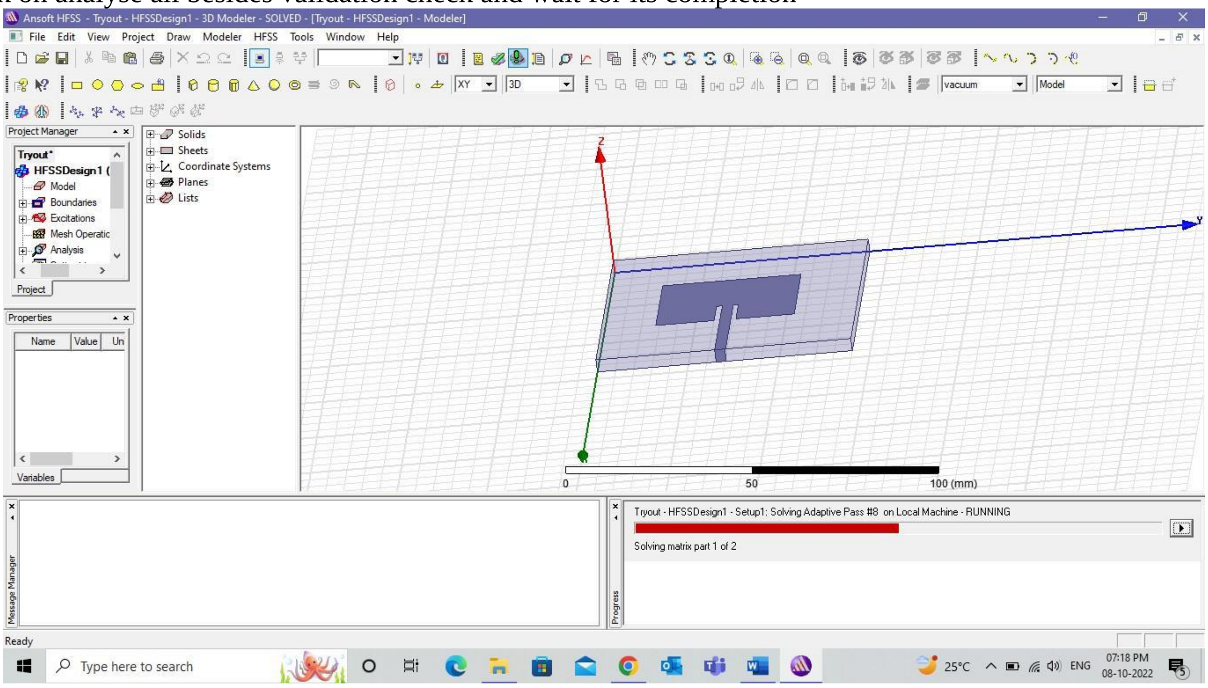




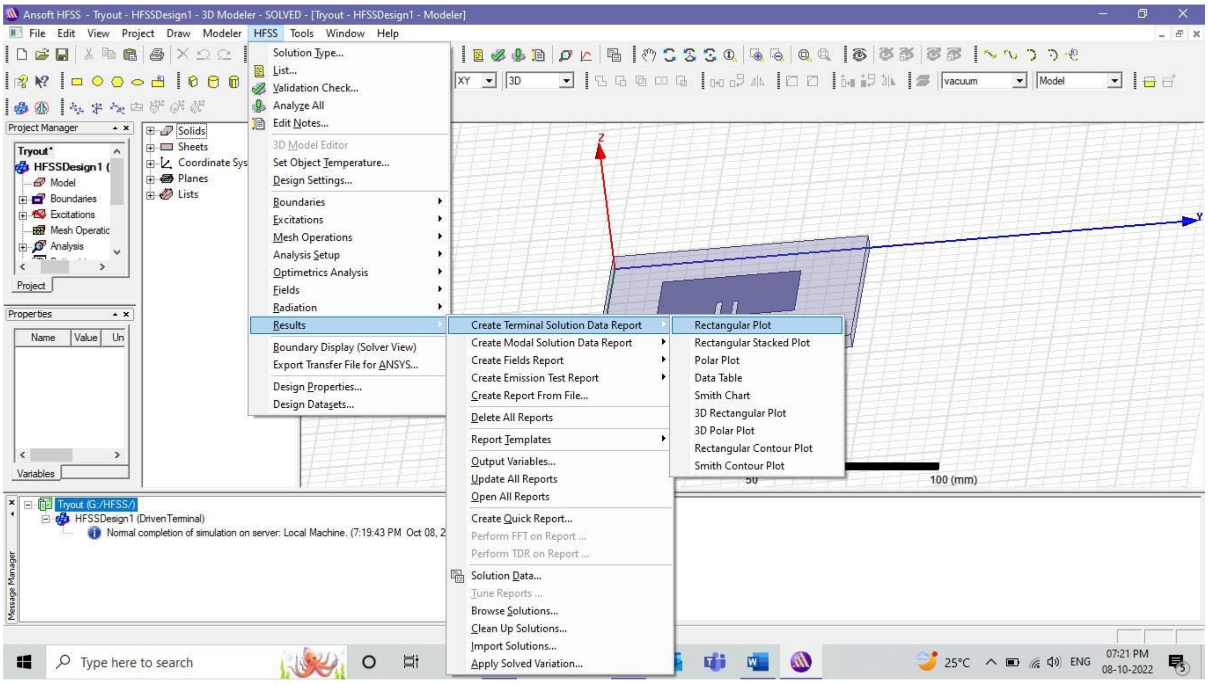
- Click on the validation check and check for the errors.



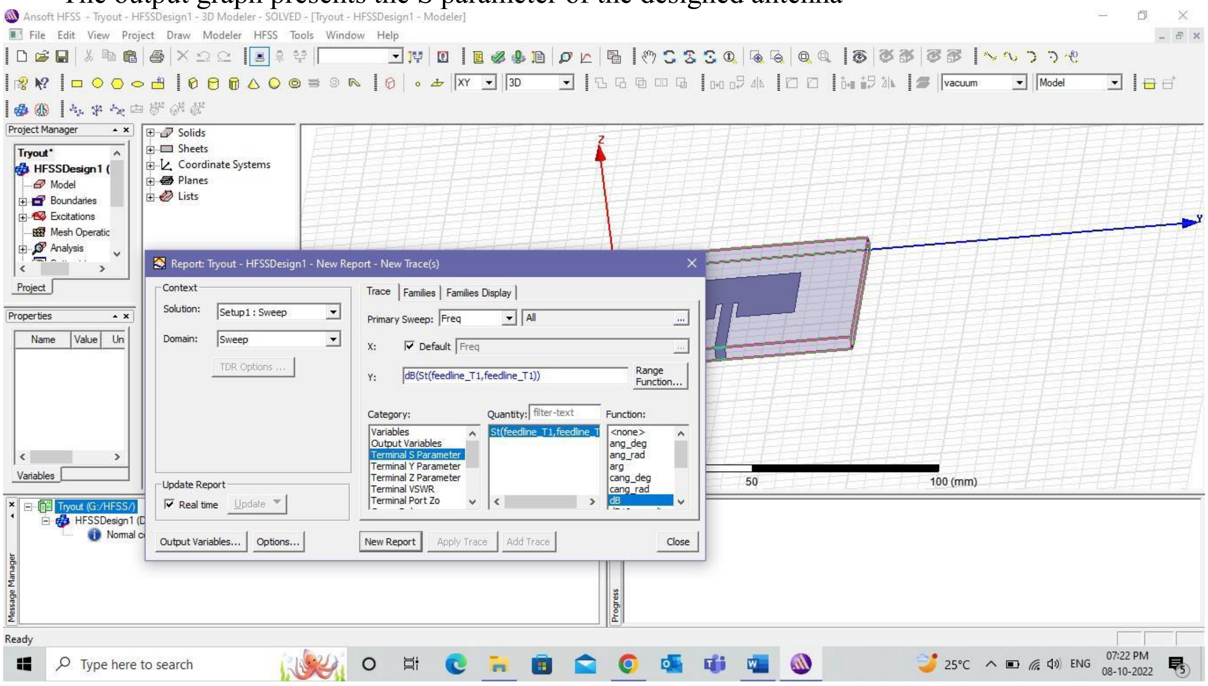
Click on analyse all besides validation check and wait for its completion

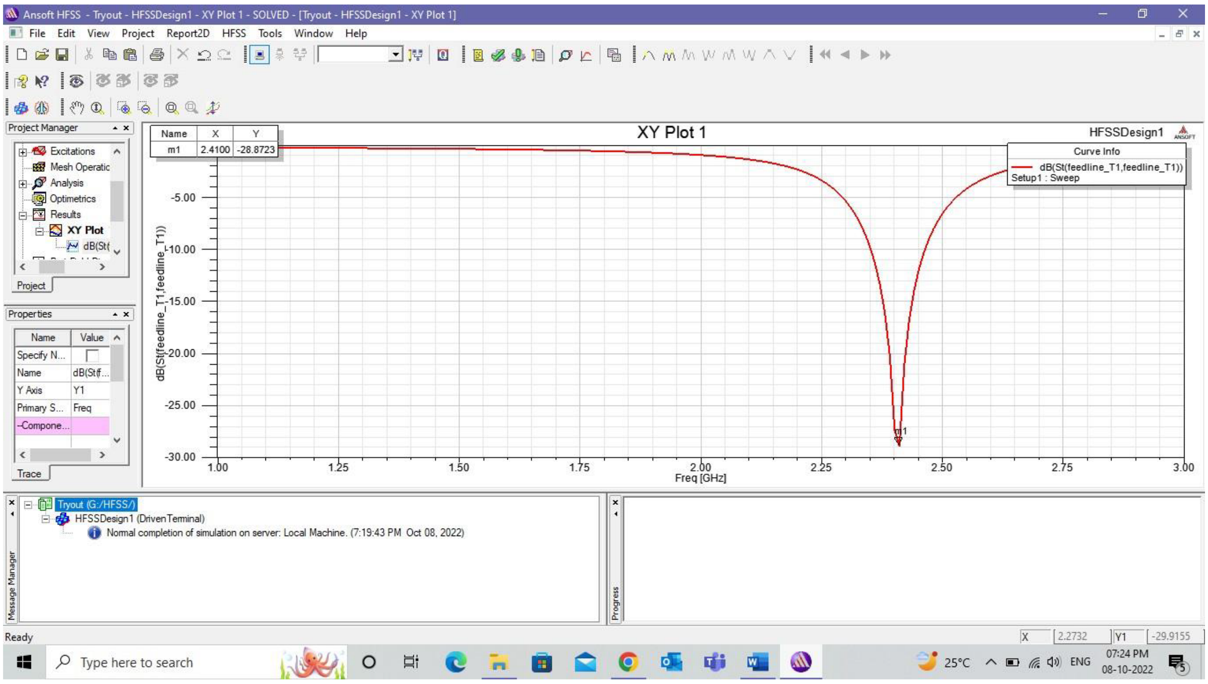


- Go to HFSS->results->create terminal solution data report->rectangular plot

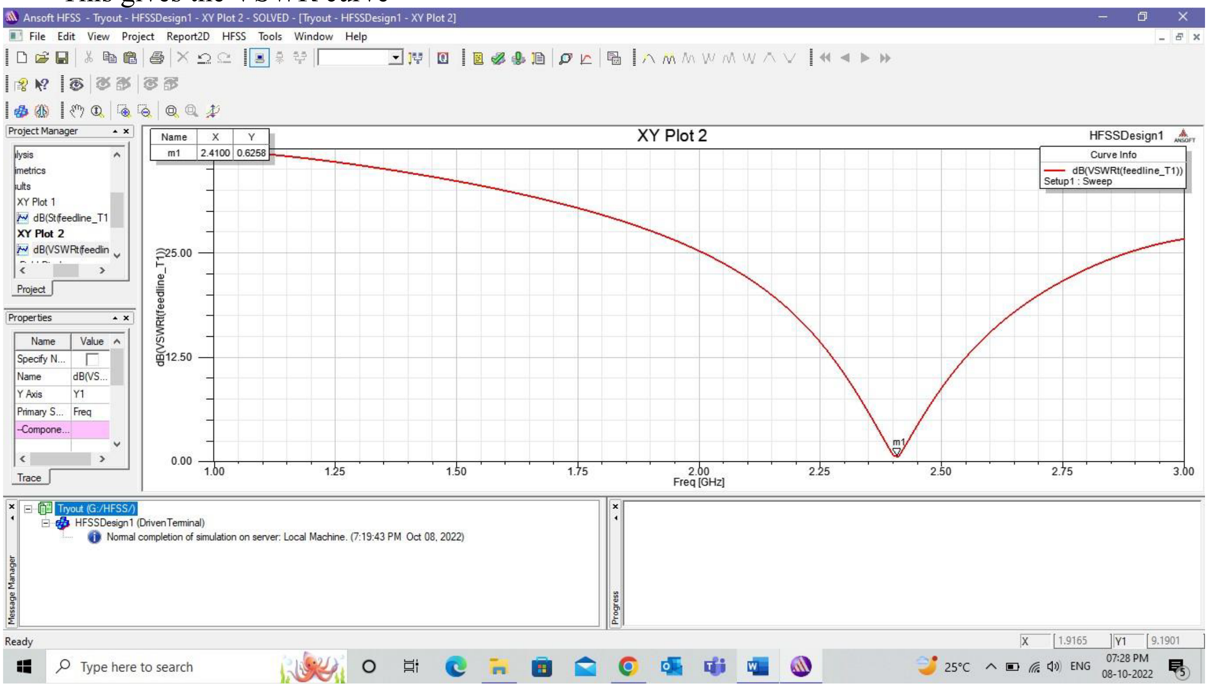


- Select category=terminal S parameter, function=DB and click new report
- • The output graph presents the S parameter of the designed antenna

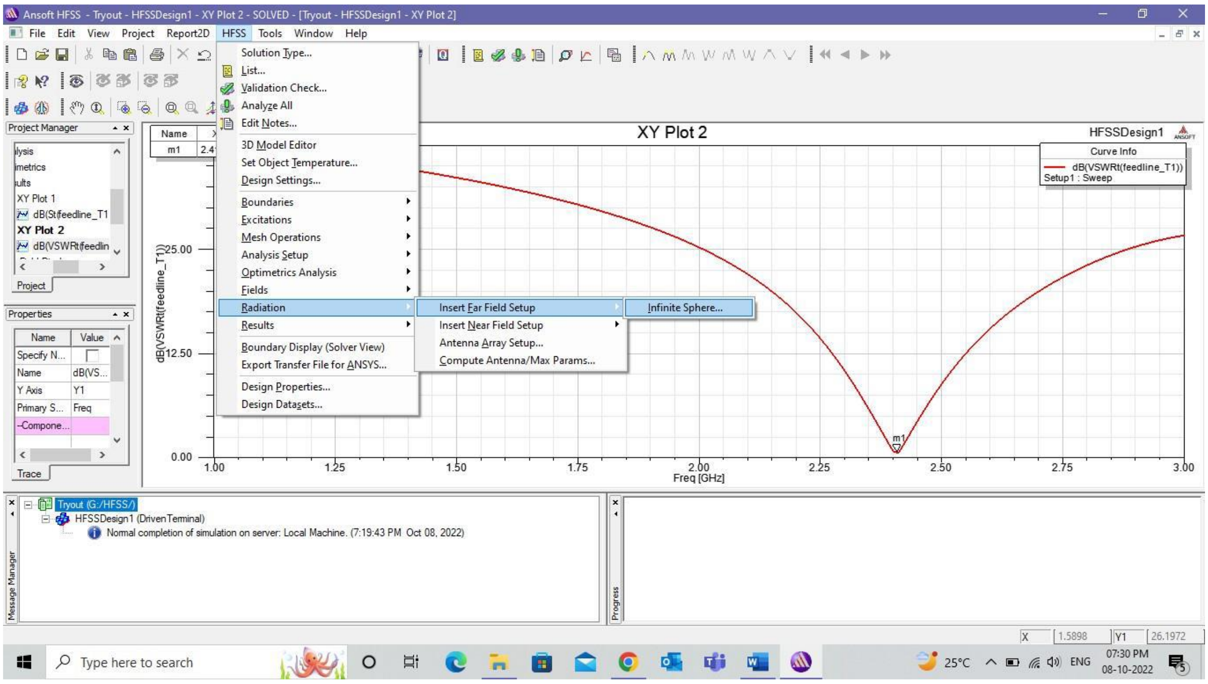


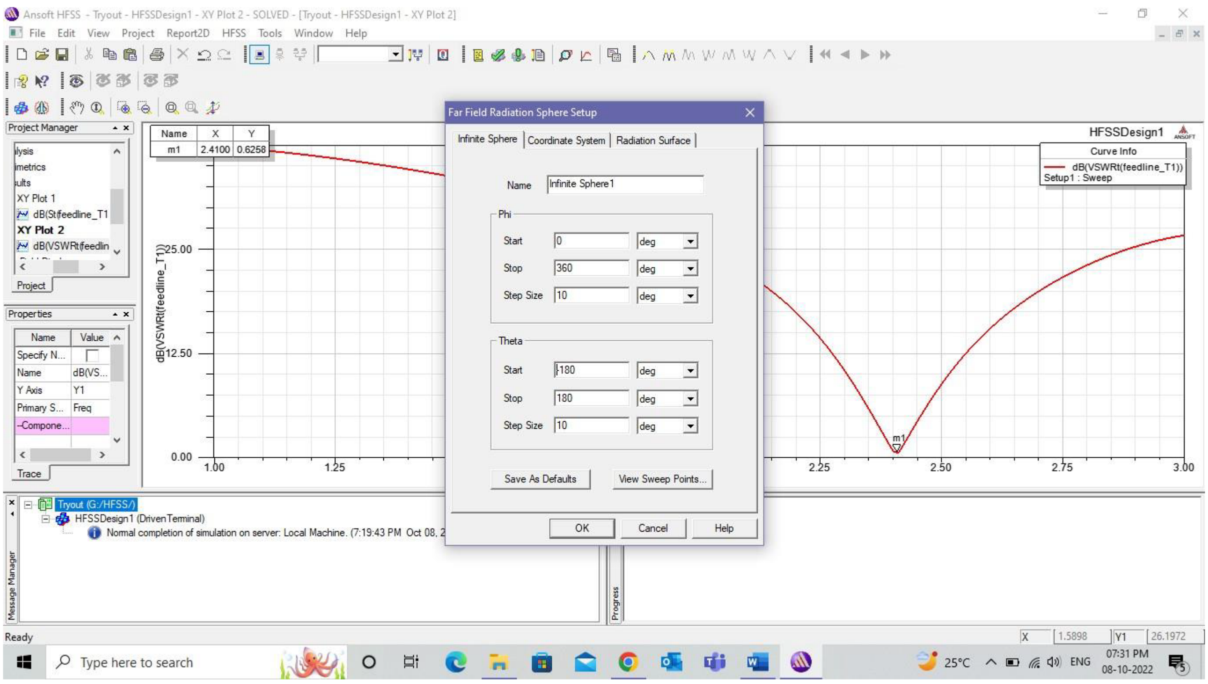


- Repeat the same step of results and select category=terminal VSWR
- • This gives the VSWR curve

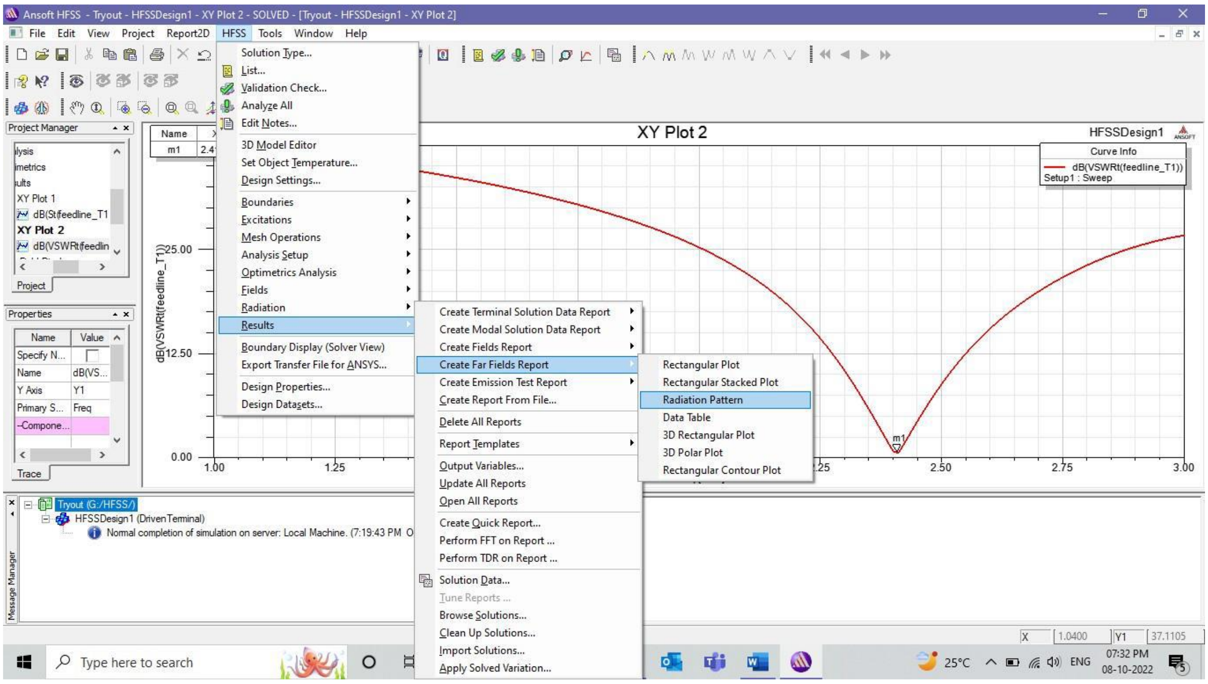


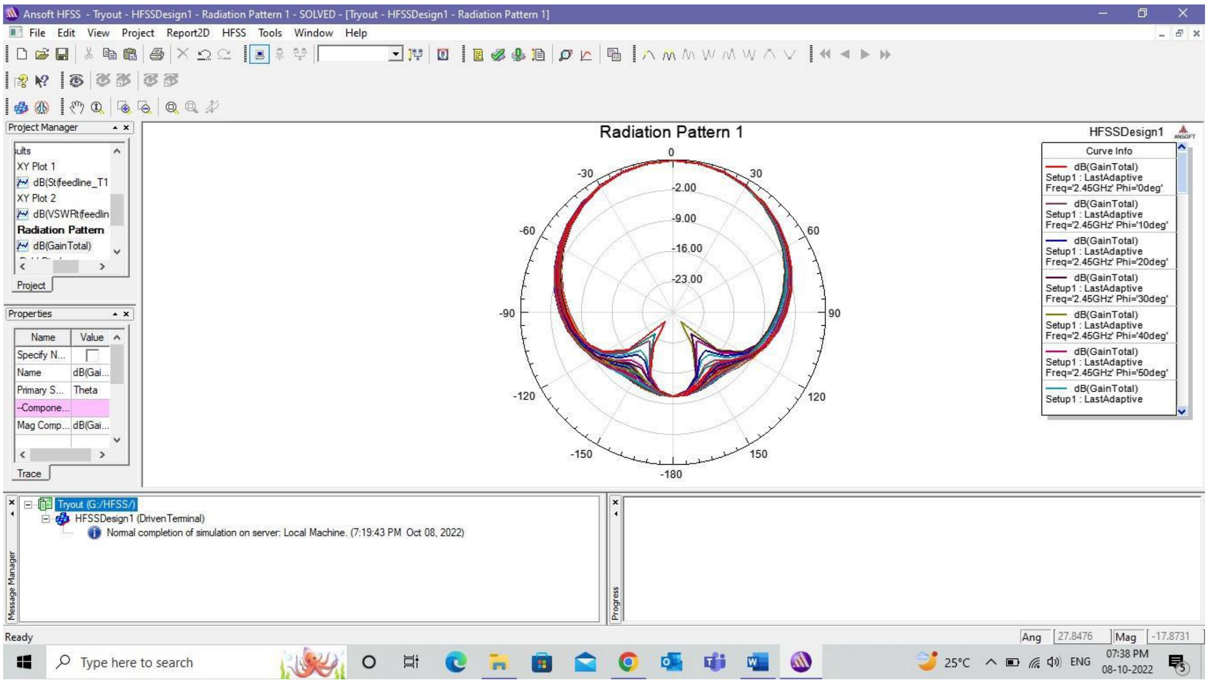
- Go to HFSS->radiation->insert far field setup->Infinite sphere
- Change phi from 0 to 360 and theta from -180 to +180



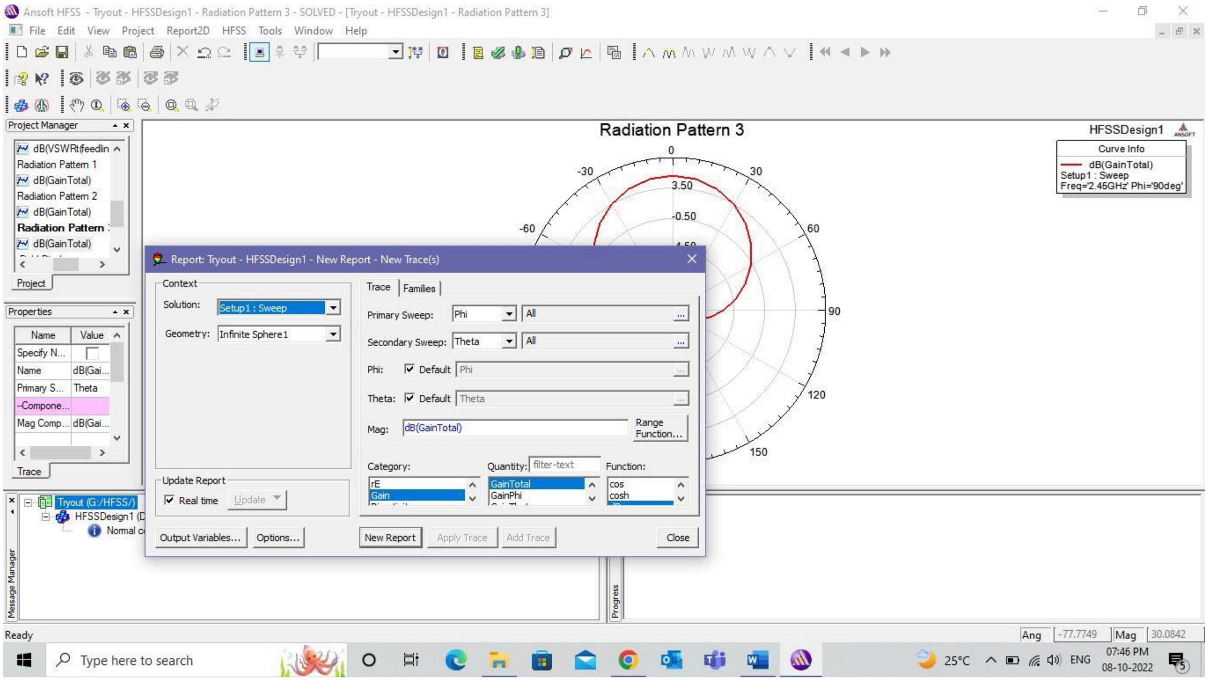


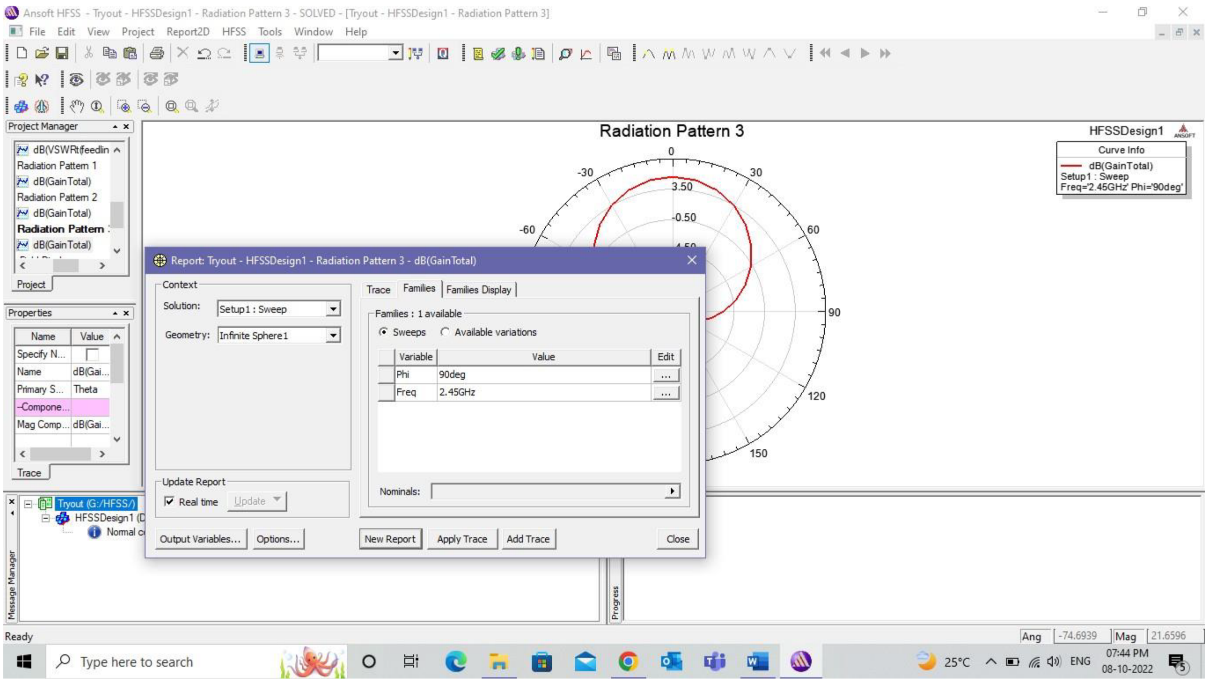
- Go to HFSS->results->create far field report->radiation pattern
- • Select category=gain, quantity=gain total, function=DB
- • The generated graph shows the radiation pattern of the designed antenna
- • This shows for all the frequencies

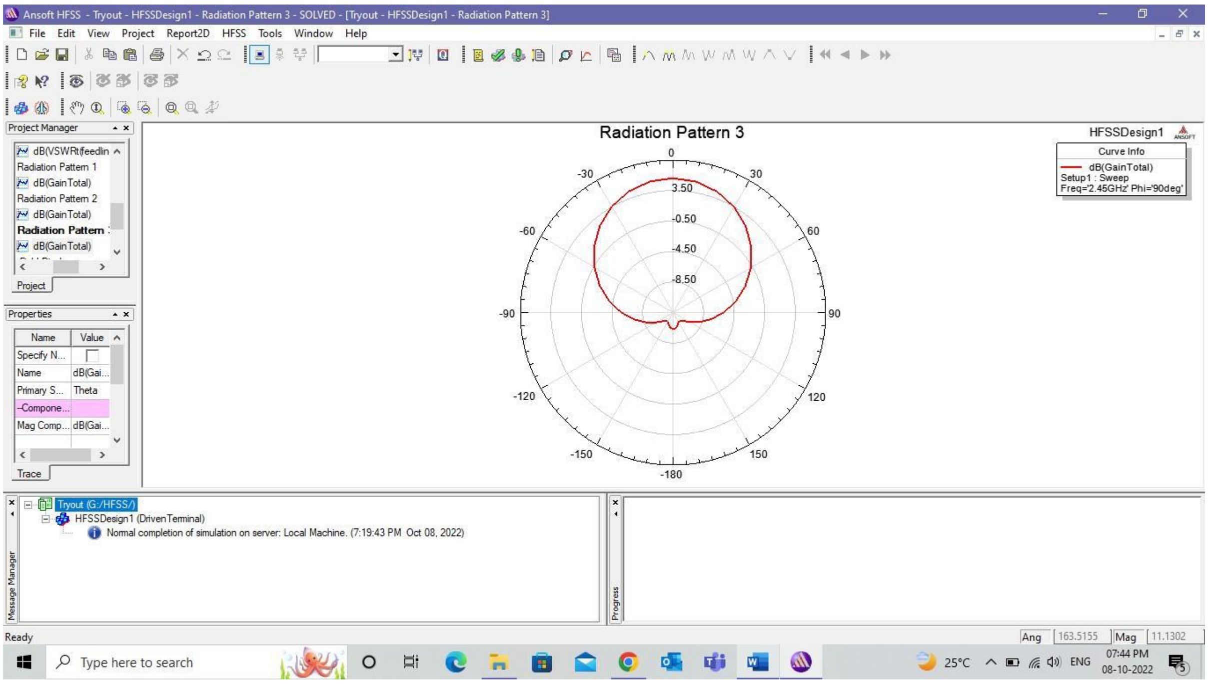




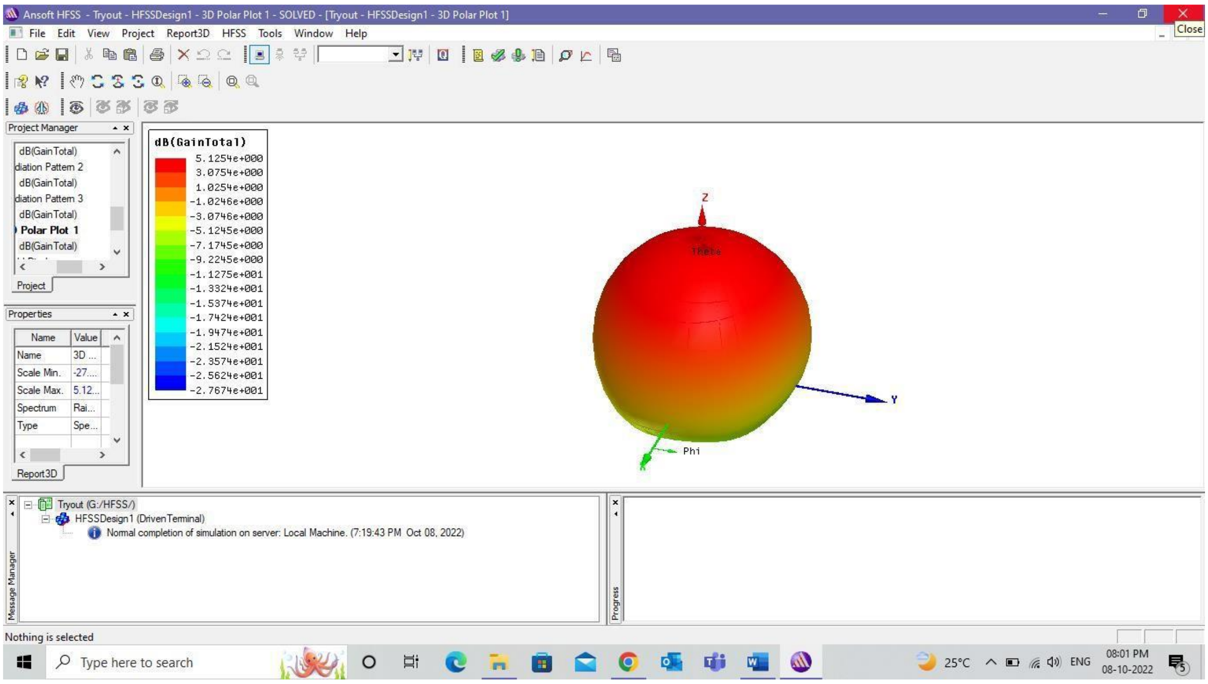
- Got to modify report and select families
- • Change the solution to sweep1, change phi to 90 degrees


Ansoft HFSS - Tryout - HFSSDesign1 - Radiation Pattern 3 - SOLVED - [Tryout - HFSSDesign1 - Radiation Pattern 3]
Radiation Pattern 3
HFSSDesign1
Curve Info
dB(GainTotal)
Setup1 : Sweep
Freq='2.45GHz' Phi='90deg'
Report: Tryout - HFSSDesign1 - New Report - New Trace(s)
Context
Solution: Setup1 : Sweep
Geometry: Infinite Sphere1
Trace
Families
Primary Sweep: Phi
Secondary Sweep: Theta
Mag: dB(GainTotal)
Category:
Quantity:
Function:
Update Report
Real time
Output Variables...
Options...
New Report
Apply Trace
Add Trace
Close
Ready


Ansoft HFSS - Tryout - HFSSDesign1 - Radiation Pattern 3 - SOLVED - [Tryout - HFSSDesign1 - Radiation Pattern 3]
Radiation Pattern 3
HFSSDesign1
Curve Info
dB(GainTotal)
Setup1 : Sweep
Freq='2.45GHz' Phi='90deg'
Report: Tryout - HFSSDesign1 - Radiation Pattern 3 - dB(GainTotal)
Context
Solution: Setup1 : Sweep
Geometry: Infinite Sphere1
Trace
Families
Families Display
Families : 1 available
Sweeps
Available variations
Variable
Value
Edit
Phi
90deg
Freq
2.45GHz
Nominals:
Update Report
Real time
Output Variables...
Options...
New Report
Apply Trace
Add Trace
Close
Ready



- Go to HFSS->results->create far field report->3D polar plot
- • Select category=gain, quantity=gain total, function=DB
- • Change the solution to sweep1, change phi to 90 degrees
- • This shows the 3d plot of the gain.



Conclusion: The basic patch antenna with a rectangular patch is designed and it's performance is measured using s11 parameter and VSWR plots.

References:

1. **Simulation of Digital Communication Systems using MATLAB: Mathuranathan Viswanathan. 2013.**
2. **R. W. Heath Jr. Introduction to Wireless Digital Communication: A Signal Processing Perspective, course notes for EE 371C taught at The University of Texas at Austin.**
3. **R. W. Heath Jr. EE 371C / EE 381V: Wireless Communications Lab.**
4. **HFSS tool.**