# INDIAN INSTITUTE OF INFORMATION TECHNOLOGY

**SURAT**

# LABORATORY MANUAL

**ELECTRONICS & COMMUNICATION ENGINEERING DEPARTMENT**

**EC: 303: COMMUNICATION ENGINEERING**

**B.TECH II-SEMESTER III**

NAME: JITANSHU J. RAUT

ROLL NO: UI21CS27

BRANCH: CSE B1

# INDIAN INSTITUTE OF INFORMATION AND TECHNOLOGY

*Certificate*

This is to certify that Mr./Ms. JITANSHU_J. RAUT of 2nd year 3rd sem Class Roll No. UI21CS27 has Satisfactory completed the course in Communication Engineering laboratory practical during the Year 20222023 .

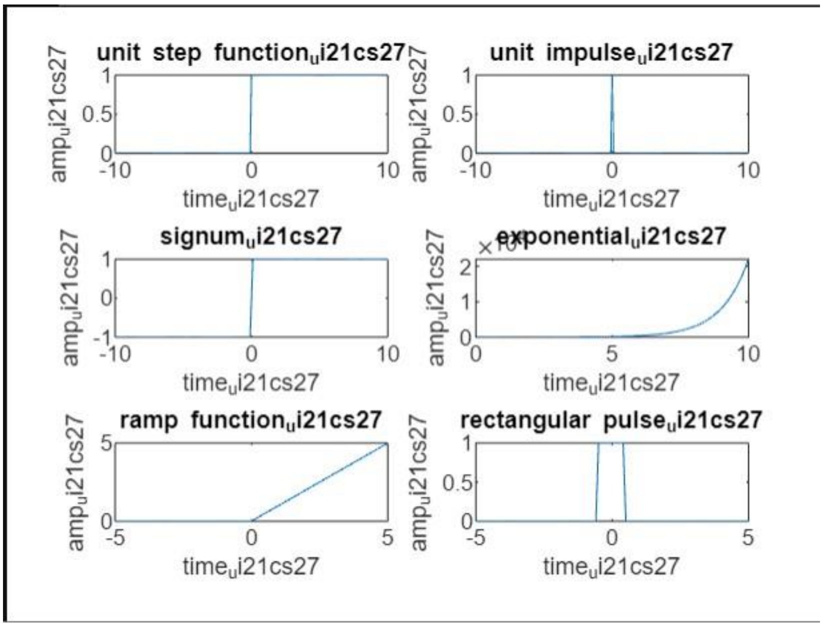
**MS. SEJAL RATHOD** **MS. SEJAL**
**RATHOD**
**(COURSE COORDINATOR)** **MR. RAHUL**
**PATEL**
**MR. DHIRAJ**
**PATEL**
**(LAB**
**INSTRUCTORS)**

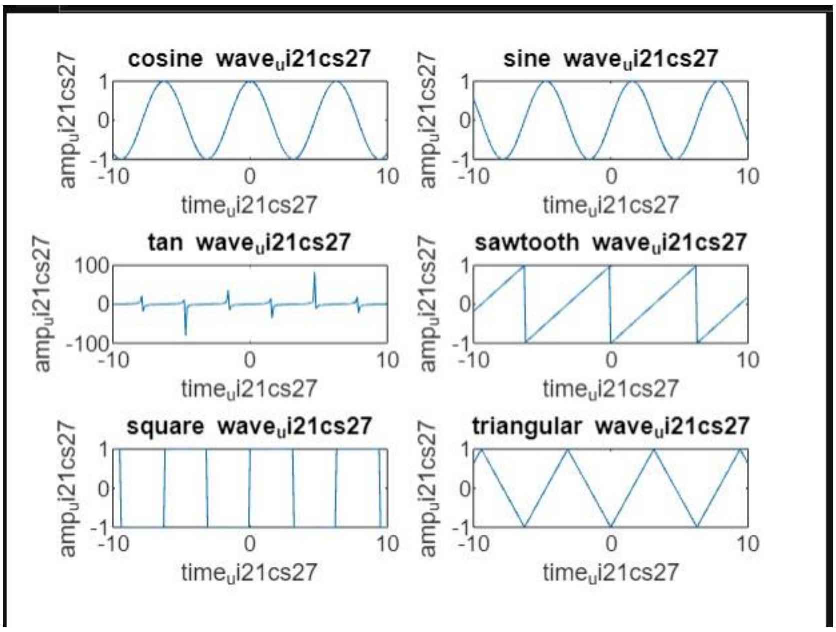
## Experiment_1:

```
t=[-10:0.1:10]; a=cos(t);
subplot(3,2,1); plot(t,a);
```

```
xlabel('time_ui21cs27');
ylabel('amp_ui21cs27');
title('cosine wave_ui21cs27');
 t=[-10:0.1:10]; a=sin(t);
subplot(3,2,2); plot(t,a);
xlabel('time_ui21cs27');
ylabel('amp_ui21cs27');
title('sine wave_ui21cs27');
  t=[-10:0.1:10]; a=tan(t);
subplot(3,2,3); plot(t,a);
xlabel('time_ui21cs27');
ylabel('amp_ui21cs27');
title('tan wave_ui21cs27');
 t=[-10:0.1:10]; a=sawtooth(t);
subplot(3,2,4); plot(t,a);
xlabel('time_ui21cs27');
ylabel('amp_ui21cs27');
title('sawtooth wave_ui21cs27');
 t=[-10:0.1:10]; a=square(t);
subplot(3,2,5); plot(t,a);
xlabel('time_ui21cs27');
ylabel('amp_ui21cs27');
title('square wave_ui21cs27');
 t=[-10:0.1:10];
a=sawtooth(t,0.5); subplot(3,2,6);
plot(t,a);
xlabel('time_ui21cs27');
ylabel('amp_ui21cs27');
title('triangular wave_ui21cs27');
t=[-10:0.1:10]; a=(t>=0);
subplot(3,2,1); plot(t,a);
xlabel('time_ui21cs27');
ylabel('amp_ui21cs27');
title('unit step
function_ui21cs27');
 t=[-10:0.1:10]; a=(t==0);
subplot(3,2,2); plot(t,a);
xlabel('time_ui21cs27');
ylabel('amp_ui21cs27');
title('unit impulse_ui21cs27');
  t=[-10:0.1:10];
a=sign(t);
subplot(3,2,3);
plot(t,a);
xlabel('time_ui21cs27');
ylabel('amp_ui21cs27');
title('signum_ui21cs27');
 t=[0:0.1:10]; a=exp(t);
subplot(3,2,4); plot(t,a);
xlabel('time_ui21cs27');
ylabel('amp_ui21cs27');
title('exponential_ui21cs27');
 a=[-5:0.1:5]; ramp_a=(a>=0).*a;
subplot(3,2,5); plot(a,ramp_a);
xlabel('time_ui21cs27');
ylabel('amp_ui21cs27');
title('ramp function_ui21cs27');
```

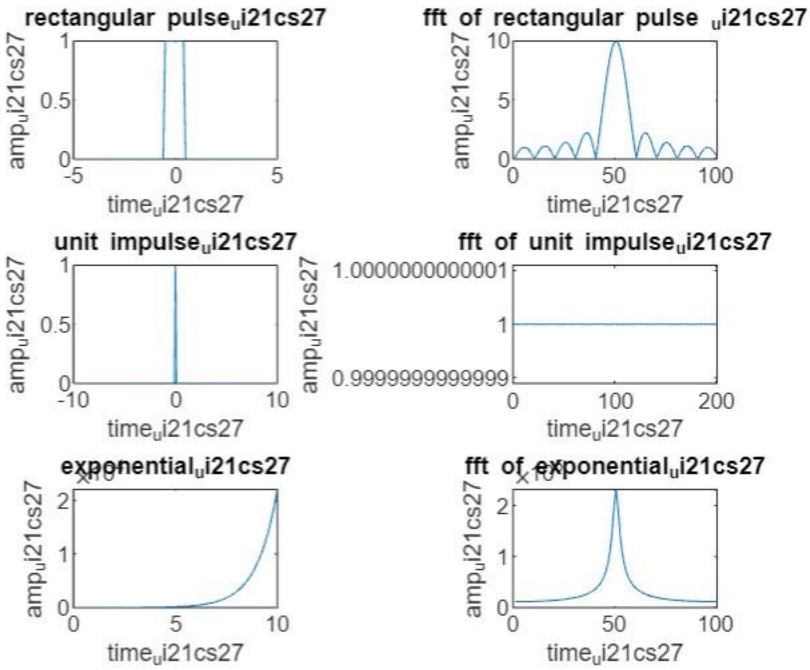
```
  t=[-5:0.1:5]; a=rectpuls(t);
subplot(3,2,6); plot(t,a);
xlabel('time_ui21cs27');
ylabel('amp_ui21cs27');
title('rectangular pulse_ui21cs27');
```



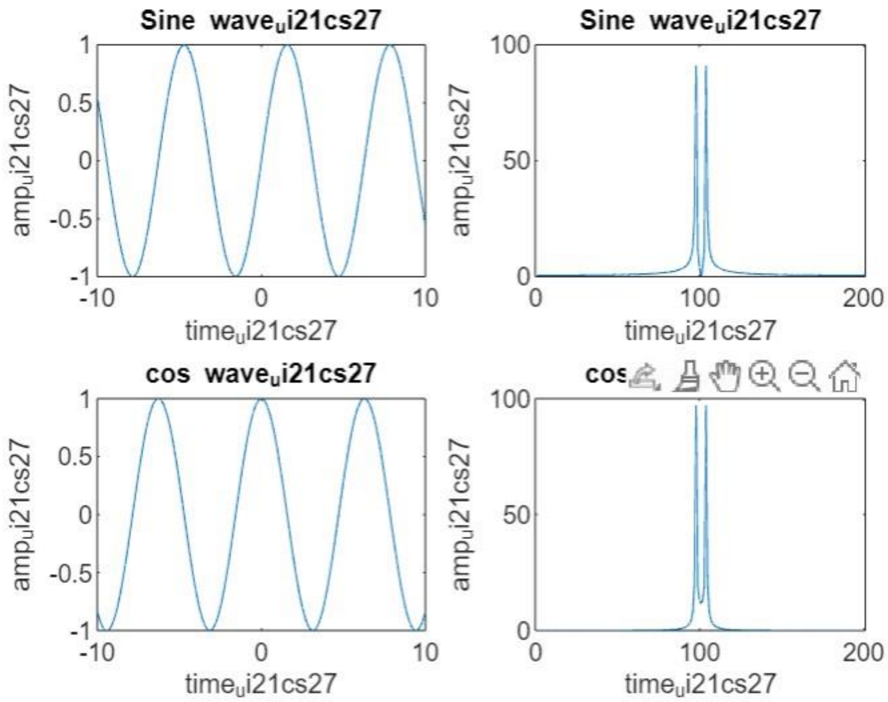



# Experiment_2:

```
t=[-5:0.1:5]; a=rectpuls(t);
subplot(3,2,1); plot(t,a);
xlabel('time_ui21cs27');
ylabel('amp_ui21cs27');
title('rectangular pulse_ui21cs27');

x=fftshift(abs(fft(a)));
subplot(3,2,2); plot(x);
xlabel('time_ui21cs27');
ylabel('amp_ui21cs27');
title('fft of rectangular pulse _ui21cs27');
  t=[-10:0.1:10]; a=(t==0);
subplot(3,2,3); plot(t,a);
xlabel('time_ui21cs27');
ylabel('amp_ui21cs27');
title('unit impulse_ui21cs27');
  x=fftshift(abs(fft(a)));
subplot(3,2,4); plot(x);
xlabel('time_ui21cs27');
ylabel('amp_ui21cs27'); title('fft of
unit impulse_ui21cs27');
  t=[0:0.1:10]; a=exp(t);
subplot(3,2,5); plot(t,a);
xlabel('time_ui21cs27');
ylabel('amp_ui21cs27');
title('exponential_ui21cs27');
  x=fftshift(abs(fft(a)));
subplot(3,2,6); plot(x);
xlabel('time_ui21cs27');
ylabel('amp_ui21cs27'); title('fft of
exponential_ui21cs27');
```




```
t=[-10:0.1:10]; y=sin(t);
subplot(2,2,1); plot(t,y);
xlabel('time_ui21cs27');
ylabel('amp_ui21cs27');
title('Sine wave_ui21cs27');
x=fftshift(abs(fft(y)));
subplot(2,2,2); plot(x);
xlabel('time_ui21cs27');
ylabel('amp_ui21cs27');
title('Sine wave_ui21cs27');
 t=[-10:0.1:10]; y=cos(t);
subplot(2,2,3); plot(t,y);
xlabel('time_ui21cs27');
ylabel('amp_ui21cs27');
title('cos wave_ui21cs27');
x=fftshift(abs(fft(y)));
subplot(2,2,4); plot(x);
xlabel('time_ui21cs27');
ylabel('amp_ui21cs27');
title('cos wave_ui21cs27');
```

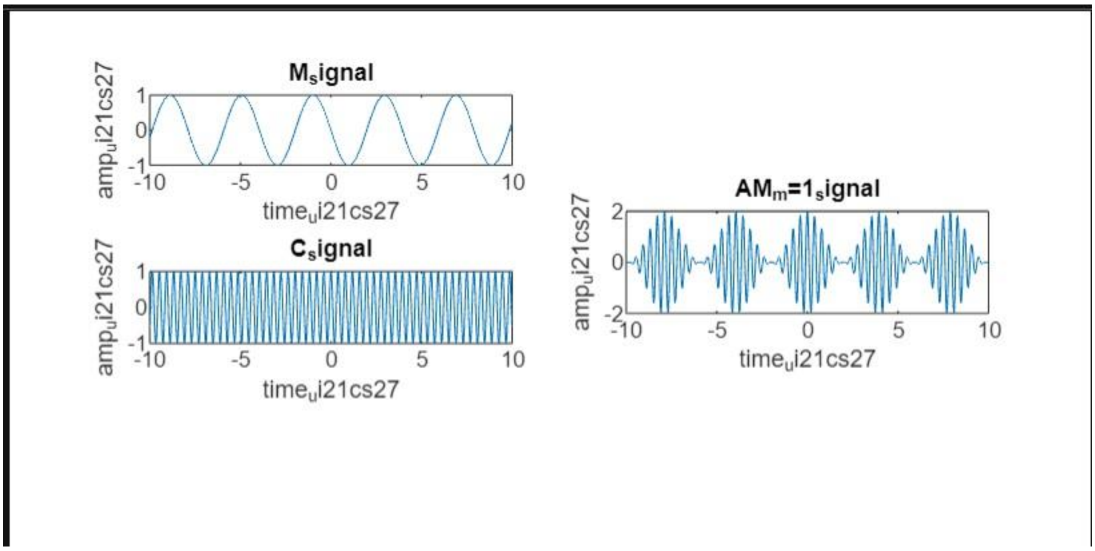



# Experiment_3:

## M==1

```
t=[-10:0.01:10]; em=1;
ec=1; p=3.14; m=em/ec;
fc=5000; fm=500;
%M_SIGNAL
a=sin(2*p*fm*t);
subplot(4,2,1);
plot(t,a);
xlabel('time_ui21cs27');
ylabel('amp_ui21cs27');
title('M_signal');
%c_signal
```

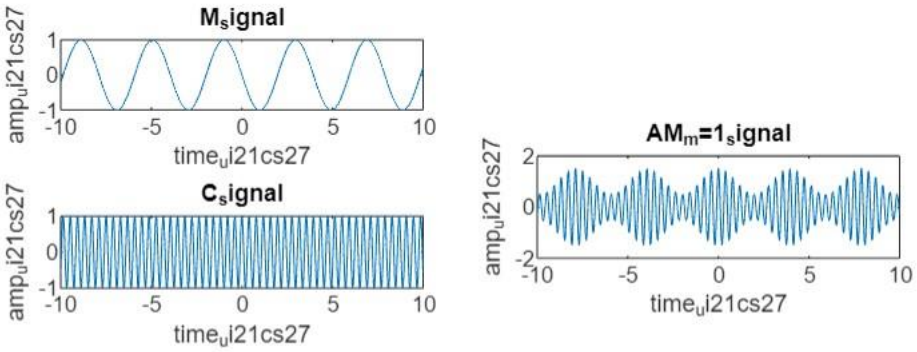
```
b=ec*cos(2*p*fc*t);
subplot(4,2,3);
plot(t,b);
xlabel('time_ui21cs27');
ylabel('amp_ui21cs27');
title('C_signal');
 %AM
s=ec*(1+m.*cos(2*p*fm*t)).*cos(2*p*fc*t);
subplot(5,2,4); plot(t,s);
xlabel('time_ui21cs27');
ylabel('amp_ui21cs27');
title('AM_m==1_signal');
```




# M<1

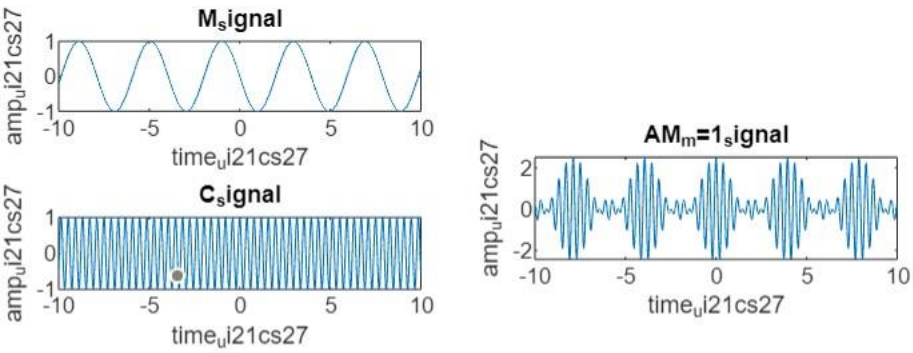
```
t=[-10:0.01:10]; em=0.5;
ec=1; p=3.14; m=em/ec;
fc=5000; fm=500;
%M_SIGNAL
a=sin(2*p*fm*t);
subplot(4,2,1);
plot(t,a);
xlabel('time_ui21cs27');
ylabel('amp_ui21cs27');
title('M_signal');
 %c_signal
b=ec*cos(2*p*fc*t);
subplot(4,2,3);
plot(t,b);
xlabel('time_ui21cs27');
ylabel('amp_ui21cs27');
title('C_signal');
 %AM
s=ec*(1+m.*cos(2*p*fm*t)).*cos(2*p*fc*t);
subplot(5,2,4); plot(t,s);
xlabel('time_ui21cs27');
```

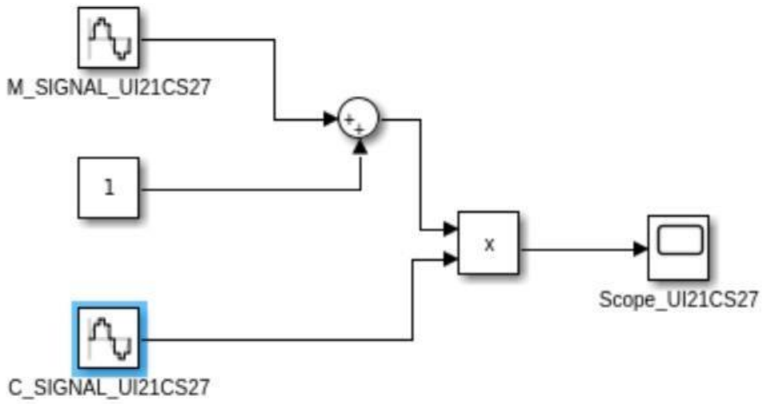
```
ylabel('amp_ui21cs27');
title('AM_m=1_signal');
```



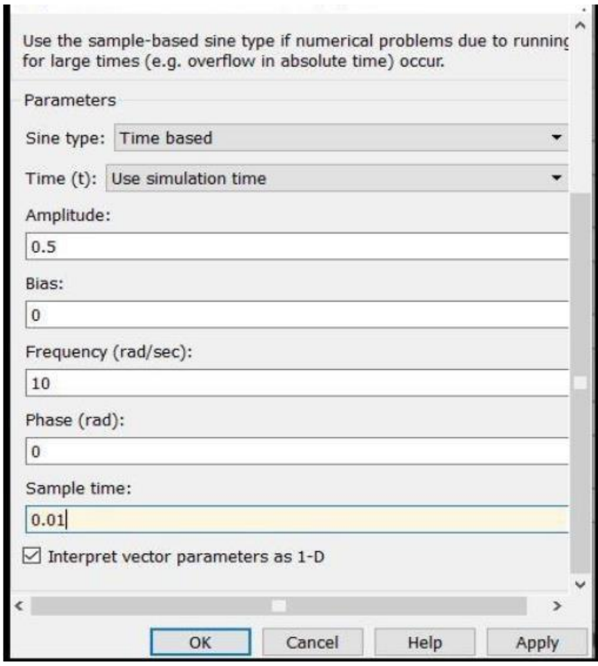

# M>1

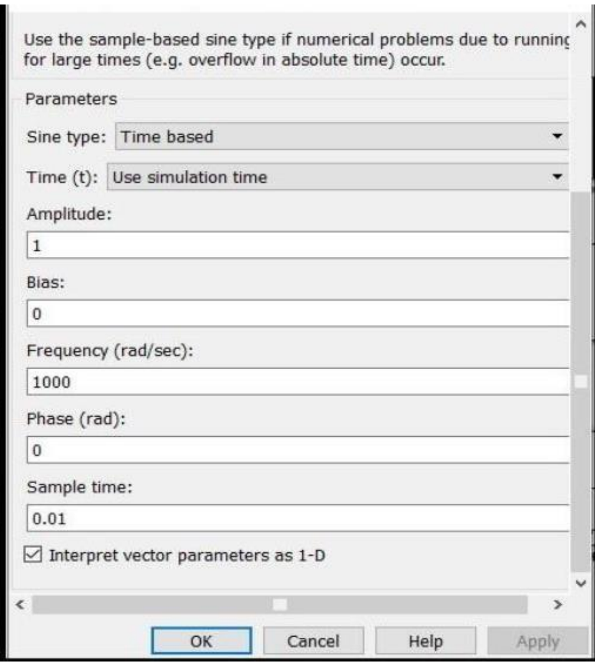
```
t=[-10:0.01:10]; em=1.5;
ec=1; p=3.14; m=em/ec;
fc=5000; fm=500;
%M_SIGNAL
a=sin(2*p*fm*t);
subplot(4,2,1);
plot(t,a);
xlabel('time_ui21cs27');
ylabel('amp_ui21cs27');
title('M_signal');
 %c_signal
b=ec*cos(2*p*fc*t);
subplot(4,2,3);
plot(t,b);
xlabel('time_ui21cs27');
ylabel('amp_ui21cs27');
title('C_signal');
 %AM
s=ec*(1+m.*cos(2*p*fm*t)).*cos(2*p*fc*t);
subplot(5,2,4); plot(t,s);
xlabel('time_ui21cs27');
ylabel('amp_ui21cs27');
title('AM_m=1_signal');
```

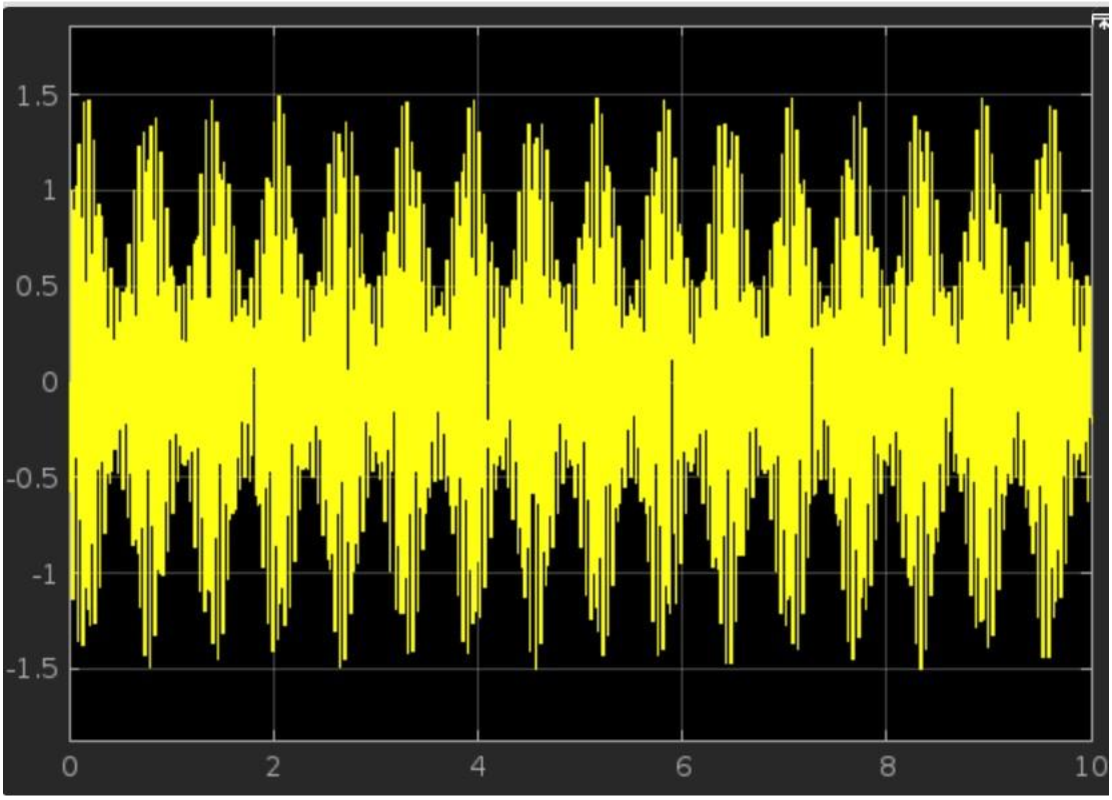


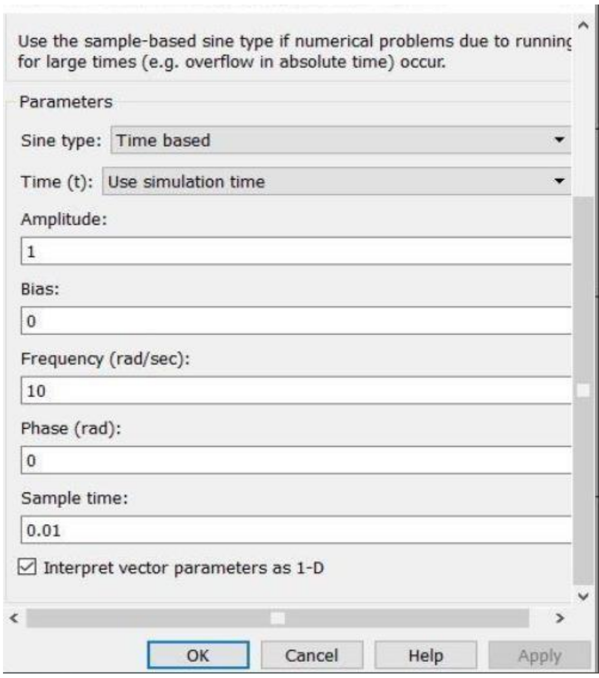

**LESS THAN 100%**

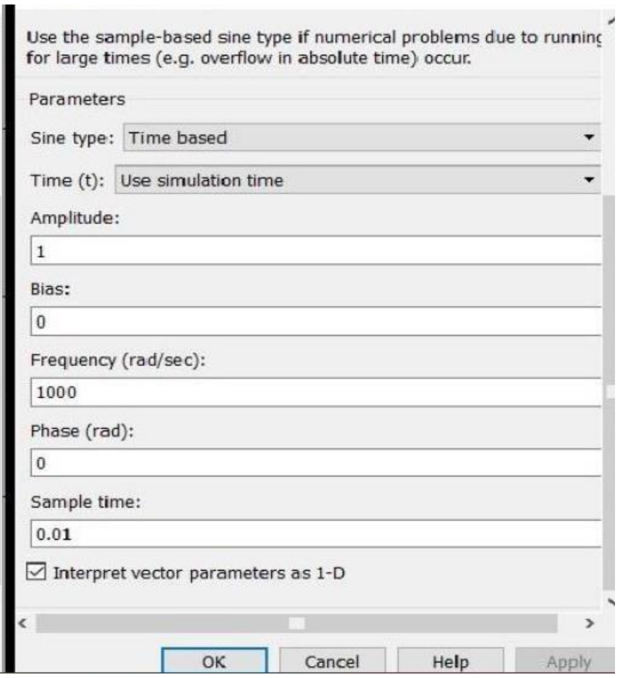


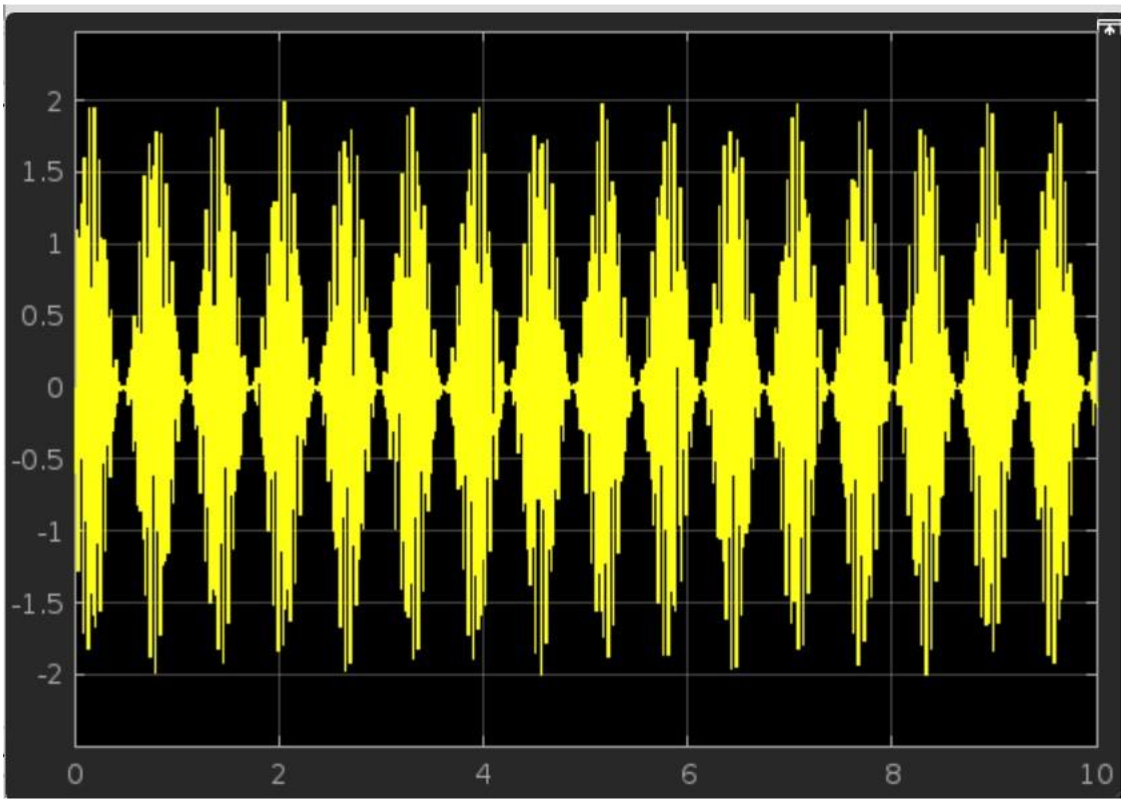



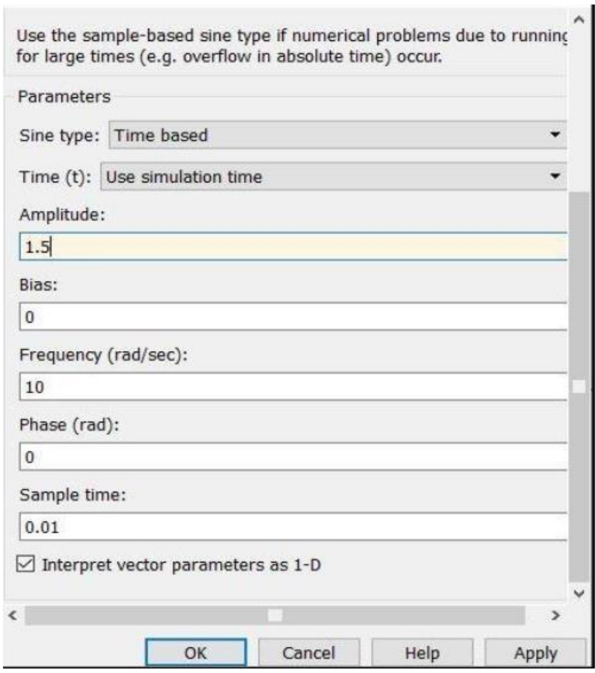

**100% MODULATION:**

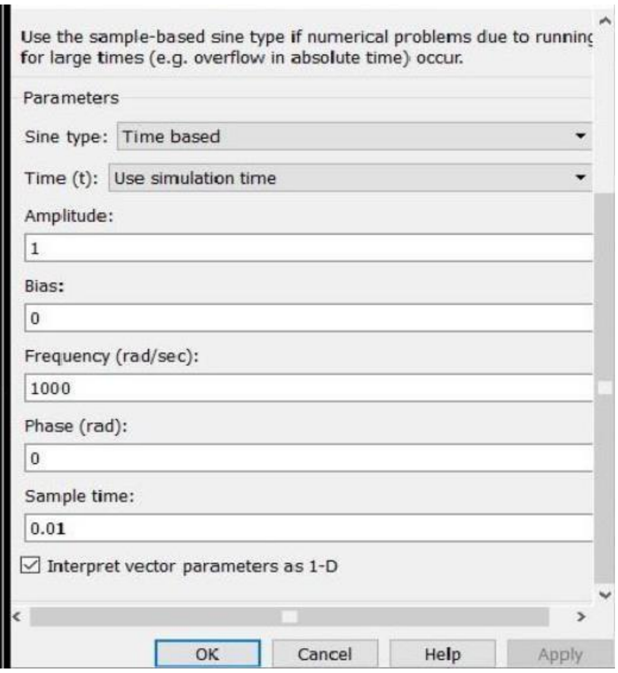


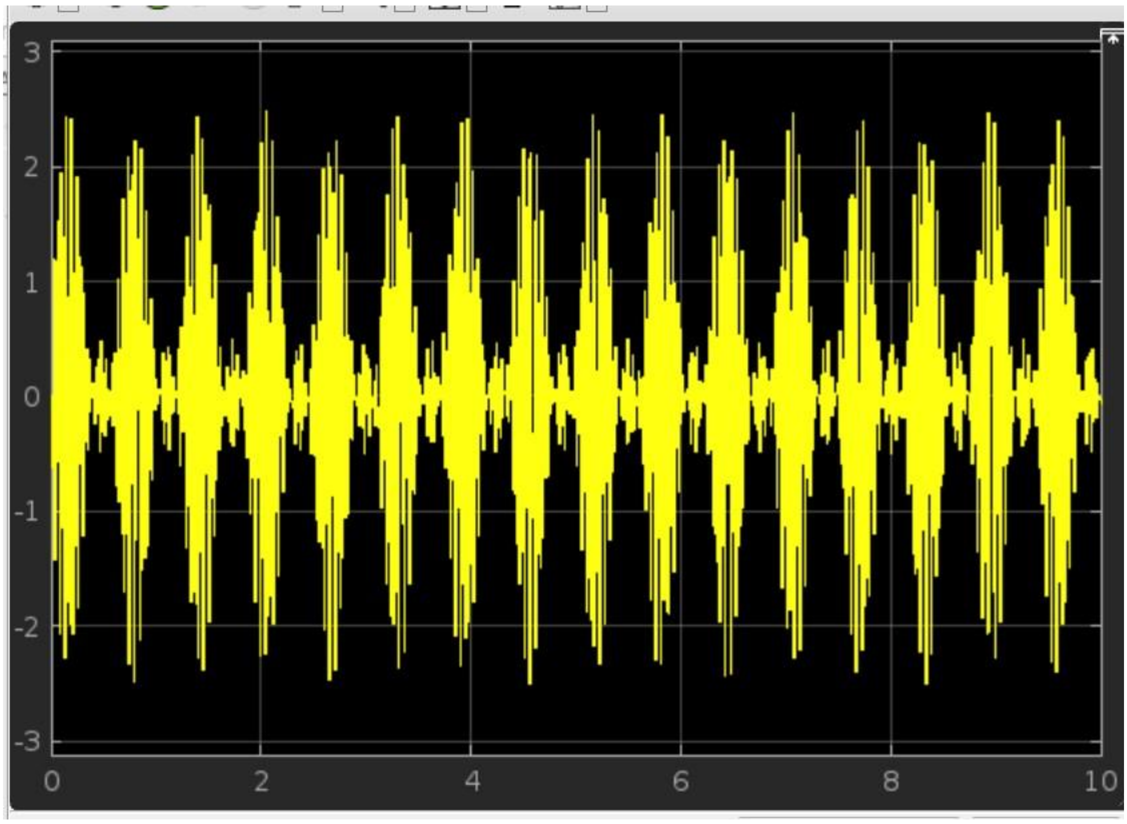




**MORE THAN 100% MODULATION:**


Use the sample-based sine type if numerical problems due to running for large times (e.g. overflow in absolute time) occur.
Parameters
Sine type: Time based
Time (t): Use simulation time
Amplitude:
1.5
Bias:
0
Frequency (rad/sec):
10
Phase (rad):
0
Sample time:
0.01
Interpret vector parameters as 1-D
OK
Cancel
Help
Apply


Use the sample-based sine type if numerical problems due to running for large times (e.g. overflow in absolute time) occur.
Parameters
Sine type: Time based
Time (t): Use simulation time
Amplitude:
1
Bias:
0
Frequency (rad/sec):
1000
Phase (rad):
0
Sample time:
0.01
Interpret vector parameters as 1-D
OK
Cancel
Help
Apply


3
2
1
0
-1
-2
-3
0
2
4
6
8
10

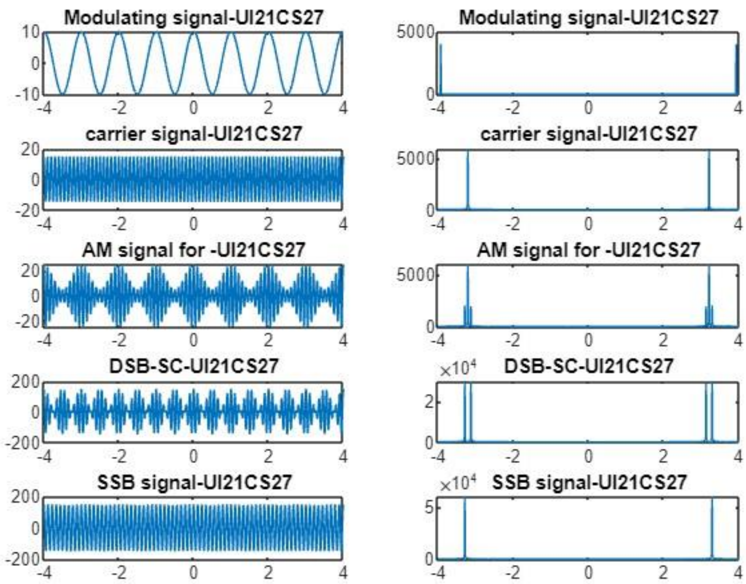
```
clc;
close all;
clear all;
Am=10;
```

## Experiment_4:

```
clc;
close all;
clear all;
Am=10;
```

```
Ac=15;  fm=1;
fc=10;
m=Am/Ac;
pi=3.14;  t=[-
4:0.01:4];
Em=Am*cos(2*pi*fm*t);
figure;
subplot(5,2,1);
plot(t,Em);
title('Modulating signal-UI21CS27');


a=abs(fft(Em));
subplot(5,2,2);
plot(t,a);
title('Modulating signal-UI21CS27');



Ec=Ac*cos(2*pi*fc*t);
subplot(5,2,3);
plot(t,Ec);
title('carrier signal-UI21CS27');
 subplot(5,2,3);
a=abs(fft(Ec));
subplot(5,2,4);
plot(t,a);
title('carrier signal-UI21CS27');


m=Am/Ac;

Eam=Ac*[1+m*cos(2*pi*fm*t)].*cos(2*pi*fc*t);

subplot(5,2,5);
plot(t,Eam);
title('AM signal for "m<1"-UI21CS27');


a=abs(fft(Eam));
subplot(5,2,6);
plot(t,a);
title('AM signal for -UI21CS27');


m=Am/Ac;

EDSB=[Ac*cos(2*pi*fc*t)].*[Am*cos(2*pi*fm*t)];

subplot(5,2,7);
plot(t,EDSB);
title('DSB-SC-UI21CS27');

   a=abs(fft(EDSB));
subplot(5,2,8);
```

```
plot(t,a);  title('DSB-
SC-UI21CS27');



Em1=Am*sin(2*pi*fm*t);

Ec1=Ac*sin(2*pi*fc*t);

ESSB=Em.*Ec+Em1.*Ec1;

subplot(5,2,9);
 plot(t,ESSB);
title('SSB signal-UI21CS27');

a=abs(fft(ESSB));

subplot(5,2,10);
 plot(t,a);
title('SSB signal-UI21CS27');
```



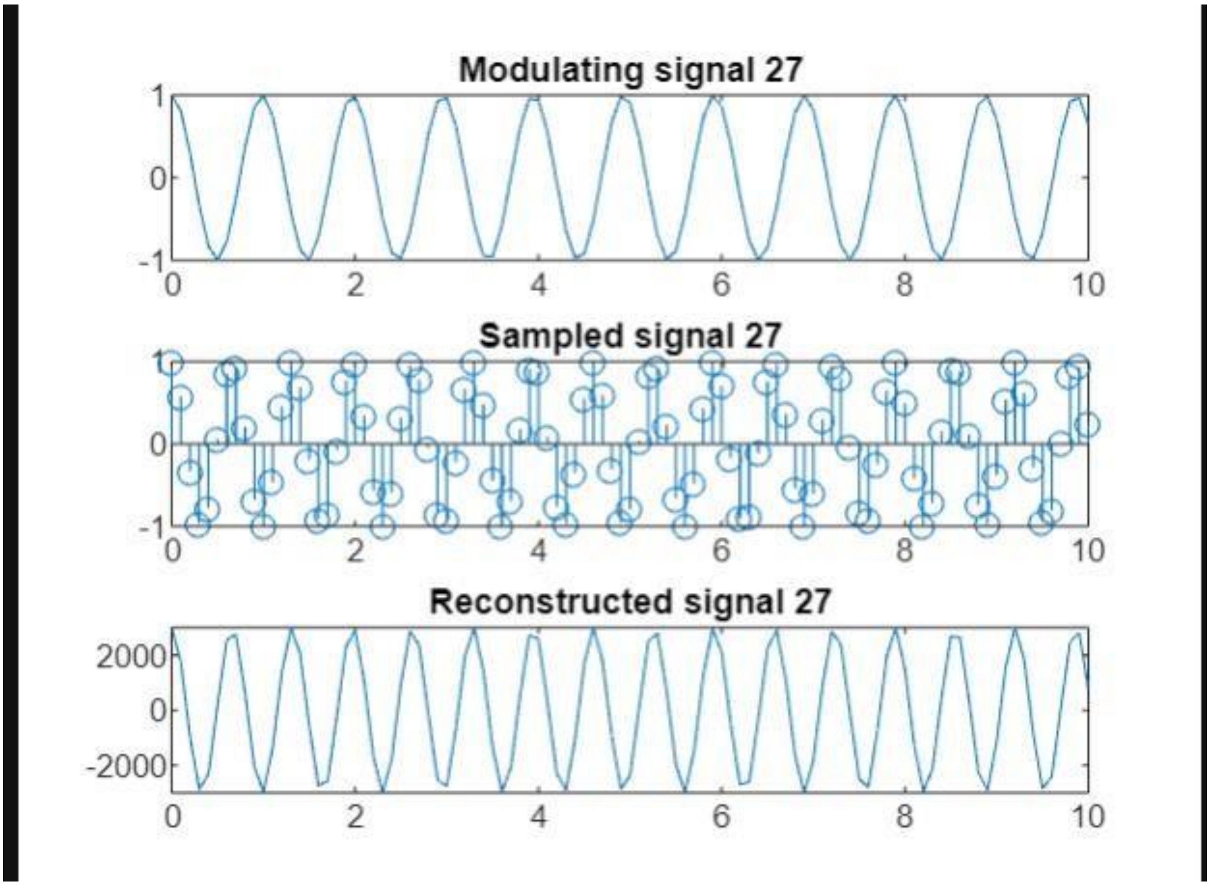
# Experiment_5:

```
% Lab 2 clc; clear all;
t=[0:0.1:10]; fm=input('Enter
frequency fm= '); fs=input('Enter
frequency fs= '); pi=3.14;
x=cos(2*fm*pi*t); subplot(3,1,1);
plot(t,x); title('Modulating
signal 27');
 y=cos(2*fs*pi*t);
subplot(3,1,2); stem(t,y);
title('Sampled signal 27');
 h=filter(fs,1,y);
subplot(3,1,3); plot(t,h);
title('Reconstructed signal 27');
```

**CASE-I: Under-Sampling**

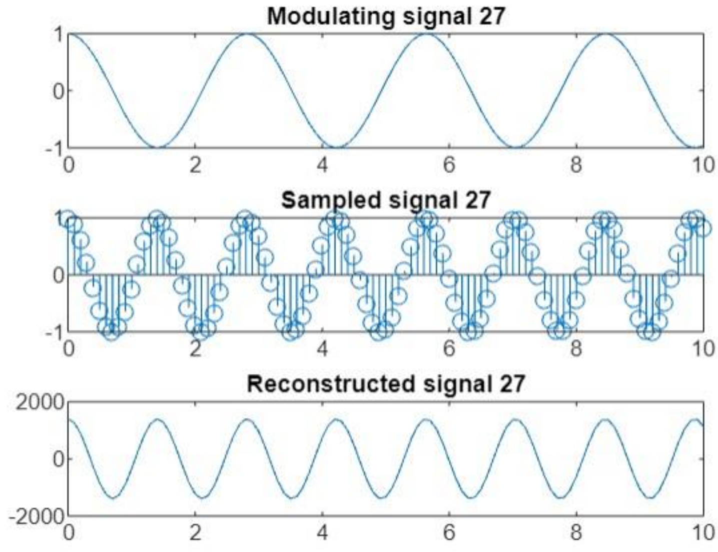
- **Fs <2*Fm**

- **Fm = 2000**
- **Fs = 3000**



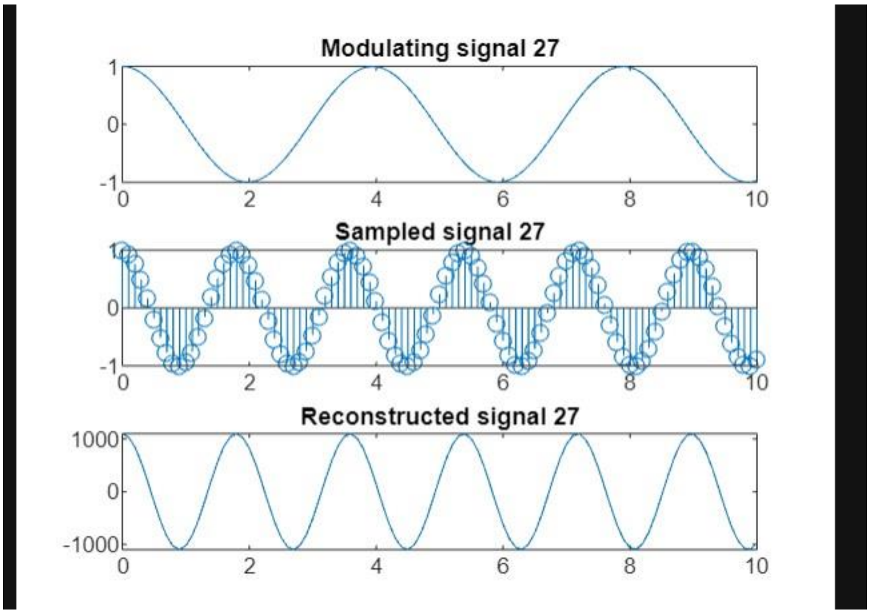

**CASE-II: Perfect Sampling**

- **Fs = 2*Fm**
- **Fs = 700**
- **Fm =1400**

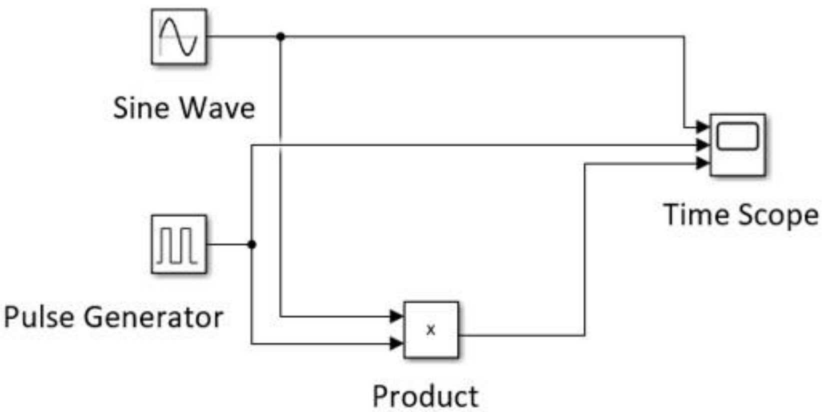



**CASE-III: Over Sampling**

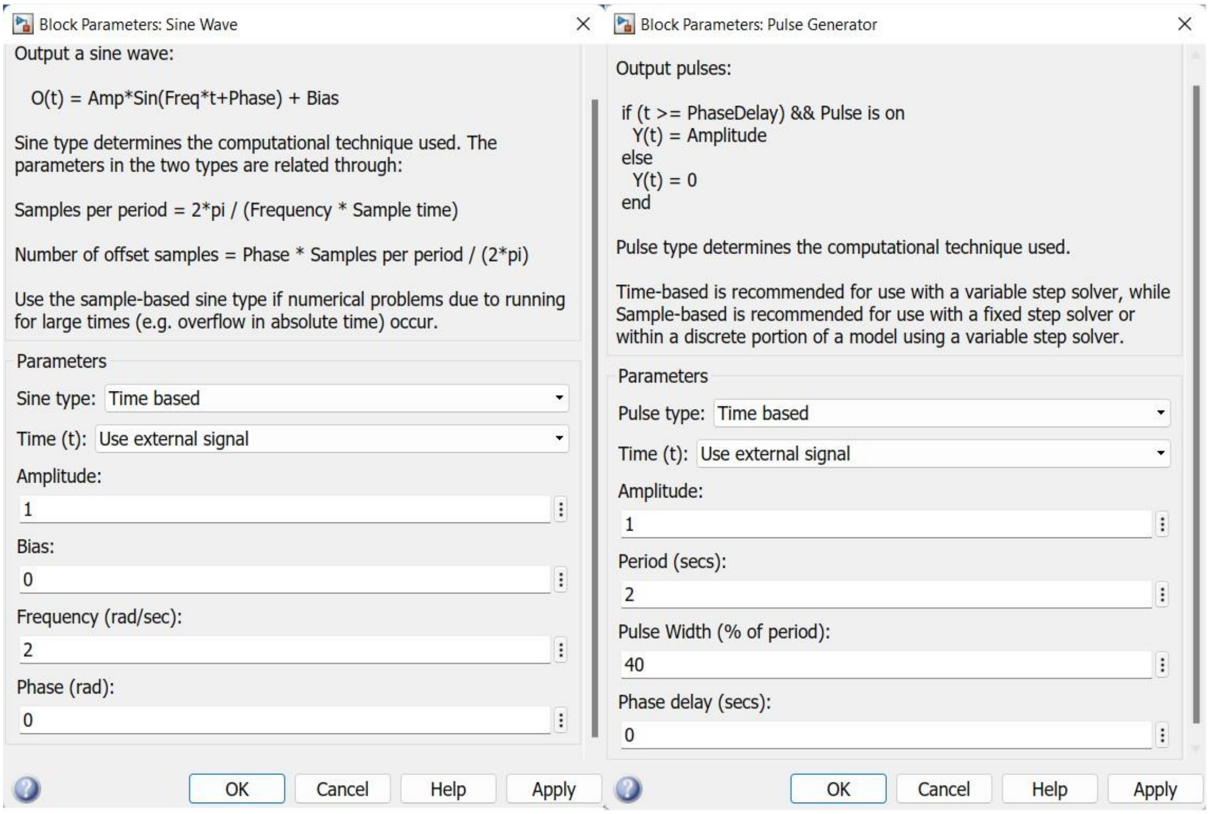
- **Fs > 2*Fm**
- **Fm= 500**
- **Fs= 1100**

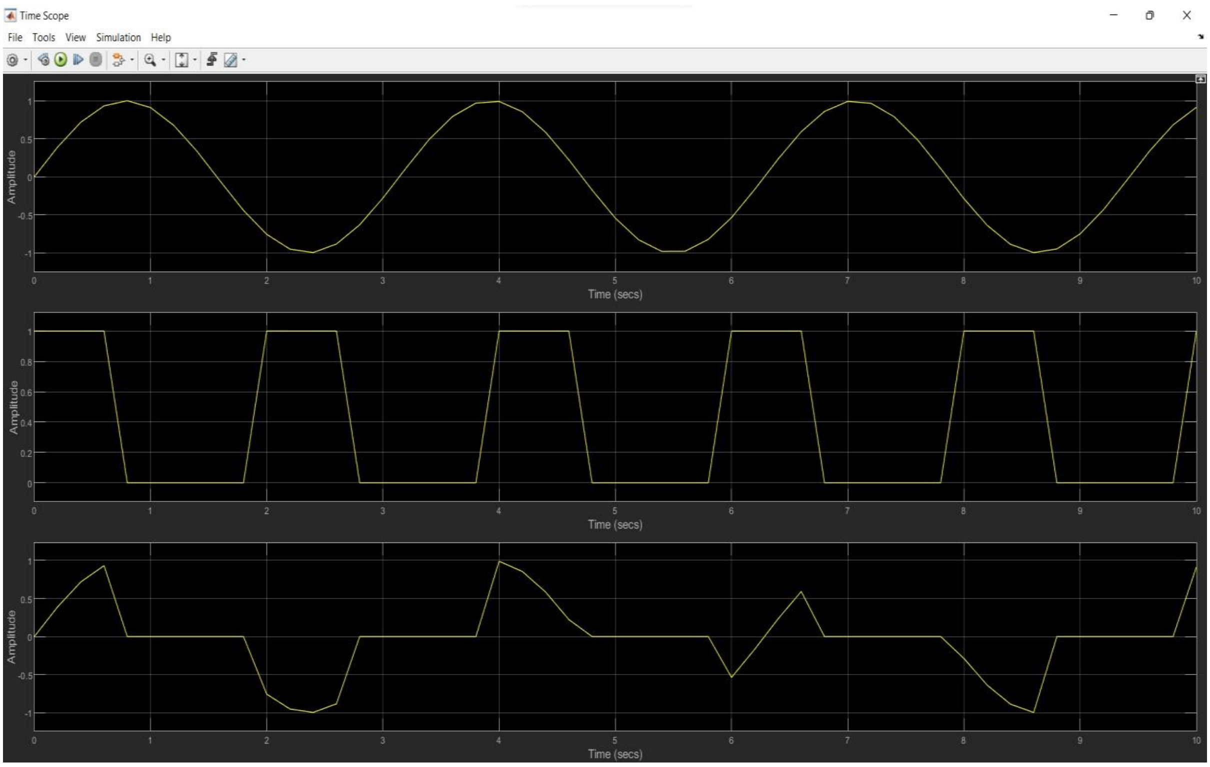



:------

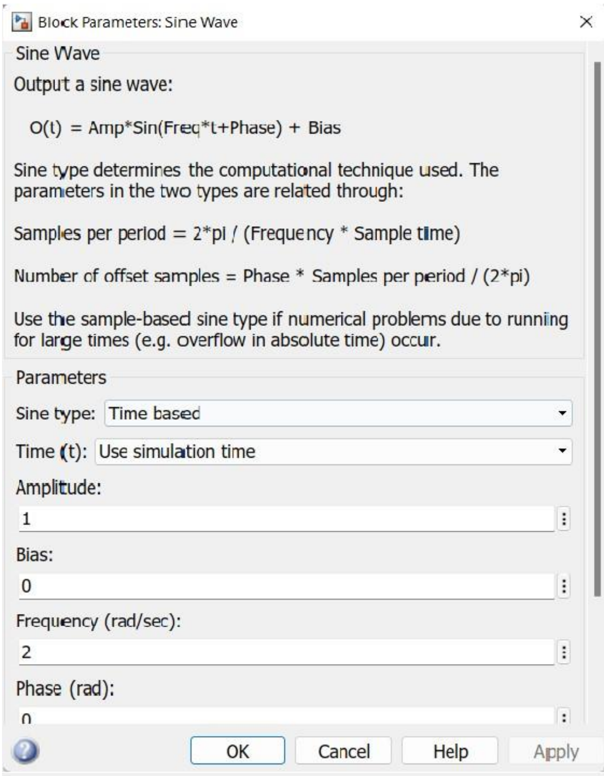


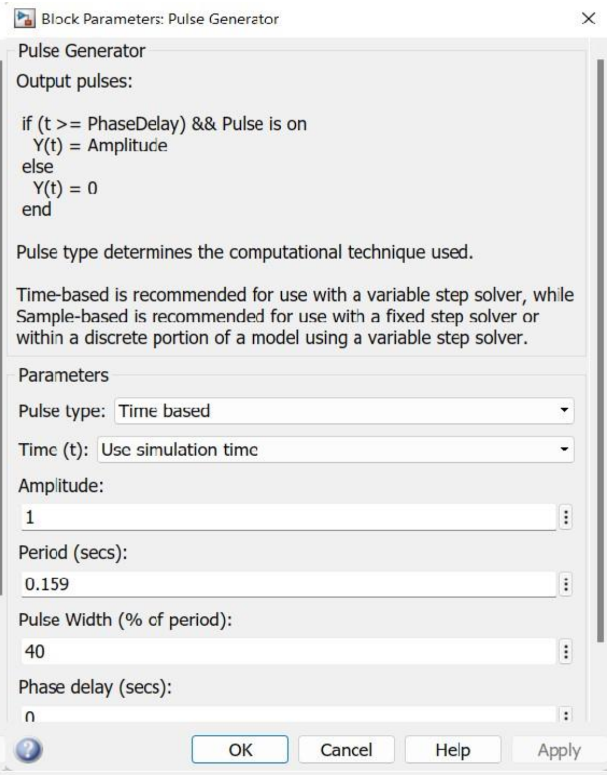

## CASE-I: Under-Sampling

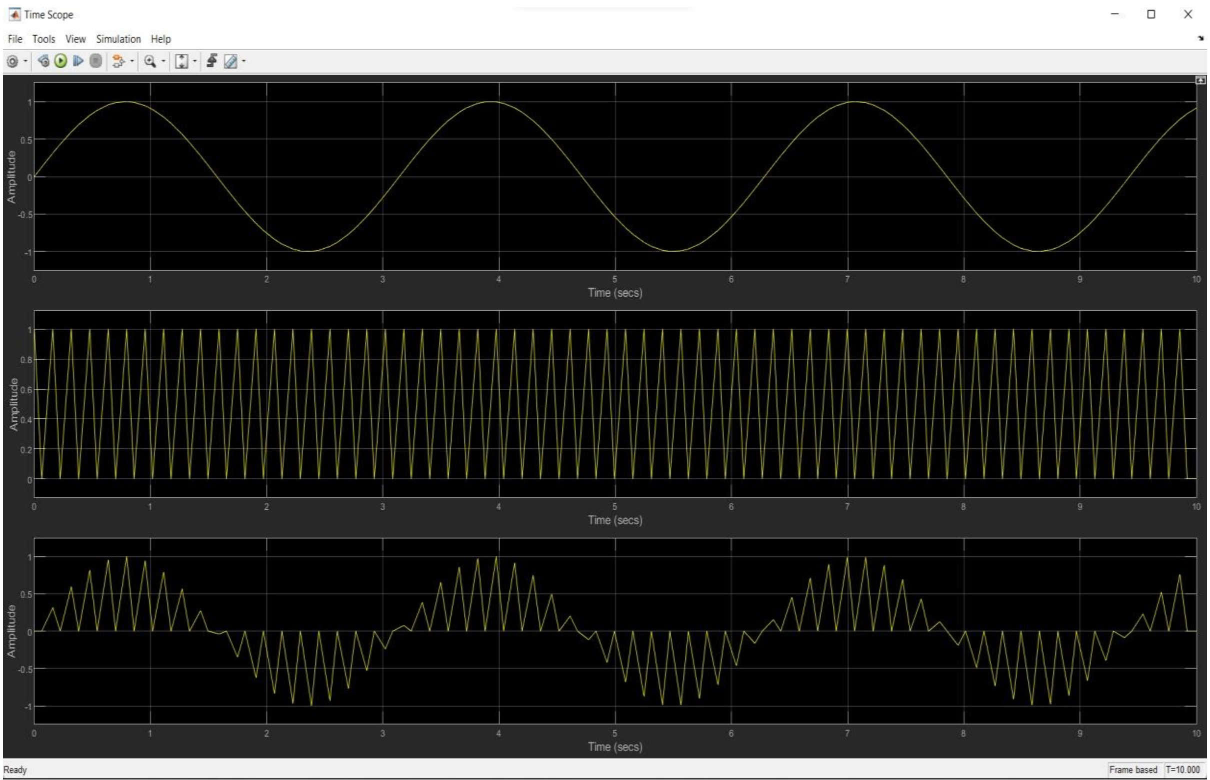


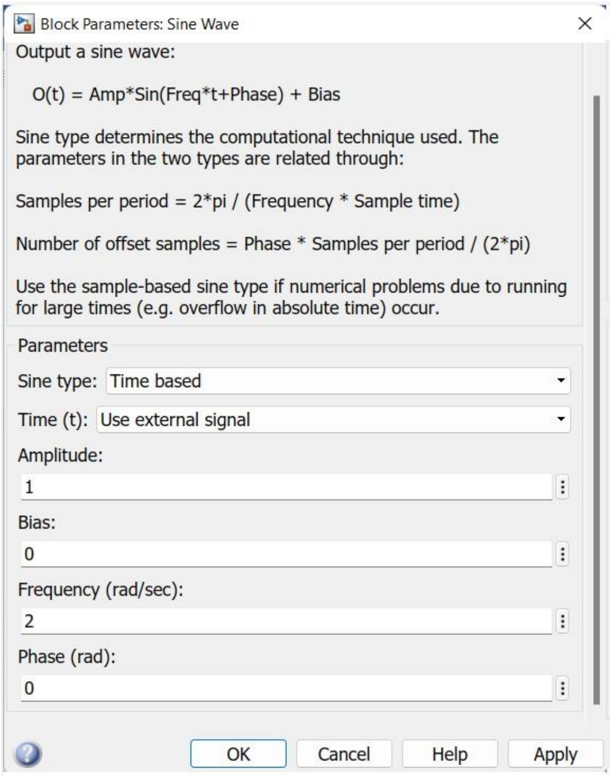

## CASE-II: Perfect Sampling

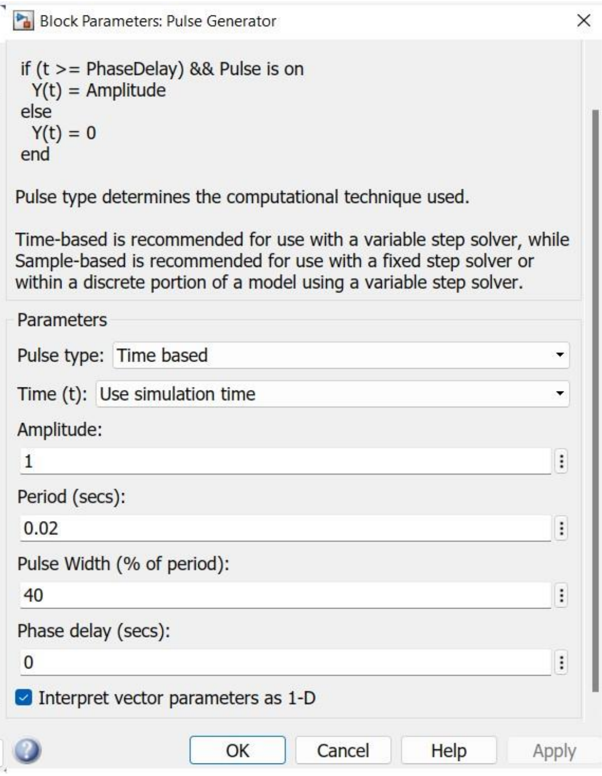


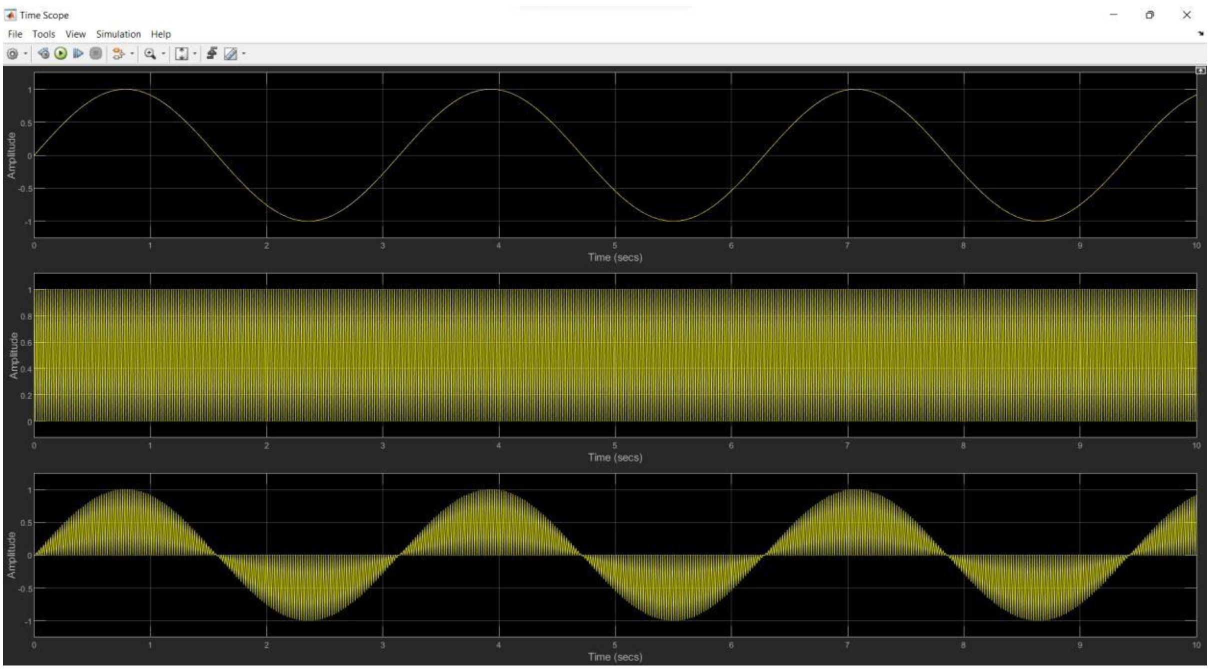




## CASE-III: Over Sampling

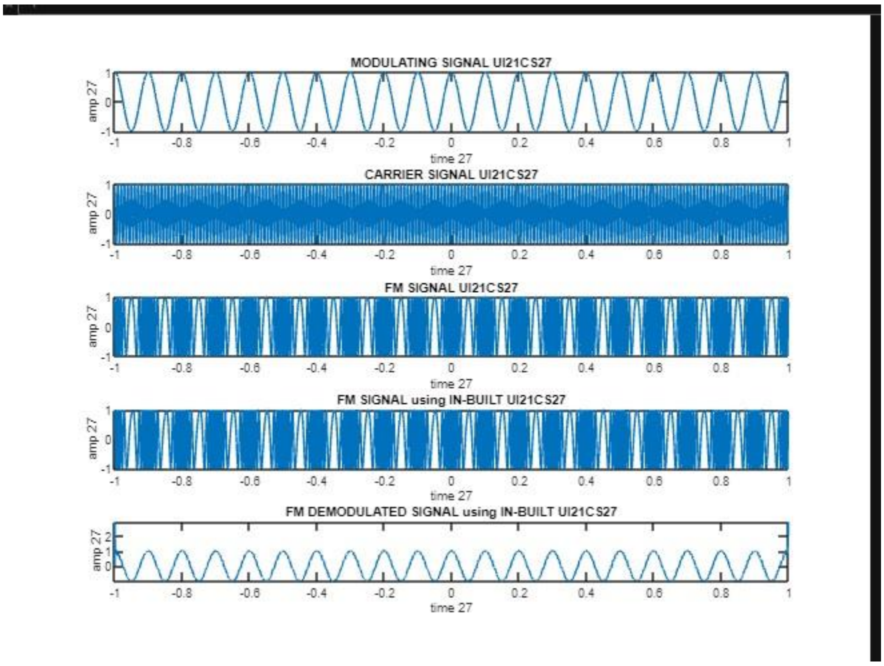

Block Parameters: Sine Wave
Output a sine wave:
O(t) = Amp*Sin(Freq*t+Phase) + Bias
Sine type determines the computational technique used. The parameters in the two types are related through:
Samples per period = 2*pi / (Frequency * Sample time)
Number of offset samples = Phase * Samples per period / (2*pi)
Use the sample-based sine type if numerical problems due to running for large times (e.g. overflow in absolute time) occur.
Parameters
Sine type: Time based
Time (t): Use external signal
Amplitude:
1
Bias:
0
Frequency (rad/sec):
2
Phase (rad):
0
OK
Cancel
Help
Apply

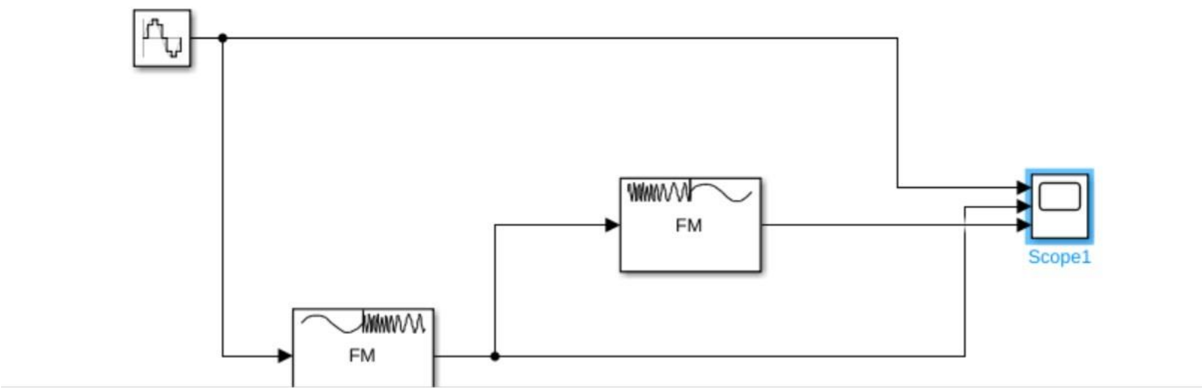

Block Parameters: Pulse Generator
if (t >= PhaseDelay) && Pulse is on
Y(t) = Amplitude
else
Y(t) = 0
end
Pulse type determines the computational technique used.
Time-based is recommended for use with a variable step solver, while Sample-based is recommended for use with a fixed step solver or within a discrete portion of a model using a variable step solver.
Parameters
Pulse type: Time based
Time (t): Use simulation time
Amplitude:
1
Period (secs):
0.02
Pulse Width (% of period):
40
Phase delay (secs):
0
Interpret vector parameters as 1-D
OK
Cancel
Help
Apply

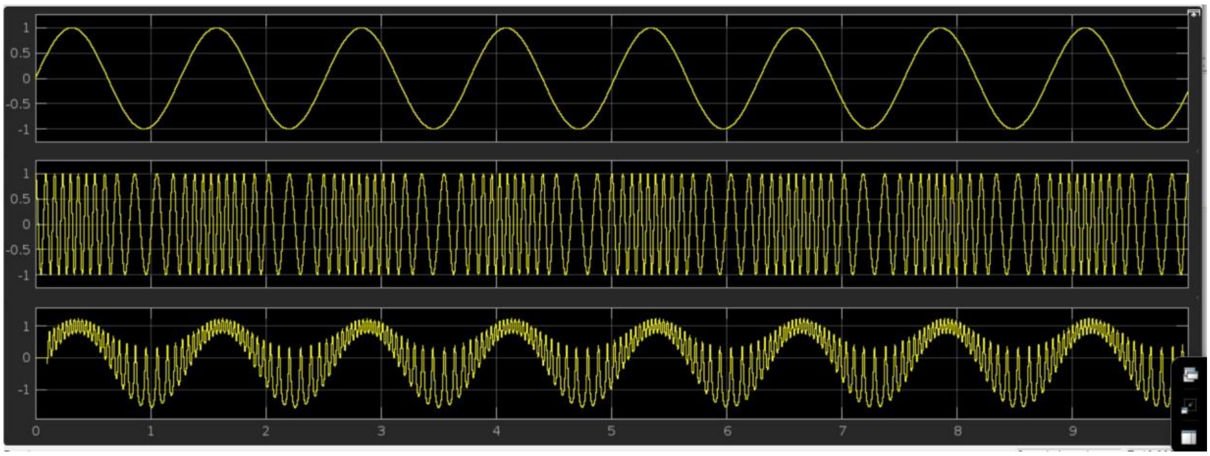

Time Scope
File
Tools
View
Simulation
Help
Amplitude
Time (secs)
Amplitude
Time (secs)
Amplitude
Time (secs)

# Experiment_6:

```
clc; clear ALL;
close all; %
MODULATING SIGNAL
figure; fs=10000; t
= -1:1/fs:1;
pi=3.14; fm=10;
fc=100; Am=1;
Ac=1;
B=8;%modulation index
freqdev=B*fm;
m=Am*cos(2*pi*fm*t);
subplot(5,1,1);
plot(t,m);
xlabel('time 27');
ylabel('amp 27');
title('MODULATING SIGNAL UI21CS27');

% CARRIER SIGNAL
c=Ac*cos(2*pi*fc*t);
subplot(5,1,2);
plot(t,c); xlabel('time
27'); ylabel('amp 27');
title('CARRIER SIGNAL UI21CS27');
 % FM
SIGNAL
s=Ac*cos(2*pi*fc*t+(B*sin(2*pi*fm*t)));
subplot(5,1,3); plot(t,s); xlabel('time
27'); ylabel('amp 27'); title('FM
SIGNAL UI21CS27');

%IN-BUILT FM SIGNAL
y=fmmod(m,fc,fs,freqdev);
subplot(5,1,4);
plot(t,y); xlabel('time
27'); ylabel('amp 27');
title('FM SIGNAL using IN-BUILT UI21CS27');

%IN-BUILT FM DEMODULATION SIGNAL
z=fmdemod(s,fc,fs,freqdev);
subplot(5,1,5); plot(t,z);
xlabel('time 27'); ylabel('amp
27');
title('FM DEMODULATED SIGNAL using IN-BUILT UI21CS27');
```


MODULATING SIGNAL UI21CS27
CARRIER SIGNAL UI21CS27
FM SIGNAL UI21CS27
FM SIGNAL using IN-BUILT UI21CS27
FM DEMODULATED SIGNAL using IN-BUILT UI21CS27
amp 27
time 27

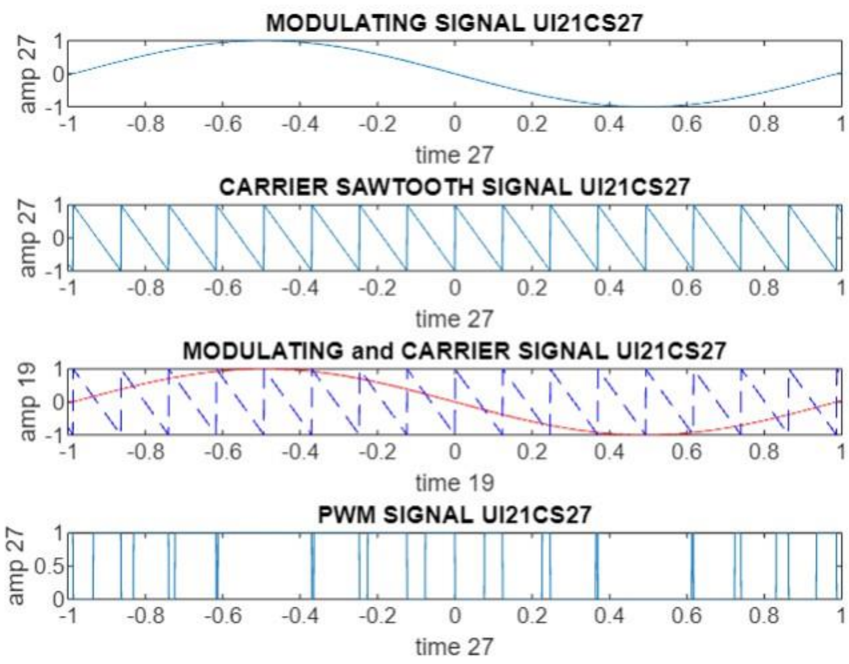

FM
FM
Scope1

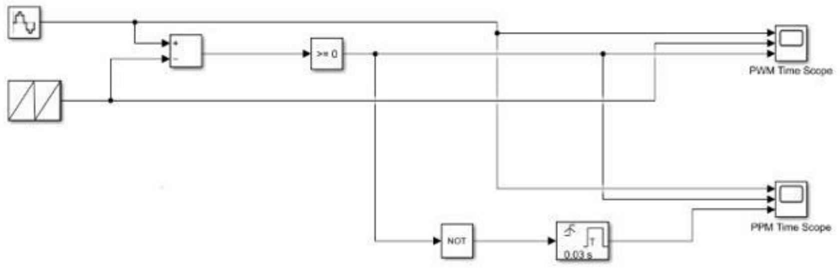


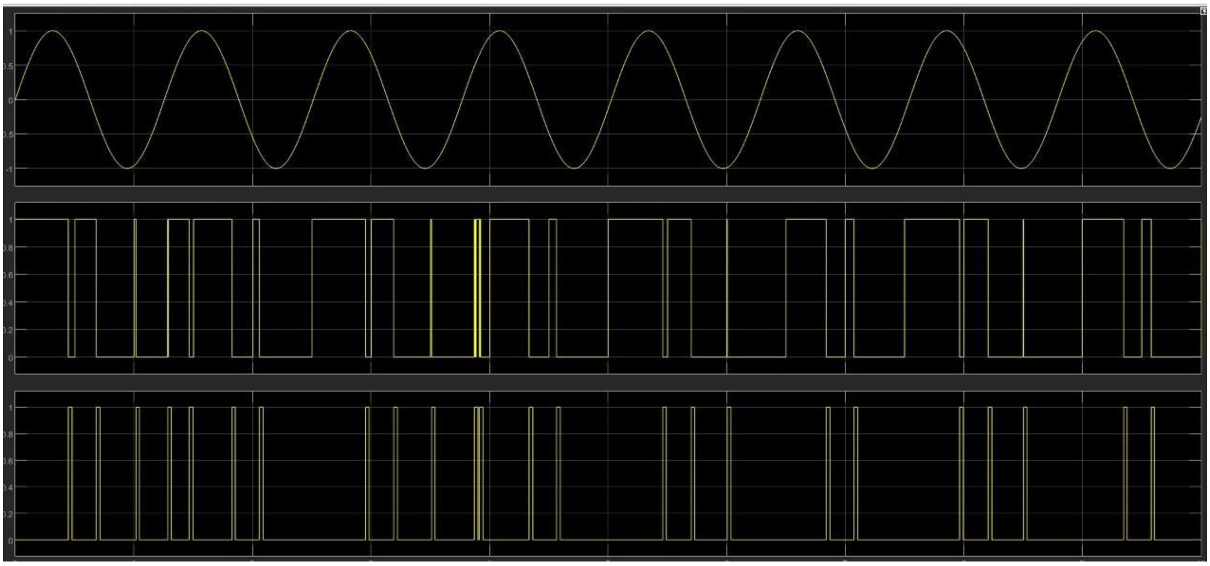
# Experiment_7:

```
clc; clear
ALL; close
all;
%LAB 7 UI21CS27 PAM - PWM - PPM SIGNAL
% MESSAGE SIGNAL
figure; fs=10000;
t = -1:0.001:1;
pi=3.14; fm=1000;
fc=16000;
m=sin(2*pi*fm*t);
subplot(4,1,1);
plot(t,m);
xlabel('time 27');
ylabel('amp 27');
title('MODULATING SIGNAL UI21CS27');

% SAWTOOTH CARRIER SIGNAL
c=sawtooth(2*pi*fc*t);
subplot(4,1,2);
```

```
plot(t,c); xlabel('time
27'); ylabel('amp 27');
title('CARRIER SAWTOOTH SIGNAL UI21CS27');
 subplot(4,1,3);
plot(t,m,'r',t,c,'b--');
xlabel('time 27');
ylabel('amp 27');
title('MODULATING and CARRIER SIGNAL UI21CS27');
n=length(c); for i = 1:n
if m(i)>=c(i)
pwm(i)=1;      else
pwm(i)=0;      end end
subplot(4,1,4); plot(t,pwm);
xlabel('time 27');
ylabel('amp 27'); title('PWM
SIGNAL UI21CS27');
```

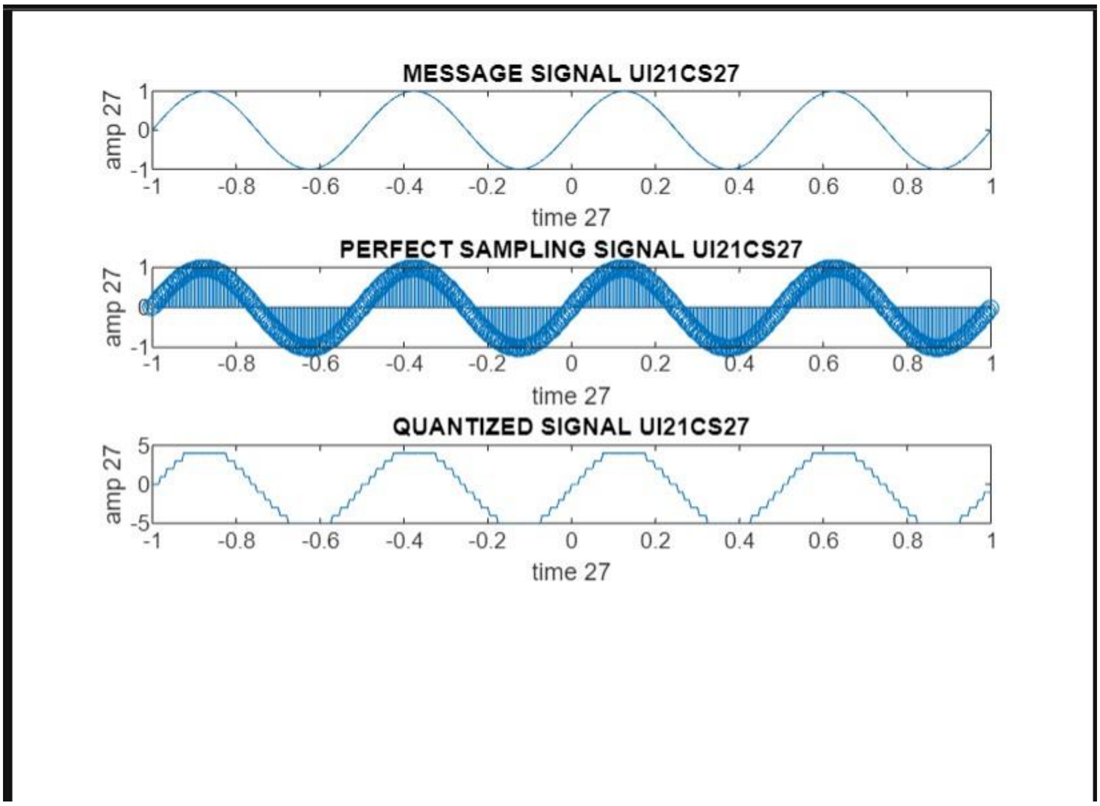


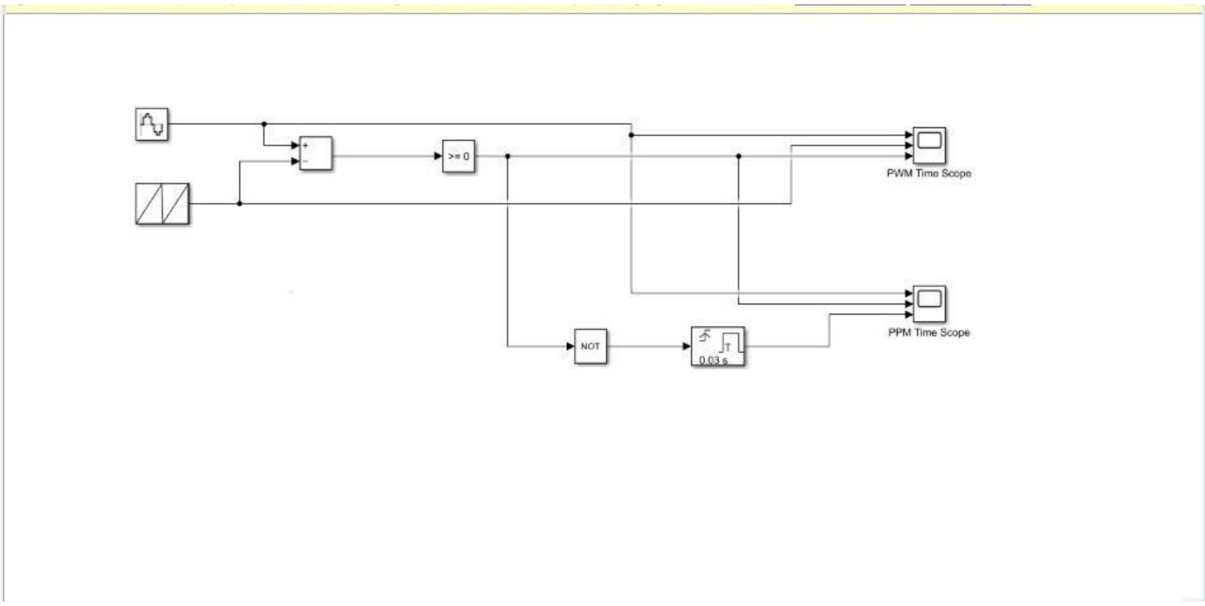

>= 0
PWM Time Scope
PPM Time Scope
NOT
0.03 s

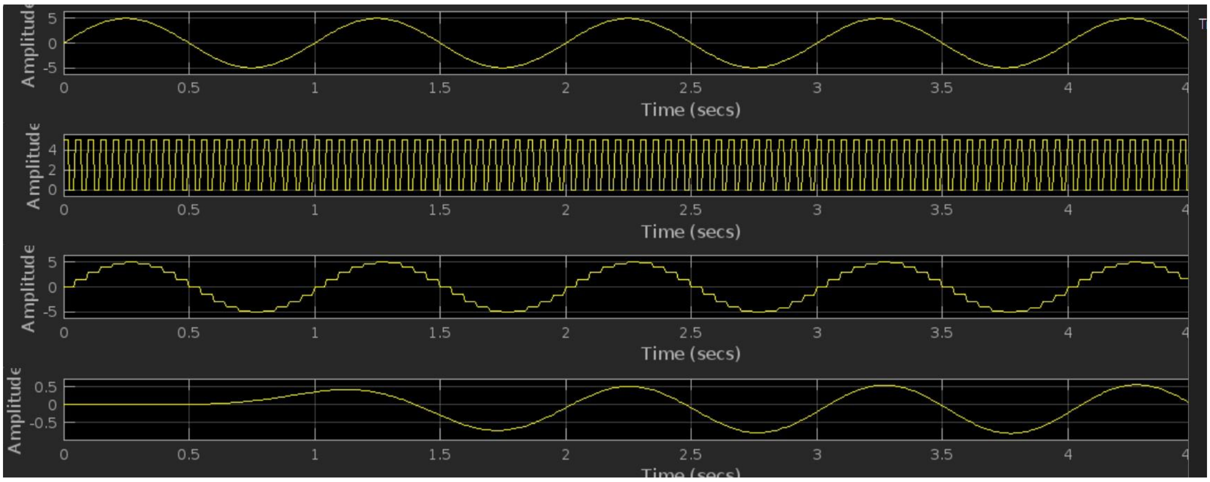


# Experiment_8:

```
 clc; clear All; close all;  figure; t = -
1:0.005:1; pi=3.14; fm=2;
m=sin(2*pi*fm*t); subplot(4,1,1);
plot(t,m); xlabel('time 27'); ylabel('amp
27'); title('MESSAGE SIGNAL UI21CS27'); %
PERFECT SAMPLING fm=1000; subplot(4,1,2);
stem(t,m); xlabel('time 27'); ylabel('amp
27'); title('PERFECT SAMPLING SIGNAL
UI21CS27'); x=floor(m/0.2); y=dec2bin(x);
subplot(4,1,3); plot(t,x); xlabel('time
27'); ylabel('amp 27'); title('QUANTIZED
SIGNAL UI21CS27'); z=bin2dec(y);s disp(z);
output1=filter(1,1,z);
[B,A]=butter(5,1/42,'low');
output2=filter(B,A,output1);
subplot(4,1,4); plot(t,output2);
```

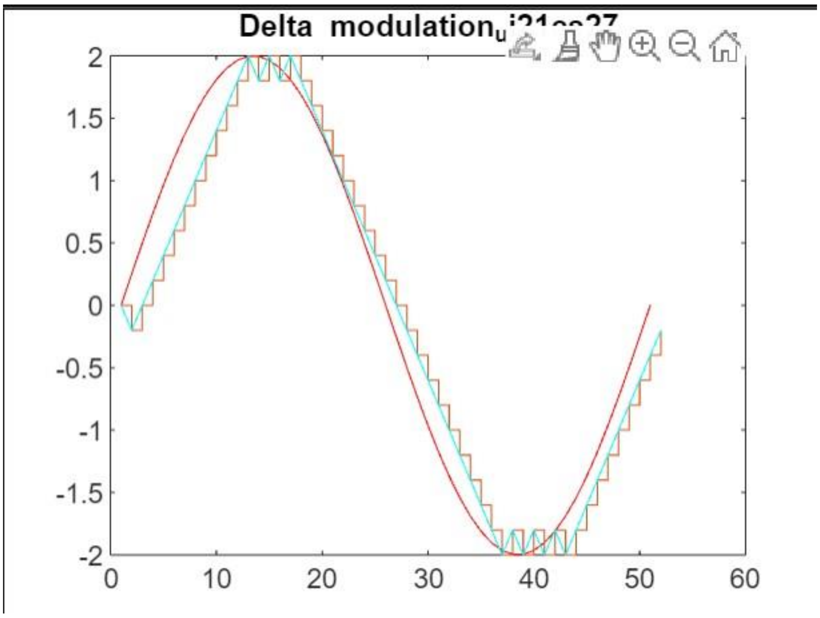

MESSAGE SIGNAL UI21CS27
amp 27
time 27
PERFECT SAMPLING SIGNAL UI21CS27
amp 27
time 27
QUANTIZED SIGNAL UI21CS27
amp 27
time 27


>= 0
PWM Time Scope
NOT
0.03 s
PPM Time Scope




# Experiment_9:

```
clc; clear all; close all;  a=2;
t=0:2*pi/50:2*pi; x = a*sin(t); l =
length(x); plot(x,'r');
title('Delta modulation_ui21cs27');
delta = 0.2;  hold on xn = 0; for
i=1:l if x(i)> xn(i) d(i) = 1;
xn(i+1) = xn(i)+delta; else d(i) =
0; xn(i+1)= xn(i)-delta; end end
stairs(xn) hold on for i=1:d if
d(i)>xn(i) d(i) = 0; xn(i+1) =
xn(i)-delta; else d(i) = 1; xn(i+1)
= xn(i)+delta; end end
plot(xn,'c');
```

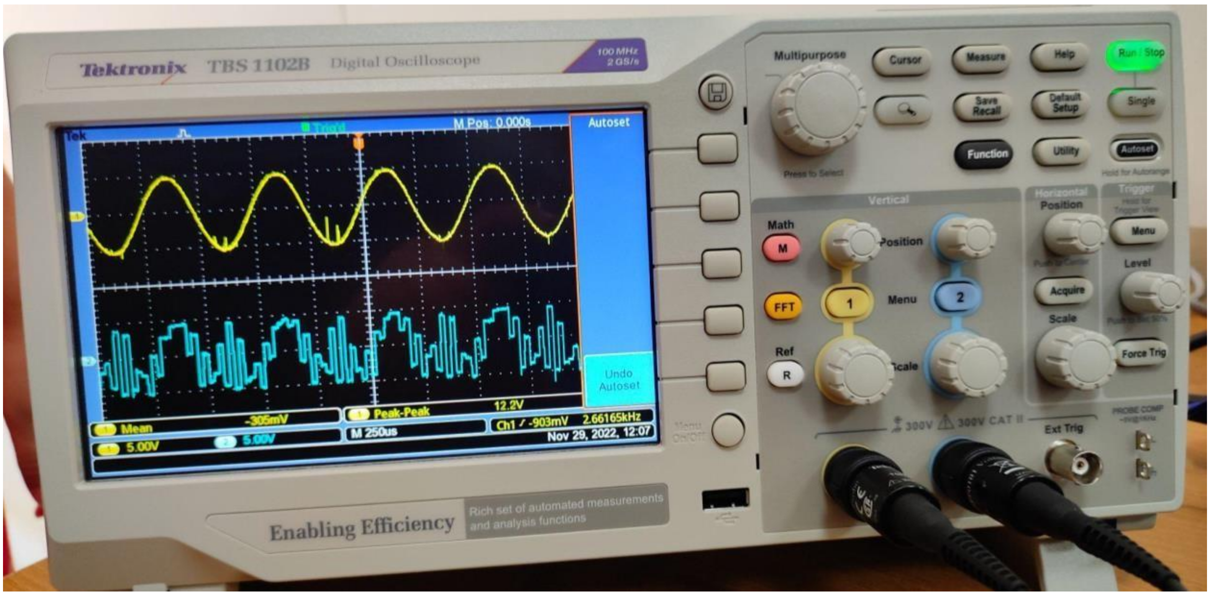



# PRACTICAL- 10

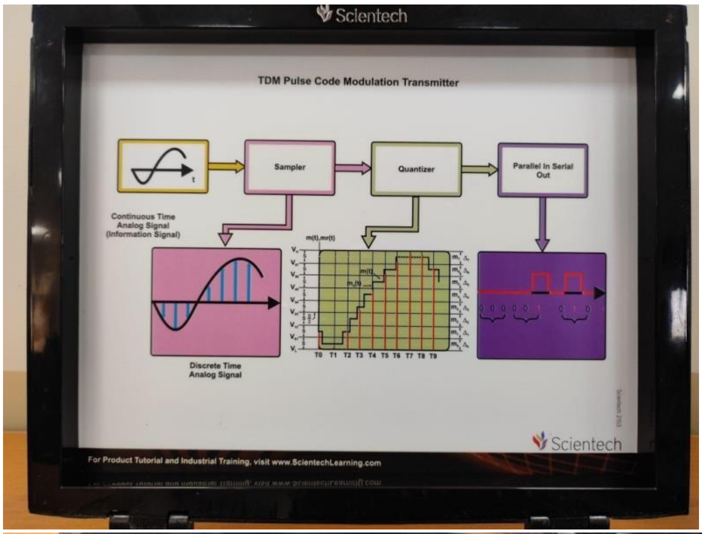
**AIM:-**The study construction and execution of PCM and PCMTDM signals using Transmission and Reciever kit.

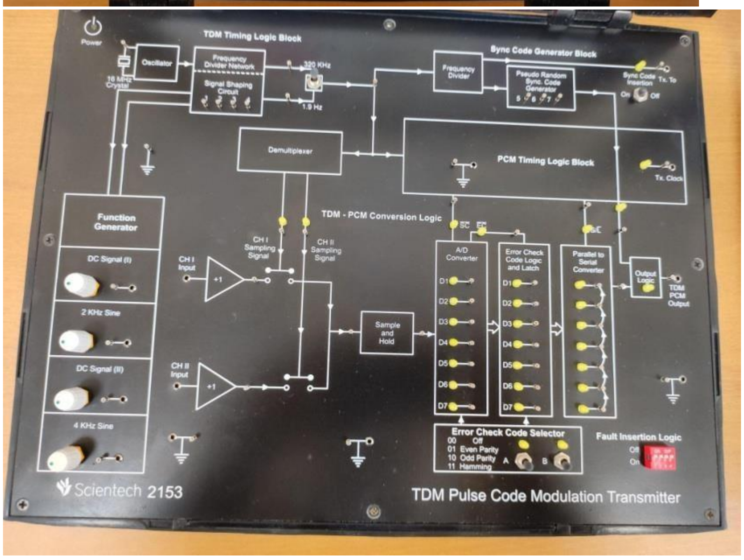
## APPARATUS:-

1.OSCILLOSCOPE

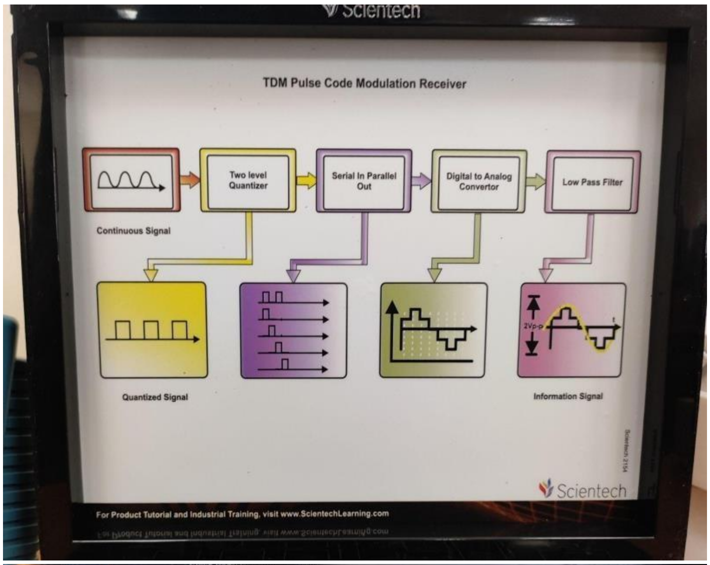


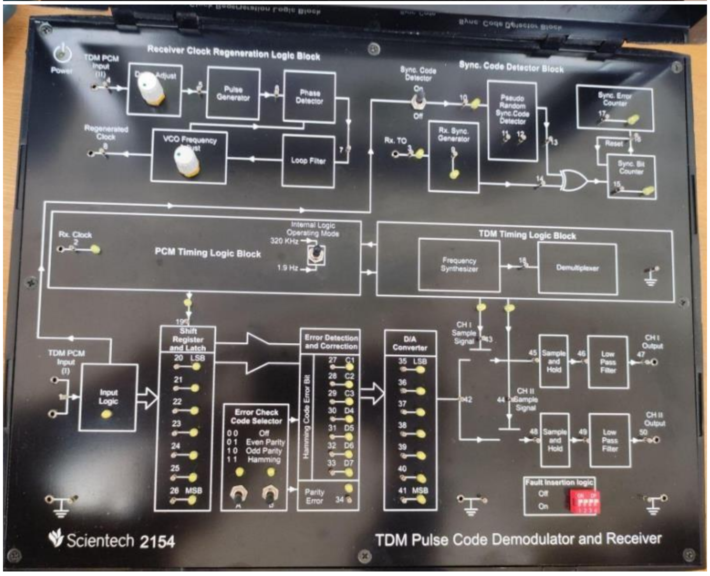

2. PCM-TDM TRANSMITTER:-



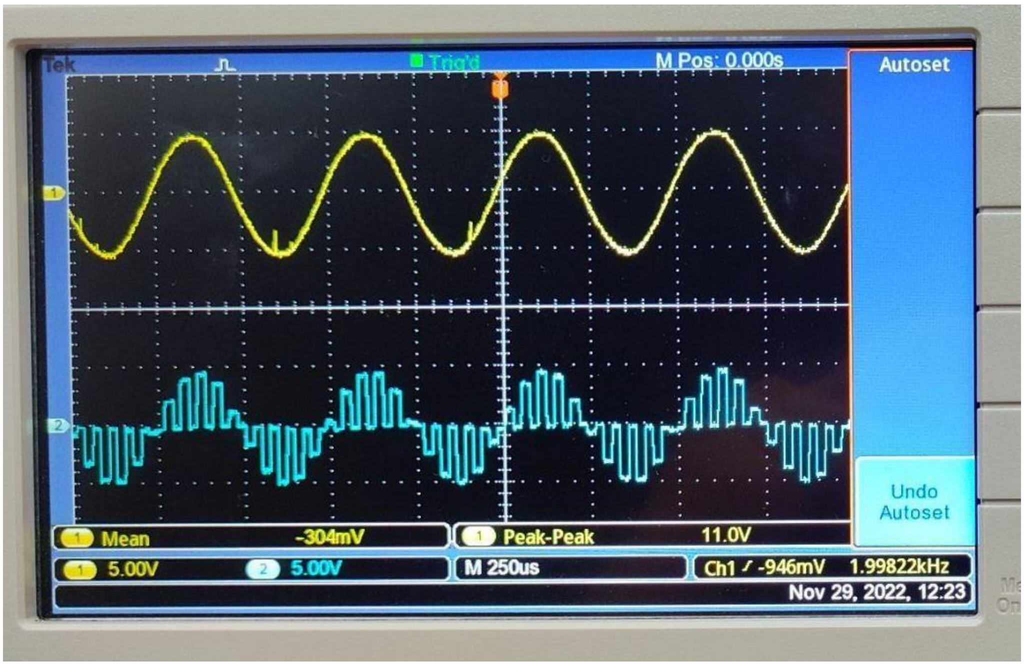

3.TDM DECODER and DEMODULATOR KIT:-


Scientech
TDM Pulse Code Modulation Receiver
Two level Quantizer
Serial In Parallel Out
Digital to Analog Convertor
Low Pass Filter
Continuous Signal
Quantized Signal
Information Signal
2Vp-p
Scientech
For Product Tutorial and Industrial Training, visit www.ScientechLearning.com

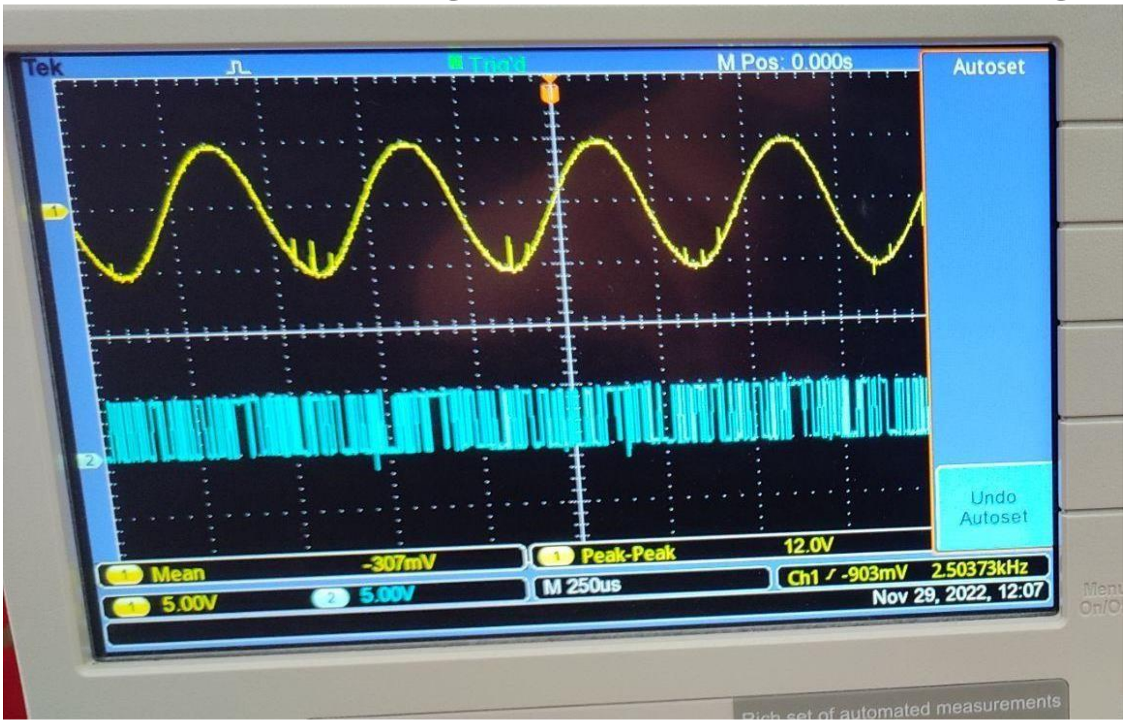

Power
TDM PCM Input
Receiver Clock Regeneration Logic Block
Pulse Generator
Phase Detector
Regenerated Clock
VCO Frequency Adjust
Loop Filter
Sync. Code Detector
On
Off
Sync. Code Detector Block
Pseudo Random Sync.Code Detector
Rx. TO
Rx. Sync. Generator
Sync. Error Counter
Reset
Sync. Bit Counter
Rx. Clock
PCM Timing Logic Block
Internal Logic Operating Mode
320 KHz
1.9 Hz
TDM Timing Logic Block
Frequency Synthesizer
Demultiplexer
TDM PCM Input
Input Logic
Shift Register and Latch
LSB
MSB
Error Detection and Correction
Hamming Code Error Bit
Error Check Code Selector
0 0 Off
0 1 Even Parity
1 0 Odd Parity
1 1 Hamming
Parity Error
D/A Converter
CH I Sample Signal
CH II Sample Signal
Sample and Hold
Low Pass Filter
CH I Output
CH II Output
Fault Insertion logic
Off
On
Scientech 2154
TDM Pulse Code Demodulator and Receiver

**OBSERVATION: (PCM TX –RX)**

MESSAGE Signal(2KHz) and its SAMPLED OUTPUT Signal:-

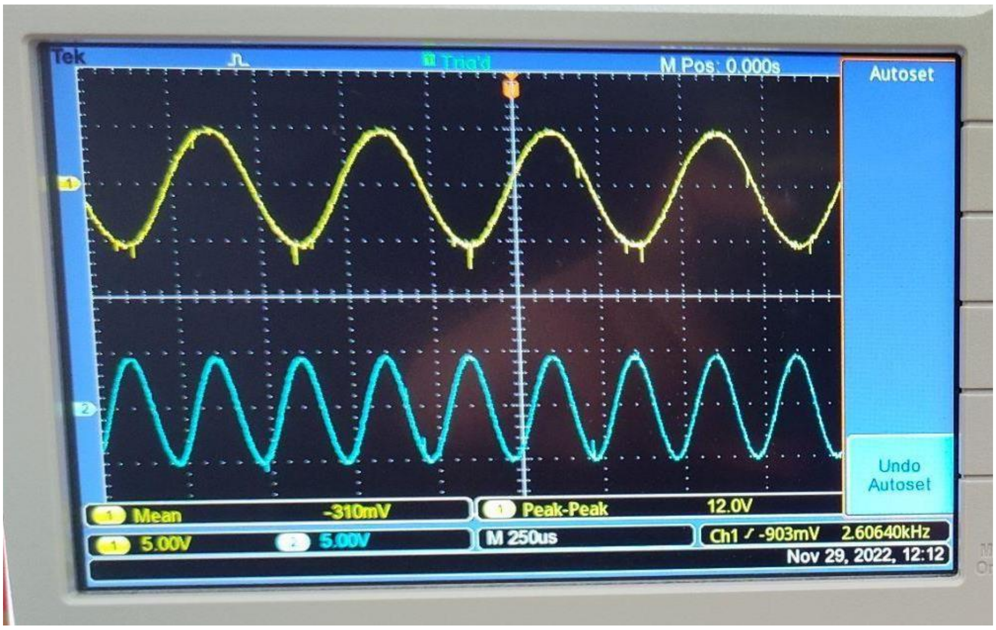



RECONSTRUCTED Signal and RECIEVED PCM  Signal:-

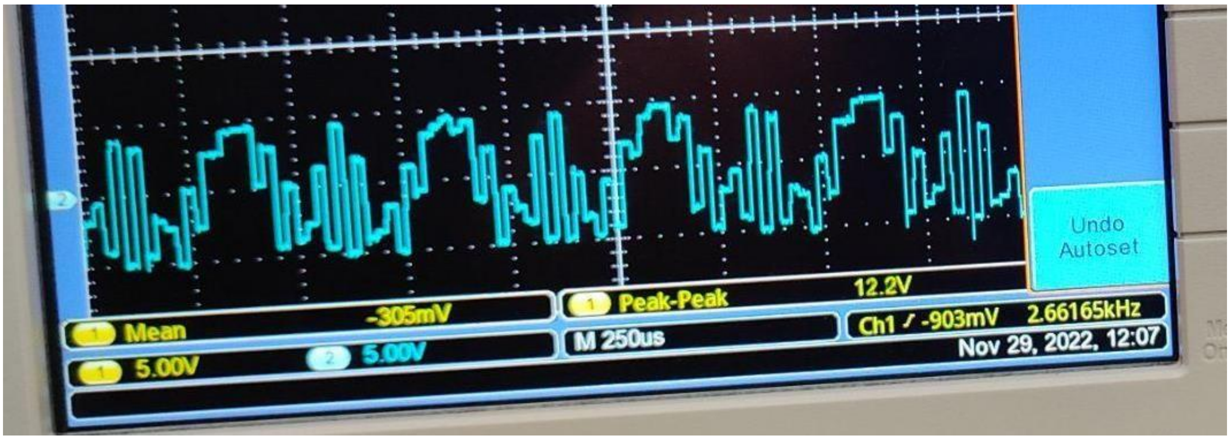



**OBSERVATION: (PCM-TDM TX –RX)**

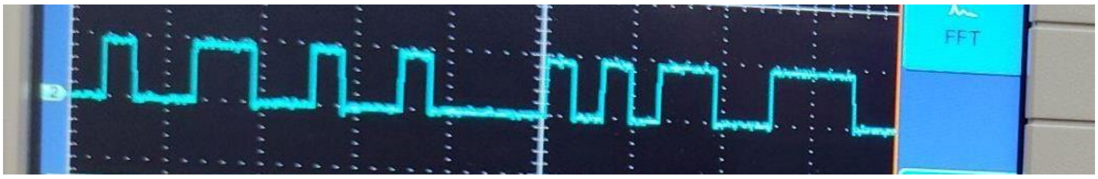
TWO MESSAGE SIGNAL (2KHz and 4KHz):-




TDM SAMPLED SIGNAL for both the SIGNALS (2KHz&4KHz):-




# PCM Signal and RECONSTRUCTED Signal:-


TDM RECONSTRUCTED SIGNAL (2KHz and 4KHz):-

**CONCLUSION:-**

We learned how to createe ,transmit and recieve PCM,PCM-TDM signaal using TDM Transmission and Recieving Kit.