Experiment 2:

Experiment 2: Generation of Continuous Time and Discrete Time Signal

Aim: To generate the continuous time and discrete time signal using SCI lab

Software Requirement: SCI Lab

Theory:

Signals are represented mathematically as functions of one or more independent variables. Here we focus attention on signals involving a single independent variable. For convenience, this will generally refer to the independent variable as time. There are two types of signals: continuous-time signals and discrete-time signals. Continuous-time signal: the variable of time is continuous. A speech signal as a function of time is a continuous-time signal. Discrete-time signal: the variable of time is discrete.

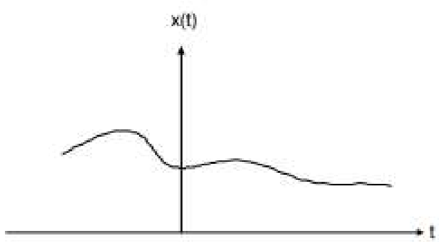


Fig.1. CT signal

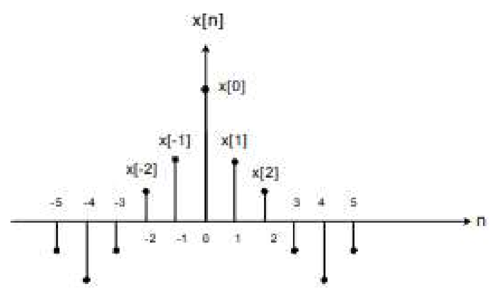


Fig.2 DT signal

To distinguish between continuous-time and discrete-time signals we use symbol t to denote the continuous variable and n to denote the discrete-time variable. And for continuous-time signals we will enclose the independent variable in parentheses (·), for discrete-time signals we will enclose the independent variable in bracket [·].

A discrete-time signal x n][may represent a phenomenon for which the independent variable is inherently discrete. A discrete-time signal x n][may represent successive samples of an underlying phenomenon for which the independent variable is continuous. For example, the processing of speech on a digital computer requires the use of a discrete time sequence representing the values of the continuous-time speech signal at discrete points of time.
The differences between continuous and discrete-time signals are as follows:

Table 1

S.No.	Continuous-time signal	Discrete-time signal
1	The continuous-time signal is an analog representation of a natural signal.	The discrete-time signal is a digital representation of a continuous-time signal.
2	The continuous-time signal can be converted into discrete-time signal by the Euler's method.	The discrete -time signal can be converted into continuous-time signal by the methods of zero-order hold or first-order hold.
3	The conversion of continuous to discrete-time signal is comparatively easy than the conversion of discrete to continuous-time signals.	The conversion of discrete to continuous-time signals is very complicated and it is done through a sample and hold process.
4	It is defined over a finite or infinite domain of sequence.	It is defined over a finite domain of sequence.
5	The value of the signal can be obtained at any arbitrary point of time.	The value of the signal can be obtained only at sampling instants of time.
6	The continuous-time signals are not used for the processing of digital signals.	The discrete-time signals are used for the processing of digital signals.
7	The continuous-time variable is denoted by a letter *t*.	The discrete-time variable is denoted by a letter *n*.
8	The independent variable encloses in the parenthesis $(\bullet)$.	The independent variable encloses in the bracket $[\bullet]$.

The differences between discrete and digital signals are as follows:

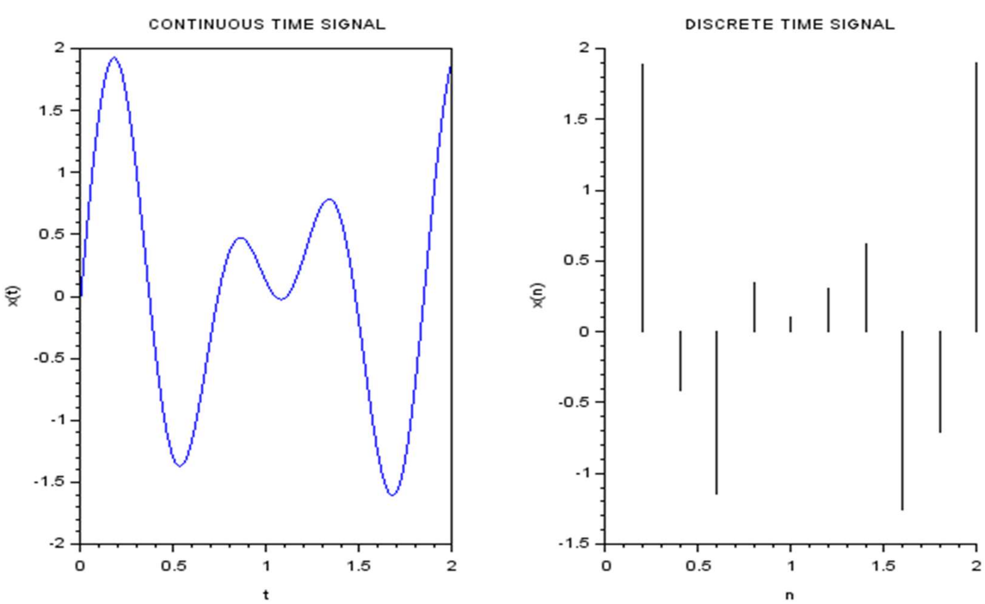
Table 2

S.No.	Discrete-time signal	Digital signal
1	The discrete-time signal is a digital representation of a continuous-time signal.	The digital signal is a form of discrete-time signal.
2	The discrete -time signal can be obtained from the continuous-time signal by the Euler's method.	The digital signal can be obtained by the process of sampling, quantization, and encoding of the discrete-time signal.
3	The discrete-time signal is a signal that has discrete in time and discrete in amplitude.	The digital signal is a signal that has discrete in amplitude and continuous in time.
4	The value of the signal can be obtained only at sampling instants of time.	The amplitude of the digital signal is either 1 or 0. That is, either OFF or ON.
5	The signals are sampled but not necessary to quantized in the discrete-time signals.	The signals are sampled and quantized in the digital signals.
6	All the discrete-time signals are digital signals.	All the digital signals are not discrete-time signals.

PROGRAM

```
// GENERATION OF CONTINUOUS TIME SIGNAL AND DISCRETE TIME SIGNAL
clear ;
clc ;
close ;
t =0:0.01:2;
x1 =sin (7* t ) +sin (10* t ) ;
subplot (1 ,2 ,1) ;
plot (t , x1 ) ;
xlabel ( 't' ) ;
ylabel ( 'x(t)') ;
title ( 'CONTINUOUS TIME SIGNAL');
n =0:0.2:2;
x2 =sin (7* n ) +sin (10* n ) ;
subplot (1 ,2 ,2) ;
plot2d3 (n , x2 ) ;
xlabel ( 'n') ;
ylabel ( 'x(n)') ;
title ( 'DISCRETE TIME SIGNAL' ) ;
```

SIMULATION RESULT



Pre-lab questions:

1. Define continuous time signal
2. How is discrete time signal generated?
3. Draw the graphical representation of continuous time signal and discrete time signal

Post-Lab questions:

1. Write any three difference between analog signal and digital signal
2. Write the code for generation of discrete signal for time interval 5μs and 7μs using subplot function
3. Give the advantages of digital signal over analog signal

Result:

EXPERIMENT 3:

EXPERIMENT 3a: SAMPLING THEOREM

Aim: To generate the sampled signal from the analog signal using SCI lab

Software Requirement: SCI Lab

Theory:

The real life signals that we encounter in our day to day basis are mostly analog signals. These signals are defined continuously in time and have infinite range of amplitude values. In order to process these signals to obtain meaningful information, they need to be converted to a format which is easily handled by computing resources like microprocessors, computers etc... The first step in this process is to convert the real-time signal into discrete-time signals. Discrete-time signals are defined only at a particular set of time instances. They can thus be represented as sequence of numbers with continuous range of values.

The process of converting an analog signal (denoted as x(t)) to a digital signal (denoted as x(n)) is called the analog-to-digital conversion (referred to as digitization), usually performed by an analog-to-digital converter (ADC). Here t is the continuous time variable and n is the sequence order. In many applications after the processing of the digital signal is performed, x(n) needs to be converted back to analog signal x(t) before it is applied to appropriate analog device. This reverse process is called digital-to-analog conversion and is typically performed using a digital-to-analog converter (DAC).

The typical block diagram of an ADC is shown in Fig. 1 below.

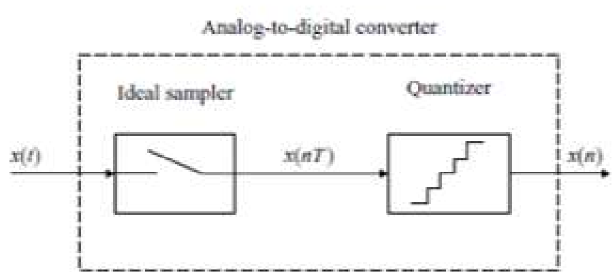


Fig:-1 Block diagram of an ADC

The process of digitization consists of first sampling (digitization in time) and quantization (digitization in amplitude). In this experiment we will study and understand the principle of sampling, while the principle of quantization will be studied in the next experiment. The sampling process depicts an analog signal as a sequence of values. The basic sampling function can be carried out with an ideal 'sample-and-hold' circuit which maintains the sampled signal until next sample is taken. An ideal sampler can be considered as a switch that periodically opens and closes every T seconds. The sampling frequency (fs in Hertz) is thus defined as

$$f_s = \frac{1}{T} \dots (1)$$

The sampled discrete time signal x(nT) , n=0,1,2,.... of the original continuous time signal x(t) is shown in Fig. 2 below.

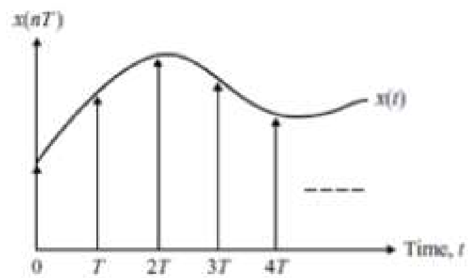


Fig:-2 Digitization of analog signal x(t) into discrete-time signal x(nT)

In order to represent an analog signal x(t) by a discrete-time signal x(nT) accurately, so that the analog signal can be exactly reconstructed back from the discrete-time signal, the sampling frequency fs must be at least twice the maximum frequency component (fM) of the original analog signal. Thus we have,

$$f_s \geq 2f_m \dots (2)$$

The minimum sampling rate is called the Nyquist rate and the above Sampling Theorem is called the Shannon's Sampling Theorem. When an analog signal is sampled at fs , frequency components higher than fs/2 fold back into the frequency range [0, fs/2]. This folded frequency components overlap with the original frequency components in the same range and leads to an undesired effect known as aliasing. In this case, the original analog signal cannot be recovered from the sample data.

Consider an analog signal of frequency 1Hz as shown in Fig. 3(a) below. The sampling frequency is 4Hz. The sampled signal is shown in Fig. 3(b), Note that an exact reconstruction of the missing samples is obtained so long as the Shannon's Sampling Theorem is satisfied.

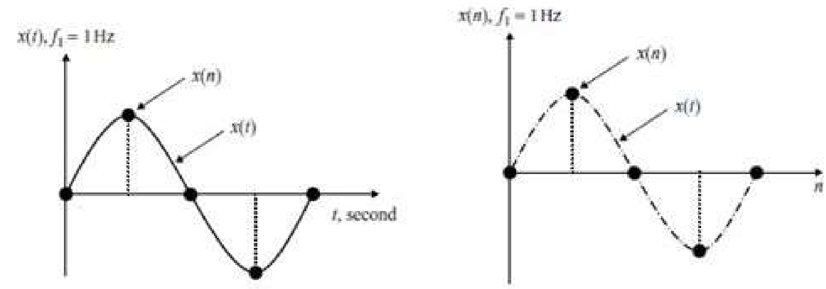


Fig:-3 Sampling of an analog signal x(t) into discrete-time signal x(nT) (a) and its exact reconstruction (b)

Now let's consider, the analog signal of frequency 5Hz as shown in Fig. 4(a) below. The sampling frequency is same as above, i.e. 4Hz. The sampled signal is shown in Fig. 4(b), Note that the reconstruction of the original analog signal is not possible since the sampling frequency does not satisfy Shannon's Sampling Theorem. In this case the reconstructed signal has a frequency of 1Hz. The signal of 5Hz is folded back as 1Hz, into the range determined by the sampling frequency leading to the problem of aliasing.

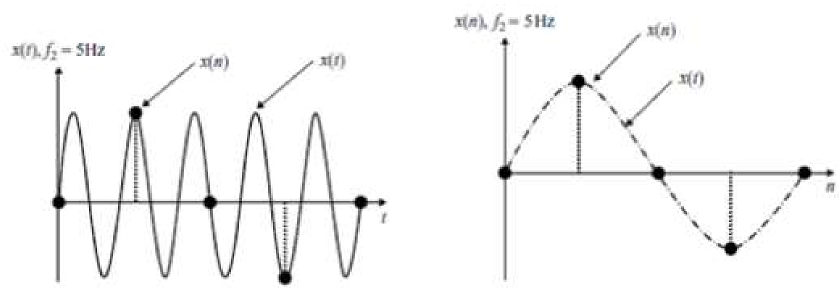


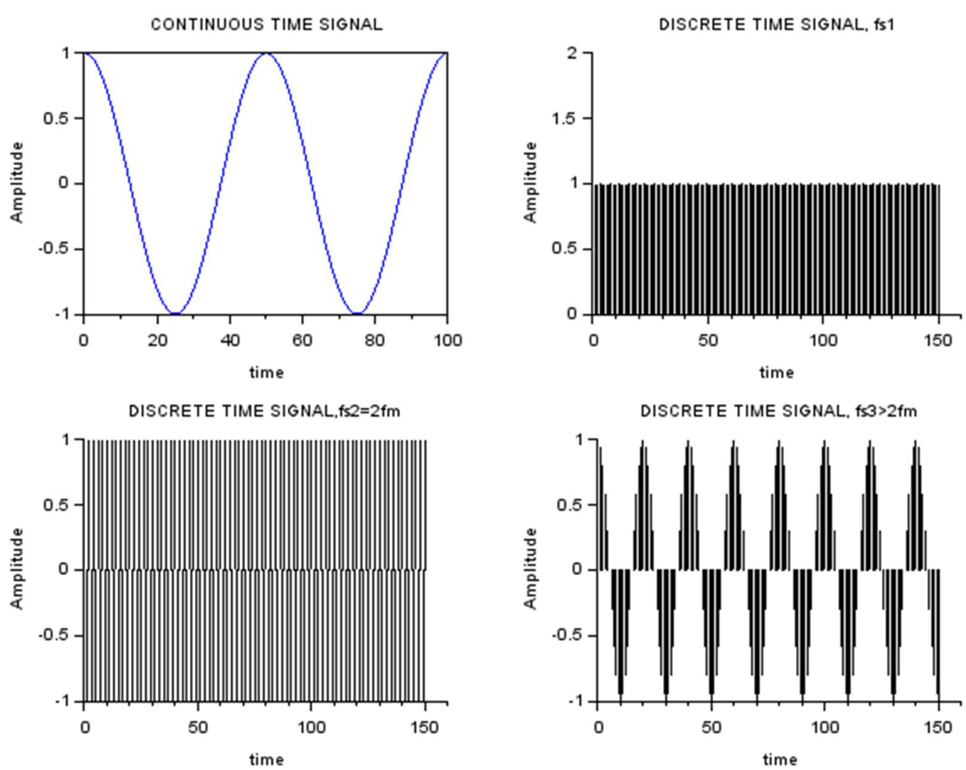
Fig:-4 Sampling of an analog signal x(t) into discrete-time signal x(nT) (a) and its inaccurate reconstruction (b)

PROGRAM

```
// PROGRAM FOR SAMPLING
// CONVERSION OF ANALOG SIGNAL TO DISCRETE SIGNAL
clc ;
clf ;
clear all;
t =0:0.01:100;
fm =0.02;
x = cos (2* %pi * fm * t );
subplot (2 ,2 ,1) ;
plot (t ,x ) ;
title ( 'CONTINUOUS TIME SIGNAL' ) ;
xlabel('time');
```

```
ylabel('Amplitude');
fs1 =0.002;
n =0:1:150;
x1 =cos (2* %pi * fm * n / fs1 ) ;
subplot (2 ,2 ,2) ;
plot2d3 (n , x1 ) ;
title ( 'DISCRETE TIME SIGNAL, fs1');
xlabel('time')
ylabel('Amplitude')
fs2 =0.04;
x2 =cos (2* %pi * fm * n / fs2 ) ;
subplot (2 ,2 ,3) ;
plot2d3 (n , x2 ) ;
title ( 'DISCRETE TIME SIGNAL,fs2=2fm' ) ;
xlabel('time')
ylabel('Amplitude')
fs3 =0.4;
x3 =cos (2* %pi * fm * n / fs3 ) ;
subplot (2 ,2 ,4) ;
plot2d3 (n , x3 ) ;
title ( 'DISCRETE TIME SIGNAL, fs3>2fm' ) ;
xlabel('time')
ylabel('Amplitude')
```

SIMULATION RESULTS



EXPERIMENT 3:

EXPERIMENT 3b: ALIASING AND ITS EFFECTS

Aim: To analyze the effects of aliasing frequencies for the various sampling frequencies of the discrete signal using SCI lab

Software Requirement: SCI Lab

Theory:

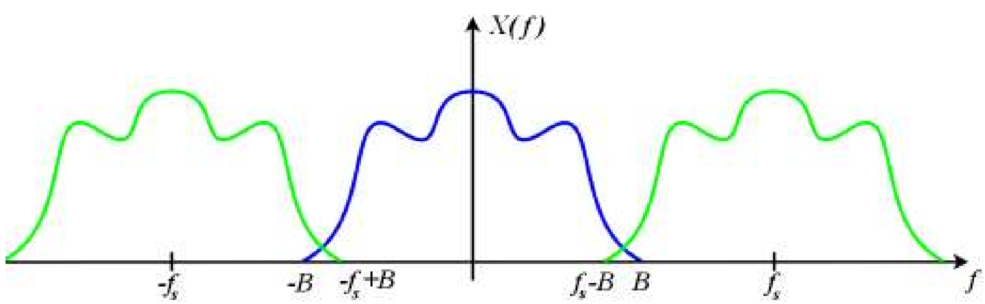
Aliasing – from alias – is an effect that makes different signals indistinguishable when sampled. It also refers to the difference between a signal reconstructed from samples and the original continuous signal, when the resolution is too low. Basically, aliasing depends on the sampling rate and freqency content of the signal.

Sampling Rate and Nyquist Frequency Limit

▪ For a given highest frequency B, we get the lower bound on the sampling frequency : 2B or Nyquist rate. For instance : for a signal whose maximum frequency is 16 KHz, we need a 32 KHz sampling rate.

▪ For a given sampling rate, we get the upper bound for frequency components : B<fs/2, or Nyquist frequency or Fmax. For instance : for a signal whose sampling rate is 48 KHz, we can sample signals up to 24 KHz.

In practice, a signal can never be perfectly bandlimited. Even if an ideal reconstruction could be made, the reconstructed signal would not be exactly the original signal. The error that corresponds to the failure of bandlimitation is referred to as aliasing.



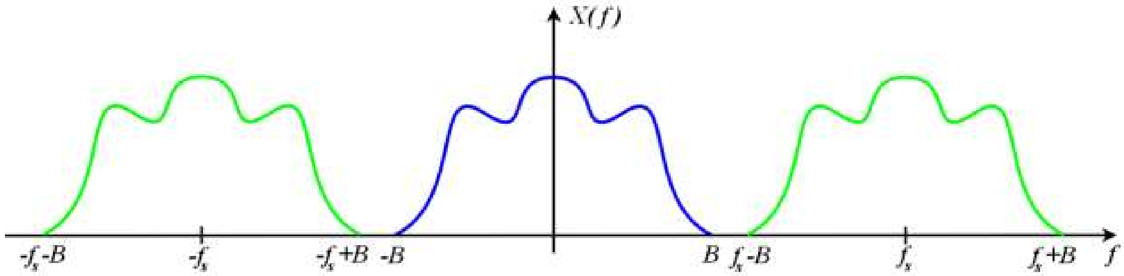
The blue sampled signal is insufficiently bandlimited. The overlapping edges of the green images are added and creating a spectrum

When a signal is sampled, its contents is reduced from real numbers to integer numbers. Values can be rounded to a superior or inferior value.

If a signal is sampled with a 32 KHz sampling rate, any frequency components above 16 KHz – Nyquist frequency, create an aliasing.

Any frequency component above fs/2 is indistinguishable from a lower-frequency component, called an alias, associated with one of the copies.

The Fourier transfom of the signal creates a symetrical image. The energy above the Nyquist frequency is transfered below this frequency.



The blue signal is bandlimited and properly sampled. The images do not overlap.

PROGRAM 1:

```
// ALIASING AND ITS EFFECTS
function [F]=aliasfrequency(f, s, s1)
if (s>2* f ) then
disp ('Aliasing not occured')
else
disp ('Aliasing occured')
end
F = f / s ;
for i =1:100
if ( abs ( F ) >0.5)
F =F - i ;
end
end
fa = F * s1 ;
disp ( fa ,'Frequency of Reconstructed Signal is ')
endfunction

f=input('Enter the frequency ');
s =240; // sampling  frequency
s1 = s;
aliasfrequency (f , s )
s =140; // sampling  frequency
s1 = s;
aliasfrequency (f ,s , s1 )
s =90; //  sampling  frequency
s1 = s;
```

```
aliasfrequency (f ,s , s1 )
s =35;// sampling  frequency
s1 = s;
aliasfrequency (f ,s , s1 )
```

SIMULATION RESULT

```
Enter the frequency: 100


  "Aliasing not occurred"



  "Frequency of Reconstructed Signal is 100."

  "Aliasing occured"


  "Frequency of Reconstructed Signal is   -40.
 "

  "Aliasing occured"


  "Frequency of Reconstructed Signal is 10.000000
"

  "Aliasing occured"



  "Frequency of Reconstructed Signal is -5.0000000"
```

Or

PROGRAM 2 :

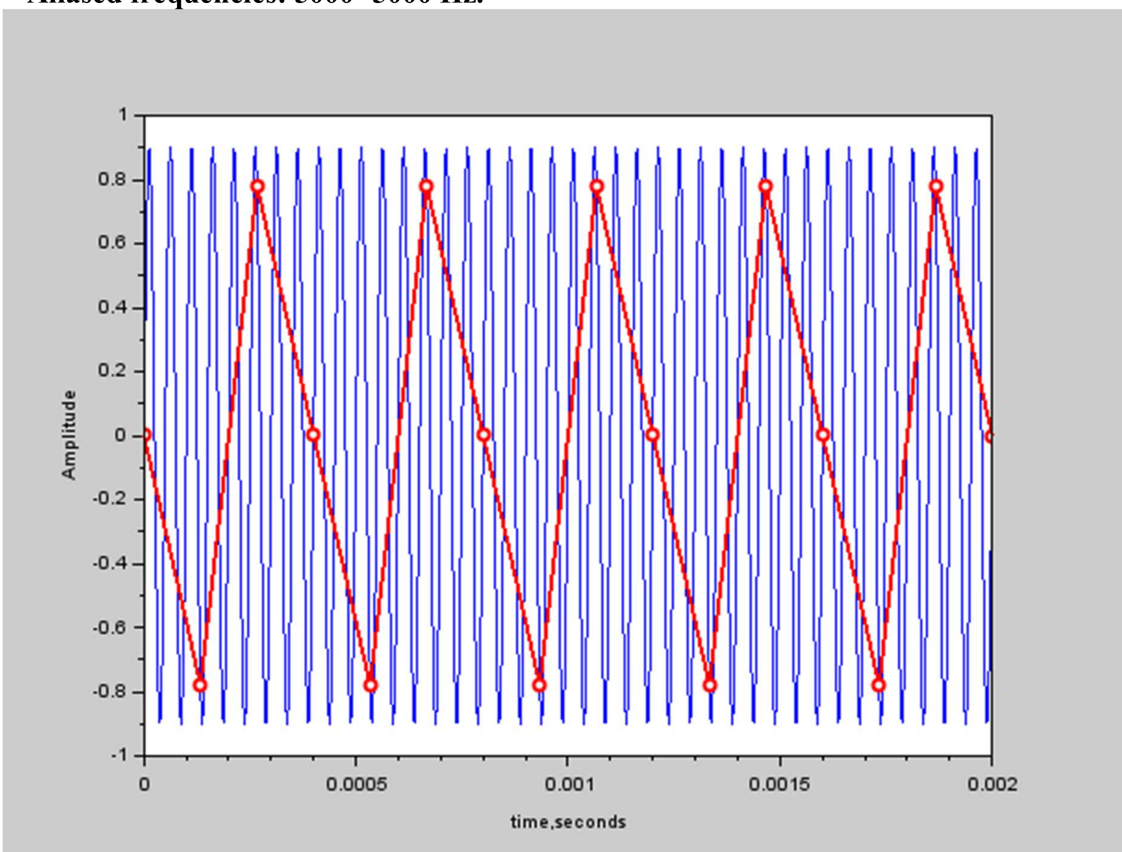
```
//------------------------------------------------------------------
//Aliasing - explore several different undersampling sins.
//Franz Hover MIT Mechanical Engineering
clear;
clf;
tfinal = .002 ;
fm = 20000*2*%pi;

fs = 7500*2*%pi; // change the sampling rate here [5500 6500 7500]

dtfi = 2*%pi/fm/20 ; // fine time scale

tfi = 0:dtfi:tfinal ;

dt = 2*%pi/fs ; // sampling time scale

t = 0:dt:tfinal ;

xfi = .9*sin(fm*tfi) ; // here are the signals

x = .9*sin(fm*t) ;

figure(1);clf;
//hold off;

//subplot(3,1,1);

plot(tfi,xfi);

//hold on;
//set(gca(),"auto_clear","on")
//mtlb_hold

plot(t,x,'ro-','LineWidth',2);
xlabel('time,seconds');
ylabel('Amplitude')

disp('Save the figure if necessary');
nfi = length(xfi);
fxfi = fft(xfi)/nfi;
wfi = [0:nfi-1]/(nfi-1)*2*%pi/dtfi ;
n = length(x);
fx = fft(x)/n ;
w = [0:n-1]/(n-1)*2*%pi/dt;
figure(2);clf;
//hold off;
plot(w/2/%pi,abs(fx),wfi/2/%pi,abs(fxfi));
//hold ;
plot(modulo(fm,fs)/2/%pi,0,'r*','LineWidth',2);
```

```
plot(modulo(-fm,fs)/2/%pi,0,'r*','LineWidth',2);
disp(sprintf('Sampling freq: %g Hz', fs/2/%pi));
disp(sprintf('Aliased frequencies: %g %g Hz.', ...
modulo(fm,fs)/2/%pi,modulo(-fm,fs)/2/%pi));
disp('Save figure if you want');
```

SIMULATION RESULT

"Sampling freq: 7500 Hz"

"Aliased frequencies: 5000 -5000 Hz."



Pre-lab questions:

1. Define Sampling rate
2. Why is signal to be sampled?
3. Define : sampling theorem
4. What is aliasing? When is aliasing occurred?
5. How to avoid aliasing?

Post-Lab questions:

1. If signal $x(t)= 5\sin(800\Pi t)$ is sampled what is the minimum sampling rate required to avoid aliasing ? Determine the discrete time signal after sampled
2. Define Alias frequency with an example
3. What will be the graphical representation of discrete time signal of x(t) for the following cases a) $f_s=f_m$ b) $f_s=2f_m$ c) $f_s<f_m$ d) $f_s>2f_m$

Result: