Institute of Information Technology (IIT)

Jahangirnagar University

Lab Report: 01

Submitted by:

Name: Zannat Hossain Tamim

Roll No: 1970

Lab Date: 29 May,2023

Submission Date: 12 June,2023

Experiment No : 01

Name of the Experiment:

Validation of sampling theorem.

Objective:

1.To understand the Sampling theorem.
2.To validate the theorem by varying sampling theorem frequency.

Theory :

Sampling theorem is a fundamental concept in digital signal processing that relates to the accurate representation and reconstruction of continuous signals using discrete samples.It states that in order to faithfully reconstruct a continuous signal from its discrete samples, the sampling rate must be at least twice the maximum frequency present in the signal. This is often referred to as the Nyquist rate or the Nyquist frequency. By sampling a signal at a rate higher than the Nyquist rate, it is possible to avoid aliasing, which is the distortion that occurs when higher frequency components of a signal fold back and appear as lower frequency components.
The Nyquist theory can be mathematically expressed as follows:

$$Fs >= 2 * F$$

Where Fs is the sampling frequency and F is the signal frequency.
In telecommunications, Nyquist theory is essential for efficient and reliable data transmission. By applying appropriate sampling rates and modulation techniques, it is possible to transmit digital information over analog channels without significant loss of data. The theory also influences the design of analog-to-digital converters (ADCs) and digital-to-analog converters (DACs) used in various devices.

Method:

First, we begin by constructing an analog signal. Next, we generate discrete sequences from this signal using different sampling frequencies: one that is less than 2F, another that is equal to 2F, and a third that is greater than 2F. Then, we proceed to reconstruct the analog signal from these discrete sequences and observe the results. We compare the accuracy of the reconstructed signals and identify the instances where noise is more pronounced or prevalent.

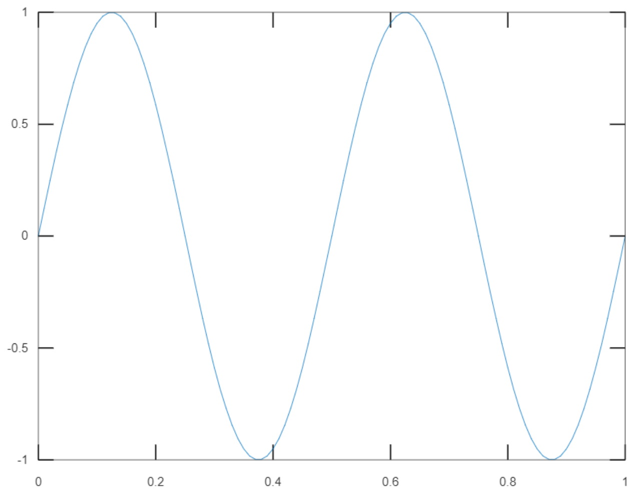
Lets a signal is Y=A*sin(2*pi*f*t+theta).

Matlab Code: (Construct a analog signal)

```
A=1;f=2;theta=0;
t=0:0.01:1;
Y=A*sin(2*pi*f*t+theta);
plot(t,Y);
```

Output :

```
octave:1> source("lab1_nyquestTheorem.m")
```

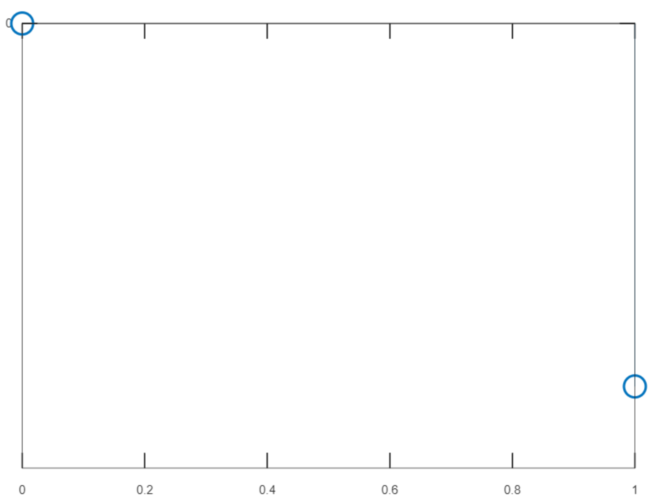


(i) Matlab Code: (Create Discrete sequence of signal for fs=.5f)

```
fs=.5f;ts=1/fs;n=fs;n1=0:ts:n*ts;
xs=A*sin(2*pi*f*n1+theta);
stem(n1,xs);
```

Output :

```
octave:2> source("lab1_nyquestTheorem.m")
```

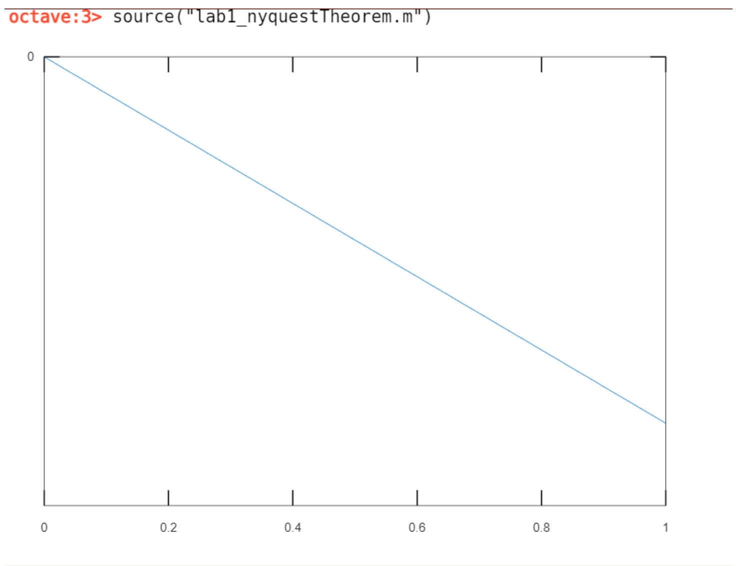


Matlab Code: (Reconstruct signal for fs=.5f)

```
t1=linspace(0,max(n1),100);
```

```
xr=interp1(n1,xs,t1,'spline');
plot(t1,xr);
```

Output :

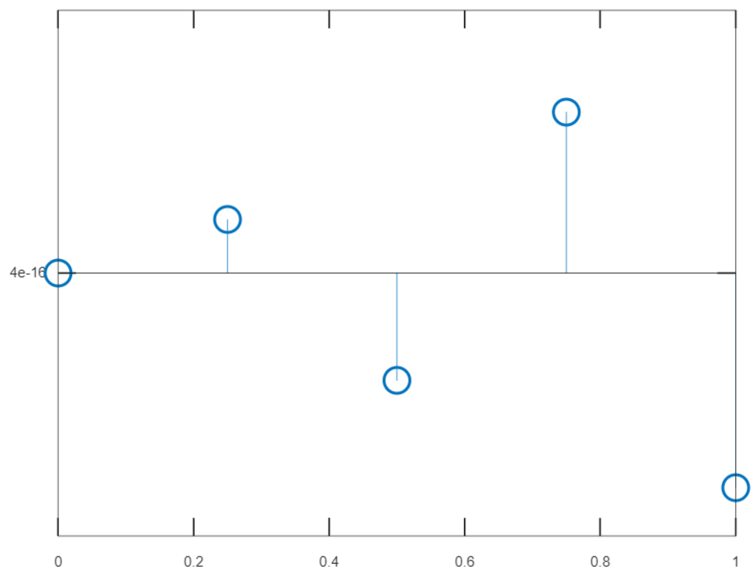


(ii) Matlab Code: (Create Discrete sequence of signal for fs=2*f)

```
fs=2*f;ts=1/fs;n=fs;n1=0:ts:n*ts;
xs=A*sin(2*pi*f*n1+theta);
stem(n1,xs);
```

Output :

```
octave:4> source("lab1_nyquestTheorem.m")
```

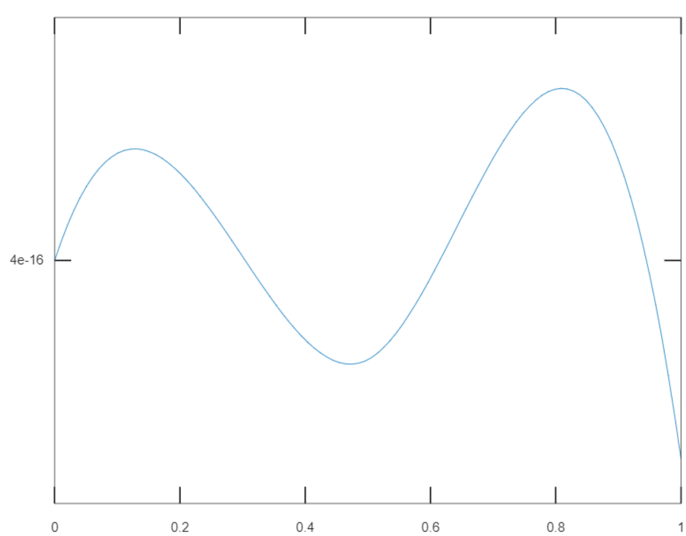


Matlab Code: (Reconstruct signal for fs=2*f)

```
t1=linspace(0,max(n1),100);
xr=interp1(n1,xs,t1,'spline');
plot(t1,xr);
```

Output :

```
octave:5> source("lab1_nyquestTheorem.m")
```



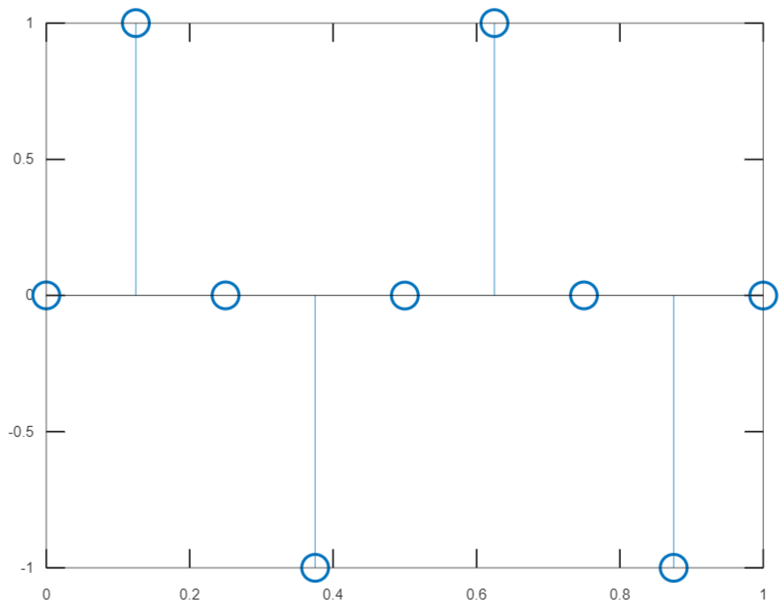
(iii) Matlab Code: (Create Discrete sequence of signal for fs=4*f)

```
fs=4*f;ts=1/fs;n=fs;n1=0:ts:n*ts;
xs=A*sin(2*pi*f*n1+theta);
```

```
stem(n1,xs);
```

Output :

```
octave:6> source("lab1_nyquestTheorem.m")
```

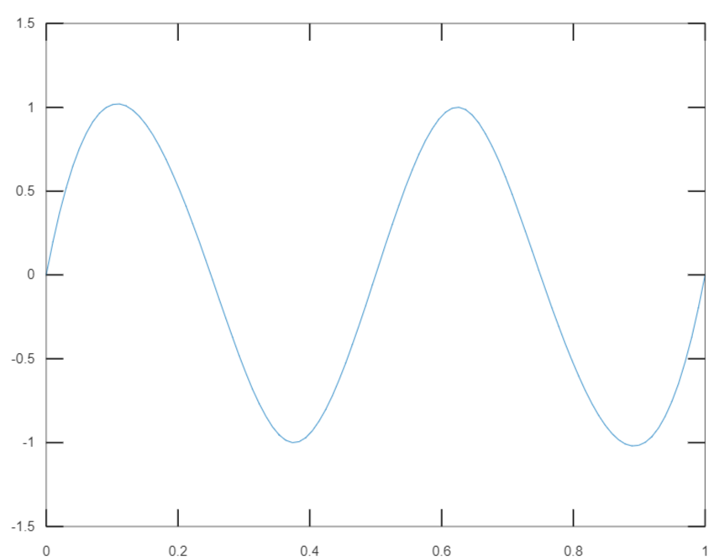


Matlab Code: (Reconstruct signal for fs=4*f)

```
t1=linspace(0,max(n1),100);
 xr=interp1(n1,xs,t1,'spline');
 plot(t1,xr);
```

Output :

```
octave:7> source("lab1_nyquestTheorem.m")
```



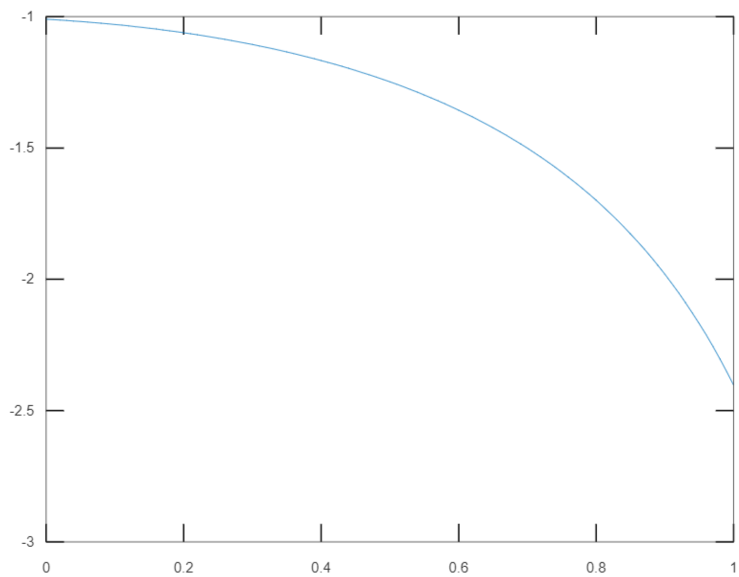
Signal 1: y=sec(2*A-5+n1).

Matlab Code: (Construct analog signal)

```
A=1;f=2;
t=0:0.01:1;
Y=sec(2*A-5+t);
plot(t,Y);
```

Output :

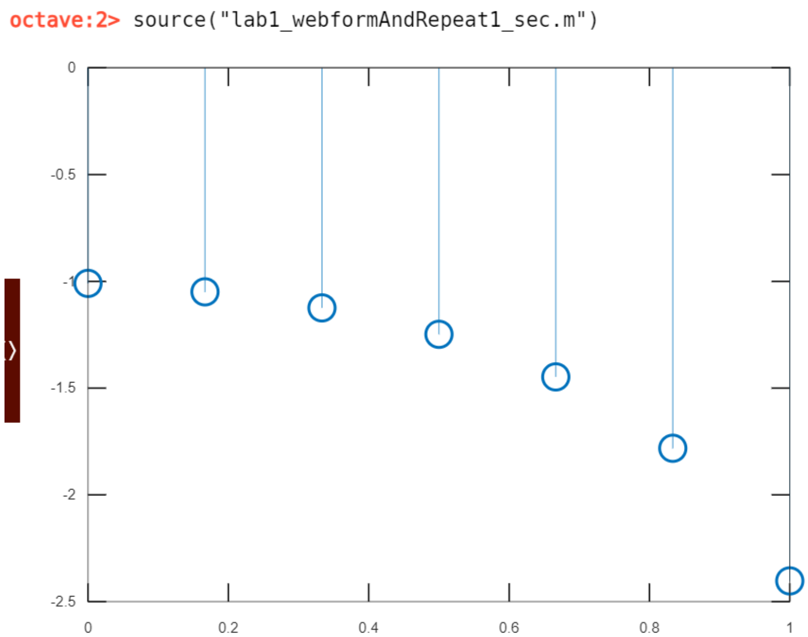
```
octave:1> source("lab1_webformAndRepeat1_sec.m")
```



(i) Matlab Code: (Create Discrete sequence of signal for fs=3*f)

```
fs=3*f;ts=1/fs;n=fs;
n1=0:ts:n*ts;
xs=sec(2*A-5+n1);
stem(n1,xs);
```

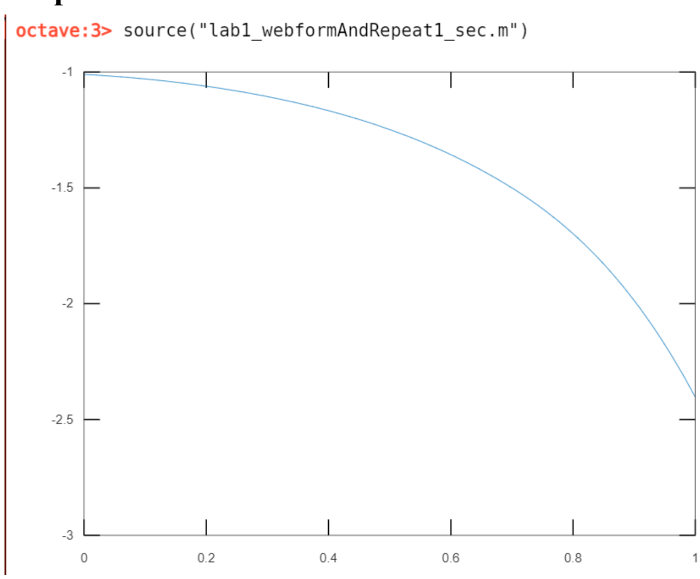
Output :



Matlab Code: (Reconstruct signal for fs=3*f)

```
t1=linspace(0,max(n1),100);
xr=interp1(n1,xs,t1,'spline');
plot(t1,xr);
```

Output :

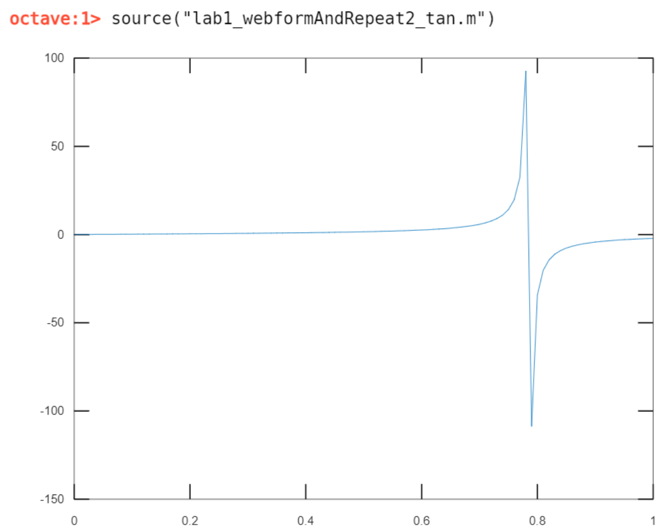


Signal 2: Y=A*tan(f*t+theta);

Matlab Code: (Construct analog signal)

```
A=1;f=2;theta=0;
 t=0:0.01:1;
 Y=A*tan(f*t+theta);
 plot(t,Y);
```

Output :

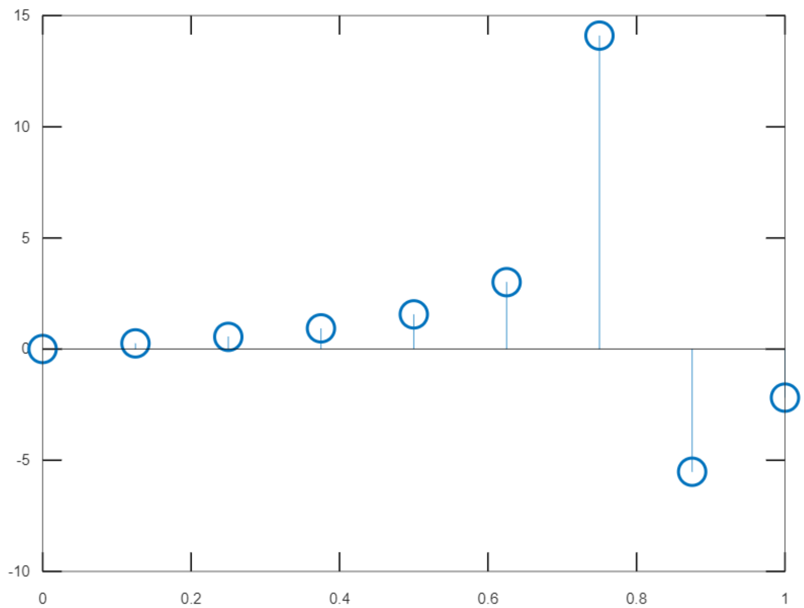


(i) Matlab Code: (Create Discrete sequence of signal for fs=1.5*f)

```
fs=4*f;ts=1/fs;n=fs;
n1=0:ts:n*ts;
 xs=A*tan(f*n1+theta);
 stem(n1,xs);
```

Output :

```
octave:2> source("lab1_webformAndRepeat2_tan.m")
```

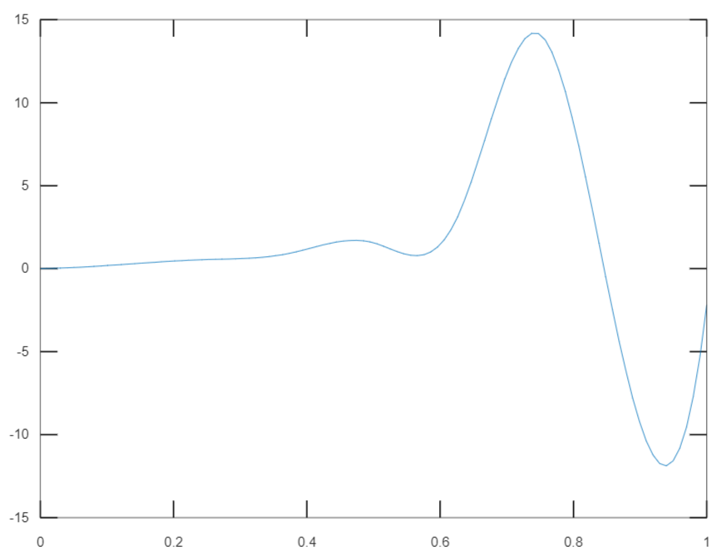


Matlab Code: (Reconstruct signal for fs=1.5*f)

```
t1=linspace(0,max(n1),100);
 xr=interp1(n1,xs,t1,'spline');
 plot(t1,xr);
```

Output :

```
octave:3> source("lab1_webformAndRepeat2_tan.m")
```



Results and Discussion:

The following plots show the original signal and the reconstructed signals for sampling frequency 3F and 1.5F.
As can be seen, as the sample rate rises, the reconstructed signals get steadily more precise. The reconstructed signal is the same as the original signal when the sampling frequency is 3F and some information is changed when sampling frequency is 1.5F .

Reference:
1.https://www.geeksforgeeks.org/analog-to-digital-conversion/[Accessed 12 June 2023,1:30pm]
2.https://uomustansiriyah.edu.iq/media/lectures/5/5_2020_12_26!03_09_06_PM.pdf[Accessed 12 June 2023,1:35pm]