Institute of Information Technology (IIT)

Jahangirnagar University

Lab Report: 01

Course Code: ICT-4104

<u>Submitted by:</u>

Name: Md Shakil Ahmed

Roll No: 2013

Lab Date: 23.05.2023

Submission Date: 13.06.2023

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:

Nyquist Theorem: The Nyquist theorem is also known as the sampling theorem. It is the principle to accurately reproduce a pure sine wave measurement, or sample, rate, which must be at least twice its frequency. The Nyquist theorem underpins all analog-to-digital conversion and is used in digital audio and video to reduce aliasing. The Nyquist theorem is also known as the Nyquist-Shannon theorem or the Whittaker-Nyquist-Shannon sampling theorem.

To avoid aliasing and accurately reconstructing the original signal, the Nyquist theorem sets the minimum sampling rate at twice the bandwidth (2B) of the signal. This sampling rate is often referred to as the Nyquist rate.

$$Fs > 2F$$

Where,
Fs= Sampling frequency;
F= Maximum frequency of input signal.

Signal Reconstruction: The sampling process produces a discrete time signal from a continuous time signal by examining the value of the continuous time signal at equally spaced points in time. Reconstruction, also known as interpolation, attempts to perform an opposite process that produces a continuous time signal coinciding with the points of the discrete time signal. Because the sampling process for general sets of signals is not invertible, there are numerous possible reconstructions from a given discrete time signal, each of which would sample to that signal at the appropriate sampling rate. This module will introduce some of these reconstruction schemes.

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.

Problem-01:

Let’s a signal 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);
xlabel('Time (sec)');
ylabel('X_A');
title('Analogue (Continuous) Input Signal', 'Linewidth',5);
```

Output:

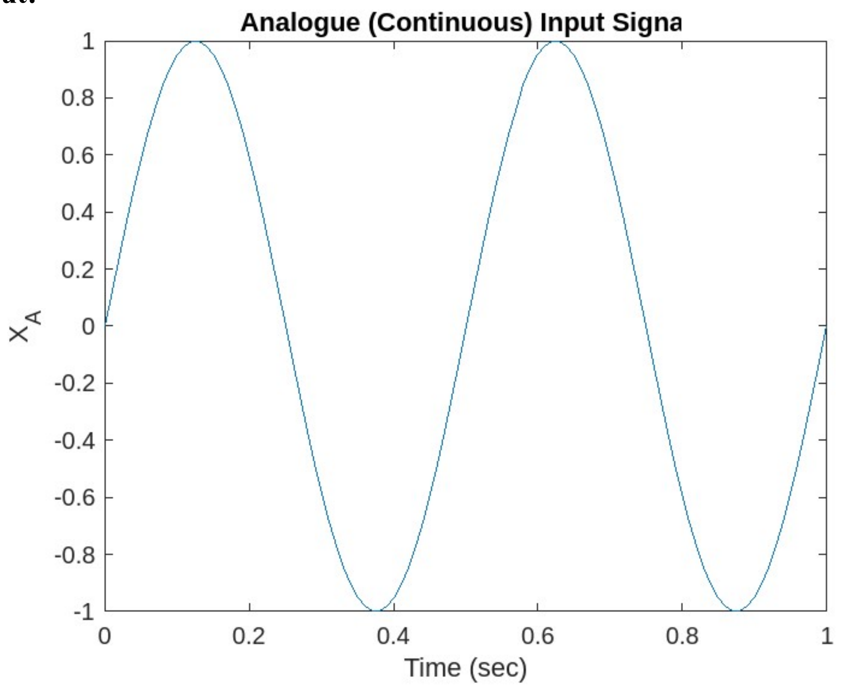

Fig-01: Analog Signal

(i) Matlab Code: (Create Discrete sequence of signal for fs=6*F)

```
Fs=6*F;Ts=1/Fs;
n=Fs;
n1=0:Ts:n*Ts;
Xs=A*sin(2*pi*F*n1+theta);
stem(n1,Xs);
xlabel('Sampling(n)');

ylabel('X_S');

title('Discrete Time Signal', 'Linewidth',5);
```

Output:

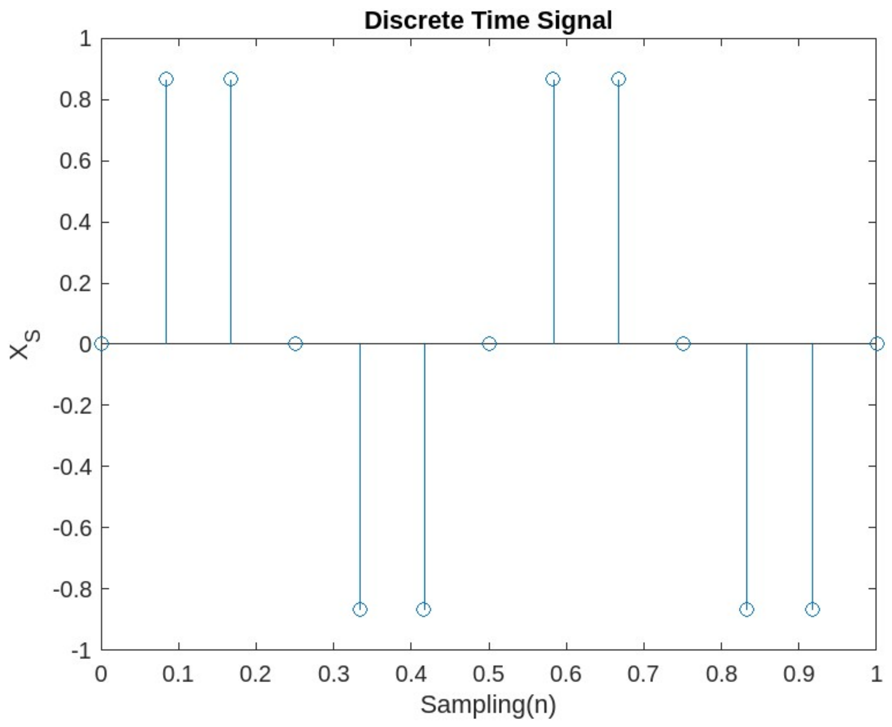

Fig-02: Digital Signal

Matlab Code: (Reconstruct signal for fs=6*F)

```
t1=linspace(0,max(n1),100);
Xr=interp1(n1,Xs,t1,'spline');
plot(t1,Xr);
xlabel('Time (sec)');
ylabel('X_A');
title('Reconstructed Signal');
```

Output:

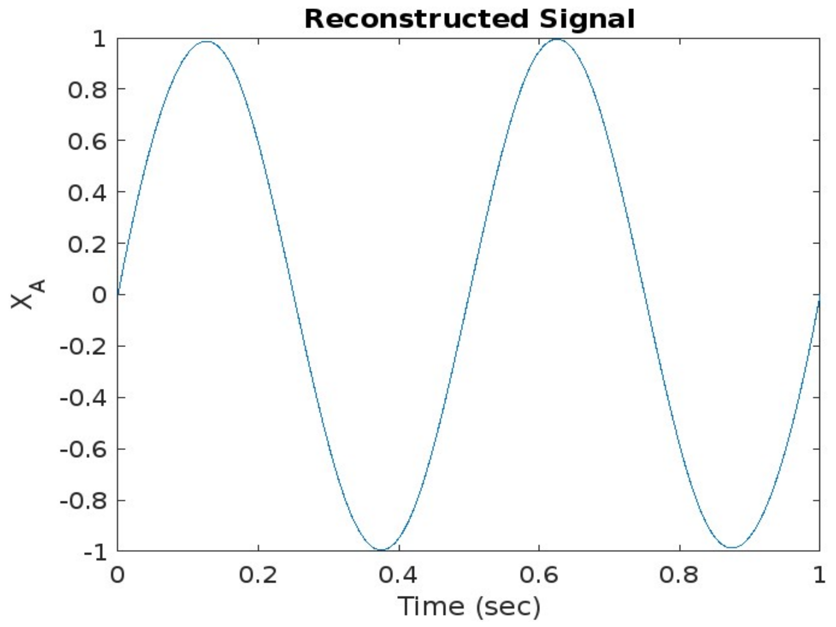

Fig-03: Reconstructed Signal

(ii) Matlab Code: (Create Discrete sequence of signal for fs=1.5*F)

```
Fs=1.5*F;Ts=1/Fs;
n=Fs;
n1=0:Ts:n*Ts;
Xs=A*sin(2*pi*F*n1+theta);
stem(n1,Xs);
xlabel('Sampling(n)');
ylabel('X_S');
title('Constructed Discrete Signal (Fs=1.5*F)', 'Linewidth',5);
```

Output:

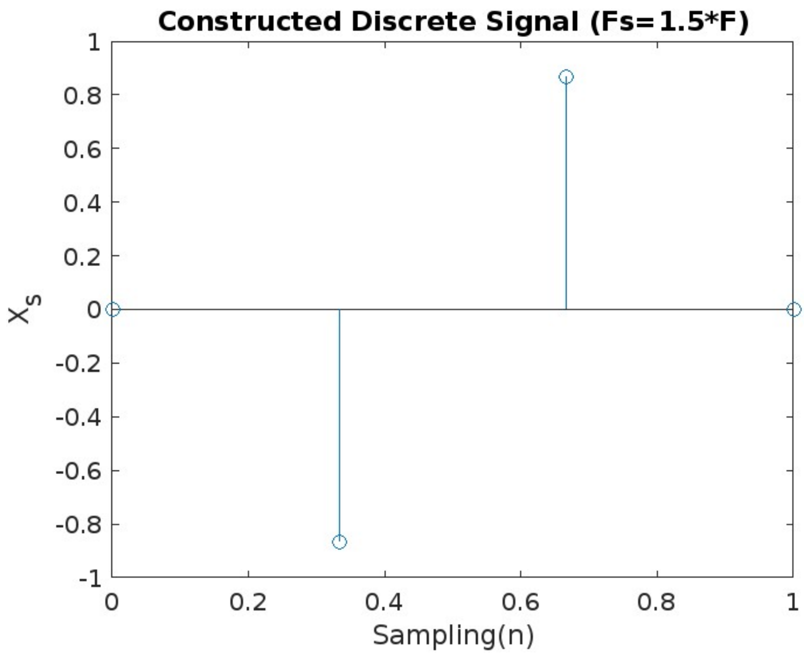

Fig-04: Discrete Signal (Fs=1.5*F)

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

```
t1=linspace(0,max(n1),100);
Xr=interp1(n1,Xs,t1,'spline');
plot(t1,Xr);
xlabel('Time (sec)');
ylabel('X_A');
title('Reconstructed Signal');
```

Output:

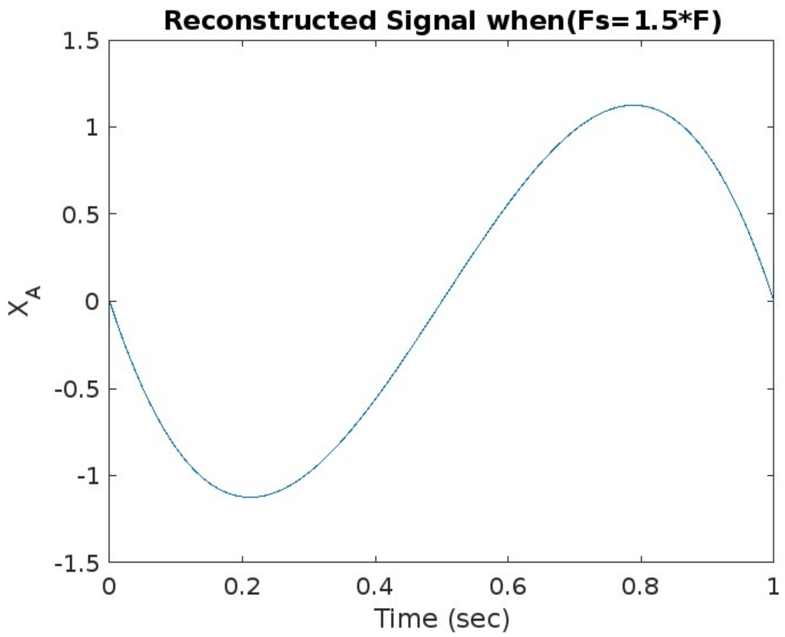

Fig-05: Reconstructed Signal (Fs=1.5*F)

(iii) Matlab Code: (Create Discrete sequence of signal for Fs=1.5*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);
xlabel('Sampling(n)');
ylabel('X_S');
title('Constructed Discrete Signal (Fs=2*F)', 'Linewidth',5);
```

Output

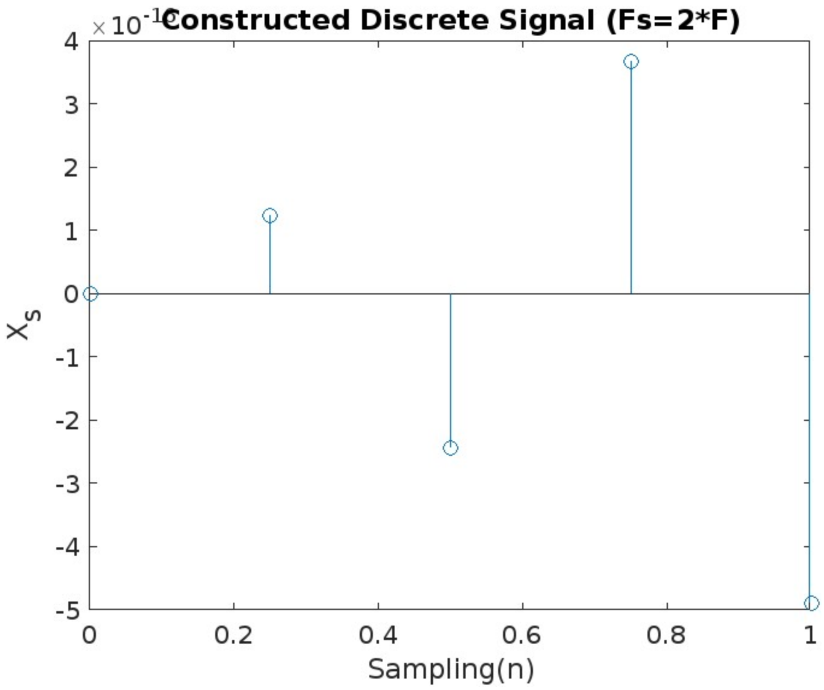

Fig-06: Discrete Signal (Fs=1.5*F)

Matlab Code: (Reconstruct signal for Fs=1.5*F)

```
t1=linspace(0,max(n1),100);
Xr=interp1(n1,Xs,t1,'spline');
plot(t1,Xr);
xlabel('Time (sec)');
ylabel('X_A');
title('Reconstructed Signal');
```

Output

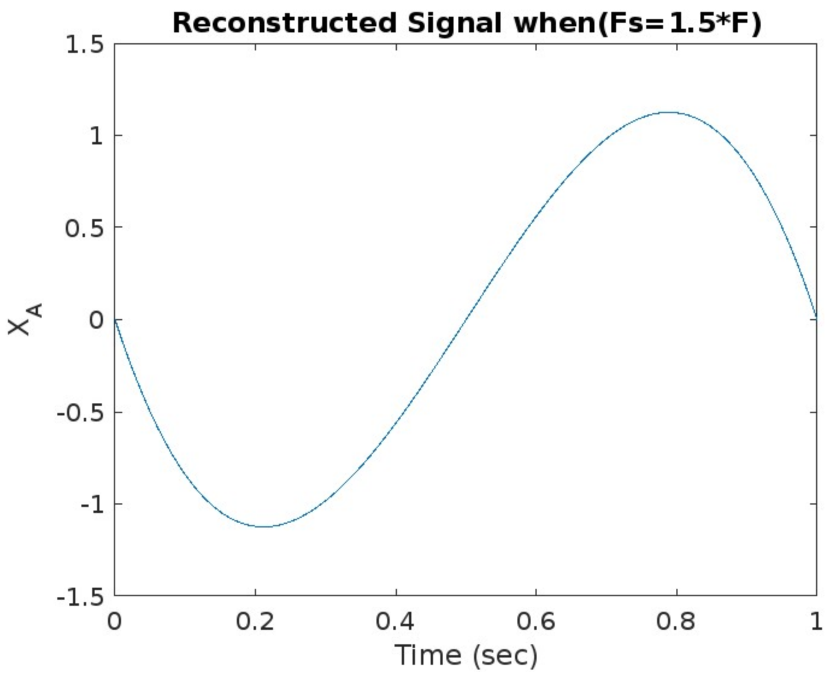

Fig-07: Reconstructed Signal (Fs=1.5*F)

Problem-02:

Signal-01 Y=A*cos(3*F*pi*t+theta);

MATLAB Code(construct a analog signal):

```
A=1.5;F=2.34;theta=0;
t=0:0.01:1;
Y=A*cos(3*F*pi*t+theta);
plot(t,Y);
xlabel('Time (sec)');
ylabel('X_A');
title('Analogue (Continuous) Input Signal', 'Linewidth',5);
```

Output

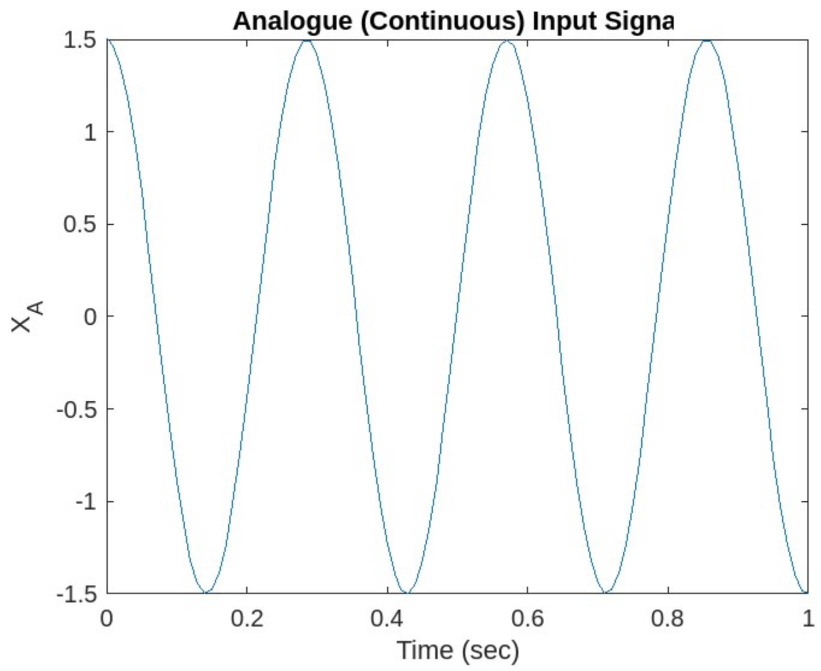

Fig-08: Analog input signal

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=A*sin(2*pi*F*n1+theta);
stem(n1,Xs);
xlabel('Sampling(n)');
ylabel('X_S');
title('Constructed Discrete Signal (Fs=3*F)', 'Linewidth',5);
```

Output

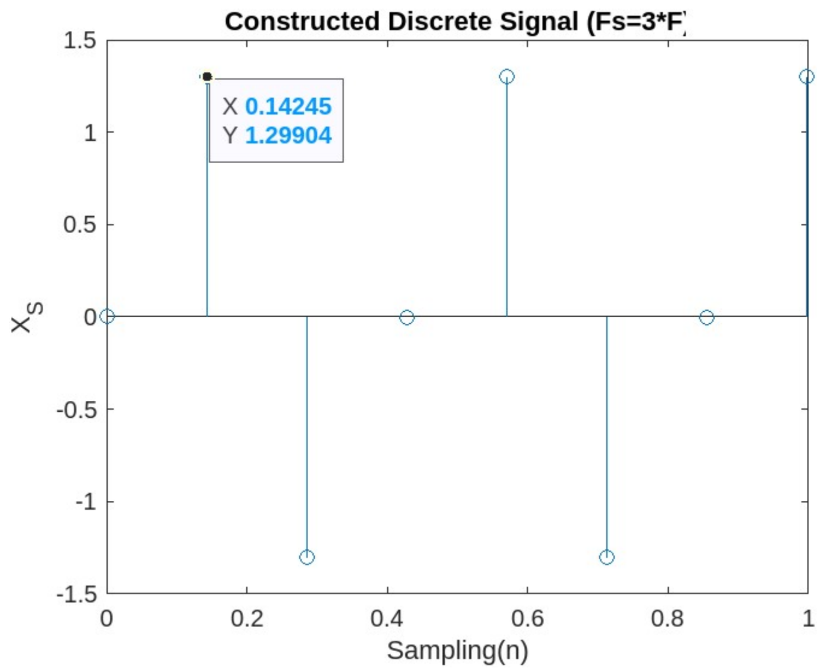

Fig-09: Discrete Signal (Fs=3*F)

Matlab Code: (Reconstruct signal for Fs=3*F)

```
t1=linspace(0,max(n1),100);
Xr=interp1(n1,Xs,t1,'spline');
plot(t1,Xr);
xlabel('Time (sec)');
ylabel('X_A');
title('Reconstructed Signal when(Fs=3*F)');
```

Output

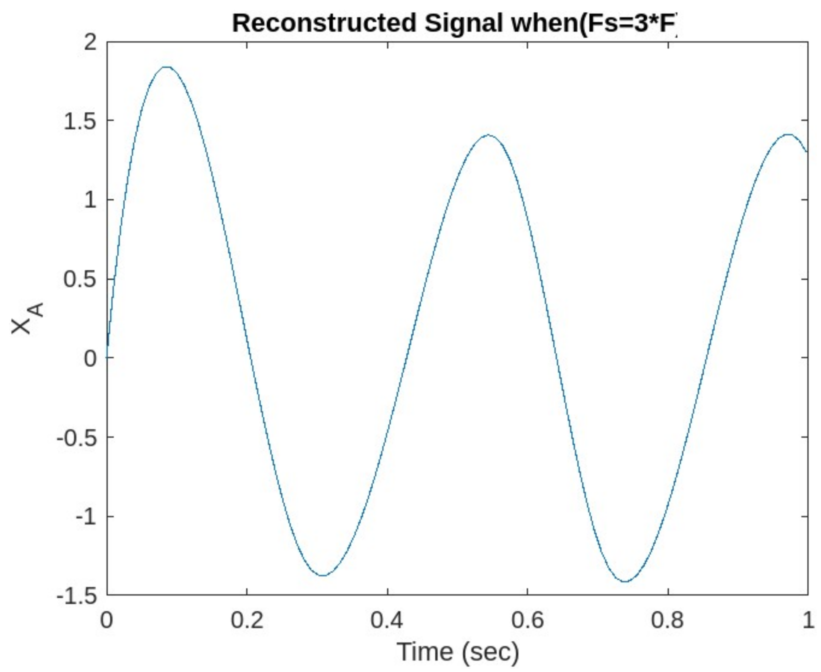

Fig-10: Reconstructed Signal (Fs=3*F)

Signal-02 Y=A*tan(F*t-theta)

MATLAB Code(construct a analog signal):

```
A=1;F=2.25;theta=0;
t=0:0.01:1;
Y=A*tan(F*t-theta);
plot(t,Y);

xlabel('Time (sec)');
ylabel('X_A');
title('Analogue (Continuous) Input Signal', 'Linewidth',5);
```

Output

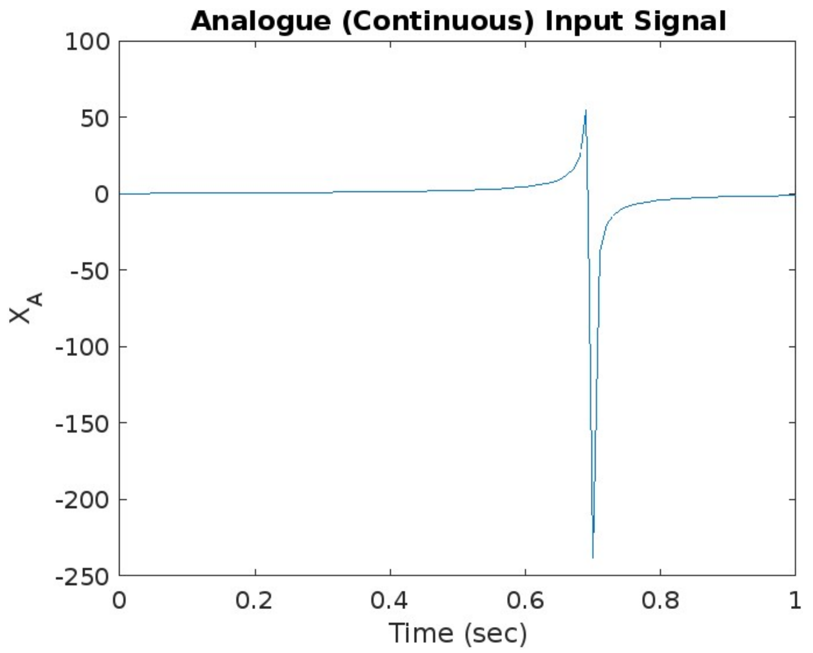

Fig-11: Analog input Signal

Matlab Code: (Create Discrete sequence of signal for Fs=3*F)

```
%discrete signal
Fs=3*F;Ts=1/Fs;
n=Fs;
n1=0:Ts:n*Ts;
Xs=A*sin(2*pi*F*n1+theta);
stem(n1,Xs);
xlabel('Sampling(n)');
ylabel('X_S');
title('Constructed Discrete Signal (Fs=3*F)', 'Linewidth',5);
```

Output

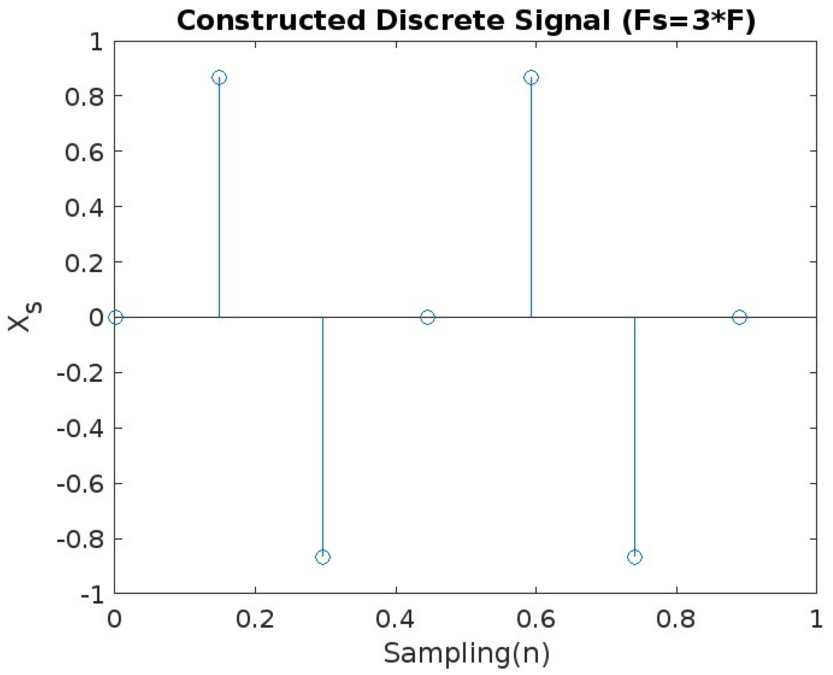

Fig-12: Discrete Signal (Fs=3*F)

Matlab Code: (Reconstruct signal for Fs=3*F)

```
%reconstructed signla
t1=linspace(0,max(n1),100);
Xr=interp1(n1,Xs,t1,'spline');
plot(t1,Xr);
xlabel('Time (sec)');

ylabel('X_A');

title('Reconstructed Signal when(Fs=3*F)');
```

Output

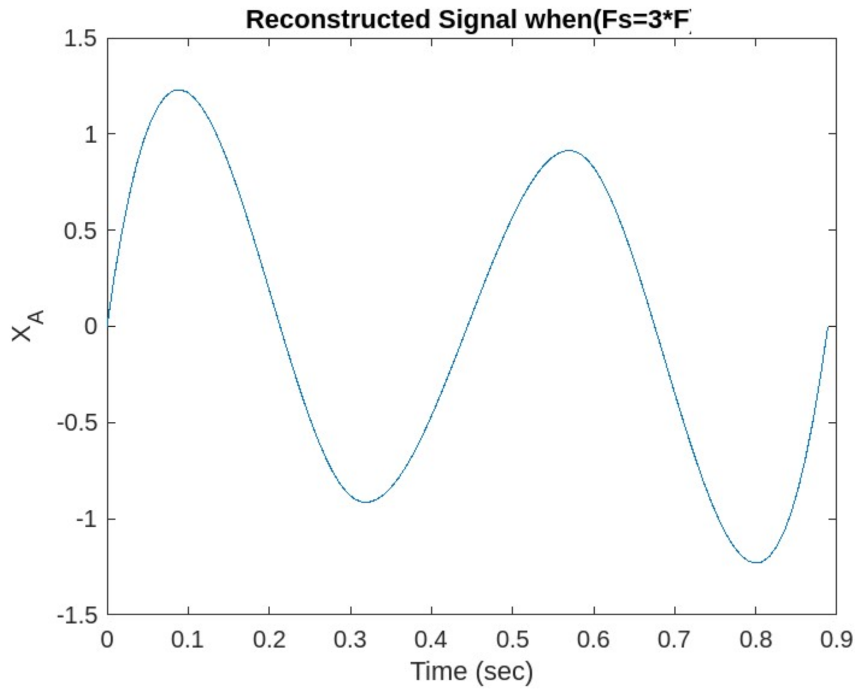

Fig-13: Reconstructed Signal (Fs=3*F)

Discussion:

When we change the sampling signal in the time domain, it can have a significant impact on the reconstructed signal. The most important factor is the sampling frequency. If the sampling frequency is too low, then aliasing will occur, which means that some of the higher frequencies in the original signal will be folded into the lower frequencies. This can cause distortion and loss of information in the reconstructed signal.

The Nyquist theorem states that in order to accurately reconstruct a continuous-time signal from its samples, the sampling frequency must be at least twice the maximum frequency present in the signal. To validate the Nyquist theorem for a given signal, I have to check if the sampling frequency satisfies the Nyquist. It calculates the maximum frequency present in the signal using the max_frequency/2 predicate. Then, it checks if the sampling frequency is greater than or equal to twice the maximum frequency as per the Nyquist theorem.

Reference:

1. https://www.geeksforgeeks.org/analog-to-digital-conversion/[Accessed 12 June 2023,10:00pm]
2. https://www.techtarget.com/whatis/definition/Nyquist-Theorem [Accessed 12 June 2023,1:00pm]