**Aim ↬ To study the concept and fundamentals of Sampling Theorem.**

**Software Required ↬** MATLAB

**Theory ↬**

The Sampling Theorem, also known as the Nyquist-Shannon Sampling Theorem, is a fundamental principle in signal processing that defines the conditions under which a continuous signal can be sampled and perfectly reconstructed from its samples. The theorem states that a signal can be completely reconstructed if it is sampled at a rate greater than twice its highest frequency component. This minimum sampling rate is known as the Nyquist rate.

Mathematically, if a continuous-time signal x(t) contains no frequencies higher than $f_m$ (the maximum frequency), it must be sampled at a frequency $f_s$ that satisfies:

$$f_s > 2f_m$$

If the sampling frequency is lower than this threshold, aliasing occurs, causing distortion in the reconstructed signal. Aliasing happens when different signals become indistinguishable upon sampling, resulting in an inaccurate representation.

To reconstruct a sampled signal, interpolation is used, typically through a low-pass filter represented by the sinc function. The reconstructed signal $x_r(t)$ can be expressed as:

$$x_r(t) = \sum_{n=-\infty}^{\infty} x[n] \cdot sinc\left(\frac{t - nT}{T}\right)$$

where T is the sampling period, and x[n]are the sampled values. This equation shows how sampled values spread over time to recreate the continuous signal.

Sampling can be uniform, with equal intervals, or non-uniform, with variable intervals based on signal characteristics. Choosing the right sampling rate is crucial to avoid aliasing and ensure accurate signal reconstruction. Anti-aliasing filters are often employed to limit bandwidth before sampling, helping maintain signal integrity in applications like telecommunications, audio processing, and image processing.

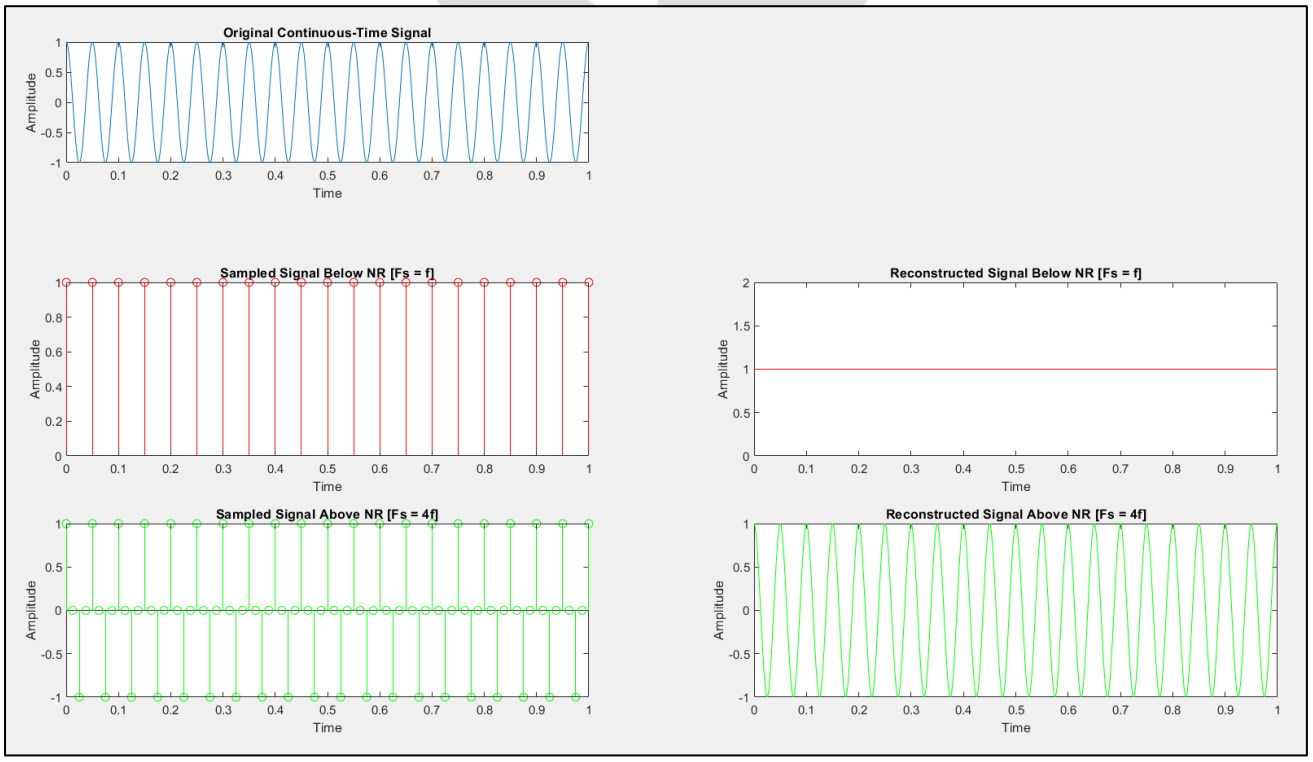
## Code ↬

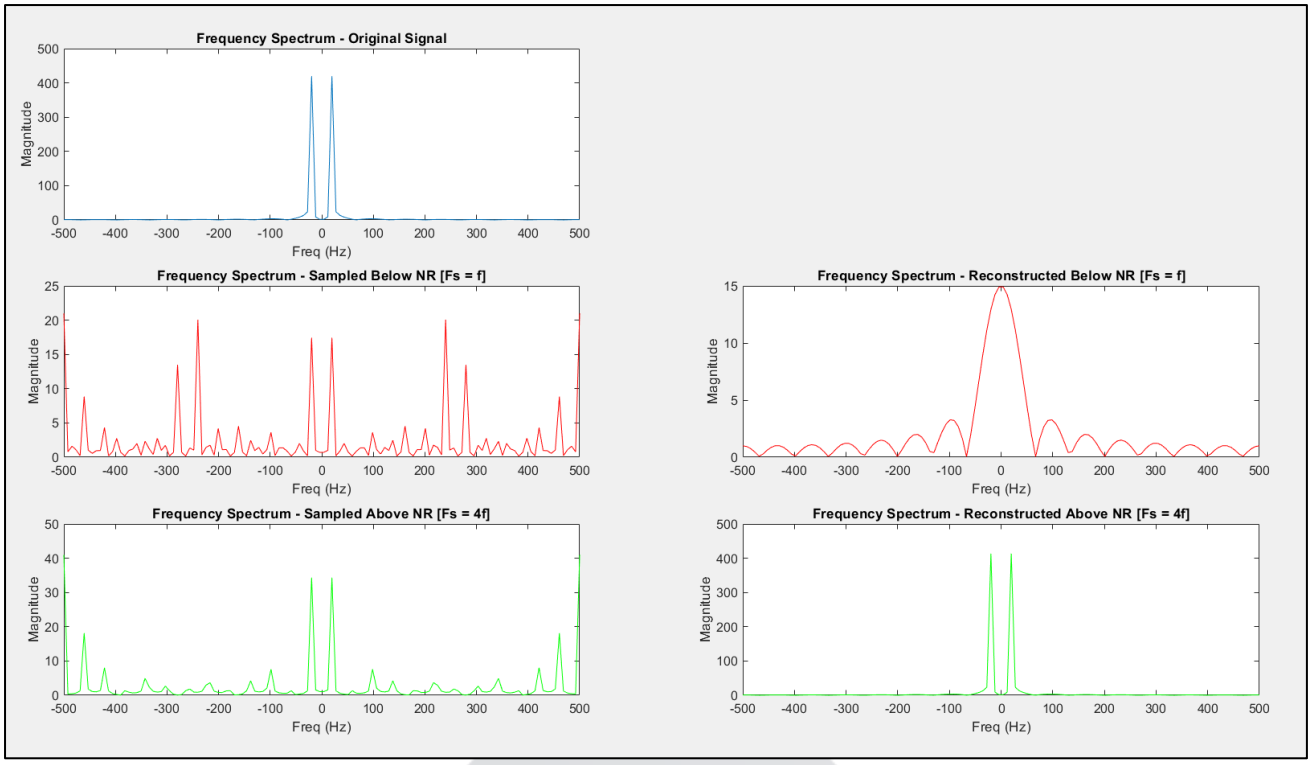
```
%Sampling Theorem
Fs = 1000;
t = 0:1/Fs:1;
f = 20;

x = cos(2*pi*f*t);

subplot(4,2,1);
plot(t, x);
xlabel('Time');
ylabel('Amplitude');
title('Original Continuous-Time Signal');

Fs1 = f;
Fs2 = 4*f;

t1 = 0:1/Fs1:1;
t2 = 0:1/Fs2:1;

x1 = cos(2*pi*f*t1);
x2 = cos(2*pi*f*t2);

subplot(3,2,3);
stem(t1, x1, 'r');
xlabel('Time');
ylabel('Amplitude');
title('Sampled Signal Below NR [Fs = f]');

subplot(3,2,5);
stem(t2, x2, 'g');
xlabel('Time');
ylabel('Amplitude');
title('Sampled Signal Above NR [Fs = 4f]');

t_interp = 0:1/Fs:1;
x1_interp = interp1(t1, x1, t_interp, 'spline');
x2_interp = interp1(t2, x2, t_interp, 'spline');

subplot(3,2,4);
```

```
plot(t_interp, x1_interp, 'r');
xlabel('Time');
ylabel('Amplitude');
title('Reconstructed Signal Below NR [Fs = f]');

subplot(3,2,6);
plot(t_interp, x2_interp, 'g');
xlabel('Time');
ylabel('Amplitude');
title('Reconstructed Signal Above NR [Fs = 4f]');

n = 128;
f_freq = linspace(-Fs/2, Fs/2, n);

X = zeros(1, n);
X1 = zeros(1, n);
X2 = zeros(1, n);
X1_interp = zeros(1, n);
X2_interp = zeros(1, n);

for k = 1:n
    freq = f_freq(k);
    X(k) = sum(x .* exp(-1i * 2 * pi * freq * t));
    X1(k) = sum(x1 .* exp(-1i * 2 * pi * freq * t1));
    X2(k) = sum(x2 .* exp(-1i * 2 * pi * freq * t2));
    X1_interp(k) = sum(x1_interp .* exp(-1i * 2 * pi * freq * t_interp));
    X2_interp(k) = sum(x2_interp .* exp(-1i * 2 * pi * freq * t_interp));
end

figure;

subplot(3,2,1);
plot(f_freq, abs(X));
xlabel('Freq (Hz)');
ylabel('Magnitude');
title('Frequency Spectrum - Original Signal');

subplot(3,2,3);
plot(f_freq, abs(X1), 'r');
xlabel('Freq (Hz)');
```

```
ylabel('Magnitude');
title('Frequency Spectrum - Sampled Below NR [Fs = f]');

subplot(3,2,4);
plot(f_freq, abs(X1_interp), 'r');
xlabel('Freq (Hz)');
ylabel('Magnitude');
title('Frequency Spectrum - Reconstructed Below NR [Fs = f]');

subplot(3,2,5);
plot(f_freq, abs(X2), 'g');
xlabel('Freq (Hz)');
ylabel('Magnitude');
title('Frequency Spectrum - Sampled Above NR [Fs = 4f]');

subplot(3,2,6);
plot(f_freq, abs(X2_interp), 'g');
xlabel('Freq (Hz)');
ylabel('Magnitude');
title('Frequency Spectrum - Reconstructed Above NR [Fs = 4f]');
```

## Output ↬






## Result ↬

The results successfully verified the need for sampling at or above the Nyquist rate to prevent aliasing and ensure accurate signal reconstruction, highlighting differences in frequency spectra.

## Conclusion ↬

In conclusion, the experiment validated the Sampling Theorem, emphasizing the importance of sampling at rates equal to or greater than twice the maximum frequency to maintain the original signal's integrity. This principle is crucial for applications in telecommunications and audio processing.

## Precautions ↬

- Ensure the sampling rate is set correctly according to the Nyquist criterion to prevent aliasing.
- Use an anti-aliasing filter prior to sampling to limit the bandwidth of the input signal.
- Verify the accuracy of the signal generation formulas to avoid computational errors.
- Label and title plots clearly to differentiate between original and reconstructed signals effectively.