**Theorem Definition**

The sampling theorem can be defined as the conversion of an analog signal into a discrete form by taking the sampling frequency as twice the input analog signal frequency. Input signal frequency denoted by Fm and sampling signal frequency denoted by Fs.

The output sample signal is represented by the samples. These samples are maintained with a gap, these gaps are termed as sample period or sampling interval (Ts). And the reciprocal of the sampling period is known as “sampling frequency” or “sampling rate”. The number of samples is represented in the sampled signal is indicated by the sampling rate.

Sampling frequency **Fs=1/Ts**

### Sampling Theorem Statement

Sampling theorem states that “continues form of a time-variant signal can be represented in the discrete form of a signal with help of samples and the sampled (discrete) signal can be recovered to original form when the sampling signal frequency Fs having the greater frequency value than or equal to the input signal frequency Fm.

**Fs ≥ 2Fm**

If the sampling frequency (Fs) equals twice the input signal frequency (Fm), then such a condition is called the Nyquist Criteria for sampling. When sampling frequency equals twice the input signal frequency is known as “Nyquist rate”.

**Fs=2Fm**

If the sampling frequency (Fs) is less than twice the input signal frequency, such criteria called an Aliasing effect.

**Fs<2Fm**

So, there are three conditions that are possible from the sampling frequency criteria. They are sampling, Nyquist and aliasing states. Now we will see the Nyquist sampling theorem.

### Nyquist Sampling Theorem

Nyquist sampling theorem states that the sampling signal frequency should be double the input signal’s highest frequency component to get distortion less output signal. As per the scientist’s name, Harry Nyquist this is named as Nyquist sampling theorem.

**Fs=2Fm**

Program:

%lab:3 Vaification of Samplig Theorey

k=input('Enter the number of cycles'); % 10

a=input('Enter the input signal amplitude'); % 50

f=input('Enter the input signal Frequency'); % 100

t=0:1/(f.\*f):k/f;

y=a\*cos(2.\*pi.\*f.\*t);

figure(1)

subplot(2,2,1);

plot(t,y);

xlabel('Time (T)');

ylabel('Amplitude (A)');

title('Input Signal [Anjan KC]');

fn=2\*f;

%Under Sampling

fs=0.75.\*fn;

tx=0:1/fs:k/f;

ys=a.\*cos(2.\*pi.\*f.\*tx);

subplot(2,2,2);

stem(tx,ys);

hold on; % Reconstructing

plot(tx,ys,'r'); % Reconstructed original in Red

xlabel('Time (T)');

ylabel('Amplitude (A)');

title('Under Sampling [Anjan KC]');

%Fine Sampling

fs=fn;

tx=0:1/fs:k/f;

ys=a.\*cos(2.\*pi.\*f.\*tx);

subplot(2,2,3);

stem(tx,ys);

hold on; % Reconstructing

plot(tx,ys,'r'); % Reconstructed original in Red

xlabel('Time (T)');

ylabel('Amplitude (A)');

title('Fine Sampling [Anjan KC]');

% over Sampling

fs=10.\*fn;

tx=0:1/fs:k/f;

ys=a.\*cos(2.\*pi.\*f.\*tx);

subplot(2,2,4);

stem(tx,ys);

hold on; % Reconstructing

plot(tx,ys,'r'); % Reconstructed original in Red

xlabel('Time (T)');

ylabel('Amplitude (A)');

title('Over Sampling [Anjan KC]');

Output:



