**PRACTICAL 1 :**

|  |  |
| --- | --- |
| **U19EC046 - Harsh Suthar** | **Date: 30-07-2021** |

**AIM**

Write a MATLAB program to sample the given continuous time signal at different sampling frequencies and reconstruct the input signal for Perfect Sampling, Under Sampling and Over Sampling.

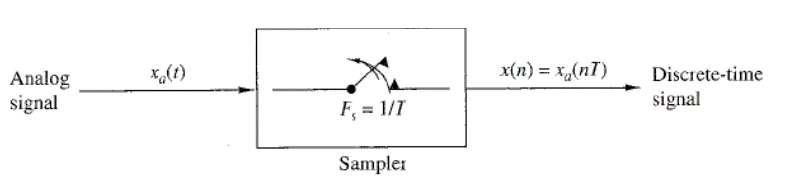
**THEORY**

* Sampling converts analog (continuous time signal) into discrete time signal.
* Obtained by taking “samples” of the continuous-time signal at discrete time instants.

𝑥𝑎 (𝑡) − 𝑐𝑜𝑛𝑡𝑖𝑛𝑢𝑜𝑢𝑠 𝑡𝑖𝑚𝑒 𝑠𝑖𝑔𝑛𝑎𝑙 (𝑖𝑛𝑝𝑢𝑡)

𝑥𝑎 (𝑛𝑇) ≡ 𝑥(𝑛)𝑑𝑖𝑠𝑐𝑟𝑒𝑡𝑒 𝑡𝑖𝑚𝑒 𝑠𝑖𝑔𝑛𝑎𝑙 (𝑜𝑢𝑝𝑢𝑡)

Where, 𝑇 is called sampling interval/sampling period its reciprocal, 1/ 𝑇 = 𝐹𝑠 is called sampling rate/sampling frequency and −∞ < 𝑛 < ∞



Given: Equally spaced samples of 𝑥(𝑡): 𝑥(𝑛𝑇) 𝑤ℎ𝑒𝑟𝑒, 𝑛 = 0, ±1, ±2, ± 3, … ….

• 𝑥(𝑡) is band-limited,

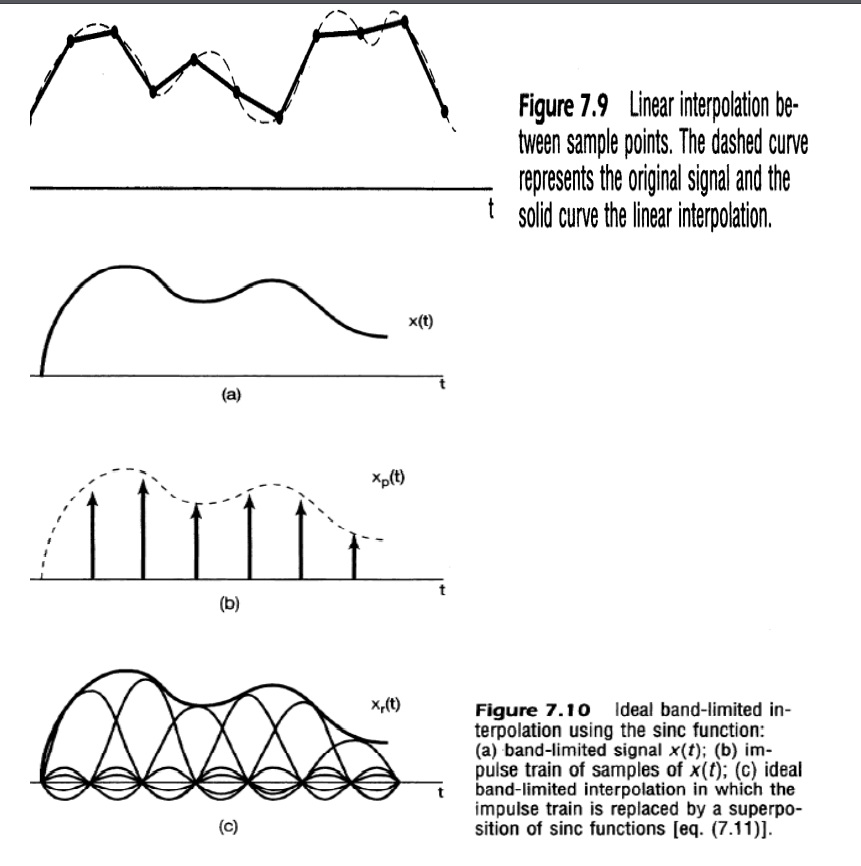
𝑋(𝜔) = 0; 𝑓𝑜𝑟 |𝜔| > 𝜔𝑚 rad/sec.

• If, 2𝜋 𝑇 ≡ 𝜔𝑠 > 2𝜔𝑚

• Then, 𝑥(𝑡) can be uniquely recovered using low pass filter.

• Interpolation: The fitting of a continuous signal to a set of sample values for reconstructing a function either approximately or exactly from its samples.

• Sample points may be connected by Straight line, higher order polynomials, other mathematical functions.



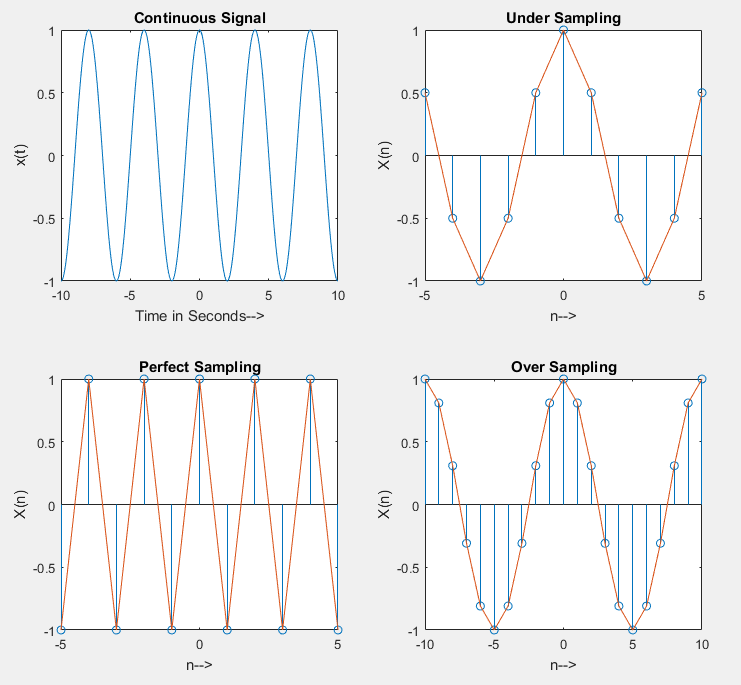
**ALGORITHM**

1. Define analog frequency fm and time window for the input sinusoidal signal.
2. Plot the input sinusoidal signal.
3. Define the sample frequency for three cases and n as per the sampling frequency.
4. Define the sampled signal using sampled frequency.\
5. Plot the sampled signal.

**CODE**

|  |
| --- |
| **clc;**  **close all;**  **t=-10:0.01:10;*%time window***  **fm=1/4;*%Analog Frequency***  ***%code for plotting input signal***  **x=cos(2\*pi\*fm\*t);*%input signal***  **subplot(2,2,1);**  **plot(t,x);**  **xlabel('Time in Seconds-->');**  **ylabel('x(t)');**  **title('Continuous Signal');**  ***%code for plotting fs<Nquist Rate***  **fs1=1.2\*fm;*%sampling freuqncy***  **n=-5:1:5;*%defining time range***  **xn1=cos(2\*pi\*n\*fm/fs1);**  **subplot(2,2,2);**  **stem(n,xn1);**  **hold on;**  **plot(n,xn1);**  **xlabel('n-->');**  **ylabel('X(n)');**  **title('Under Sampling');**  ***%code for plotting fs=Nquist Rate***  **fs2=2\*fm;*%sampling freuqncy***  **xn2=cos(2\*pi\*n\*fm/fs2);**  **subplot(2,2,3);**  **stem(n,xn2);**  **hold on;**  **plot(n,xn2);**  **xlabel('n-->');**  **ylabel('X(n)');**  **title('Perfect Sampling');**  ***%code for plotting fs>Nquist Rate***  **fs3=10\*fm;*%sampling freuqncy***  **n=-10:1:10;*%defining time range***  **xn3=cos(2\*pi\*n\*fm/fs3);**  **subplot(2,2,4);**  **stem(n,xn3);**  **hold on;**  **plot(n,xn3);**  **xlabel('n-->');**  **ylabel('X(n)');**  **title('Over Sampling');** |

**OUTPUT**



**CONCLUSION**

From result we can see that the signal can be totally remade in the event that where the Sampling frequency is more prominent than Nyquist Rate, in other cases aliasing is found and reproduction is fruitless.