**U19EC045 | DCOM | LAB X**

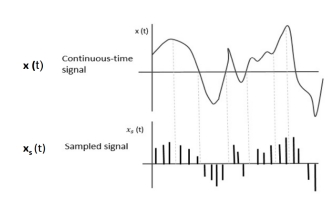
**AIM**

## To study and simulate Sampling theorem. Also observe the effect of over and under-sampling.

**THEORY**

### **Sampling:**

* The process of measuring the instantaneous values of continuous-time signal in a discrete form.
* Sample is a piece of data taken from the whole data which is continuous in the time domain.
* When a source generates an analog signal and if that has to be digitized, having 1s and Os i.e., High or Low, the signal has to be discretized in time.
* This discretization of analog signal is called as Sampling.



When **x(**t) is multiplied by a periodic impulse train, the sampled signal **xs**(t) is obtained.

### **Sampling rate:** To discretize the signals, the gap between the samples should be fixed. That gap can be termed as a sampling period or sampling time

### IMG_257

### **Sampling frequency:** It is the reciprocal of the sampling period. This sampling frequency, or Sampling rate. It denotes the number of samples taken per second.

Ts -sampling time

fs -sampling frequency or the sampling rate

The rate of sampling should be such that the data in the message signal should neither be lost nor get over-lapped.

### **Nyquist criterion:** It states that a message waveform can be correctly reconstructed, if sampling frequency is greater than double the highest frequency.

### **Nyquist Rate:**

### Suppose that a signal is band-limited and W is the highest frequency.

### IMG_258

### Therefore, for effective reproduction of the original signal, the sampling rate should be twice the highest frequency.

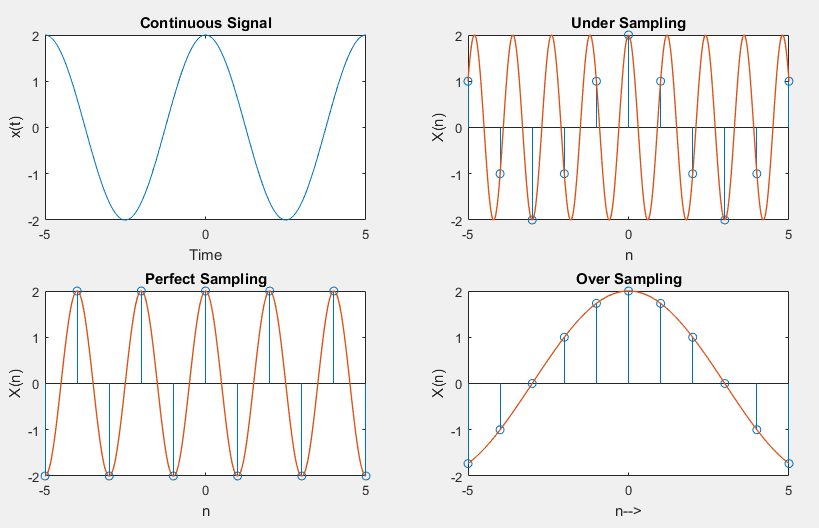
Where, fS is the sampling rate and **W** is the highest frequency

This rate of sampling is called as **Nyquist rate** and theorem called, **Sampling Theorem**.

**MATLAB CODE**

|  |
| --- |
| ***%user inputs for frequenct and amplitede***  **fm=input('Enter Signal frequency : ');**  **Am=input('Enter Signal Amplitude : ');**  ***%0.01 spaced time array***  **t=(-1/fm):0.01:(1/fm);**  ***%original signal***  **x=Am\*cos(2\*pi\*fm\*t);**  ***%plots***  **subplot(2,2,1);**  **plot(t,x);**  **xlabel('Time');**  **ylabel('x(t)');**  **title('Continuous Signal');**  ***%fs < 2fm undersampling case***  **fs\_under=1.2\*fm;**  **n=(-1/fm):1:(1/fm);**  **xn\_under=Am\*cos(2\*pi\*n\*fm/fs\_under);**  **subplot(2,2,2);**  **stem(n,xn\_under);**  **hold on;**  **plot(t,Am\*cos(2\*pi\*t\*fm/fs\_under) , 'LineWidth', 1)**  **xlabel('n');**  **ylabel('X(n)');**  **title('Under Sampling');**  ***%fs = 2fm ideal case***  **fs\_ideal=2\*fm;**  **xn\_ideal=Am\*cos(2\*pi\*n\*fm/fs\_ideal);**  **subplot(2,2,3);**  **stem(n,xn\_ideal);**  **hold on;**  **plot(t,Am\*cos(2\*pi\*t\*fm/fs\_ideal) , 'LineWidth', 1)**  **xlabel('n');**  **ylabel('X(n)');**  **title('Perfect Sampling');**  ***%fs > 2fm oversampling case***  **fs\_over=12\*fm;**  **xn\_over=Am\*cos(2\*pi\*n\*fm/fs\_over);**  **subplot(2,2,4);**  **stem(n,xn\_over);**  **hold on;**  **plot(t,Am\*cos(2\*pi\*t\*fm/fs\_over) , 'LineWidth', 1)**  **xlabel('n-->');**  **ylabel('X(n)');**  **title('Over Sampling');** |

**OUTPUT**



**CONCLUSION**

If the signal x (t) is sampled above the Nyquist rate, the original signal can be recovered, and if it is sampled below the Nyquist rate, the signal cannot be recovered.