# Experiment # 1

# Sampling and Quantization

**Objective:**

Study sampling, quantization and re-construction of a signal using Matlab Simulation.

**Sampling Theorem and Quantization:**

The sampling theorem states that if the highest frequency in the signal spectrum is B (Hz), the signal can be reconstructed from its samples, taken at a rate not less than 2B samples per second. This means that in order to transmit the information in a continuous-time signal, we need only transmit its samples as shown in figure 1. Unfortunately, the sample values are still not digital because they lie in a continuous range and can take on any one of the infinite values in the range. This difficulty is resolved by a process called "**quantization**", where each sample is approximated, or rounded off to the nearest quantized level, as shown in figure 1. Amplitudes of the signal m(t) lie in the range (-mp, mp), which is partitioned into L intervals, each of magnitude $\Delta v = 2m_p / L$. Each sample amplitude is approximated to the midpoint of the interval in which the sample value falls. Each sample is now approximated to one of the L numbers. The information is thus digitized.

The quantized signal is an approximation of the original one. We can improve the accuracy of the quantized signal to any desired degree by increasing the number of levels L.
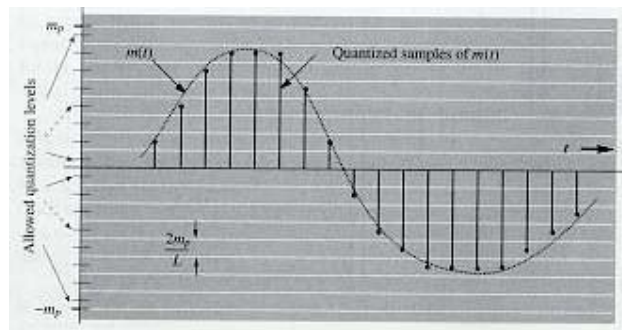


Figure 1: Sampling of a Signal.

### MATLAB functions used in this Simulation:

The details for the matlab commands and functions that may be useful in this simulation experiment are given below.


### zeros

ZEROS  Zeros array.

  ZEROS(N) is an N-by-N matrix of zeros.

  ZEROS(M,N) or ZEROS([M,N]) is an M-by-N matrix of zeros.


### ones

ONES   Ones array.

  ONES(N) is an N-by-N matrix of ones.

  ONES(M,N) or ONES([M,N]) is an M-by-N matrix of ones.


### filter

FILTER One-dimensional digital filter.

  Y = FILTER(B,A,X) filters the data in vector X with the

  filter described by vectors A and B to create the filtered

  data Y.  The filter is a "Direct Form II Transposed"

  implementation of the standard difference equation:


  a(1)*y(n) = b(1)*x(n) + b(2)*x(n-1) + ... + b(nb+1)*x(n-nb)

              - a(2)*y(n-1) - ... - a(na+1)*y(n-na)


  If a(1) is not equal to 1, FILTER normalizes the filter

  coefficients by a(1).


### plot

PLOT   Linear plot.

  PLOT(X,Y) plots vector Y versus vector X. If X or Y is a matrix,

  then the vector is plotted versus the rows or columns of the matrix,

  whichever line up.  If X is a scalar and Y is a vector, length(Y)

  disconnected points are plotted.

**stem**

STEM   Discrete sequence or "stem" plot.

   STEM(Y) plots the data sequence Y as stems from the x axis

   terminated with circles for the data value.


   STEM(X,Y) plots the data sequence Y at the values specified

   in X.


**quant**

QUANT Discretize values as multiples of a quantity.

   QUANT(X,Q) takes these inputs,

          X - Matrix, vector or scalar.

          Q - Minimum value.

        and returns values in X rounded to nearest multiple of Q


**fft**

FFT Discrete Fourier transform.

   FFT(X) is the discrete Fourier transform (DFT) of vector X.  For

   matrices, the FFT operation is applied to each column. For N-D

   arrays, the FFT operation operates on the first non-singleton

   dimension.


   FFT(X,N) is the N-point FFT, padded with zeros if X has less

   than N points and truncated if it has more.


**ifft**

IFFT Inverse discrete Fourier transform.

   IFFT(X) is the inverse discrete Fourier transform of X.


   IFFT(X,N) is the N-point inverse transform.


   IFFT(X,[],DIM) or IFFT(X,N,DIM) is the inverse discrete Fourier

   transform of X across the dimension DIM.

*Exercise:*

The following Signal is given:

$X = \cos(2\pi f_1 T) + (2/3)*\cos(2\pi f_2 T) - (2.5/3)*\cos(2\pi f_3 T)$

using Matlab;

(1) Generate a train of pulses, use the impulse response of a comb filter $H(z)=1/(1-(z)^{-4})$

(2) Sample the given signal

(3) Quantize the sampled signal

(4) Use a low pass filter to reconstruct the original signal.

Submit the hard copy with figures and soft copy of the program.