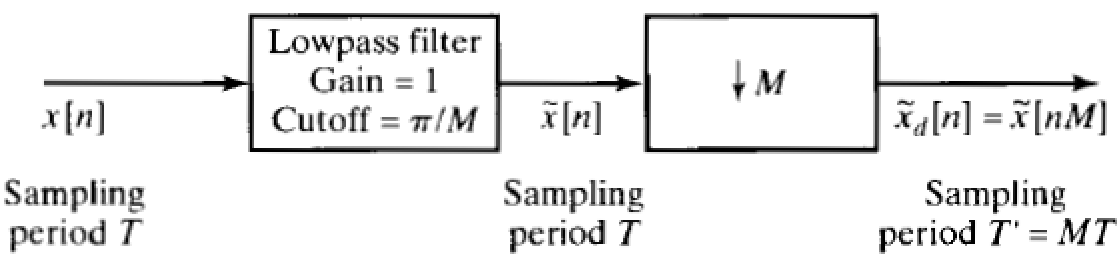
Lab- (DSP)

Sampling and Quantization

In this lab you will gain some practical knowledge about how to handle real world signals. In any modern signal processing system (e.g. in a telecommunication system), signal acquisition, its processing and efficient storage/transmission are the critical steps. In the class lectures, you have gained the knowledge about sampling of a continuous-time signal, change of sampling rates and their hierarchical criteria. Similarly you are familiar with the significance of quantization of a discrete signal. This lab covers the above mentioned tasks i.e., change of sampling rate and quantization.

Change of Sampling Rate



You are familiar with the above figure for downsampling. For a discrete-time signal to be sampled by the factor of M, you need to first pass the signal from the low pass filter with a certain cutoff frequency to avoid the frequency aliasing in the downsampled signal.

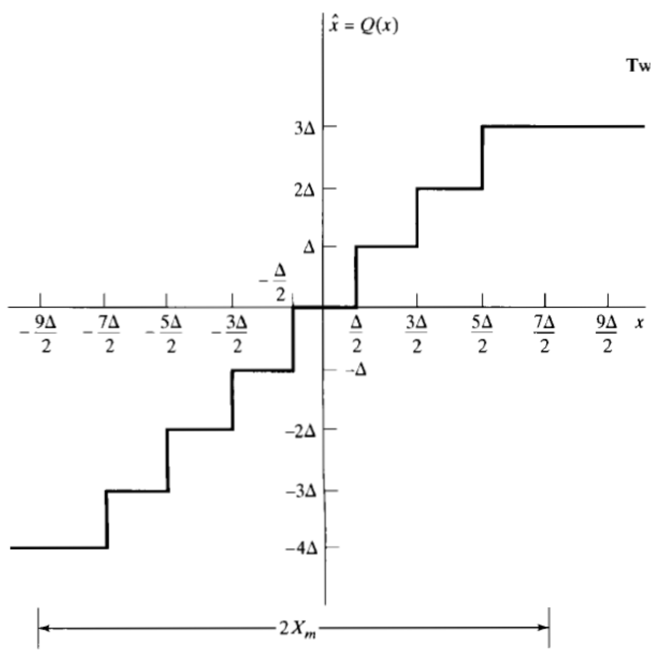
TASK-1:

You are given a speech signal. Consider it a discrete-time signal with the sampling frequency $f_s = 16\ kHz$.

1. Load the signal in Matlab using the function *wavread*. Listen to the signal using *wavplay*.

2. Design a 6th order low-pass butterworth filter. Hint: see Matlab help for *butter* and *filter*. The butter command takes the normalized cutoff frequency (in the range 0-1) as an input argument where the maximum 1 means $f_S/2$
3. Consider the maximum frequency of the speech signal $f_N = f_S/2$. Apply the filter.
4. Now downsample the filtered signal by the factor of 2 i.e., *M*=2. Do this manually by picking up every alternative sample and storing it in a different array. (use a loop)
5. See the Matlab help for the function *downsample*. Apply this function for downsampling the signal by the factors *M* = 3,5,10. Listen to the output signal in every case and prepare your conclusions. Also plot the spectrum of the input and output signal in a subplots for original and three cases for different *M*.
6. For *M* = 10, avoid the anti-aliasing filter and directly downsample the speech to listen if there is any difference. Also plot the spectrum in subplot for input and output signal

Quantization



The above mentioned figure a quantizer function you have learned recently. Given below is the quantization function in Matlab (algorithm)

```
X=(1:99)*(8/100)-4;
N = 8; %number of quantization levels.
% find the highest value point in the signal, round it to the upper limit.
% find the lowest value point in the signal, round it to the lower limit.
```

```
qstep = (high-low)/N;
Q = floor((X-low)/qstep);
low = low + qstep/2;
Y = low + qstep*Q;
```

TASK-2

1. Redo the above code in Matlab, plot the original and quantized signal in a subplot for N = 8, 16, 32, 64.
2. Explain each and every step in the code.