ECE4270
Fundamentals of DSP
Lecture 14

Changing the Sampling Rate Using Digital Filtering (II) & A/D Converter

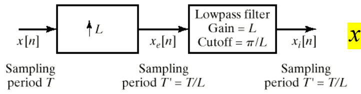
School of ECE
Center for Signal and Information Processing
Georgia Institute of Technology

Overview of Lecture

- Changing sampling rates
  - Decimation (last Lecture)
  - Interpolation
  - Non-Integer Rate Change
- Over-sampling to ease filtering
- Representation of A-to-D Converter
- Probabilistic analysis of quantization
  - Model
  - Signal-to-noise ratio
- Variation of SNR with Signal Level

Interpolation - I

$x_c(nT)$



$x_c(nT/L)$

$$x_e[n] = \sum_{k=-\infty}^{\infty} x[k]\delta[n-kL] = \begin{cases} x[n/L], & n = 0, \pm L, \ldots \\ 0, & \text{otherwise} \end{cases}$$

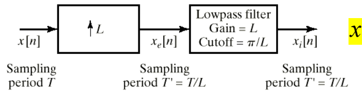
$$x_i[n] = \sum_{k=-\infty}^{\infty} x[k] \frac{\sin[\pi(n-kL)/L]}{[\pi(n-kL)/L]} \quad \text{or since}$$

$$x_c(t) = \sum_{k=-\infty}^{\infty} x[k] \frac{\sin[\pi(t-kT)/T]}{[\pi(t-kT)/T]} \quad \longleftarrow \text{Sampling Theorem}$$

$$x_i[n] = \sum_{k=-\infty}^{\infty} x[k] \frac{\sin[\pi(nT/L-kT)/T]}{[\pi(nT/L-kT)/T]} = x_c(nT/L)$$

Interpolation - II

$x_c(nT)$



$x_c(nT/L)$

$$X_e(e^{j\omega}) = \sum_{n=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} x[k]\delta[n-kL]e^{-j\omega n}$$

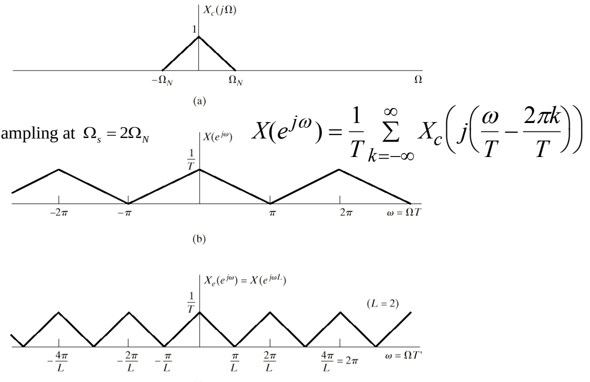
$$= \sum_{k=-\infty}^{\infty} x[k]e^{-j\omega Lk} = X(e^{j\omega L})$$

$$X_e(e^{j\Omega T/L}) = X(e^{j\Omega T}) = \frac{1}{T}\sum_{k=-\infty}^{\infty} X_c\left(j\left(\Omega - \frac{2\pi k}{T}\right)\right)$$

$$X_i(e^{j\Omega T/L}) = H_i(e^{j\Omega T/L})X_e(e^{j\Omega T/L})$$

$$= \frac{1}{(T/L)}\sum_{k=-\infty}^{\infty} X_c\left(j\left(\Omega - \frac{2\pi k}{T/L}\right)\right)$$

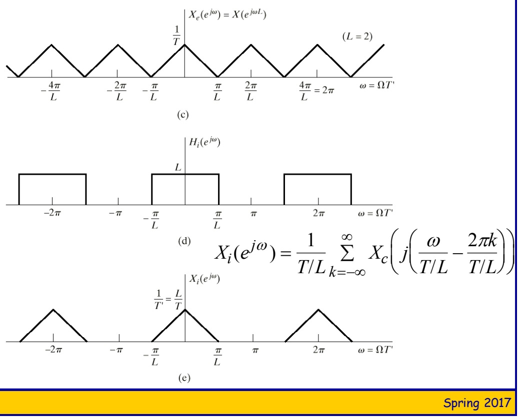
Interpolation - III



Sampling at $\Omega_s = 2\Omega_N$

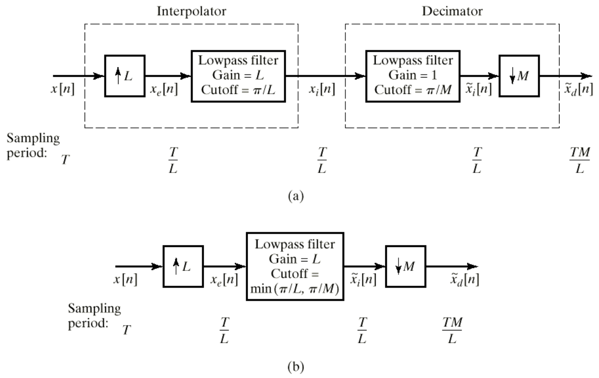
$$X(e^{j\omega}) = \frac{1}{T}\sum_{k=-\infty}^{\infty} X_c\left(j\left(\frac{\omega}{T} - \frac{2\pi k}{T}\right)\right)$$

Interpolation - IV



$$X_i(e^{j\omega}) = \frac{1}{T/L}\sum_{k=-\infty}^{\infty} X_c\left(j\left(\frac{\omega}{T/L} - \frac{2\pi k}{T/L}\right)\right)$$

Non-Integer Rate Change - I


Interpolator

Decimator

$x[n]$

$\uparrow L$

$x_e[n]$

Lowpass filter
Gain = L
Cutoff = π/L

$x_i[n]$

Lowpass filter
Gain = 1
Cutoff = π/M

$\tilde{x}_i[n]$

$\downarrow M$

$\tilde{x}_d[n]$

Sampling period: T $\frac{T}{L}$ $\frac{T}{L}$ $\frac{T}{L}$ $\frac{TM}{L}$

(a)

$x[n]$

$\uparrow L$

$x_e[n]$

Lowpass filter
Gain = L
Cutoff = $\min(\pi/L, \pi/M)$

$\tilde{x}_i[n]$

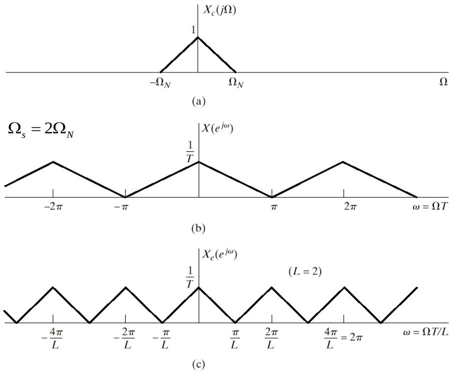
$\downarrow M$

$\tilde{x}_d[n]$

Sampling period: T $\frac{T}{L}$ $\frac{T}{L}$ $\frac{TM}{L}$

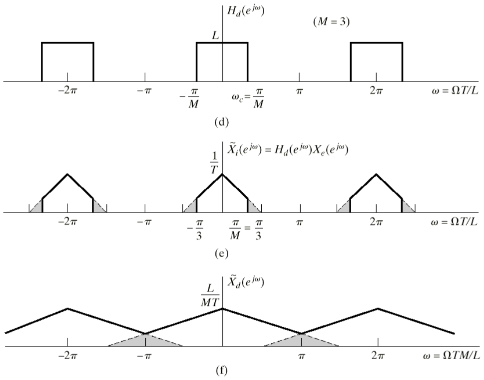
(b)

Non-Integer Rate Change - II



Sampling at $\Omega_s = 2\Omega_N$

Non-Integer Rate Change - III

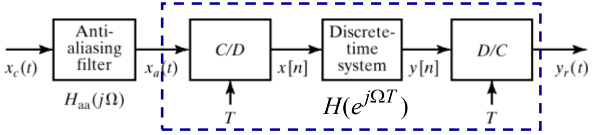


(d)

(e)

(f)

Anti-Alias Pre-filtering



$$Y_r(j\Omega) = H(e^{j\Omega T})X_a(j\Omega) \text{ if } X_a(j\Omega) = 0 \quad \text{for } |\Omega| \geq \Omega_N$$

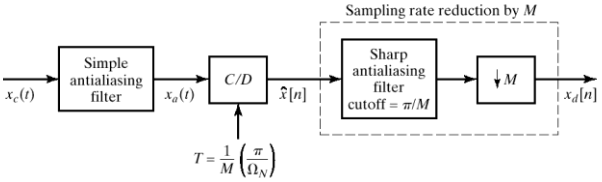
- What is the overall effective frequency response?

$$X_a(j\Omega) = H_{aa}(j\Omega)X_c(j\Omega)$$

$$\Rightarrow \quad X_a(j\Omega) = 0 \quad \text{for } |\Omega| \geq \Omega_N$$

$$Y_r(j\Omega) = \underbrace{H(e^{j\Omega T})H_{aa}(j\Omega)}_{H_{\text{eff}}(j\Omega)}X_c(j\Omega)$$

Oversampling Eases Filtering - I

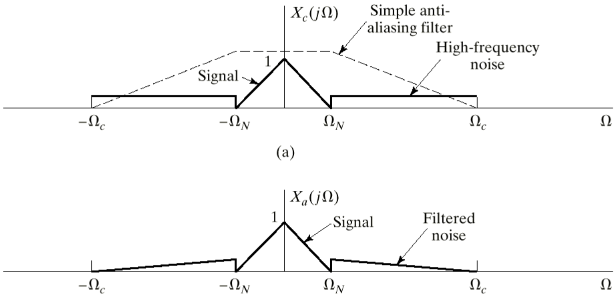


$$X_a(j\Omega) = H_{aa}(j\Omega)X_c(j\Omega)$$

$$\text{Choose } \; H_{aa}(j\Omega) = 0 \quad \text{for } |\Omega| \geq M\Omega_N$$

$$\Rightarrow \; X_a(j\Omega) = 0 \quad \text{for } |\Omega| \geq M\Omega_N$$

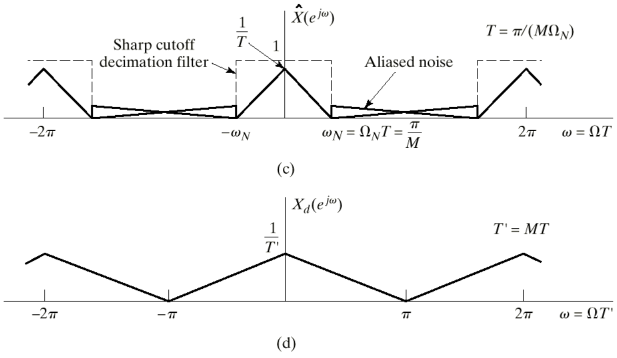
Oversampling Eases Filtering - II



(a)

(b)

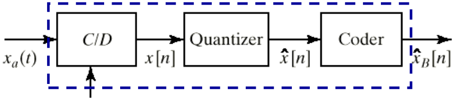
Oversampling Eases Filtering - III



Digital Processing of Analog Signals

- Practical considerations in implementations:
  - The input signal cannot be perfectly bandlimited
  - A-to-D and D-to-A converters have finite-precision output and input respectively
  - Only finite-precision arithmetic is available for computations

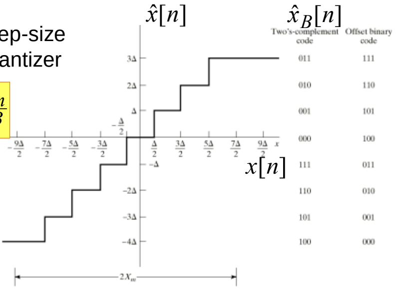
Representation of A-to-D Converter



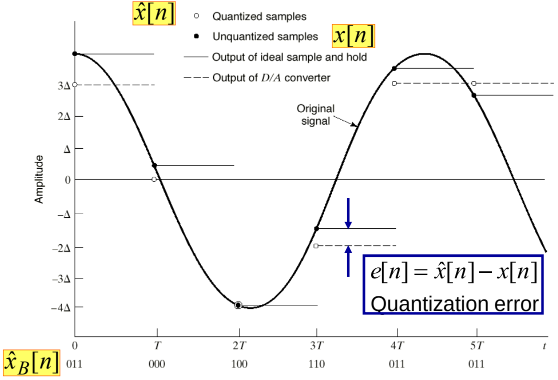
T

Quantization step-size for (*B*+1)-bit quantizer

$$\Delta = \frac{2X_m}{2^{B+1}} = \frac{X_m}{2^B}$$



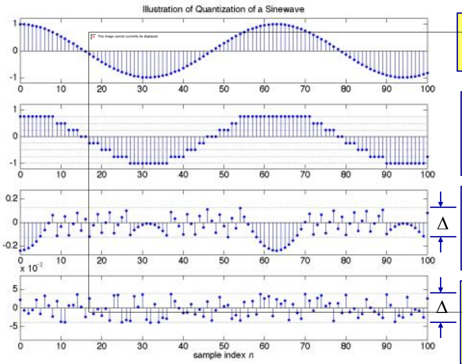
A-to-D and D-to-A Conversion



$e[n] = \hat{x}[n] - x[n]$

Quantization error

Quantization of a Sine Wave



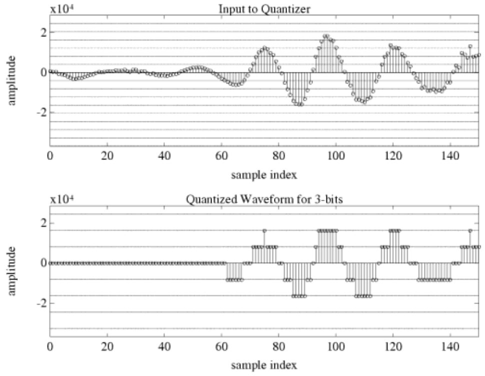
Unquantized sinewave

3-bit quantization waveform

3-bit quantization error

8-bit quantization error

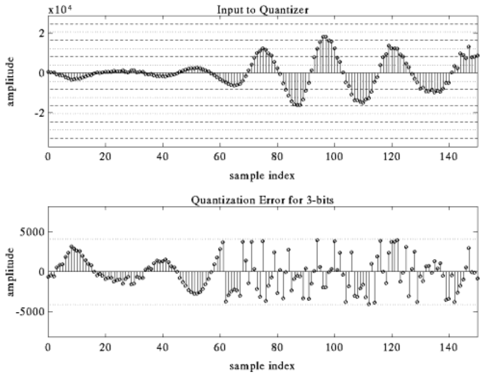
3-Bit Speech Quantization



Input to quantizer

3-bit quantization waveform

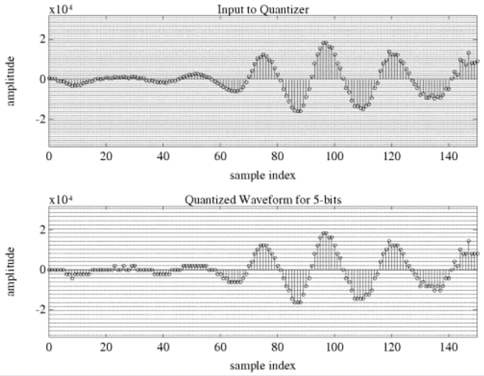
3-Bit Speech Quantization Error



Input to quantizer

3-bit quantization error

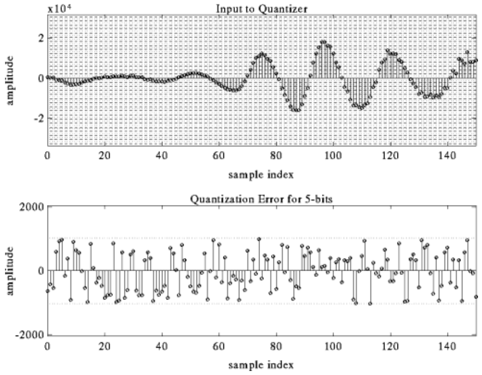
5-Bit Speech Quantization



Input to quantizer

5-bit quantization waveform

5-Bit Speech Quantization Error



Input to quantizer

5-bit quantization error

Quantization Error

- Each sample is quantized and each sample has a quantization error defined as

$$e[n] = \hat{x}[n] - x[n]$$

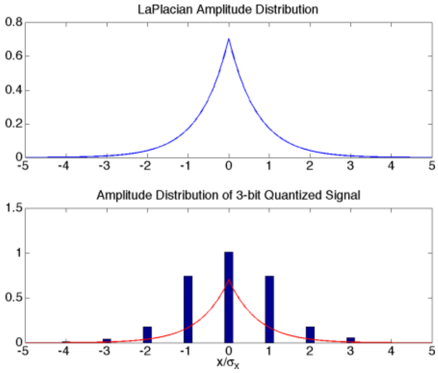
- Since each sample falls in an interval of length Δ, and the quantized sample falls in the middle of that interval,

$$-(\Delta/2) < e[n] \le (\Delta/2).$$

- We call this “quantization noise” because it seems to vary randomly. Clearly, the strength (power) of this noise is proportional to Δ; i.e.,

$$\sigma_e^2 = K\Delta^2$$

Typical Amplitude Distributions



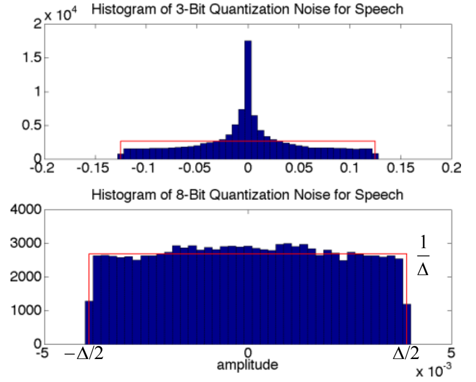
Laplacian distribution is often used as a model for speech signals

A 3-bit quantized signal has only 8 different values

Probabilistic Model for Quantization

- We observed that the quantization error has very complicated variations that suggest a random or noise-like character.
- Random signals are represented by probability distributions and averages such as
  - Mean and mean-square (average power)
  - Histograms
  - Autocorrelation function
  - Power spectrum
- This is a good way to think about quantization noise.

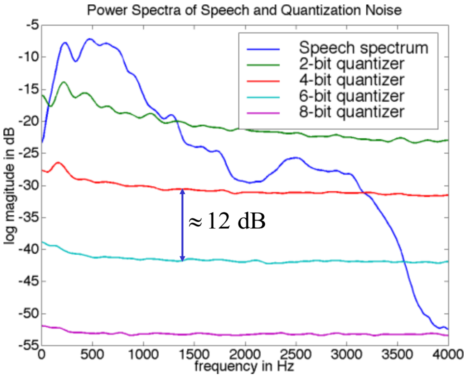
Histograms of Quantization Noise



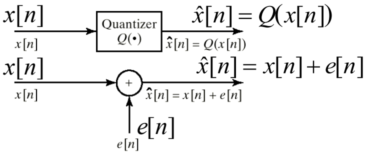
3-bit quantization histogram

8-bit quantization histogram

Spectra of Quantization Noise



Linear Noise Model



- Error is uncorrelated with the input.
- Error is uniformly distributed over the interval

$$-(\Delta/2) < e[n] \le (\Delta/2).$$

- Error is stationary white noise, (i.e. flat spectrum)

$$P_e(\omega) = \sigma_e^2 = \int_{-\Delta/2}^{\Delta/2} \frac{1}{\Delta} e^2 de = \frac{\Delta^2}{12}, \qquad |\omega| \le \pi$$

Quantizer Signal-to-Noise Ratio

- Assume $2^{(B+1)}$ levels and amplitude range $2X_m$. Then using a probabilistic analysis we obtain

$$\Rightarrow \underbrace{\Delta = \frac{2X_m}{2^{(B+1)}}}_{\text{step size}} = 2^{-B}X_m \qquad \Rightarrow \qquad \underbrace{\sigma_e^2 = \frac{2^{-2B}X_m^2}{12}}_{\text{noise power}}$$

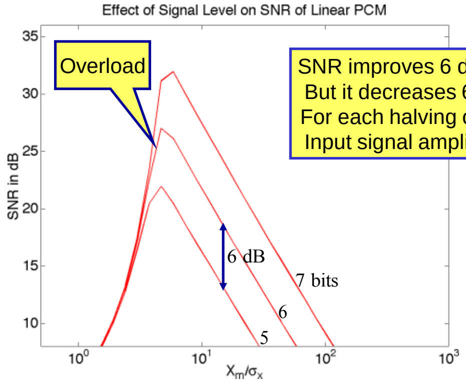
- Therefore the quantizer SNR is:

$$\text{SNR} = 10\log_{10}\left(\frac{\sigma_x^2}{\sigma_e^2}\right) = 10\log_{10}\left(\frac{12\cdot\boxed{2^{2B}}\sigma_x^2}{X_m^2}\right)$$

$$= \boxed{6.02B} + 10.8 - 20\log_{10}\left(\frac{X_m}{\sigma_x}\right)$$

(in dB)

Variation of SNR with Signal Level



Overload

SNR improves 6 dB/bit,
But it decreases 6 dB
For each halving of the
Input signal amplitude