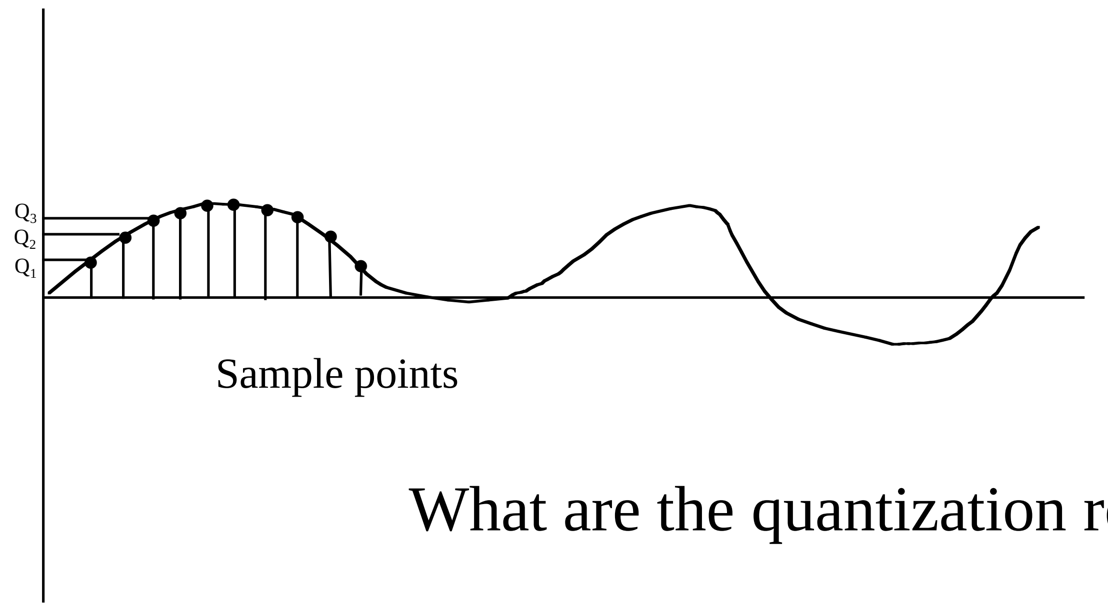
Lecture 4: Quantization

Eytan Modiano
AA Dept.

Sampling

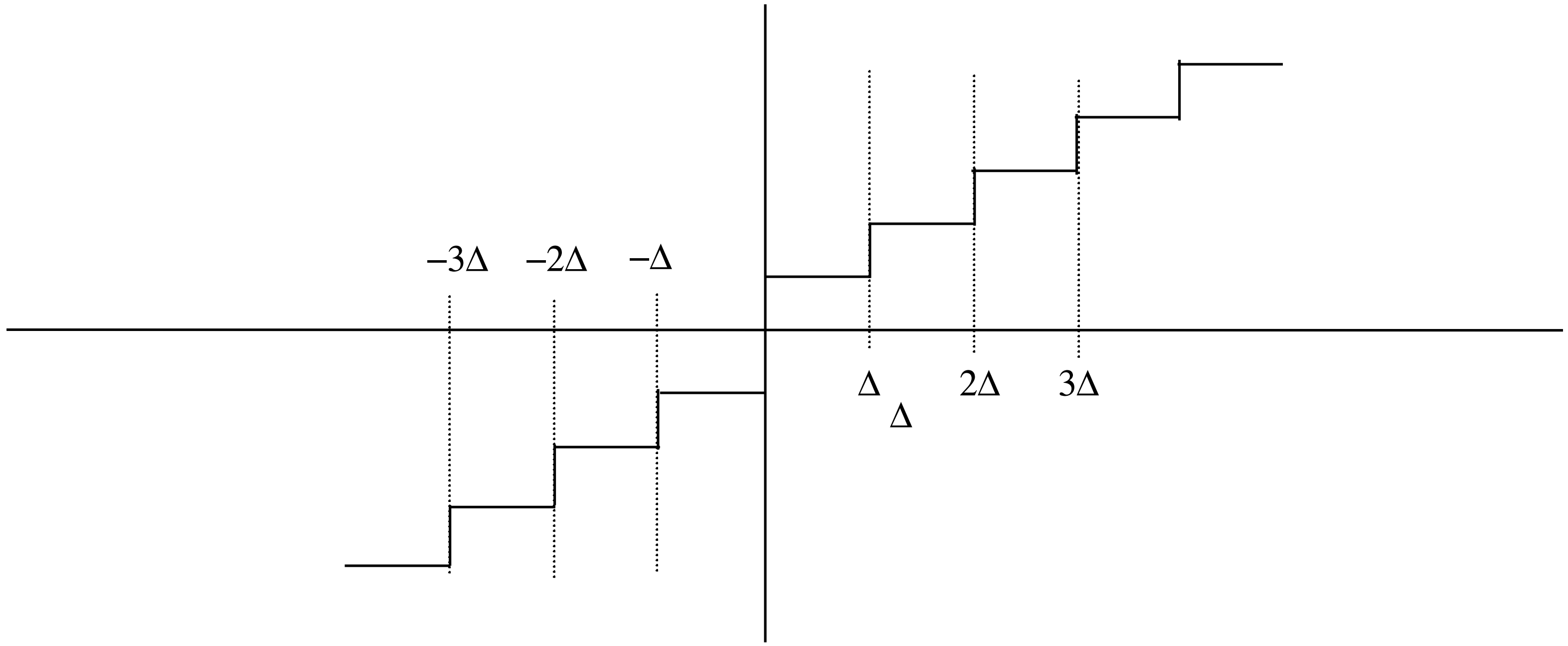
- **Sampling provides a discrete-time representation of a continuous waveform**
  - **Sample points are real-valued numbers**
  - **In order to transmit over a discrete valued system we must first convert into discrete valued numbers**



What are the quantization regions

What are the quantization levels

Uniform Quantizer



- **All quantization regions are of equal size (Δ)**
  - **Except first and last regions if samples are not finite valued**

- **With N quantization regions, use $\log_2(N)$ bits to represent each quantized value**

Quantization Error

$e(x) = Q(x) - x$

Squared error: $D = E[e(x)^2] = E[(Q(x)-x)^2]$

SQNR: $E[X^2]/E[(Q(x)-x)^2]$

Example

- **X is uniformly distributed between -A and A**
  - **f(x) = 1/2A, -A<=x<=A and 0 otherwise**
- **Uniform quantizer with N levels => Δ = 2A/N**
  - **Q(x) = quantization level = midpoint of quantization region in which x lies**
- **D = E[e(x)²] is the same for quantization regions**

$$D = E[e(x)^2 \mid x \in R_i] = \int_{-\Delta/2}^{\Delta/2} x^2 f(x)dx = \frac{1}{\Delta}\int_{-\Delta/2}^{\Delta/2} x^2 dx = \frac{\Delta^2}{12}$$

$$E[X] = \frac{1}{2A}\int_{-A}^{A} x^2 dx = \frac{A^2}{3}$$

$$SQNR = \frac{A^2/3}{\Delta^2/12} = \frac{A^2/3}{(2A/N)^2/12} = N^2, (\Delta = 2A/N)$$

Quantizer design

- **Uniform quantizer is good when input is uniformly distributed**
- **When input is not uniformly distributed**
  - **Non-uniform quantization regions**
    - **Finer regions around more likely values**
  - **Optimal quantization values not necessarily the region midpoints**
- **Approaches**
  - **Use uniform quantizer anyway**
    - **Optimal choice of Δ**
  - **Use non-uniform quantizer**
    - **Choice of quantization regions and values**
  - **Transform signal into one that looks uniform and use uniform quantizer**

Optimal uniform quantizer

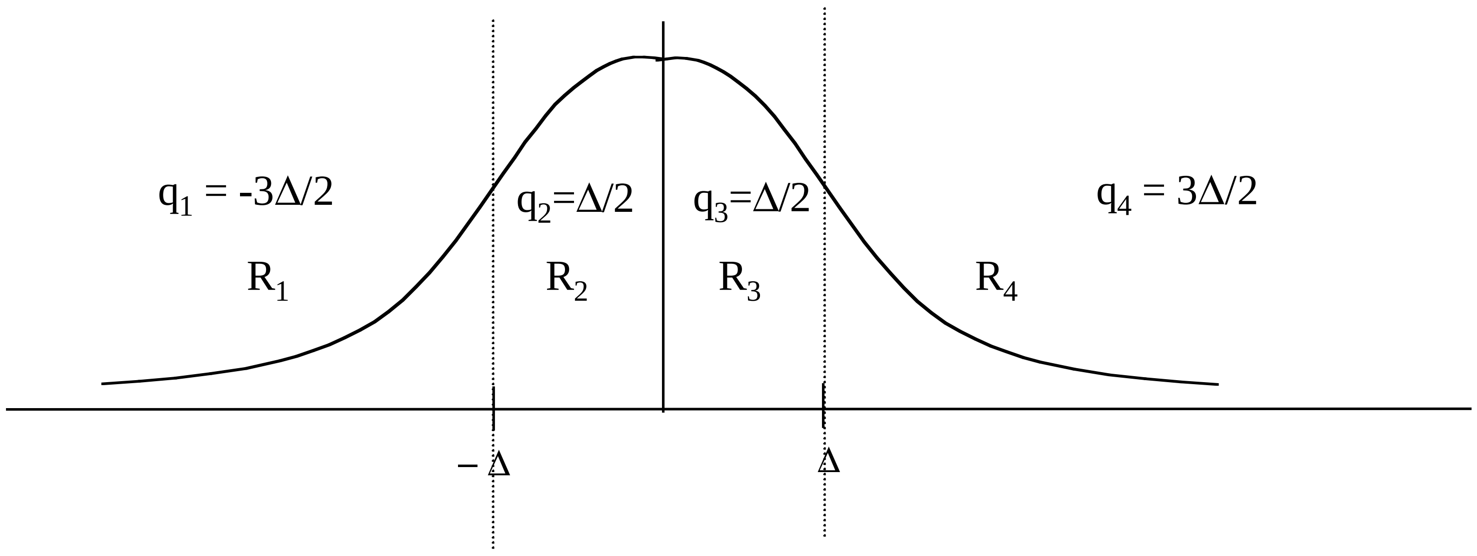
- **Given the number of regions, N**
  - **Find the optimal value of Δ**
  - **Find the optimal quantization values within each region**
  - **Optimization over N+2 variables**

- **Simplification: Let quantization levels be the midpoint of the quantization regions (except first and last regions, when input not finite valued)**

- **Solve for Δ to minimize distortion**
  - **Solution depends on input pdf and can be done numerically for commonly used pdfs (e.g., Gaussian pdf, table 6.2, p. 296 of text)**

Uniform quantizer example

- **N=4, X~N(0,1)** $$f_x(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-x^2/2\sigma^2}, \quad \sigma^2 = 1$$
- **From table 6.2, Δ=0.9957, D=0.1188, H(Q)= 1.904**
  - **Notice that H(Q) = the entropy of the quantized source is < 2**
  - **Two bits can be used to represent 4 quantization levels**
  - **Soon we will learn that you only need H(Q) bits**



Non-uniform quantizer

- Quantization regions need not be of same length
- Quantization levels need not be at midpoints
- Complex optimization over 2N variables

- Approach:
  - Given quantization regions, what should the quantization levels be?
  - What should the quantization regions be?

- Solve for quantization levels first (given region (a_{i-1}, a_i))
  - Minimize distortion

Optimal quantization levels

- **Minimize distortion, D**
  - **Optimal value affects distortion only within its region**

$$D_R = \int_{a_{i-1}}^{a_i} (x - \hat{x}_i)\ f_x(x)dx$$

$$\frac{dD_R}{d\hat{x}_i} = \int_{a_{i-1}}^{a_i} 2(x - \hat{x}_i)^2 f_x(x)dx = 0$$

$$\hat{x}_i = \int_{a_{i-1}}^{a_i} x f_x(x \mid a_{i-1} \le x \le a_i)dx$$

  –

$$\hat{x}_i = E[X \mid a_{i-1} \le x \le a_i]$$

- **Quantization values should be the “centroid” of their regions**
  - **The conditional expected value of that region**

- **Approach can be used to find optimal quantization values for the uniform quantizer as well**

Optimal quantization regions

- **Take derivative of D with respect to a_i**
  - **Take derivative with respect to integral boundaries**

$$\frac{dD}{da_i} = f_x(a_i)[(a_i - \hat{x}_i)^2 - (a_i - \hat{x}_{i+1})^2] = 0$$

$$a_i = \frac{\hat{x}_i + \hat{x}_{i+1}}{2}$$

  - **Boundaries of the quantization regions are the midpoint of the quantization values**
- **Optimality conditions:**
  1. **Quantization values are the "centroid" of their region**
  2. **Boundaries of the quantization regions are the midpoint of the quantization values**
  3. **Clearly 1 depends on 2 and visa-versa. The two can be solved iteratively to obtain optimal quantizer**

Finding the optimal quantizer

- **Start with arbitrary regions (e.g., uniform Δ)**

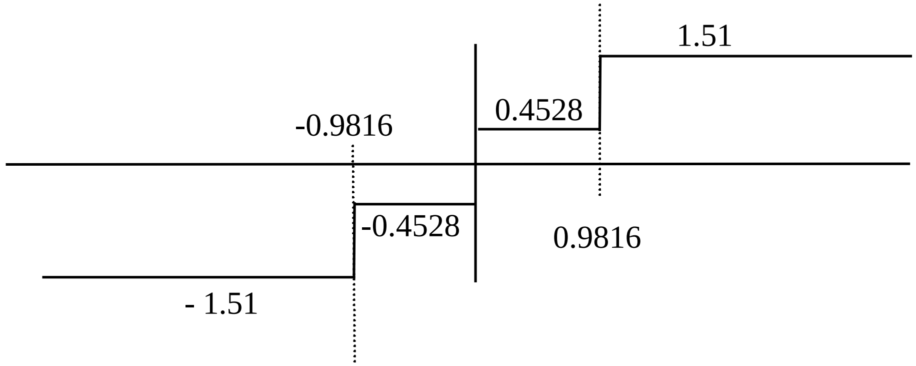
  A) Find optimal quantization values ("centroids")

  B) Use quantization values to get new regions ("midpoints")

  – **Repeat A & B until convergence is achieved**

- **Can be done numerically for known distributions**
  – **Table 6.3 (p. 299) gives optimal quantizer for Gaussian source**

- **E.g., N=4,**
  – **D = 0.1175, H(x) = 1.911**
  – **Recall: uniform quantizer, D= 0.1188, H(x) = 1.904 (slight improvement)**



Companders

- **Non-uniform quantizer can be difficult to design**
  - **Requires knowledge of source statistics**
  - **Different quantizers for different input types**

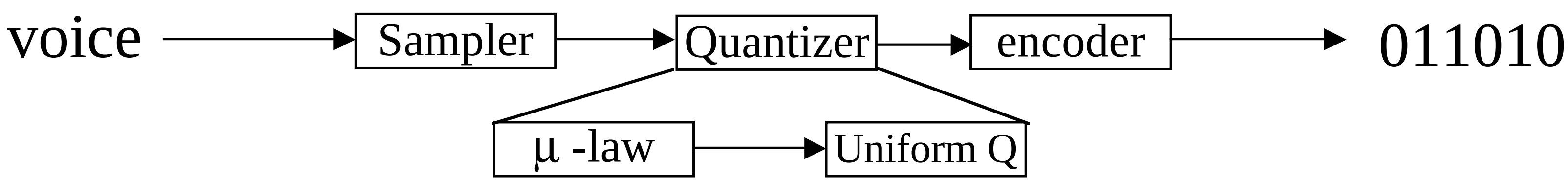
- **Solution: Transfer input signal into one that looks uniform and then use uniform quantizer**

- **μ-law compander**

$$g(x) = \frac{Log(1+\mu|x|)}{Log(1+\mu)}\operatorname{sgn}(x)$$

  - **μ controls the level of compression**
  - **μ = 255 typically used for voice**

Pulse code modulation



- **Uniform PCM: x(t) ∈ [Xmin, Xmax]**
  - **N = 2^V quantization levels, each level encoded using v bits**
  - **SQNR: same as uniform quantizer**

$$SQNR = \frac{E[X^2] \times 3 \times 4^v}{X_{MAX}^2}$$

  - **Notice that increasing the number of bits by 1 decreases SQNR by a factor of 4 (6 dB)**

Speech coding

- PCM with $\mu = 255$
- Uniform quantizer with 128 levels, $N = 2^7$, 7 bits per sample
- Speech typically limited to 4KHZ
  - Sample at 8KHZ => Ts = 1/8000 = 125 μs

    8000 samples per second at 7 bits per sample => 56 Kbps

- Differential PCM
  - Speech samples are typically correlated
  - Instead of coding samples independently, code the difference between samples
  - Result: improved performance, lower bit rate speech