EECS4214

Digital Communications

Sunila Akbar

Department of Electrical Engineering and Computer Science
York University

January 29, 2025
Week 4

Week 4 - Lecture 8

1 Review of the Last Lecture

2 Quantization

Review of the Last Lecture

Review

- Sampling

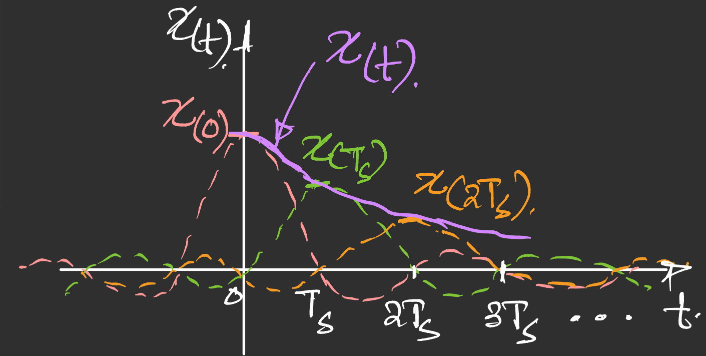
  $f_s \geq 2W$ W is the message signal's BW

- We can reconstruct the original sig. by LPF with filter gain of T_s and cutoff freq. of W.

- We have seen the reconstruction in time-domain (summation of sinc WFs)

Will review in the lecture!

$$\boxed{\tilde{x}(t) = \frac{1}{T_s}\sum_{k=-\infty}^{\infty} x(kT_s)\,\text{sinc}\left(\frac{t-kT_s}{T_s}\right)}$$



The signal at sampling instants $t = nT_s$.

$$x(nT_s) = \frac{1}{T_s}\sum_{k=-\infty}^{\infty} x(kT_s)\,\text{sinc}\left(\frac{nT_s - kT_s}{T_s}\right)$$

$$= \frac{1}{T_s}\sum_{k=-\infty}^{\infty} x(kT_s)\,\underbrace{\text{sinc}(n-k)}$$

$= 1$ for $k = n$

$= 0$ for $k \neq n$.

$$= \frac{1}{T_s}\, x(nT_s).$$

Can compensate in LPF.

$$H_{\text{LPF}}(f) = \begin{cases} 1, & -W \le f \le W \\ 0, & \text{otherwise} \end{cases}$$

T_s.

Remember:

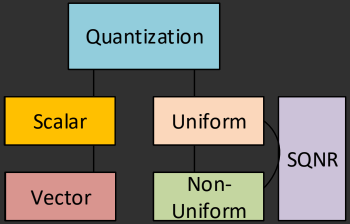
$$\text{sinc}\; nx = \begin{cases} 1, & n = 0 \\ 0, & n = \pm 1, \pm 2, \pm 3, \dots \end{cases}$$

For other t's its the superposition of all sincs & we get $x(t)$!

Quantization

Quantization

The process of transforming sampled amplitude value/s of a message signal into a discrete amplitude value is referred to as Quantization

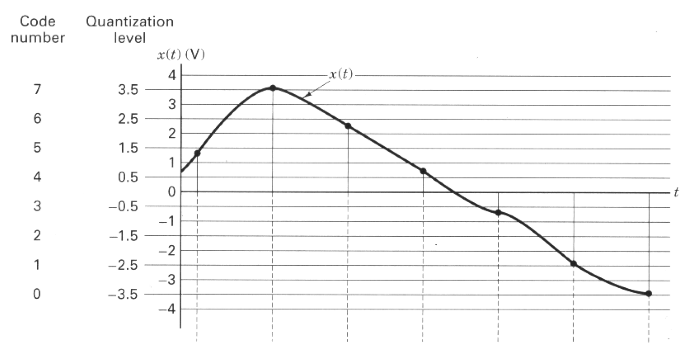


Quantization Types

- **Scalar Quantization:** A single sample (scalar value) is mapped to the closest value from a finite set of discrete levels
- **Vector Quantization:** A group of samples (a vector) is mapped to the closest vector from a finite set of representative vectors (codebook)

Scalar Quantization

A single sample (scalar value) is mapped to the closest value from a finite set of discrete levels



CV →
DV →

Natural sample value	1.3	3.6	2.3	0.7	−0.7	−2.4	−3.4
Quantized sample value	1.5	3.5	2.5	0.5	−0.5	−2.5	−3.5
Code number	5	7	6	4	3	1	0
PCM sequence	101	111	110	100	011	001	000

the closest value

Scalar Quantization

Example: Landline Telephony

- Message signal bandwidth = 4kHz
- Sampling frequency = 8 kHz
- Levels = 256 (8 bits per sample)
- Bit rate = ?

samples/sec

8 bits/sample

$f_s \geq 2(4\,\text{kHz}) = \underbrace{8\,\text{kHz}}$.

Quantized to 256 levels. $= 2^{8}$

$R_b = 8000$ samples/sec $\times$ 8 bits/sample $= 64$ kbits/sec.

TYPICALLY 56 kbps 8 kbps signalling

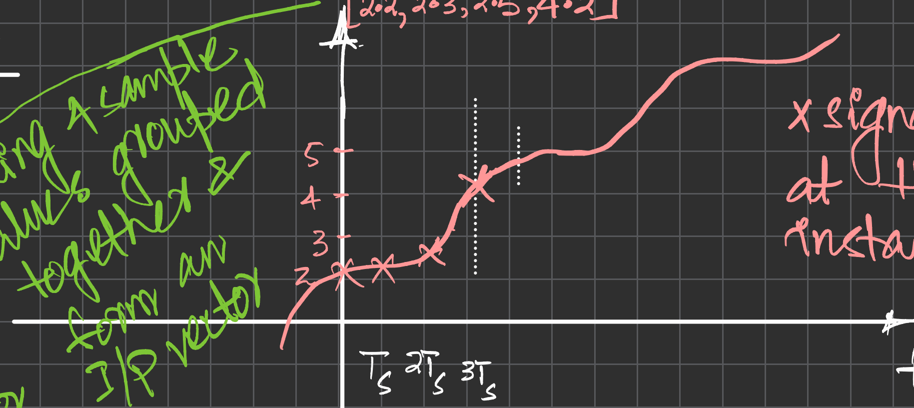
Vector Quantization

- In Vector Quantization (VQ), each input vector consists of multiple sample values grouped together into a vector
- Instead of quantizing individual scalar values (as in scalar quantization), VQ treats a group of values (a vector) as a single entity and maps it to a representative codebook vector
- **For example**, in image compression, instead of quantizing each pixel separately, a block of pixels (e.g., a 4×4 block) can be treated as a vector and mapped to a representative codeword
- This helps in preserving structural patterns and reducing redundancy more effectively than scalar quantization

Vector Quantization

- Assign a codeword depending on to which codevector this I/P vector is the nearest!

Taking 4 sample values grouped together & form an I/P vector



x signal's value at the sampling instants (kT_s)

- If we have 'k' codevectors.

$\Rightarrow k = 2^c$

'c' bits/codeword.

✓ Based on the I/p vector (4-dimensional space here), a codeword is nearest to the code book vector

✓ The number of code vectors are limited (k)

say $8 = 2^3$

3 bits/code vector

Vector Quantization

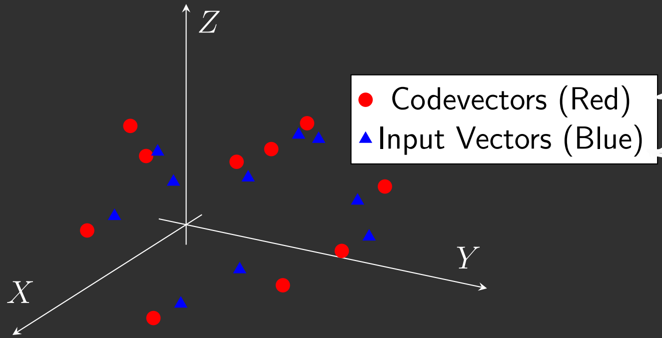
The **key operation** in vector quantization is the quantization of a *random vector* by encoding it as a *binary codeword*

Each input vector can be viewed as a point in an *n-dimensional space*

- Let k be the number of quantization regions in this n-dimensional space
- Each region is associated with a representative *codevector*
- The total number of codevectors in the *codebook* is k
- The dimensionality of each codevector (i.e., the number of samples in a vector) is n

Vector Quantization

The **vector quantizer** works by *partitioning the space into a set of non-overlapping n-dimensional regions*. To encode an input vector, it is compared against the stored reference vectors in the *codebook*. The closest matching *codevector* (typically based on a distance metric like *Euclidean distance*) is selected, and its corresponding *binary codeword* is used to represent the input vector



Limited (k)

Random can take any value

Vector Quantization

Example: NA-TDMA

- Message signal bandwidth = 4kHz
- Sampling frequency = 8 kHz
- Levels = 8192 (13 bits per sample) $= 2^{13}$
- Bit rate = ? = 8 ksamples/sec × 13 bits/sample = 104 K bits/sec

13 bits/sample

If Scalar Quantiz.

Vector quantization is a lossy compression technique used in speech and image decoding

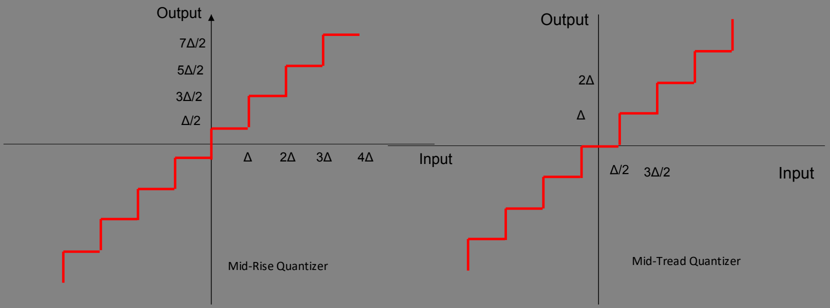
With Vector Quantization 8 Kbits/sec

via compression →

Quantization Types

- **Uniform Quantization:** When the signal amplitude is discretized uniformly, i.e., the difference between any two quantization levels is same
- **Non-Uniform Quantization:** When the signal amplitude is not discretized uniformly, i.e., the difference between any two quantization levels is not the same

Uniform Quantization



Input-Output Characteristics of Uniform Quantizers

- Mid-Rise: Even number of quantization levels
- Mid-Tread: Odd number of quantization levels

N-Level Uniform Quantizer Analysis

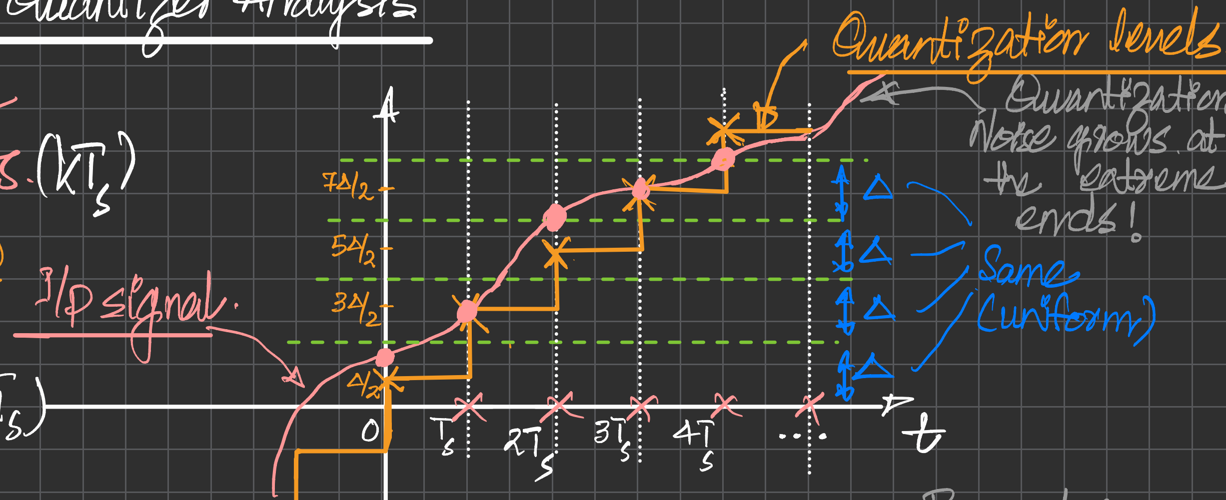
- The mid-point of each interval is selected as the quantized output of each interval. At this point the quantization error is zero
- Quantization error varies from $-\Delta/2$ and $+\Delta/2$
- Quantization error continues to grow at the extreme ends

N-level Uniform Quantizer Analysis

- I/p values at the sampling instants. (kT_s)

X. Quantized value of the I/p sig. at the sampling instants (kT_s)

x_q ↓ can only take odd integral multiples of $\Delta/2$!



- sampled I/p signal at kT_s $(k \in \mathbb{Z})$

Remember.

- I/p is a random variable "X"
- & so is the quantized value "$\hat{X}_q$"

$$\text{Quantization Error} = x_{(kT_s)} - x_q$$

$$\text{Max}^m \text{ Quantization Error} = \Delta/2 = X - \hat{X}_q$$

Quantization Error Varies from $-\Delta/2$ to $\Delta/2$.

Signal-to-Quantization Noise Ratio (SQNR)

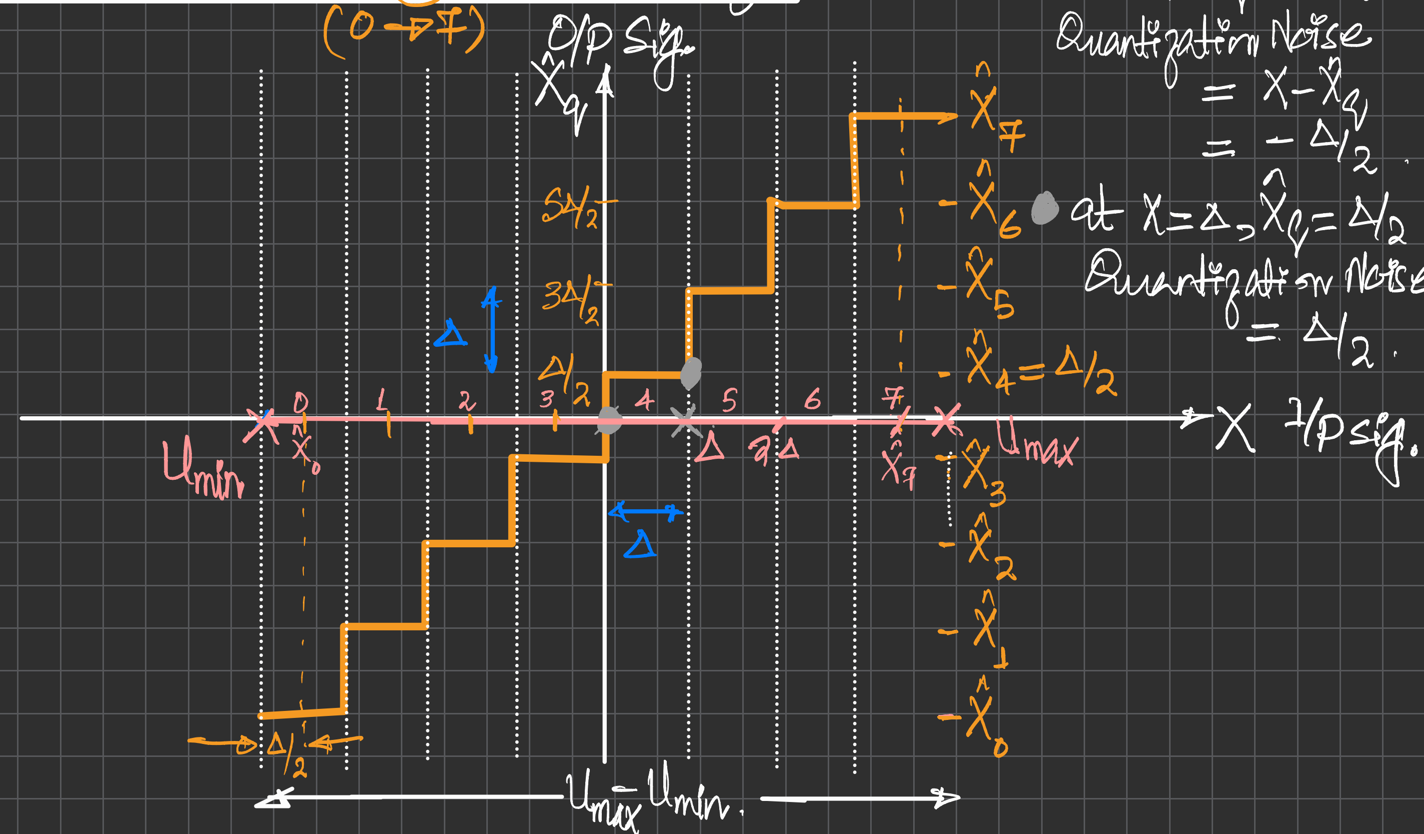
- X = Input signal is a random variable and follows a PDF of $f_X(x)$
- $\hat{X}_q$ = Quantized output signal
- $n_q = X - \hat{X}_q$ = Quantization noise error

$$\boxed{\mathrm{SQNR} = \frac{\mathbb{E}[X^2]}{\mathbb{E}[n_q^2]}}$$

- $\mathbb{E}[X^2] = \int_{-\infty}^{\infty} x^2 f_X(x)dx$
- $\mathbb{E}[n_q^2] = \int_{-\infty}^{\infty} (X - \hat{X}_q)^2 f_X(x)dx = \frac{\Delta^2}{12}$

Transfer Function of an 8-level Quantizer

(0 → 7)



- at $X = 0$, $\hat{X}_q = \Delta/2$.
  Quantization Noise $= X - \hat{X}_q$
  $= -\Delta/2$.
- at $X = \Delta$, $\hat{X}_q = \Delta/2$.
  Quantization Noise
  $= \Delta/2$.

*. The midpoint of each interval is selected as the quantized O/p of each interval.

For finding SQNR, lets find $E[n_q^2]$ first:

0 to (8-1) = 0 to 7.
in the 8-Level Quantizer.

$$E[n_q^2] = \int_{-\infty}^{\infty} (x - \hat{x}_q)^2 f_{X}(x)\, dx$$

$$= \int_{-\infty}^{\hat{x}_0 - \Delta/2} (x - \hat{x}_0)^2 f_{X}(x)\, dx + \sum_{i=0}^{N-1} \int_{\hat{x}_i - \frac{\Delta}{2}}^{\hat{x}_i + \Delta/2} (x - \hat{x}_i)^2 f_{X}(x)\, dx$$

$$+ \int_{\hat{x}_{N-1} + \Delta/2}^{\infty} (x - \hat{x}_{N-1})^2 f_{X}(x)\, dx$$

Consider the simplifying assumptions as in the next slide!

We will continue from here!

Thank You

Happy Learning

☺