# EE6424 Digital Audio Signal Processing
# Part 2
# Lecture 3:
# Sampling and Quantization of Speech

# Outline of lecture

- Digitization (Analog to digital)
  - Sampling (Continuous time $\rightarrow$ Discrete time)
  - Quantization (Continuous amplitude $\rightarrow$ Discrete amplitude)
  - Coding (Discrete amplitude $\rightarrow$ binary digit)
- Scalar Quantization
  - Mechanism of scalar quantization
  - Quantization noise (SQNR)
  - Companding
- Vector Quantization
  - The LBG algorithm

EE6424 Part 3: Lecture 3.1

# DIGITIZATION

# Digital speech

- A speech signal, in the form of an **acoustic** sound pressure wave, can be changed into a **processable** object by converting it into an **electrical** signal using a microphone.
- The electrical signal is usually transformed from the **analog** into a **digital** signal prior to almost all speech processing.
- Speech coding, speech enhancement, speech and speaker recognition, among others, involve highly **sophisticated algorithms** which cannot otherwise be **realized** using analog techniques.
- Analog-to-digital (**A/D**) conversion, commonly referred to as the **digitization**, consists of three processes
  - **Sampling** (continuous time → discrete time)

    Sampling is the process of converting a **continuous** time signal as a **periodic sequence** of values.

– **Quantization** (continuous amplitude → finite set of discrete values)

  Quantization involves representing the sampled values by one from a finite set of **values**.

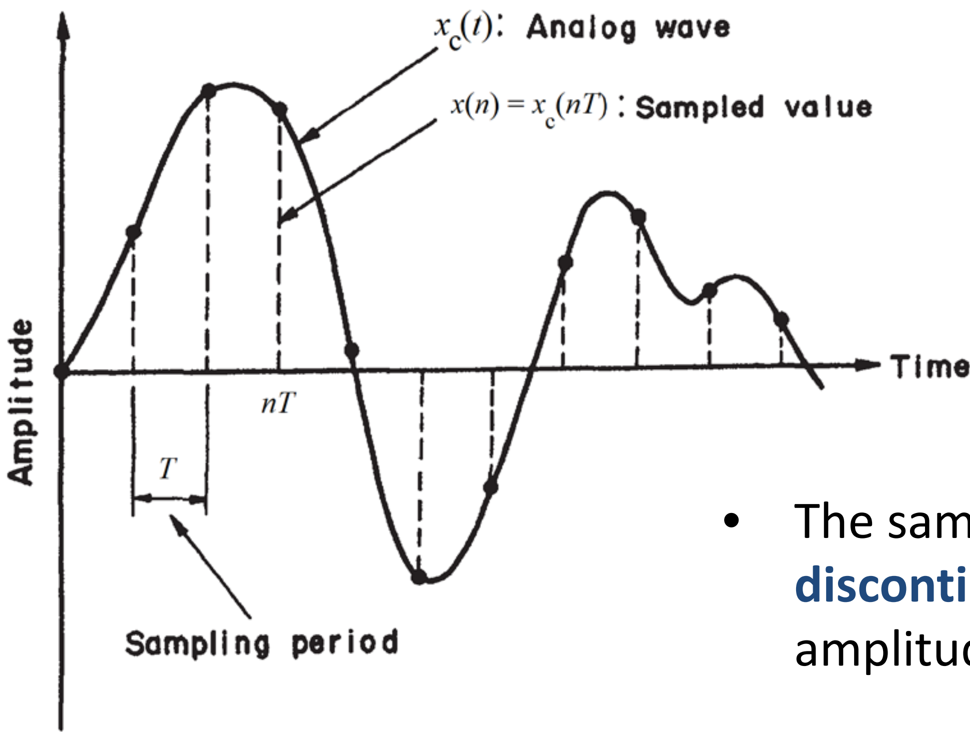
– **Coding** (finite set of binary codes)

  Coding is concerned with the assignment of **binary codes** to each value in the finite set.

- The digitization process thus converts a **continuous-time continuous amplitude** speech signal into a **sequence** of **binary codes** (or **bit stream**).

# Sampling

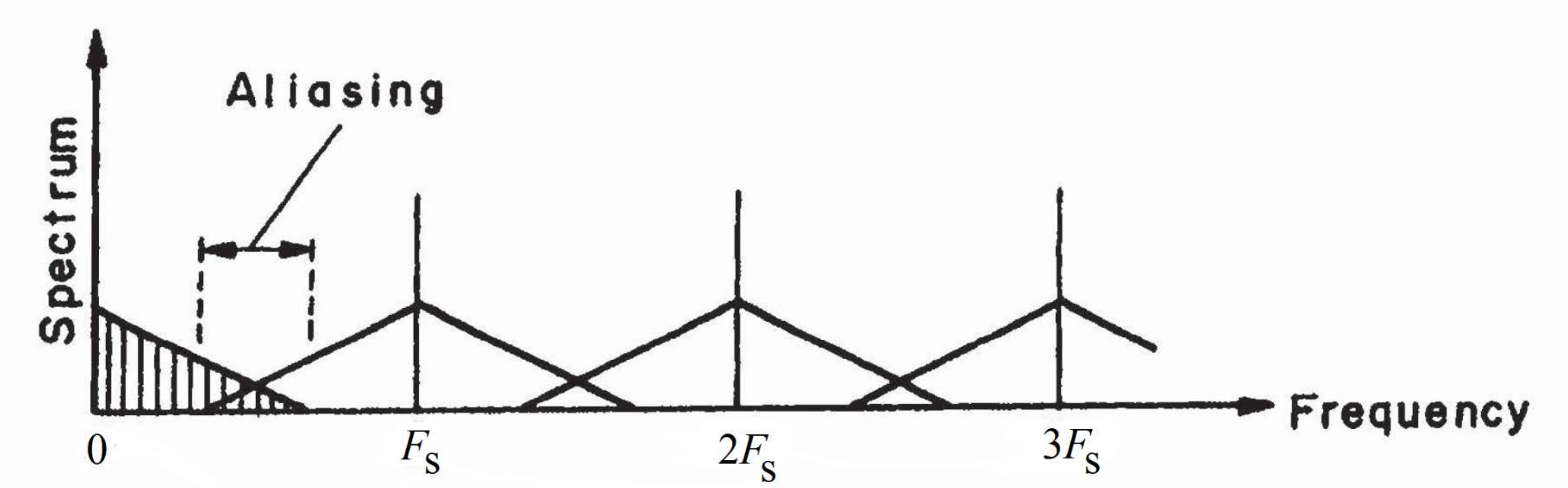
- Given an analog input $x_c(t)$, the sampler produces a number $x(n) = x_c(nT)$ at a periodic time $nT$, where $n$ is an integer (the discrete time index).




- $T$ (seconds) is called the **sampling period**, while $F_s = 1/T$ (Hz) is referred to as the **sampling frequency**.

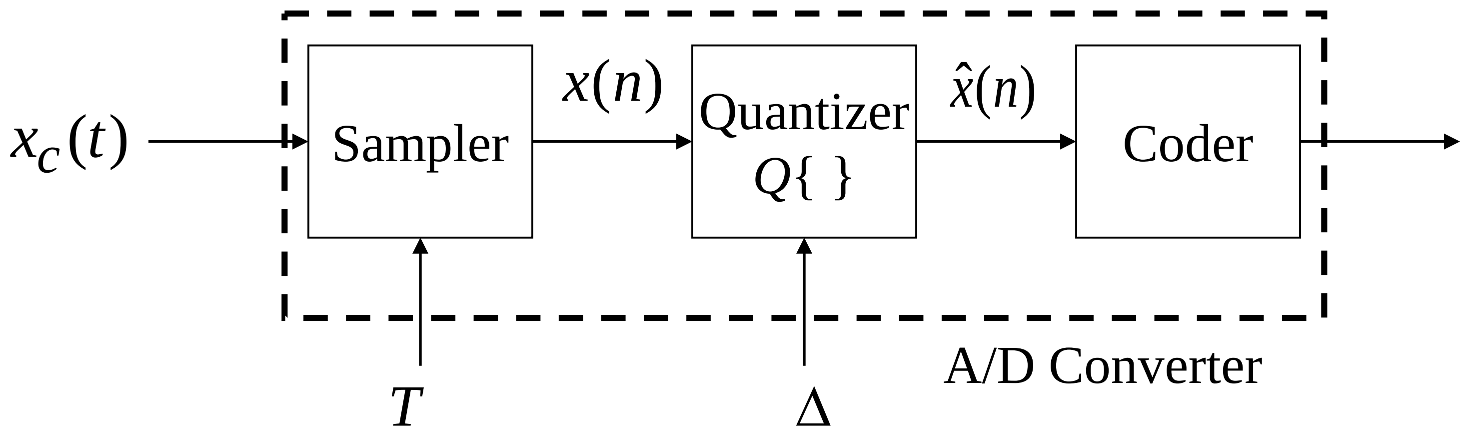
- The sampled signal $x(n)$, which is **discontinuous** in time but **continuous** in amplitude is called a **discrete-time signal**.

- Sampling a band-limited signal can be achieved without loss of information, as long as the **Shannon rule** is followed.
- The **sampling frequency** $F_s$ must be at least **twice** as high as the highest frequency (Nyquist rate).
- Incorrect sampling (when signal bandwidth is larger than $F_s/2$), **aliasing distortion** occurs.



# Quantization

- The **quantizer** takes the numbers $x(n)$ and assigned quantized values $\hat{x}(n)$ according to a **non-linear discrete-output mapping function** $Q\{\cdot\}$.
- Quantization resulted in **noise** being added to sampled signal $x(n)$ with continuous amplitude.
- $\Delta$ is the **quantization step-size**. The value is decided such that the SNR of the quantized signal is sufficiently large.



- The coder assigns a **binary code** (quantization index) to each quantization level.
- These **codewords** are chosen to correspond to the **quantized amplitudes** such that **arithmetic** can be done directly on the codewords.
- The fine distinction between **quantized samples** and **coded samples** (i.e., base-10 versus base-2 numbers) could generally be ignored.
- Speech signal represented by **binary coded quantized samples** is called **pulse-code modulation** (or just **PCM**) because binary numbers can be transmitted as on/off pulse amplitude modulation.

# Bit rate

- Let $B$ denotes the **number of bits** use to represent the **quantized samples** (i.e., the length of the codewords) and $F_s$ the sampling rate in Hz (or samples per second).
- The **bit rate** (or data rate), measured in bits per second (bps) of a **sampled** and **quantized** speech signal is

$$I = B \times F_s$$

- The standard values for **sampling** and **quantizing** sound signals (singing, instrumental music) are $B = 16$ and $F_s$ = 44.1 kHz or 48 kHz.

$$bit\ rate = 16 \times 44100 = 705{,}600\ bps$$

$$bit\ rate = 16 \times 48000 = 768{,}000\ bps$$

- This value is **more** than **adequate** and much more **than desired** for most **speech** applications.

# Telephone vs. wideband speech

- Human voice ranges from **50 Hz** to **10 kHz**, with **99%** of voice information being below **4 kHz**.
- **Speech** signals are often sampled at **8 kHz**. A low-pass filter is used to **remove spectral components** above the frequency of interest (e.g., 4 kHz) to avoid aliasing distortion.
- This is usually done by sampling at a very **high sampling rate** and applying a **digital** low-pass filter (instead of an **analog** filter) before down-sampling.
- The telephone network limits the bandwidth of speech signals to between the ranges from **300 Hz** to **3400 Hz**. This is called the **telephone bandwidth**.
- **Wideband speech**, as opposed to telephone speech, uses a bandwidth of **50 Hz** to **7000 Hz** and a sampling frequency of 16 kHz.

EE6424 Part 3: Lecture 3.2

# SCALAR QUANTIZATION

# The process of quantization

- Quantization is the process of converting samples of **discrete-time** signal (continuous amplitude) into a **digital** signal with **reduced resolution** (discrete amplitude).
- An analog sample can be considered as having **infinite resolution** as it requires infinite number of bits to represent.
- The use of **16 bits/sample** provides a quality that is considered high.
- **Human ear** is widely believed to be unable to perceive any loss of information with resolution higher than **24 bits/sample**.
- During quantization, the entire **continuous** amplitude range is divided into **finite number** of sub-ranges, referred to as the **quantization intervals**.

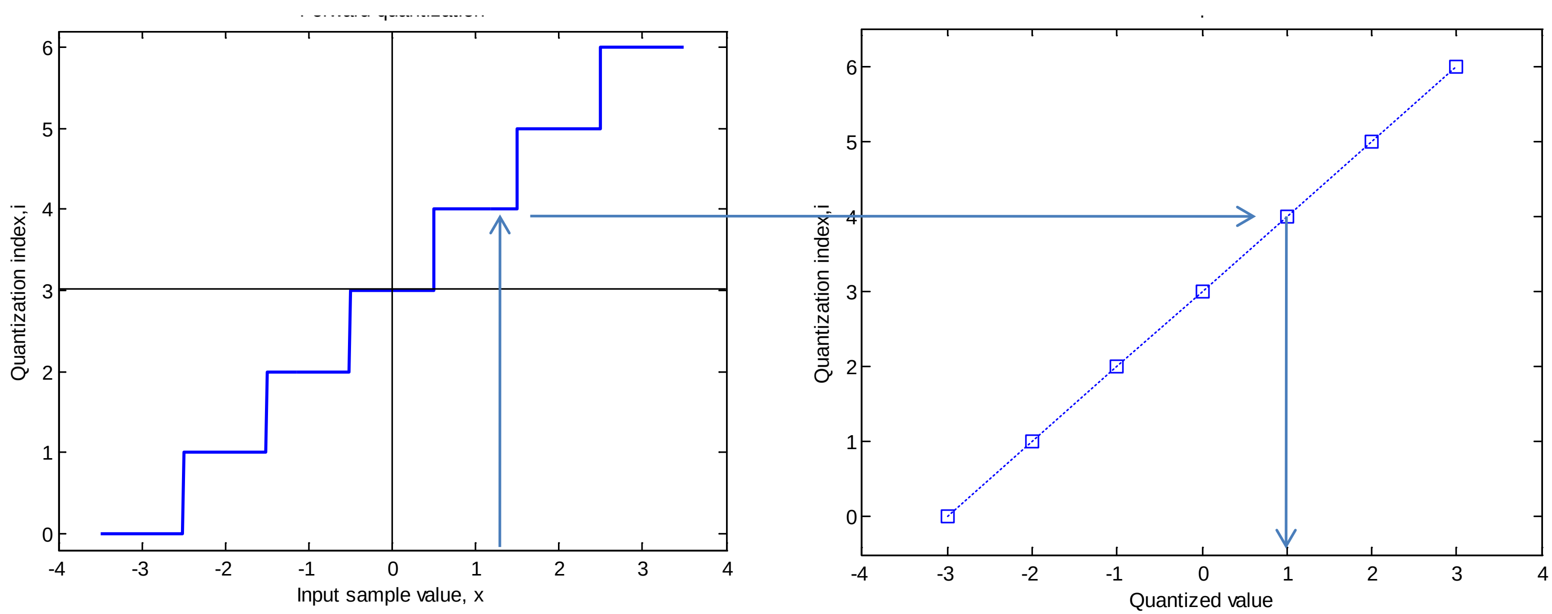
- The mechanism of quantization
  - Divide the real number line into **quantization intervals**. This determines the resolution.
  - Associate each interval with a quantization index, each corresponding to a **binary code**.
  - Map each interval to a **quantized value**, as given by a **quantization table**.

| Input interval | Binary code | Quantized value |
|---|---|---|
| [-3.5, -2.5) | 101 | -3 |
| [-2.5, -1.5) | 110 | -2 |
| [-1.5, -0.5) | 111 | -1 |
| [-0.5, 0.5) | 000 | 0 |
| [0.5, 1.5) | 001 | 1 |
| [1.5, 2.5) | 010 | 2 |
| [2.5, 3.5) | 011 | 3 |

The notation [a, b) is used to indicate an interval from a to b that is inclusive of a but exclusive of b.

- Each interval is defined by its left and right boundaries and associated with a **binary code** and a **quantized value** on the right panel. See example below for the case of 7 quantization intervals.

- Samples with their amplitudes fall into the same **quantization interval** are assigned the same **quantized values**.



# Quantization step-size

- The quantized value is usually taken as the **middle point** in the quantization **interval**.

- The size of the quantization interval is referred to as the **quantization step size**

$$\Delta = b_{i+1} - b_i$$

- Here, $b_i$ and $b_{i+1}$, for $i = 1, 2, \dots, M$, indicate the left and right **boundaries** of the $i$th quantization interval, and $M$ is the **number of quantization levels**.

- When a signal is assumed to be quantized by $B$ bits, the number of levels is usually set to

$$M = 2^B$$

- This ensures the most efficient use of the binary codes available.

- Both quantization bits $B$ and step size $\Delta$ are selected together to properly **cover the range of the signal**. For example, we could choose to cover 95% of a normal distribution.

- Assuming that $|x(n)| \leq x_{\max}$, where the signal amplitude is bounded within $\pm x_{\max}$, we have

$$2x_{\max} = \Delta(2^B)$$

- The quantization step size given the binary resources (usually in terms of bit rate) available:

$$\Delta = \frac{2x_{\max}}{2^B}$$

- Alternatively, for a given quantization step size and signal amplitude, the quantization bits (i.e., resolution) $B$ must satisfies

$$2x_{\max} = \Delta(2^B) \quad \rightarrow \quad B > \log_2\left(\frac{2x_{\max}}{\Delta}\right)$$

- Another way to express the above condition is by taking the **round number** ($B$ has to be an integer) towards plus infinity (e.g., the **ceil** function in MATLAB)

$$B = \left\lceil \log_2\left(\frac{2x_{\max}}{\Delta}\right) \right\rceil$$

# Quantization noise

- When a signal is digitized (sampling followed by quantization), samples are represented on a linear scale with **finite number of bits**, an irreversible **quantization noise** is introduced.

- **Quantization noise** (**error** or **distortion**) is defined as the difference between the continuous input $x(n)$ and quantized value $\hat{x}(n)$

amplitude

$$q(n) = \hat{x}(n) - x(n)$$

- Notice that the input $x(n)$ has continuous amplitude while the quantized value $\hat{x}(n)$ is discrete and is drawn from a finite set of values.

- The quantization process $Q\{\cdot\}$ distorts the input continuous values $x(n)$ by an **additive noise** $q(n)$:

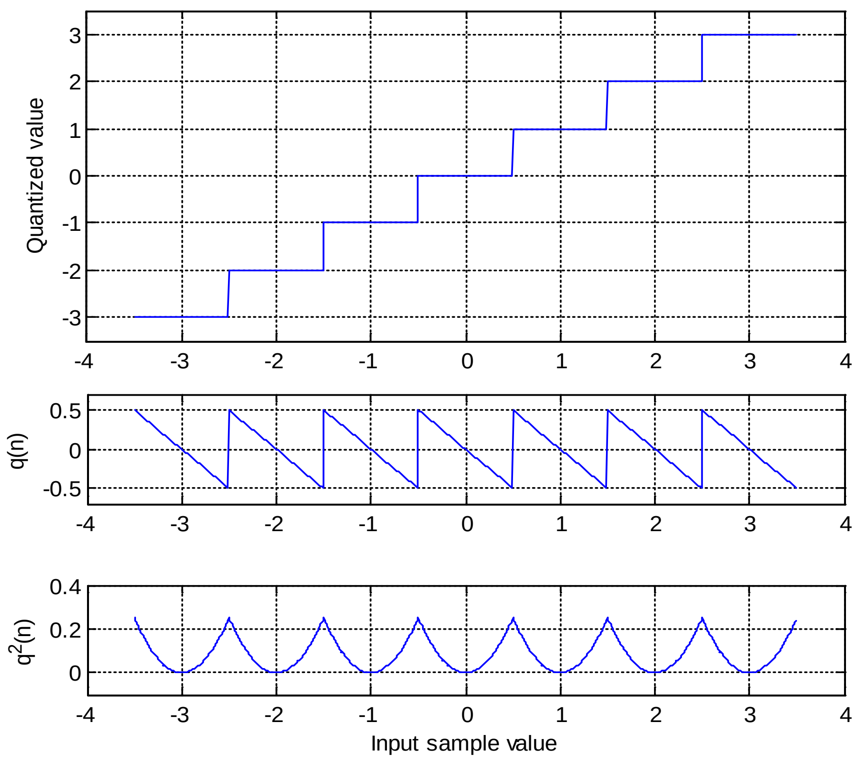
$$\hat{x}(n) = Q\{x(n)\} = x(n) + q(n)$$

- The quantization error manifests itself as the presence of **noise over the signal**.
- The **amplitude** of the quantization noise is determined by the **step size**, which in turn is determined by the **maximum amplitude** $x_{\max}$ of the input signal and **quantization resolution** $B$.
- Let the quantized value be at the middle of the quantization interval, and all quantization intervals have the same length (uniform quantization), the **quantization noise** satisfies:

$$-\frac{\Delta}{2} \leq q(n) \leq \frac{\Delta}{2}$$

# Example

- The quantization step $\Delta$ is set to 1. Therefore the maximum value of quantization error $|q(n)|$ is $\Delta/2 = 0.5$.




- For each interval, the value of the quantization error goes from $\Delta/2 = 0.5$ to $-\Delta/2 = -0.5$.

- The quantization error is **uniformly distributed** from $\Delta/2 = 0.5$ to $-\Delta/2 = -0.5$.

# Property of $q(n)$

- We assume that the **quantization noise** $q(n)$ as a **white noise** with the following properties
  - **Stationary** and **white**

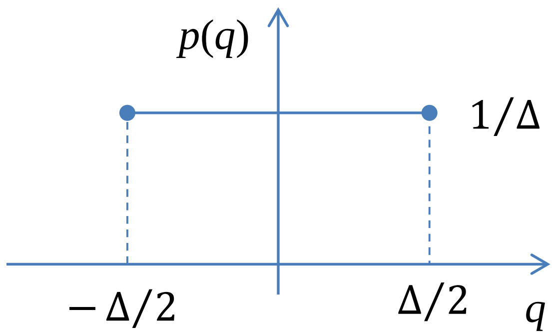
$$\gamma_{qq}(l) = E\{q(n)q(n-l)\} = 0 \;\; \forall\, l \neq 0$$

where $E\{x\}$ denotes the mean of $x$.

  - **Uncorrelated** with the input (assume stationary input $x(n)$)

$$\gamma_{xq}(l) = E\{x(n)q(n-l)\} = 0 \;\; \forall l$$

  - **Uniformly** distributed




$$p(q) = \begin{cases} \frac{1}{\Delta} & \text{for } -\frac{\Delta}{2} \leq q(n) \leq \frac{\Delta}{2} \\ 0 & \text{otherwise} \end{cases}$$

# Mean-square quantization error (MSQE)

- The **quantization error** $q(n)$ is **uniformly** distributed in the interval from $-\Delta/2$ to $\Delta/2$ ,
- Its **variance**, also referred to as the **mean-square quantization error** (MSQE) is computed (assuming zero mean) as follows

$$\sigma_q^2 = E\left\{q^2\right\} = \int_{-\Delta/2}^{\Delta/2} q^2 p(q)\, dq = \int_{-\Delta/2}^{\Delta/2} q^2 \frac{1}{\Delta}\, dq = \frac{1}{\Delta}\int_{-\Delta/2}^{\Delta/2} q^2\, dq$$

$$= \frac{1}{\Delta}\left[\frac{(\Delta/2)^3}{3} - \frac{(-\Delta/2)^3}{3}\right]$$

$$= \frac{\Delta^2}{24} + \frac{\Delta^2}{24} = \frac{\Delta^2}{12}$$

- Let

$$\Delta = \frac{2x_{\max}}{2^B}$$

- The **MSQE** becomes dependent on the **maximum amplitude** $x_{\max}$ of the input signal and the **design parameter** *B*

$$\sigma_q^2 = \frac{\Delta^2}{12} = \left[\frac{2x_{\max}}{2^B}\right]^2 \frac{1}{12} = \left[\frac{4x_{\max}^2}{2^{2B}}\right]\frac{1}{12} = \frac{x_{\max}^2}{3\times 2^{2B}}$$

# Signal-to-quantization noise ratio

- The signal-to-quantization noise ratio (**SQNR** or simply **SNR**) is defined as the ratio between the variances of the input signal and the quantization error, as follows

$$\text{SQNR} = \frac{\sigma_x^2}{\sigma_q^2} = \frac{E\left\{x^2\left(n\right)\right\}}{E\left\{q^2\left(n\right)\right\}}$$

- We assume that the input $x(n)$ and error $q(n) = Q\{x(n)\} - x(n)$, have zero mean.

- The denominator, $E\{q^2(n)\}$, is also referred to as the mean-square quantization error (**MSQE**).

- Using this result, the SQNR could be expressed as

$$\text{SQNR} = \sigma_x^2 \times \left[\frac{x_{\max}^2}{3 \times 2^{2B}}\right]^{-1} = \frac{3 \times 2^{2B}}{x_{\max}^2/\sigma_x^2}$$

- It is customary to represent the SQNR in **dB scale** by taking $10 \times \log_{10}$

$$\begin{aligned}\text{SQNR} &= 10 \times \log_{10}\frac{\sigma_x^2}{\sigma_q^2}\ dB \\ &= 10 \times \left[\log_{10}3 + B\log_{10}4 - 2\log_{10}\frac{x_{\max}}{\sigma_x}\right] \\ &= 6.02\,B + 4.77 - 20\log_{10}\frac{x_{\max}}{\sigma_x}\end{aligned}$$

# Loading factor

- The ratio $x_{\max}/\sigma_x$ is referred to as the **loading factor** of the quantizer.
- For an input $x(n)$ with **uniform distribution**, the loading factor can be computed from the variance and the maximum amplitude ($x_{\max}$ being also the peak signal amplitude)

$$\sigma_x^2 = \frac{(2x_{\max})^2}{12}$$

$$\sigma_x = \frac{2x_{\max}}{\sqrt{12}}$$

$$Loading\ factor = \frac{x_{\max}}{\sigma_x} = \frac{\sqrt{12}}{2} = 1.7$$

- For **uniformly** distributed **input**, the last term in the SQNR formula is

$$20\log_{10}\left(\frac{x_{\max}}{\sigma_x}\right) = 20\log_{10}\left(\frac{\sqrt{12}}{2}\right) = 4.77$$

- The SQNR becomes

$$\begin{aligned} \mathrm{SQNR} &= 6.02B + 4.77 - 4.77 \text{ dB} \\ &= 6.02B \text{ dB} \end{aligned}$$

- The above leads to the conclusion that **increasing the resolution by one bit increases the SNR by 6.02 dB**.

- The number of bits *B* must be decided so that the **SQNR** of the quantized signal is sufficiently large.

# Overload factor

- For **non-uniform** input, it is common to assume a **loading factor of 4**

$$x_{\max} = 4\sigma_x$$

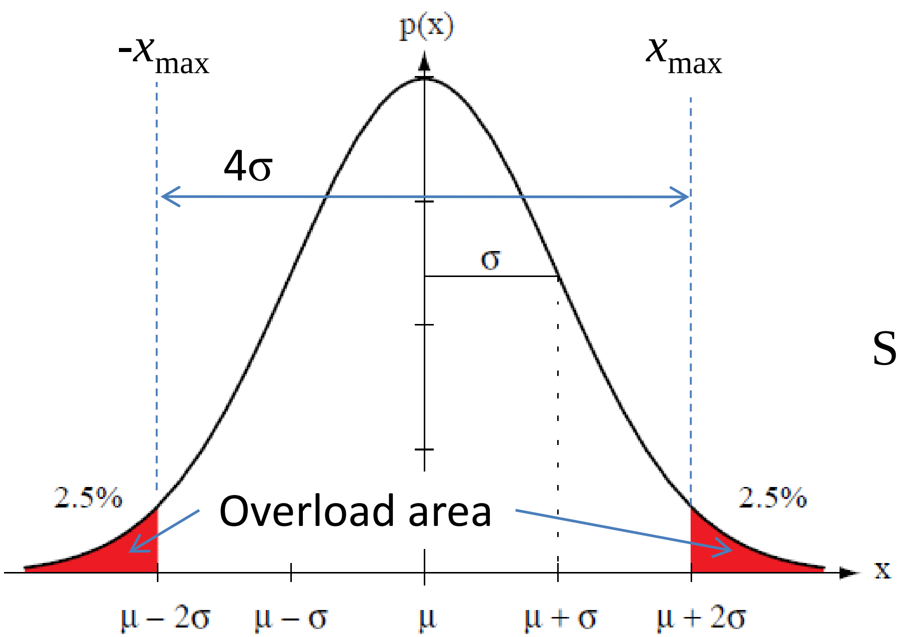
The maximum amplitude is four times the standard deviation of the input signal.

$$\begin{aligned}\text{SQNR} &= 6.02B + 4.77 - 20\log_{10} 4 \text{ dB} \\ &= 6.02B - 7.27 \text{ dB}\end{aligned}$$

- For the case when the amplitude distribution follows a **Gaussian** distribution, a **loading factor of 4** cover **99%** of the possible input values.

- The SQNR reduces as compared to **uniform** input (having a **smaller** loading factor of 1.7).

- The left over <u>≤1%</u> with value $|x| > 4\sigma_x$, which falls into the **overload area**, will be mapped to the same quantized value as $\pm x_{\max}$.
- This introduces the so-called **overload error** (or noise), which added up to the quantization noise and thereby **reducing** the overall **SQNR**.




- This example shows for the case of $x_{\max} = 2\sigma_x$ with a loading factor of 2.

$$\begin{aligned} \mathrm{SQNR} &= 6.02B + 4.77 - 20\log_{10} 2 \text{ dB} \\ &= 6.02B - 1.25 \text{ dB} \end{aligned}$$

# Loading factor vs. overload noise

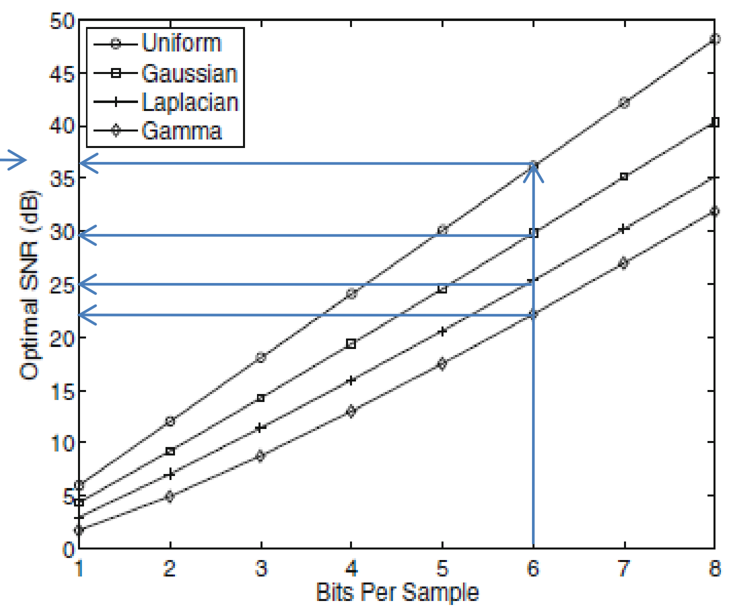
- Most signals, speech especially, have a **wide dynamic range**. The amplitudes vary greatly between **voiced** and **unvoiced** sounds.
  - A **large loading factor** reduces the overload noise.
  - This increases the quantization step size $\Delta = 2x_{\max}/2^B$ and therefore the **quantization noise** (for the same $B$).
  - The quantization noise is the same whether the signal sample is large or small.
  - For a given peak-to-peak value, we could reduce the noise by **adding more bits**.
- Another alternative is use **log-scale quantization** via **companding**.
- In companding, signal amplitude is **compressed** or **warped** so as to approach to that of uniform distribution before **quantization**.

- Companding is motivated by the fact that **optimum SQNR** is obtained for **uniform input**. Deviation from uniform distribution reduces the SQNR.

- The following figure show the optimal SQNR for **uniform**, **Gaussian**, **Laplacian**, and **Gamma** distributions. [Source: Y. You, *Audio Coding: Theory and Applications*, Springer, 2010.]

6 x 6.02 = 36.1 dB

The SQNR is lower for the same number of bits used for non-uniform signal.



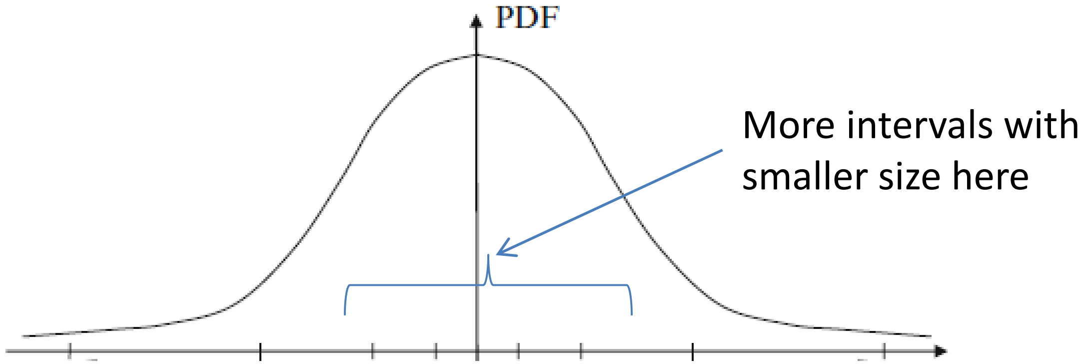
# Companding

- Companding is commonly used in the telephone system.
- The amplitude of the signal is compressed by logarithmic transformation before **uniform quantization**.
- At the decoding stage, the amplitude is exponentially expanded.
- The combined process is called **companding**:

$$\text{Compressor} \longrightarrow Q\{\cdot\} \longrightarrow \text{Expander}$$

- The **effective decision boundaries** when seen from the expander output is **logarithmic**.
- The overall process can be seen as a **logarithmic scale** quantization as opposed to **linear scale** (or uniform) quantization.

- The distribution of speech samples follows a Laplacian (or Gaussian) distribution, with heavy **concentration** around the **mode**.
- The compressor **warps** the input distribution such that it is closer to a uniform distribution.
- The consequence is that **more** quantization steps with **smaller** size are placed in high density area around the mode.



- Companding leads to a more **efficient** use of binary bits.

- An **8-bit log PCM** could give the speech quality almost equivalent to a **12-bit linear PCM**.

- From speech coding perspective, compression (reduction in the bit rate) is achieved by removing **statistical redundancy** due to **non-uniform** distribution of the input signal samples.

- Two kinds of transformation formulae commonly used are
  - $\mu$-law
  - A-law

  The actual difference between the two is marginal.

# μ-law companding

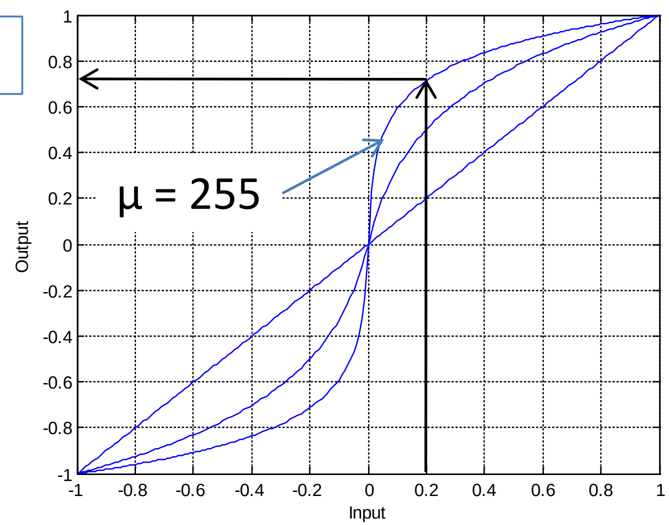
- Compressor (a)

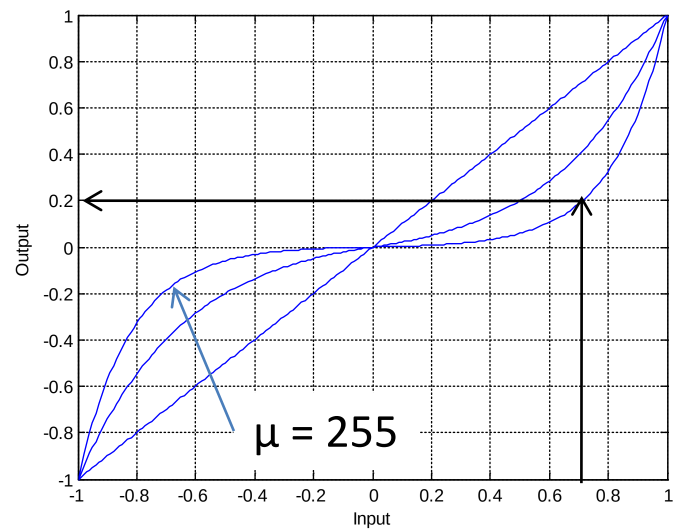
$$y = f(x) = \mathrm{sign}(x)\frac{\ln\left(1+\mu|x|\right)}{\ln\left(1+\mu\right)}, \quad -1 \le x \le 1$$

- Expander (b)

$$x = f^{-1}(y) = \mathrm{sign}(y)\frac{\left(1+\mu\right)^{|y|}-1}{\mu}, \quad -1 \le y \le 1$$

(a)




(b)

- The larger $\mu$ becomes, the larger the amount of **amplitude compression**.
- Typically, values between 100 and 500 are used for $\mu$. In particular, 8-bit, $\mu$ = 256, with sampling frequency at 8 kHz is commonly used for digital telephony.

EE6424 Part 3: Lecture 3.3

# VECTOR QUANTIZATION

# Vector quantization

- **Scalar quantization** (SQ) quantizes a signal **one sample** at a time. Furthermore, the mapping of a sample value is not influenced by previous or following sample values.

- **Vector quantization** (VQ) quantizes a block (or **vector**) of input samples each time, usually leading to a higher efficiency especially for correlated signals.

- Let $\mathbf{x}$ be an $N$-dimensional vector (i.e., we are interested in quantizing $N$ samples at a time):

$$\mathbf{x} = \left[ x_0,\ x_1,\ \ldots,\ x_{N-1} \right]^{\mathrm{T}}$$

- Notice that $\mathbf{x}$ as defined above is a **column** vector. The transpose operator $\mathrm{T}$ at the upper right corner turns the row vector (takes less writing space) into a column vector. This notation is commonly used in engineering to represent vector variables.

# Quantization regions

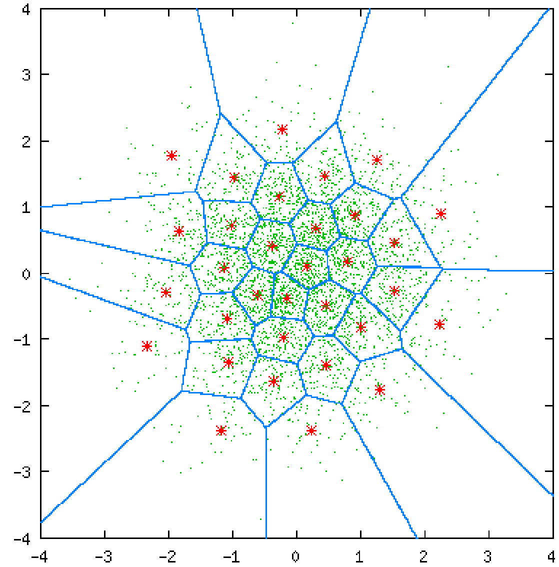
- The vector $\mathbf{x}$ is drawn from a vector space $\Omega$, which can be divided into a set of $M$ **regions** $\{\delta_0, \delta_2, \ldots, \delta_{M\text{-}1}\}$ in a **mutually exclusive** and **collectively exhaustive** way.

(a) The regions form the entire vector space.

$$\Omega = \bigcup_{i=0}^{M-1} \delta_i$$

(b) The regions are mutually exclusive (or disjoint).

$$\delta_i \cap \delta_j = \varnothing \quad \forall i \neq j$$




- The mutually exclusive property implies that a vector can fall to one and only one region among the $M$ regions (similar to the case of SQ).

# VQ codebook

- A representative vector $\mathbf{r}_i$ is assigned to each region $\delta_i$ , $i = 0, 1, \ldots, M\text{-}1$. The set of $M$ representative vectors $\{\mathbf{r}_0, \mathbf{r}_1, \ldots, \mathbf{r}_{M\text{-}1}\}$ are referred to as the **VQ codebook**.
- An input vector $\mathbf{x}$ is quantized based on its distance $d(\mathbf{x}, \mathbf{r}_i)$ from the representative vectors $\mathbf{r}_i$ as follows:

$$Q(\mathbf{x}) = \mathbf{r}_i \quad \text{if and only if} \quad d(\mathbf{x}, \mathbf{r}_i) < d(\mathbf{x}, \mathbf{r}_j) \; \forall i \neq j$$

- The **quantized vector** is the representative vector $\mathbf{r}_i$ of the region where the input sample falls into. In uniform SQ, the center of the interval is used as the quantized value.

# VQ design

- The goal of VQ design is to find the set of regions $\{\delta_0, \delta_1, \ldots, \delta_{M\text{-}1}\}$ and the codebook $\{\mathbf{r}_0, \mathbf{r}_1, \ldots, \mathbf{r}_{M\text{-}1}\}$ that minimize the total quantization error over a training set $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k, \ldots, \mathbf{x}_L\}$ of $L$ vectors:

$$Err = \sum_{k=1}^{L} d\left[\mathbf{x}_k, \hat{\mathbf{x}}\left(\mathbf{x}_k\right)\right]$$

- The distance or error is defined as the Euclidean distance of the input vector $\mathbf{x}_k$ and the representative vector $\mathbf{r}_i$ as follows

$$d\left(\mathbf{x}, \mathbf{r}_i\right) = \sum_{n=1}^{N}\left(x_n - r_{i,n}\right)^2$$

- The **Linde-Buzo-Gray** (LBG) algorithm, also known as k-means algorithm, is commonly used to find the optimal VQ code book.

# The LBG (k-means) algorithm

- Step 1: Initialize the VQ codebook (randomly)
- Step 2: Quantize each training vector $\mathbf{x}_k$ using the current codebook

$$Q(\mathbf{x}_k) = \mathbf{r}_i \quad \text{if and only if} \quad d(\mathbf{x}_k, \mathbf{r}_i) < d(\mathbf{x}_k, \mathbf{r}_j) \; \forall i \neq j$$

- Step 3: Update the VQ codebook $\{\mathbf{r}_0, \mathbf{r}_1, \ldots, \mathbf{r}_{M\text{-}1}\}$ , in which the new representative vectors are taken as the centroids of the regions. Here, $L_i$ is the number of training samples assigned to the $i$th region in Step 2.

$$\mathbf{r}_i = \frac{1}{L_i} \sum_{\mathbf{x}_k \in \delta_i} \mathbf{x}_k, \quad i = 0, 1, \ldots, M-1$$

- Step 4: Calculate the total quantization error over all the $L$ training samples:

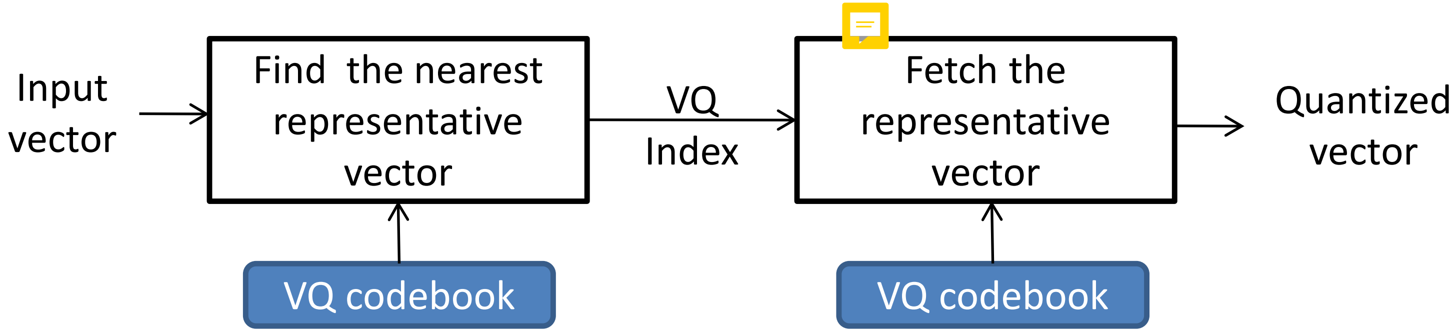
$$Err = \sum_{k=1}^{L} d\left[\mathbf{x}_k, \hat{\mathbf{x}}(\mathbf{x}_k)\right]$$

Go to Step 2 if the change in $Err$ is greater than the pre-determined threshold $\varepsilon$

$$\frac{Err(t-1)-Err(t)}{Err(t)} > \varepsilon$$

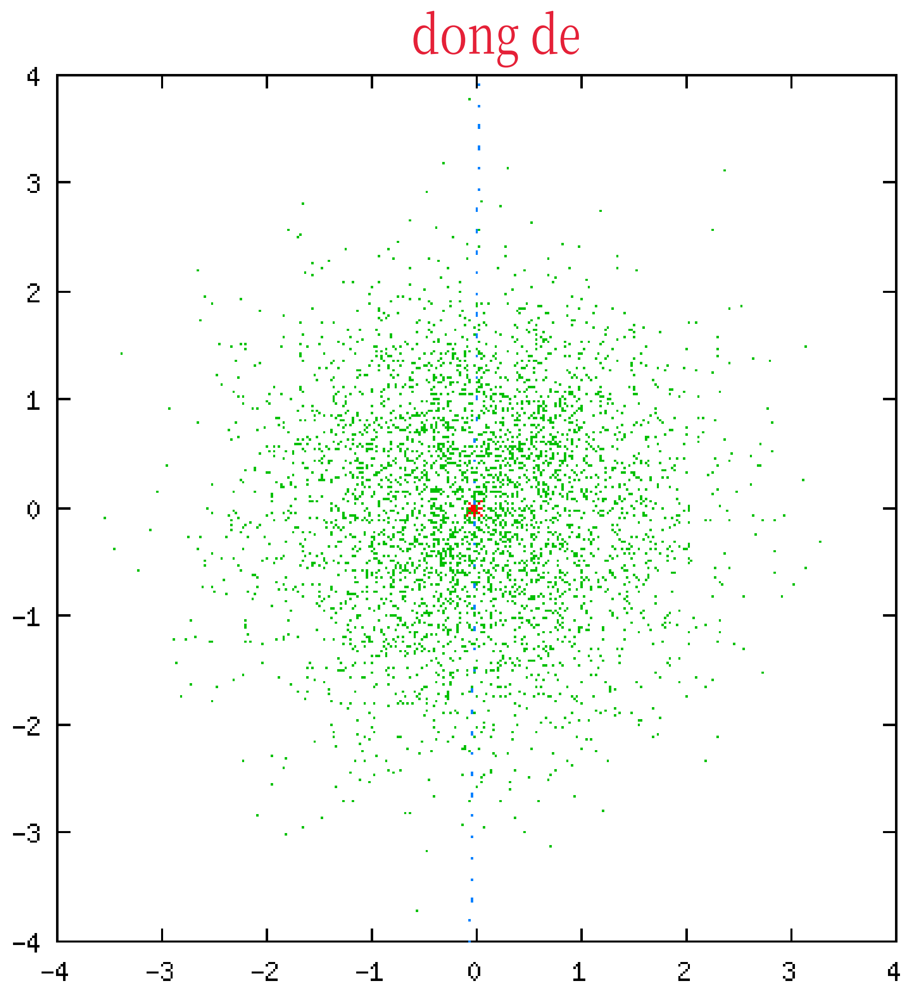
- Step 5: Stop

- VQ is used, for example in speech coding, where blocks of samples are quantized. The indices are sent to the receiving end where quantized samples are recovered using the same codebook.



# VQ Codebook example

- The following shows the VQ training process in the **two-dimensional** vector space.
- The red dots represent the **centroids** of the regions, and the region boundaries are represented as straight lines.
- **Training** data are represented as smaller dots in green
- Source: http://www.data-compression.com/vq.shtml



# Summary

- **Sampling and quantization** of the input signal into **digital form** is the necessary process in any applications of digital signal processing
- Sampling is a time-domain operation whereby a **continuous** time signal is represented as a **periodic sequence** of numbers.
- Sampling a band-limited signal can be achieved without loss of information, as long as the **Shannon rule** is followed.
- Quantization involves representing the sampled values by one from a finite set of **values**.
- The quantized values could be represented as **binary codes** (assuming that we use binary number system as it commonly used nowadays in digital transmission system).

- The quantization process $Q\{\cdot\}$ distorts the input continuous values $x(n)$ by an **additive noise** $q(n)$.

- **Uniform** (or linear) quantization produces highest SQNR (1 bit $\longrightarrow$ 6dB) when the input signal has a **uniform** distribution.

- For non-uniform input, **companding** could be used, by which statistical redundancy is reduced by warping the input samples toward uniform distribution.

- Vector quantization (VQ) quantizes blocks of samples as opposed to scalar quantization (SQ) which quantizes the input signal one sample at a time.