Chapter 2
Scalar Quantization

An audio signal is a representation of sound waves usually in the form of sound pressure level that varies with time. Such a signal is continuous both in value and time, hence carries an infinite amount of information.

The first step of significant compression is accomplished when a continuous-time audio signal is converted into a discrete-time signal using sampling. In what constitutes uniform sampling, the simplest sampling method, the continuous-time signal is sampled at a regular interval T, called **sampling period**. According to Nyquist–Shannon **sampling theorem** [65, 68], the original continuous-time signal can be perfectly reconstructed from the sampled discrete-time signal if the continuous-time signal is band-limited and its bandwidth is no more than half of the **sample rate** $(1/T)$. Therefore, sampling accomplishes a tremendous amount of lossless compression if the source signal is ideally bandlimited.

After sampling, each sample of the discrete-time signal has a value that is continuous, so the number of possible distinct output values is infinite. Consequently, the number of bits needed to represent and/or convey such a value exactly to a recipient is unlimited.

For the human ear, however, an exact continuous sample value is unnecessary because the resolution that the ear can perceive is very limited. Many believe that it is less than 24 bits. So a simple scheme of replacing an analog sample value with an integer value that is closet to it would not only satisfy the perceptual capability of the ear, but also removes a tremendous deal of imperceptible information from a continuously valued signal. For example, the hypothetical "analog" samples in the left column of Table 2.1 may be represented by the respective integer values in the right column. This process is called *quantization*.

The underlying mechanism for quantizing the sample values in Table 2.1 is to divide the real number line into real intervals and then map each of such interval to an integer value. This is shown in Table 2.2, which is call a **quantization table**. The quantization process actually involves three steps as shown in Fig. 2.1 and explained below:

Forward Quantization. A source sample value is used to look up the left column to find the interval, referred to as *decision interval*, that it falls into and the corresponding index, referred to as *quantization index*, in the center column is then identified. This mapping is referred to as encoder mapping.

Y. You, *Audio Coding: Theory and Applications*, DOI 10.1007/978-1-4419-1754-6_2,

Table 2.1 An example of mapping "analog" sample values to integer values that would take place in a process called quantization

"Analog" sound pressure level	Integer sound pressure level
$-3.4164589759\cdots$	-3
$-3.124341\cdots$	-3
$-2.14235\cdots$	-2
$-1.409086743\cdots$	-1
$-0.61341984378562890423\cdots$	-1
$0.37892458\cdots$	0
$0.61308\cdots$	1
$1.831401348156\cdots$	2
$2.8903219654710\cdots$	2
$3.208913064\cdots$	3

Table 2.2 Quantization table that maps source sample intervals in the *left column* to integer values in the *right column*

Sample value interval	Index	Integer value
$(-\infty, -2.5)$	0	-3
$[-2.5, -1.5)$	1	-2
$[-1.5, -0.5)$	2	-1
$[-0.5, 0.5)$	3	0
$[0.5, 1.5)$	4	1
$[1.5, 2.5)$	5	2
$[2.5, \infty)$	6	3

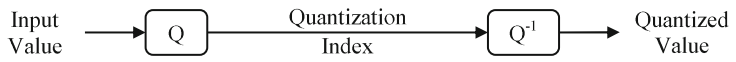

Fig. 2.1 Quantization involves an encoding or forward quantization stage represented by "Q", which maps an input value to the quantization index, and a decoding or inverse quantization stage represented by "Q^{-1}", which maps the quantization index to the quantized value

Index Transmission. The quantization index is transmitted to the receiver.

Inverse Quantization. Upon receiving the quantization index, the receiver uses it to read out the integer value, referred to as the *quantized value*, in the right column. This mapping is referred to as decoder mapping.

The quantization table above maps sound pressure levels with infinite range and resolution into seven integers, which need only 3 bits to represent, thus achieving a great deal of data compression. However, this comes with a price: much of the original resolution is lost forever. This loss of information may be significant, but it was done on purpose: those lost pieces of information are irrelevant to our needs or perception, we can afford to discard them.

2.1 Scalar Quantization

To pose the quantization process outlined above mathematically, let us consider a source random variable X with a **probability density function** (**PDF**) of $p(X)$. Suppose that we wish to quantize this source with M **decision intervals** defined by the following $M + 1$ endpoints

$$b_q, \quad q = 0, 1, \ldots, M, \tag{2.1}$$

referred to as **decision boundaries**, and with the following M **quantized values**,

$$\hat{x}_q, \quad q = 1, 2, \ldots, M, \tag{2.2}$$

which are also called **output values** or **representative values**. A source sample value x is quantized to the **quantization index** q if and only if x falls into the qth decision interval

$$\delta_q = [b_{q-1}, b_q), \tag{2.3}$$

so the operation of **forward quantization** is

$$q = Q(x), \quad \text{if and only if} \quad b_{q-1} \le x < b_q. \tag{2.4}$$

The quantized value can be reconstructed from the quantization index by the following **inverse quantization**

$$\hat{x}_q = Q^{-1}(q), \tag{2.5}$$

which is also referred to as **backward quantization**. Since q is a function of x as shown in (2.4), $\hat{x}_q$ is also a function of x and can be written as:

$$\hat{x}(x) = \hat{x}_q = Q^{-1}\left[Q(x)\right]. \tag{2.6}$$

This quantization scheme is called **scalar quantization** (**SQ**) because the source signal is quantized one sample each time.

The function in (2.6) is another approach to describing the **input–output map of a quantizer**, in addition to the *quantization table*. Figure 2.2 is such a function that describe the quantization map of Table 2.2.

The quantization operation in (2.4) obviously causes much loss of information, the reconstructed quantized value obtained in (2.5) or (2.6) is different than the input to the quantizer. The difference between them is called **quantization error**

$$q(x) = \hat{x}(x) - x. \tag{2.7}$$

It is also referred to as **quantization distortion** or **quantization noise**.

Equation (2.7) may be rewritten as

$$\hat{x}(x) = x + q(x), \tag{2.8}$$

so the quantization process is often modeled as an additive noise process as shown in Fig. 2.3.

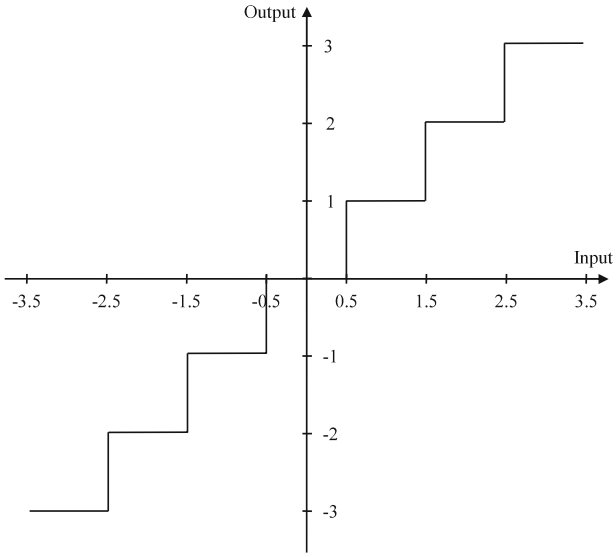

Fig. 2.2 Input–output map for the quantizer shown in Table 2.2

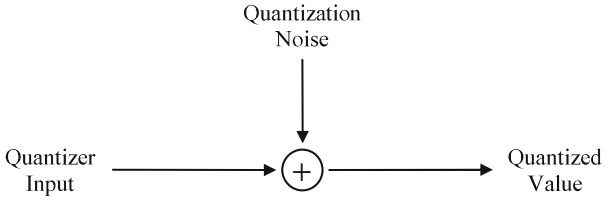

Fig. 2.3 Additive noise model for quantization

The average loss of information introduced by quantization may be characterized by average quantization error. Among the many norms that may be used to measure this error, the L-2 norm or Euclidean distance is usually used and is called **mean squared quantization error** (**MSQE**):

$$\begin{aligned}\sigma_q^2 &= \int_{-\infty}^{\infty} q^2(x)\,p(x)\mathrm{d}x \\ &= \int_{-\infty}^{\infty} (\hat{x}(x)-x)^2\,p(x)\mathrm{d}x \\ &= \sum_{q=1}^{M} \int_{b_{q-1}}^{b_q} (\hat{x}(x)-x)^2\,p(x)\mathrm{d}x \end{aligned} \tag{2.9}$$

Since $\hat{x}(x) = \hat{x}_q$ is a constant within the decision interval $[b_{q-1}, b_q)$, we have

$$\sigma_q^2 = \sum_{q=1}^{M} \int_{b_{q-1}}^{b_q} (\hat{x}_q - x)^2 p(x) \mathrm{d}x. \tag{2.10}$$

The MSQE may be better appreciated when compared with the power of the source signal. This may be achieved using **signal-to-noise ratio** (**SRN**) defined below

$$\text{SNR (dB)} = 10 \log_{10} \left(\frac{\sigma_x^2}{\sigma_q^2} \right), \tag{2.11}$$

where σ_x^2 is the variance of the source signal.

It is obvious that the smaller the decision intervals, the smaller the error term $(\hat{x}_q - x)^2$ in (2.10), thus the smaller the mean squared quantization error σ_q^2. This indicates that σ_q^2 is inversely proportional to the number of decision intervals M. The placement of each individual decision boundary and the quantized value also play major roles in the final σ_q^2. The problem of quantizer design may be posed in a variety of ways, including:

- Given a fixed M:

$$M = \text{Constant}, \tag{2.12}$$

  find the optimal placement of decision boundaries and quantized values so that σ_q^2 is minimized. This is the most widely used approach.
- Given a distortion constraint:

$$\sigma_q^2 < \text{Threshold}, \tag{2.13}$$

  find the optimal placement of decision boundaries and quantized values so that M is minimized. A minimal M means a minimal number of bits needed to represent the quantized value, hence a minimal bit rate.

2.2 Re-Quantization

The quantization process was presented above with the assumption that the source random variable or sample values are continuous or analog. Quantization by name usually gives the impression that it were only for quantizing analog sample values. When dealing with such analog source sample values, the associated forward quantization is referred to as **ADC** (**analog-to-digital conversion**) and the inverse quantization as **DAC** (**digital-to-analog conversion**).

Table 2.3 An quantization table for "re-quantizing" a discrete source

Decision interval	Quantization index	Re-quantized value
[0, 10)	0	5
[10, 20)	1	15
[20, 30)	2	25
[30, 40)	3	35
[40, 50)	4	45
[50, 60)	5	55
[60, 70)	6	65
[70, 80)	7	75
[80, 90)	8	85
[90, 100)	9	95

Discrete sources sample values can also be further quantized. For example, consider a source that takes integer sample values between 0 through 100. If it is decided, for some reason, that this resolution is too much or irrelevant for a particular application and sample values spaced at an interval of 10 are really what are needed, a quantization table shown in Table 2.3 can be established to **re-quantize** the integer sample values.

With discrete sources sample values, the formulation of quantization process in Sect. 2.1 is still valid with the replacement of probability density function with probability distribution function and integration with summation.

2.3 Uniform Quantization

Both quantization Tables 2.2 and 2.3 are the embodiment of *uniform quantization*, which is the simplest among all quantization schemes. The decision boundaries of a uniform quantizer are equally spaced, so its decision intervals are all of the same length and can be represented by a constant called *quantization step size*. For example, the quantization step size for Table 2.2 is 1 and for Table 2.3 is 10.

When an analog signal is uniformly sampled and subsequently quantized using a uniform quantizer, the resulting digital representation is called **pulse-code modulation** (**PCM**). It is the default form of representation for many digital signals, such as speech, audio, and video.

2.3.1 Formulation

Let us consider a **uniform quantizer** that covers an interval of $[X_{\min}, X_{\max}]$ of a random variable X with M decision intervals. Since its **quantization step size** is

$$\Delta = \frac{X_{\max} - X_{\min}}{M}, \tag{2.14}$$

its decision boundaries can be represented as

$$b_q = X_{\min} + \Delta \cdot q, \quad q = 0, 1, \ldots, M. \tag{2.15}$$

The mean of an decision interval is often selected as the quantized value for that interval:

$$\hat{x}_q = X_{\min} + \Delta \cdot q - 0.5\Delta, \quad q = 1, 2, \ldots, M. \tag{2.16}$$

For such a quantization scheme, the MSQE in (2.10) becomes

$$\sigma_q^2 = \sum_{q=1}^{M} \int_{X_{\min}+\Delta\cdot(q-1)}^{X_{\min}+\Delta\cdot q} (X_{\min} + \Delta \cdot q - 0.5\Delta - x)^2 p(x) \mathrm{d}x. \tag{2.17}$$

Let

$$y = X_{\min} + \Delta \cdot q - 0.5\Delta - x,$$

(2.17) becomes

$$\sigma_q^2 = \sum_{q=1}^{M} \int_{-0.5\Delta}^{0.5\Delta} y^2 p\left[X_{\min} + \Delta \cdot q - (y + 0.5\Delta)\right]^2 \mathrm{d}y. \tag{2.18}$$

Plugging in (2.15), (2.18) becomes

$$\sigma_q^2 = \sum_{q=1}^{M} \int_{-0.5\Delta}^{0.5\Delta} x^2 p\left[b_q - (x + 0.5\Delta)\right] \mathrm{d}x. \tag{2.19}$$

Plugging in (2.16), (2.18) becomes

$$\sigma_q^2 = \sum_{q=1}^{M} \int_{-0.5\Delta}^{0.5\Delta} x^2 p(\hat{x}_q - x) \mathrm{d}x. \tag{2.20}$$

2.3.2 Midtread and Midrise Quantizers

There are two major types of uniform quantizers. The one shown in Fig. 2.2 is called **midtread** because it has zero as one of its quantized values. It is useful for situations where it is necessary for the zero value to be represented. One such example is control systems where a zero value needs to be accurately represented. This is also

important for audio signals because the zero value is needed to represent the absolute quiet. Due to the midtreading of zero, the number of decision intervals (M) is odd if a symmetric sample value range ($X_{\min} = -X_{\max}$) is to be covered.

Since both the decision boundaries and the quantized values can be represented by a single step size, the implementation of the midtread uniform quantizer is simple and straight forward. The forward quantizer may implemented as

$$q = \text{round}\left(\frac{x}{\Delta}\right) \tag{2.21}$$

where round(·) is the **rounding function** which returns the integer that is closest to the input. The corresponding inverse quantizer may be implemented as

$$\hat{x}_q = q\Delta. \tag{2.22}$$

The other uniform quantizer does not have zero as one of its quantized values, so is called **midrise**. This is shown in Fig. 2.4. Its number of decision intervals is even if a symmetric sample value range is to be covered. The forward quantizer may implemented as

$$q = \begin{cases} \text{truncate}\left(\frac{x}{\Delta}\right) + 1, & \text{if } x > 0; \\ \text{truncate}\left(\frac{x}{\Delta}\right) - 1, & \text{otherwise;} \end{cases} \tag{2.23}$$

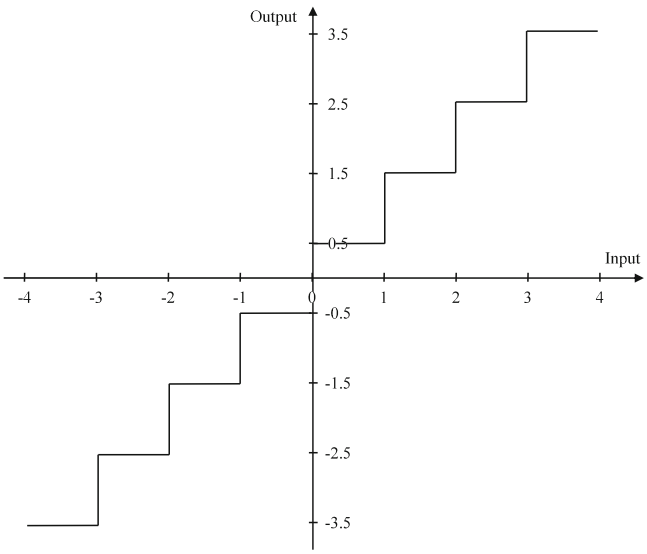

Fig. 2.4 An example of midrise quantizer

where truncate(·) is the **truncate function** which returns the integer part of the input, without the fractional digits. Note that $q = 0$ is forbidden for a midrise quantizer. The corresponding inverse quantizer is expressed below

$$\hat{x}_q = \begin{cases} (q - 0.5)\Delta, & \text{if } q > 0; \\ (q + 0.5)\Delta, & \text{otherwise.} \end{cases} \tag{2.24}$$

2.3.3 Uniformly Distributed Signals

As seen in (2.20), the MSQE of a uniform quantizer depends on the probability density function. When this density function is uniformly distributed over $[X_{\min}, X_{\max}]$:

$$p(x) = \frac{1}{X_{\max} - X_{\min}}, \quad x \in [X_{\min}, X_{\max}], \tag{2.25}$$

(2.20) becomes

$$\begin{aligned} \sigma_q^2 &= \frac{1}{X_{\max} - X_{\min}} \sum_{q=1}^{M} \int_{-0.5\Delta}^{0.5\Delta} y^2 \mathrm{d}x \\ &= \frac{1}{X_{\max} - X_{\min}} \sum_{q=1}^{M} \frac{\Delta^3}{12} \\ &= \frac{M}{X_{\max} - X_{\min}} \frac{\Delta^3}{12} \end{aligned}$$

Due to the step size given in (2.14), the above equation becomes

$$\sigma_q^2 = \frac{\Delta^2}{12}. \tag{2.26}$$

For the uniform distribution in (2.25), its variance (signal power) is

$$\sigma_x^2 = \frac{1}{X_{\max} - X_{\min}} \int_{X_{\min}}^{X_{\max}} x^2 dx = \frac{(X_{\max} - X_{\min})^2}{12}, \tag{2.27}$$

so the signal-to-noise ratio (SNR) of the uniform quantizer is

$$\begin{aligned}\text{SNR (dB)} &= 10\log_{10}\left(\frac{\sigma_x^2}{\sigma_q^2}\right)\\ &= 10\log_{10}\left(\frac{(X_{\max}-X_{\min})^2}{12}\frac{12}{\Delta^2}\right)\\ &= 20\log_{10}\left(\frac{X_{\max}-X_{\min}}{\Delta}\right)\end{aligned} \tag{2.28}$$

Due to the step size given in (2.14), the above SNR expression becomes

$$\text{SNR (dB)} = 20\log_{10}(M) = \frac{20}{\log_2(10)}\log_2(M) \approx 6.02\log_2(M). \tag{2.29}$$

If the quantization indexes are represented using *fixed-length codes*, each codeword can be represented using

$$R = \text{ceil}\left[\log_2(M)\right] \text{ bits}, \tag{2.30}$$

which is referred as **bits per sample** or **bit rate**. Consequently, (2.29) becomes

$$\text{SNR (dB)} = \frac{20}{\log_2(10)}R \approx 6.02R \text{ dB}, \tag{2.31}$$

which indicates that, for each additional bit allocated to the quantizer, the SNR is increased by about 6.02 dB.

2.3.4 Nonuniformly Distributed Signals

Most signals, and audio signals in particular, are rarely uniformly distributed. As indicated by (2.20), the contribution of each quantization error to the MSQE is weighted by the probability density function. A nonuniform distribution means that the weighting is different now, so a different MSQE is expected and is discussed in this section.

2.3.4.1 Granular and Overload Error

A nonuniformly distributed signal, such as Gaussian, is usually not bounded, so the dynamic range $[X_{\min}, X_{\max}]$ of a uniform quantizer cannot cover the whole range of the source signal. This is illustrated in Fig. 2.5. The areas beyond $[X_{\min}, X_{\max}]$ are called **overload areas**. When a source sample falls into an overload area, the quantizer can only assign either the minimum or the maximum quantized value to it:

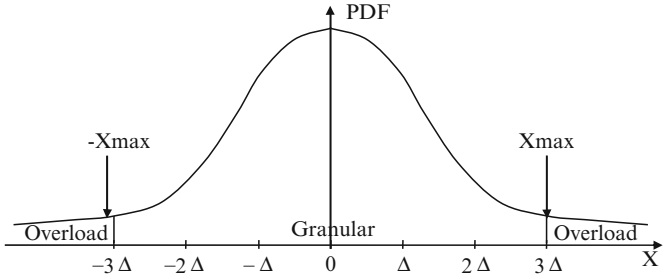

Fig. 2.5 Overload and granular quantization errors

$$\hat{x}(x) = \begin{cases} X_{\max} - 0.5\Delta, & \text{if } x > X_{\max}; \\ X_{\min} + 0.5\Delta, & \text{if } x < X_{\min}. \end{cases} \tag{2.32}$$

This introduces additional quantization error, called **overload error** or **overload noise**. The mean squared overload error is obviously the following

$$\begin{aligned} \sigma^2_{q(overload)} = & \int_{X_{\max}}^{\infty} [x - (X_{\max} - 0.5\Delta)]^2 \, p(x)\mathrm{d}x \\ & + \int_{-\infty}^{X_{\min}} [x - (X_{\min} + 0.5\Delta)]^2 \, p(x)\mathrm{d}x. \end{aligned} \tag{2.33}$$

The MSQE given in (2.17) only accounts for quantization error within $[X_{\min}, X_{\max}]$, which is referred to as **granular error** or **granular noise**. The total quantization error is

$$\sigma^2_{q(\text{total})} = \sigma^2_q + \sigma^2_{q(\text{overload})}. \tag{2.34}$$

For a given PDF $p(x)$ and the number of decision intervals M, (2.17) indicates that the smaller the quantization step size Δ is, the smaller the granular quantization noise σ^2_q becomes. According to (2.14), however, the smaller quantization step size Δ also translates into smaller $-X_{\min}$ and $X_{\max}$ for a fixed M. Smaller $-X_{\min}$ and $X_{\max}$ obviously leads to larger overload areas, hence a larger overload quantization error $\sigma^2_{q(overload)}$. Therefore, the choice of Δ, or equivalently the range $[X_{\min}, X_{\max}]$ of the uniform quantizer, represents a trade-off between granular and overload quantization errors.

This trade-off is, of course, relative to the effective width of the given PDF, which may be characterized by its variance σ. The ratio of the quantization range $[X_{\min}, X_{\max}]$ over the signal variance

$$F_{\text{l}} = \frac{X_{\max} - X_{\min}}{\sigma}, \tag{2.35}$$

called the **loading factor**, is apparently a good description of this trade-off. For Gaussian distribution, a loading factor of 4 means that the probability of input

samples going beyond the range is 0.045. For a loading factor of 6, the probability reduces to 0.0027. For most applications, 4σ loading is sufficient.

2.3.4.2 Optimal SNR and Step Size

To find the optimal quantization step size Δ that gives the minimum total MSQE $\sigma^2_{q(total)}$, let us drop (2.17) and (2.33) into (2.34) to obtain

$$\begin{aligned}\sigma^2_{q(total)} = &\sum_{q=1}^{M} \int_{X_{\min}+\Delta\cdot(q-1)}^{X_{\min}+\Delta\cdot q} [x - (X_{\min} + \Delta\cdot q - 0.5\Delta)]^2\, p(x)dx \\ &+ \int_{X_{\max}}^{\infty} [x - (X_{\max} - 0.5\Delta)]^2\, p(x)dx \\ &+ \int_{-\infty}^{X_{\min}} [x - (X_{\min} + 0.5\Delta)]^2\, p(x)dx. \end{aligned} \tag{2.36}$$

Usually, a uniform quantizer is symmetrically designed such that

$$-X_{\min} = X_{\max}. \tag{2.37}$$

Then (2.14) becomes

$$\Delta = \frac{2X_{\max}}{M}. \tag{2.38}$$

Replacing all $X_{\min}$ and $X_{\max}$ with Δ using the above equations, we have

$$\begin{aligned}\sigma^2_{q(total)} = &\sum_{q=1}^{M} \int_{(q-1-0.5M)\Delta}^{(q-0.5M)\Delta} [(q - 0.5 - 0.5M)\Delta - x]^2\, p(x)dx \\ &+ \int_{0.5M\Delta}^{\infty} [x - 0.5(M-1)\Delta]^2\, p(x)dx \\ &+ \int_{-\infty}^{-0.5M\Delta} [x + 0.5(M-1)\Delta]^2\, p(x)dx. \end{aligned} \tag{2.39}$$

Assuming a symmetric PDF:

$$p(-x) = p(x) \tag{2.40}$$

and doing a variable change of $y = -x$ in the last term of (2.39), it turns out that this last term becomes the same as the second term, so (2.39) becomes

$$\sigma^2_{q(total)} = \sum_{q=1}^{M} \int_{(q-1-0.5M)\Delta}^{(q-0.5M)\Delta} [(q - 0.5 - 0.5M)\Delta - x]^2 \, p(x)dx$$
$$+2 \int_{0.5M\Delta}^{\infty} [x - 0.5(M-1)\Delta]^2 \, p(x)dx \tag{2.41}$$

Now that both (2.39) and (2.41) are only a function of Δ, their minimum can be found by setting their respective first order derivative against Δ to zero:

$$\frac{\partial}{\partial \Delta} \sigma^2_{q(total)} = 0. \tag{2.42}$$

This equation can be solved using a variety of numerical methods, see [76], for example.

Figure 2.6 shows optimal SNR achieved by a uniform quantizer at various bits per sample (see (2.30)) for Gaussian, Laplacian, and Gamma distributions [33]. The SNR given in (2.31) for uniform distribution, which is the best SNR that a uniform quantizer can achieve, is plotted as the bench mark. It is a straight line in the form of

$$\text{SNR(R)} = a + bR \text{ (dB)}, \tag{2.43}$$

with a slope of

$$b = \frac{20}{log_2(10)} \approx 6.02 \tag{2.44}$$

and an intercept of

$$a = 0. \tag{2.45}$$

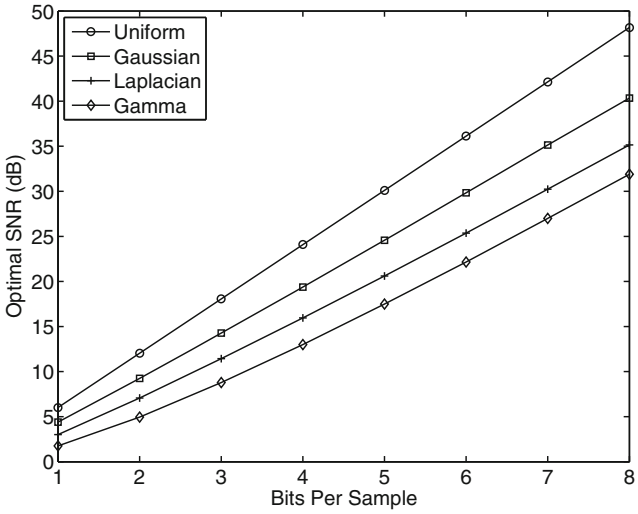

Fig. 2.6 Optimal SNR achieved by a uniform quantizer for uniform, Gaussian, Laplacian, and Gamma distributions

Apparently, the curves for other PDF's also seem to fit a straight line with different slopes and intercepts. Notice that both the slope b and the intercept a decrease as the **peakedness** or **kurtosis** of the PDF increases in the order of uniform, Gaussian, Laplacian, and Gamma, indicating that the overall performance of a uniform quantizer is inversely related to PDF kurtosis. This degradation in performance is mostly reflected in the intercept a. The slope b is only moderately affected.

There is, nevertheless, reduction in slope when compared with the uniform distribution. This reduction indicates that the quantization performance for other distributions relative to the uniform distribution becomes worse at higher bit rates.

Figure 2.7 shows the optimal step size normalized by the signal variance, $\Delta_{\text{opt}}/\sigma_x$, for Gaussian, Laplacian, and Gamma distributions as a function of the number of bits per sample [33]. The data for uniform distribution is used as the benchmark. Due to (2.14), (2.27) and (2.30), the normalized quantization step size for the uniform distribution is

$$\log_{10}\left(\frac{\Delta}{\sigma_x}\right) = \log_{10}\left(\frac{2}{\sqrt{3}}\right) - \frac{\log_2(M)}{\log_2 10} = \log_{10}\left(\frac{2}{\sqrt{3}}\right) - \frac{R}{\log_2 10}, \tag{2.46}$$

so it is a straight line. Apparently, as the peakedness or kurtosis increases in the order of uniform, Gaussian, Laplacian, and Gamma distributions, the step size also increases. This is necessary for optimal balance between granular and overload quantization errors: an increased kurtosis means that the probability density is spread more toward the tails, resulting more overload error, so the step size has to be increased to counteract this increased overload error.

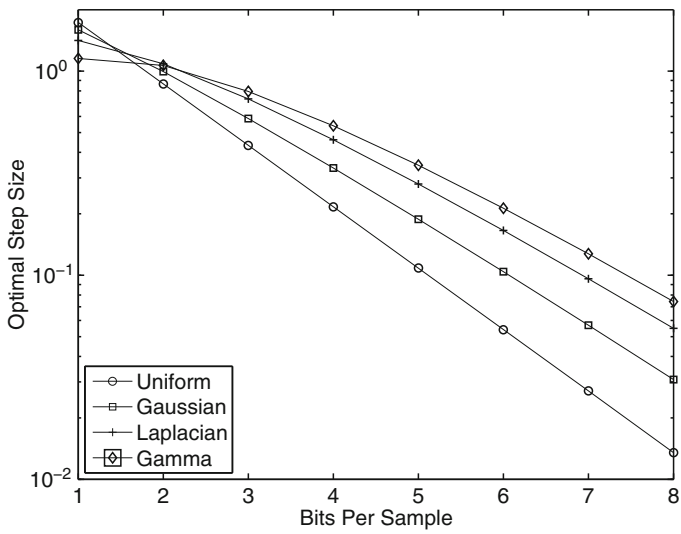

Fig. 2.7 Optimal step size used by a uniform quantizer to achieve optimal SNR for uniform, Gaussian, Laplacian, and Gamma distributions

The empirical formula (2.43) is very useful for estimating the minimal total MSQE for a particular quantizer, given the signal power and bit rate. In particular, dropping in the SNR definition in (2.11) to (2.43), we can represent the total MSQE as

$$10\log_{10}\sigma_q^2 = 10\log_{10}\sigma_x^2 - a - bR \tag{2.47}$$

or

$$\sigma_q^2 = 10^{-0.1(a+bR)}\sigma_x^2. \tag{2.48}$$

2.4 Nonuniform Quantization

Since the MSQE formula (2.10) indicates that the quantization error incurred by a source sample x is weighted by the PDF $p(x)$, one approach to reduce MSQE is to reduce quantization error in densely distributed areas where the weight is heavy. Formula (2.10) also indicates that the quantization error incurred by a source sample value x is actually the distance between it and the quantized value $\hat{x}$, so large quantization errors are caused by input samples far away from the quantized value, i.e., those which are near the decision boundaries. Therefore, reducing quantization errors in densely distributed areas necessitates using smaller decision intervals. For a given number of decision intervals M, this also means that larger decision intervals need to be placed to the rest of the PDF support so that the whole input range is covered.

From the perspective of resource allocation, each quantization index is a piece of bit resource that is allocated in the course of quantizer design, and there are only M pieces of resources. A quantization index is one-to-one associated with a quantized value and decision interval, so a piece of resource is considered as consisting of a set of quantization index, quantized value, and a decision interval. The problem of quantizer design may be posed as optimal allocation of these resources to minimize the total MSQE. To achieve this, each piece of resources should be allocated to carry the same share of quantization error contribution to the total MSQE. In other words, the MSQE conbribution carried by individual pieces of resources should be "equalized".

For a uniform quantizer, its resources are allocated uniformly, except for the first and last quantized values which cover the overload areas. As shown at the top of Fig. 2.8, its resources in the tail areas of the PDF are not fully utilized because low probability density or weight causes them to carry too little MSQE contribution. Similarly, its resources in the head area are over utilized because high probability density or weight causes them to carry too much MSQE contribution. To reduce the overall MSQE, those mis-allocated resources need to be re-distribute in such a way that the MSE produced by individual pieces of resource are equalized. This is shown at the bottom of Fig. 2.8.

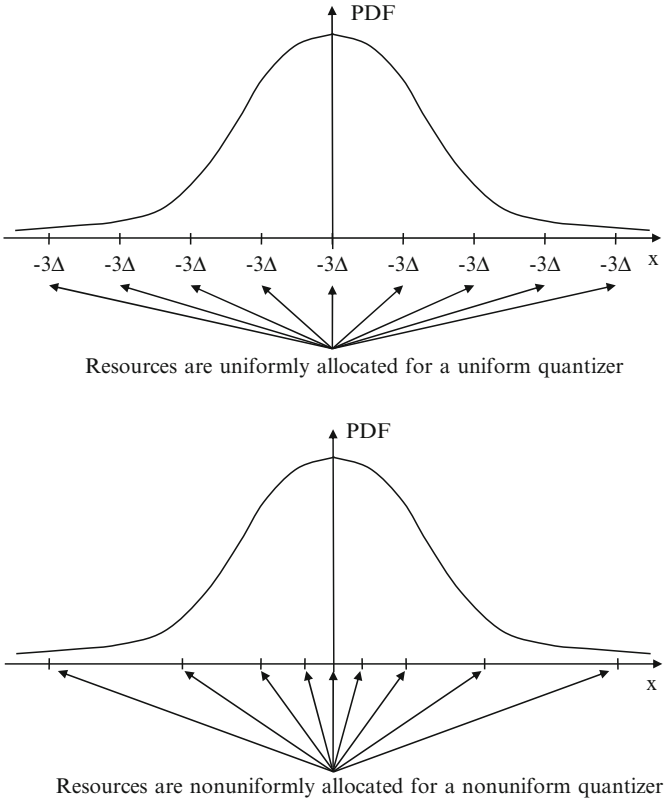

Fig. 2.8 Quantization resources are under-utilized by the uniform quantizer (*top*) in the tail areas and over-utilized in the head area of the PDF. These resources are re-distributed in the nonuniform quantizer (*bottom*) so that individual pieces of resources carry the same amount of MSQE contribution, leading to smaller MSQE

The above two considerations indicates that the MSQE can be reduced by assigning the size of decision intervals inversely proportional to the probability density. The consequence of this strategy is that the more densely distributed the PDF is, the more densely placed the decision intervals can be, thus the smaller the MSQE becomes.

One approach to nonuniform quantizer design is to post it as an optimization problem: finding the quantization intervals and quantized values that minimizes the MSQE. This leads to the Lloyd-Max algorithm. Another approach is to transform the source signal through a nonlinear function in such a way that the transformed signal has a PDF that is almost uniform, then a uniform quantizer may be used to deliver improved performance. This leads to companding.

2.4.1 Optimal Quantization and Lloyd-Max Algorithm

Given a PDF $p(x)$ and a number of decision intervals M, one approach to the design of a nonuniform quantizer is to find the set of decision boundaries $\{b_q\}_0^M$ and quantized values $\{\hat{x}_q\}_1^M$ such that the MSQE in (2.10) is minimized. Towards the solution of this optimization problem, let us first consider the following partial derivative

$$\begin{aligned}\frac{\partial \sigma_q^2}{\partial \hat{x}_q} &= 2\int_{b_{q-1}}^{b_q} (\hat{x}_q - x)p(x)dx \\ &= 2\hat{x}_q \int_{b_{q-1}}^{b_q} p(x)\mathrm{d}x - 2\int_{b_{q-1}}^{b_q} xp(x)\mathrm{d}x. \end{aligned} \tag{2.49}$$

Setting it to zero, we have

$$\hat{x}_q = \frac{\int_{b_{q-1}}^{b_q} xp(x)\mathrm{d}x}{\int_{b_{q-1}}^{b_q} p(x)\mathrm{d}x}, \tag{2.50}$$

which indicates that the quantized value for each decision interval is the centroid of the probability mass in the interval.

Let us now consider another partial derivative

$$\frac{\partial \sigma_q^2}{\partial b_q} = (\hat{x}_q - b_q)^2 p(b_q) - (\hat{x}_{q+1} - b_q)^2 p(b_q) \tag{2.51}$$

Setting it to zero, we have

$$b_q = \frac{1}{2}(\hat{x}_q + \hat{x}_{q+1}), \tag{2.52}$$

which indicates that the decision boundary is simply the midpoint of the neighboring quantized values.

Solving (2.50) and (2.52) would give us the optimal set of decision boundaries $\{b_q\}_0^M$ and quantized values $\{\hat{x}_q\}_1^M$ that minimizes σ_q^2. Unfortunately, to solve (2.50) for $\hat{x}_q$ we need b_{q-1} and b_q, but to solve (2.52) for b_q we need $\hat{x}_q$ and $\hat{x}_{q+1}$. The problem is a little difficult.

2.4.1.1 Uniform Quantizer as a Special Case

Let us consider a simple case where the probability distribution is uniform as given in (2.25). For such a distribution, (2.50) becomes

$$\hat{x}_q = \frac{b_{q-1} + b_q}{2}. \tag{2.53}$$

Incrementing q for this equation, we have

$$\hat{x}_{q+1} = \frac{b_q + b_{q+1}}{2}. \tag{2.54}$$

Dropping (2.53) and (2.54) into (2.52), we have

$$4b_q = b_{q-1} + b_q + b_q + b_{q+1}, \tag{2.55}$$

which leads us to

$$b_{q+1} - b_q = b_q - b_{q-1}. \tag{2.56}$$

Let us denote

$$b_q - b_{q-1} = \Delta, \tag{2.57}$$

plugging it into (2.56), we have

$$b_{q+1} - b_q = \Delta. \tag{2.58}$$

Therefore, we can conclude by induction on q that all decision boundaries are uniformly spaced.

For quantized values, let us subtract (2.53) from (2.54) to give

$$\hat{x}_{q+1} - \hat{x}_q = \frac{b_{q+1} - b_q + b_q - b_{q-1}}{2}. \tag{2.59}$$

Plugging in (2.57) and (2.58), we have

$$\hat{x}_{q+1} - \hat{x}_q = \Delta, \tag{2.60}$$

which indicates that the quantized values are also uniformly spaced. Therefore, uniform quantizer is optimal for uniform distribution.

2.4.1.2 Lloyd-Max Algorithm

Lloyd-Max algorithm is an iterative procedure for solving (2.50) and (2.52) for an arbitrary distribution, so an optimal quantizer is also referred to as **Lloyd-Max quantizer**. Note that its convergence is not proven, but only experimentally found.

Before presenting the algorithm, let us first note that we already know the first and last decision boundaries:

$$b_0 = X_{\min} \text{ and } b_M = X_{\max}. \tag{2.61}$$

For unbounded inputs, we may set $X_{\min} = -\infty$ and/or $X_{\max} = \infty$. Also, we rearrange (2.52) into

$$\hat{x}_{q+1} = 2b_q - \hat{x}_q, \tag{2.62}$$

The algorithm involves the following iterative steps:

1. Make a guess for $\hat{x}_1$.
2. Let $q = 1$.
3. Plugging $\hat{x}_q$ and b_{q-1} into (2.50) to solve for b_q. This may be done by integrating the two integrals in (2.50) forward from b_{q-1} until the equation holds.
4. Plugging $\hat{x}_q$ and b_q into (2.62) to get a new $\hat{x}_{q+1}$.
5. Let $q = q + 1$.
6. Go back to step 3 unless $q = M$.
7. When $q = M$, calculate

$$\theta = \hat{x}_M - \frac{\int_{b_{M-1}}^{b_M} xp(x)\mathrm{d}x}{\int_{b_{M-1}}^{b_M} p(x)\mathrm{d}x} \tag{2.63}$$

8. Stop if

$$|\theta| < \text{predetermined threshold.} \tag{2.64}$$

9. Decrease $\hat{x}_1$ if $\theta > 0$ and increase $\hat{x}_1$ otherwise.
10. Go back to step 2.

A little explanation is in order for (2.63). The iterative procedure provides us with an $\hat{x}_M$ upon entering step 7, which is used as the first term to the right of (2.63). On the other hand, since we know b_M from (2.61), we can use it with b_{M-1} provided by the procedure to obtain another estimate of $\hat{x}_M$ using (2.50). This is given as the second term on the right side of (2.63). The two estimates of the same $\hat{x}_M$ should be equal if equations (2.50) and (2.52) are solved. Therefore, we stop the iteration at step 8 when the absolute value of their difference is smaller than some predetermined threshold.

The adjustment procedure for $\hat{x}_1$ at step 9 can also be easily explained. The iterative procedure is started with a guess for $\hat{x}_1$ at step 1. Based on this guess, a whole set of decision boundaries $\{b_q\}_0^M$ and quantized values $\{\hat{x}_q\}_1^M$ are obtained from step 2 through step 8. If the guess is off, the whole set derived from it is off. In particular, if the guess is too large, the resulting $\hat{x}_M$ will be too large. This will cause $\theta > 0$, so $\hat{x}_1$ needs to be reduced; and vice versa.

2.4.1.3 Performance Gain

Figure 2.9 shows optimal SNR achieved by Lloyd-Max algorithm for uniform, Gaussian, Laplacian, and Gamma distributions against the number of bits per sample [33]. Since the uniform quantizer is optimal for uniform distribution, its optimal SNR curve in Fig. 2.9 is the same as in Fig. 2.6, thus can serve as the reference. Notice that the optimal SNR curves for the other distributions are closer to this curve in Fig. 2.9 than in Fig. 2.6. This indicates that, for a given number of bits per sample, optimal nonuniform quantization achieves better SNR than optimal uniform quantization.

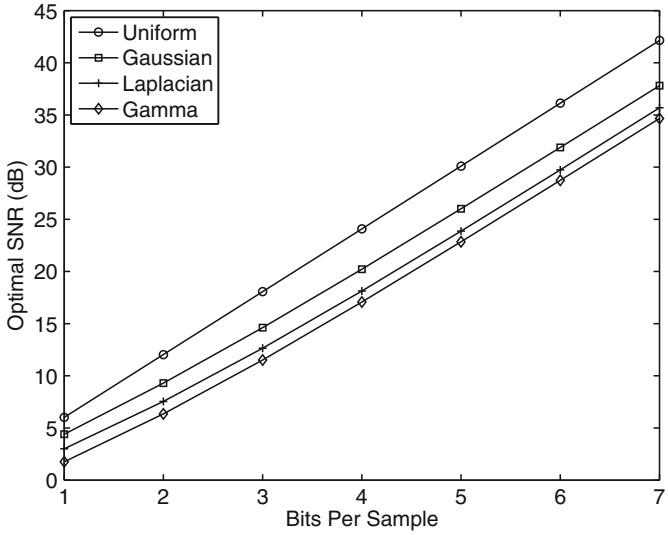

Fig. 2.9 Optimal SNR versus bits per sample achieved by Lloyd-Max algorithm for uniform, Gaussian, Laplacian, and Gamma distributions

Apparently, the optimal SNR curves in Fig. 2.9 also fit straight lines well, so can be approximated by the same equation given in (2.43) with improved slope b and intercept a. The improved performance of nonuniform quantization results in better fitting to a straight line than those in Fig. 2.6.

Similar to uniform quantization in Fig. 2.6, both the slope b and the intercept a decrease as the *peakedness* or *kurtosis* of the PDF increases in the order of uniform, Gaussian, Laplacian, and Gamma, indicating that the overall performance of a Lloyd–Max quantizer is inversely related to PDF kurtosis. Compared with the uniform distribution, all other distributions have reduced slopes b, indicating that their performance relative to the uniform distribution becomes worse as the bit rate increases. However, the degradations of both a and b are less conspicuous than those in Fig. 2.6.

In order to compare the performance between Lloyd-Max quantizer and uniform quantizer, Fig. 2.10 shows optimal SNR gain of Lloyd-Max quantizer over uniform quantizer for uniform, Gaussian, Laplacian, and Gamma distributions:

$$\text{Optimal SNR Gain} = \text{SNR}_{\text{Nonuniform}} - \text{SNR}_{\text{Uniform}},$$

where $\text{SNR}_{\text{Nonuniform}}$ is taken from Fig. 2.9 and $\text{SNR}_{\text{Uniform}}$ from Fig. 2.6. Since the Lloyd-Max quantizer for uniform distribution is a uniform quantizer, the optimal SNR gain is zero for uniform distribution. It is obvious that the optimal SNR gain is more profound when the distribution is more peaked or is of larger kurtosis.

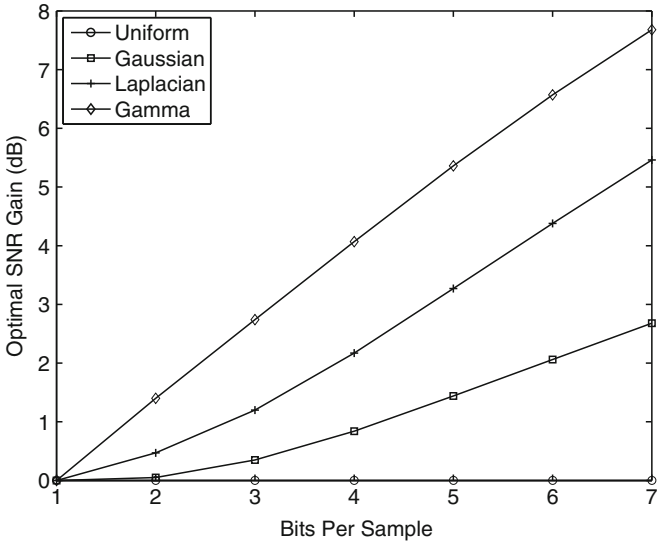

Fig. 2.10 Optimal SNR gain of Lloyd-Max quantizer over uniform quantizer for uniform, Gaussian, Laplacian, and Gamma distributions

2.4.2 Companding

Finding the whole set of decision boundaries $\{b_q\}_0^M$ and quantized values $\{\hat{x}_q\}_1^M$ for an optimal nonuniform quantizer using Lloyd-Max algorithm usually involves a large number of iterations, hence may be computationally intensive, especially for a large M. The storage requirement for these decision boundaries and quantization values may also become excessive, especially for the decoder. Companding is an alternative.

Companding is motivated by the observation that a uniform quantizer is simple and effective for a matching uniformly distributed source signal. For a nonuniformly distributed source signal, one could use a nonlinear function $f(x)$ to convert it into another one with a PDF similar to a uniform distribution. Then the simple and effective uniform quantizer could be used. After the quantization indexes are transmitted to and subsequently received by the decoders, they are first inversely quantized to reconstruct the uniformly quantized values and then the inverse function $f^{-1}(x)$ is applied to produce the final quantized values. This process is illustrated in Fig. 2.11.

The nonlinear function in Fig. 2.11 is called a **compressor** because it usually has a shape similar to that shown in Fig. 2.12 that stretches the source signal when its sample value is small and compresses it otherwise. This shape of compression is to match the typical shape of PDF, such as Gaussian and Laplacian, which has large probability density for small absolute sample values and tails off towards large absolute sample values, in order to make the converted signal have a PDF similar to a uniform distribution.

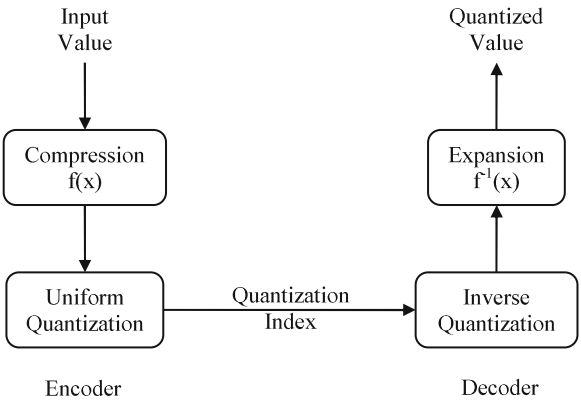

Fig. 2.11 The source sample value is first converted by the compressor into another one with a PDF similar to a uniform distribution. It is then quantized by a uniform quantizer and the quantization index is transmitted to the decoder. After inverse quantization at the decoder, the uniformly quantized value is converted by the expander to produce the final quantized value

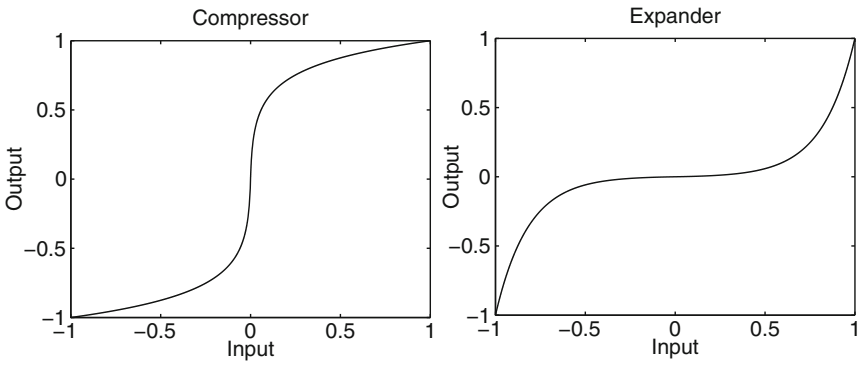

Fig. 2.12 μ-Law companding deployed in North American and Japanese telecommunication systems

The inverse function is called an **expander** because the inverse of compression is expansion. After the compression-expansion, hence "**companding**", the effective decision boundaries when viewed from the expander output is nonuniform, so the overall effect is nonuniform quantization.

When companding is actually used in speech and audio applications, additional considerations are given to the perceptual properties of the human ear. Since the perception of loudness by the human ear may be considered as logarithmic, logarithmic companding is widely used.

2.4.2.1 Speech Processing

In speech processing, the μ-law companding, deployed in North American and Japanese telecommunication systems, has a compression function given by [33]

$$y = f(x) = \text{sign}(x)\frac{\ln(1+\mu|x|)}{\ln(1+\mu)}, \quad -1 \le x \le 1; \tag{2.65}$$

where $\mu = 256$ and x is the normalized sample value to be compounded and is limited to 13 magnitude bits. Its corresponding expanding function is

$$x = f^{-1}(y) = \text{sign}(y)\frac{(1+\mu)^{|y|}-1}{\mu}, \quad -1 \le y \le 1. \tag{2.66}$$

Both functions are plotted in Fig. 2.12.

A similar companding, called A-law companding, is deployed in Europe, whose compression function is

$$y = f(x) = \frac{\text{sign}(x)}{1+\ln(A)} \begin{cases} A|x|, & 0 \le |x| \le \frac{1}{A}; \\ 1+\ln(A|x|), & \frac{1}{A} < |x| \le 1; \end{cases} \tag{2.67}$$

where $A = 87.7$ and the normalized sample value x is limited to 12 magnitude bits. Its corresponding expanding function is

$$x = f^{-1}(y) = \text{sign}(y) \begin{cases} \frac{1+\ln(A)}{A}|y|, & 0 \le |y| \le \frac{1}{1+\ln(A)}; \\ \frac{e^{|y|(1+\ln(A))-1}}{A+A\ln(A)}, & \frac{1}{1+\ln(A)} < |y| \le 1. \end{cases} \tag{2.68}$$

It is usually very difficult to implement both the logarithmic and exponential functions used in the companding schemes above, especially on embedded microprocessors with limited resources. Many such processors even do not have a floating point unit. Therefore, the companding functions are usually implemented using piece-wise linear approximation. This is adequate due to the fairly low requirement for speech quality in telephonic systems,

2.4.2.2 Audio Coding

Companding is not as widely used in audio coding as in speech processing, partly due to higher quality requirement and wider dynamic range which renders implementation more difficult. However, MPEG 1&2 Layer III [55, 56] and MPEG 2&4 AAC [59, 60] use the following exponential compression function to quantize MDCT coefficients:

$$y = f(x) = \text{sign}(x)|x|^{3/4}, \tag{2.69}$$

which may be considered as an approximation to the logarithmic function. The allowed compressed dynamic range is $-8191 \le y \le 8191$. The corresponding expanding function is obviously

$$x = f^{-1}(y) = \text{sign}(y)|y|^{4/3}. \tag{2.70}$$

The implementation cost for the above exponential function is a remarkable issue in decoder development. Piece-wise linear approximation may lead to degradation in audio quality, hence may be unacceptable for high fidelity application. Another alternative is to store the exponential function as a quantization table. This amounts to $13 \times 3 = 39\,\text{KB}$ if each of the 2^{13} entries in the table are stored using 24 bits.

The most widely used companding in audio coding is the companding of quantization step sizes of uniform quantizers. Since quantization step sizes are needed in the inverse quantization process in the decoder, they need to be packed into the bit stream and transmitted to the decoder. Transmitting these step sizes with arbitrary resolution is out of the question, so it is necessary that they be quantized.

The perceived loudness of quantization noise is usually considered as logarithmically proportional to the quantization noise power, or linearly proportional to the quantization noise power in decibel. Due to (2.28), this means the perceived loudness is linearly proportional to the quantization step size in decibel. Therefore, almost all audio coding algorithms use logarithmic companding to quantize quantization step sizes:

$$\delta = f(\Delta) = \log_2(\Delta), \tag{2.71}$$

where Δ is the step size of a uniform quantizer. The corresponding expander is obviously

$$\Delta = f^{-1}(\delta) = 2^{\delta}. \tag{2.72}$$

Another motivation for logarithmic companding is to cope with the wide dynamic range of audio signals, which may amount to more than 24 bits per sample.

http://www.springer.com/978-1-4419-1753-9

Audio Coding
Theory and Applications
You, Y.
2010, XVI, 344 p., Hardcover
ISBN: 978-1-4419-1753-9