

2

Introduction to Speech-coding Techniques

2.1 A primer on digital signal processing

2.1.1 Introduction

At the beginning of the 20th century, all devices performing some form of signal processing (recording, playback, voice or video transmission) were still using analogue technology (i.e., media information was represented as a continuously variable physical signal). It could be the depth of a groove on a disk, the current flowing through a variable resistance microphone, or the voltage between the wires of a transmission line. In the 1960s, the PCM (pulse code modulation) of audio began to be used in telecom switching equipment. Since 1980 the spectacular performance advances of computers and processors led to an ever-increasing use of digital signal processing.

Today speech signals sampled at 8 kHz can be correctly encoded and transmitted with an average of 1 bit per sample (8 kbit/s) and generic audio signals with 2 bits per sample. Speech coders leverage the redundancies within the speech signal and the properties (the weaknesses) of human ears to reduce the bitrate. Speech coding can be very efficient because speech signals have an underlying vocal tract production model; this is not the case for most audio signals, such as music.

This chapter will first explain in more detail what a 'digital' signal is and how it can be obtained from a fundamentally analogue physical input that is a continuously variable function of time. We will introduce the concepts of sampling, quantization, and transmitted bandwidth. These concepts will be used to understand the basic speech-coding schemes used today for telephony networks: the ITU-T A-law and μ -law encodings at 64 kbits per second (G.711).

At this point the reader may wonder why there is such a rush toward fully digital signal processing. There are multiple reasons, but the key argument is that all the signal transformations that previously required discrete components (such as bandpass filters, delay lines, etc.) can now be replaced by pure mathematical algorithms applied to the digitized signal. With the power of today's processors, this results in a spectacular gain in the size of digital-processing equipment and the range of operations that can be applied to a given signal (e.g., acoustic echo cancellation really becomes possible only with digital processing). In order to understand the power of fully digital signal processing, we will introduce the 'Z transform', the fundamental tool behind most signal-processing algorithms.

We will then introduce the key algorithms used by most voice coders:

- Adaptive quantizers.
- Differential (and predictive ...) quantization.
- Linear prediction of signal.
- Long-term prediction for speech signal.
- Vector quantization.
- Entropy coding.

There are two major classes of voice coders, which use the fundamental speech analysis tools in different ways:

- Waveform coders.
- Analysis by synthesis voice coders.

After describing the generic implementation of each category, the detailed properties of the most well known standardized voice coders will be presented.

We will conclude this chapter by a presentation of speech quality assessment methods.

2.1.2 Sampling and quantization

Analog-to-digital conversion is the process used to represent an infinite precision quantity, originally in a time-varying analog form (such as an electrical signal produced by a microphone), by a finite set of numbers at a fixed sample rate, each sample representing the state of the original quantity at a specific instant. Analog-to-digital conversion is mandatory in order to allow computer-based signal analysis, since computers can only process numbers.

Analog-to-digital conversion is characterized by:

- The rate of **sampling** (i.e., how often the continuously variable quantity is measured).
- The **quantization** method (i.e., the number of discrete values that are used to express the measurement (typically a certain number of bits), and how these values are distributed

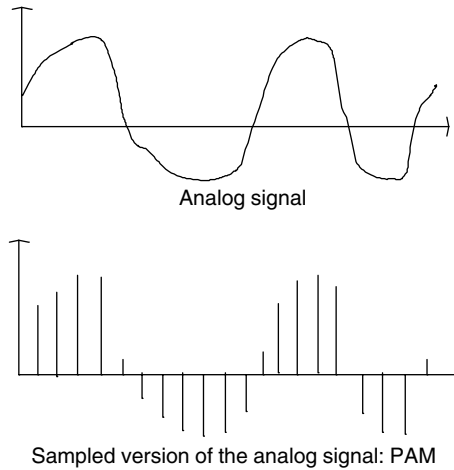


Figure 2.1 Pulse amplitude modulation.

(linearly on the measurement scale, or with certain portions of the measurement scale using a more precise scale than others)).

Mathematically, the sampling process can be defined as the result of the multiplication of an infinite periodical pulse train of amplitude 1 (with a period corresponding to the sampling period), by the original **continuous-time** signal to be sampled. This leads to the **PAM (pulse amplitude modulation) discrete time** representation of the signal (Figure 2.1).

From the PAM signal, it is possible to regenerate a continuous time signal. This is required each time the result of the signal-processing algorithm needs to be played back. For instance, a simple **discrete-to-continuous (D/C)** converter could generate linear ramps linking each pulse value, then filter out the high frequencies generated by the discontinuities.

Analog-to-digital conversion loses some information contained in the original signal, which can never be recovered (this is obvious in Figure 2.2). It is very important to choose the sample rate and the quantization scale appropriately, as this directly influences the quality of the output of the signal-processing algorithm [A2, B1, B2].

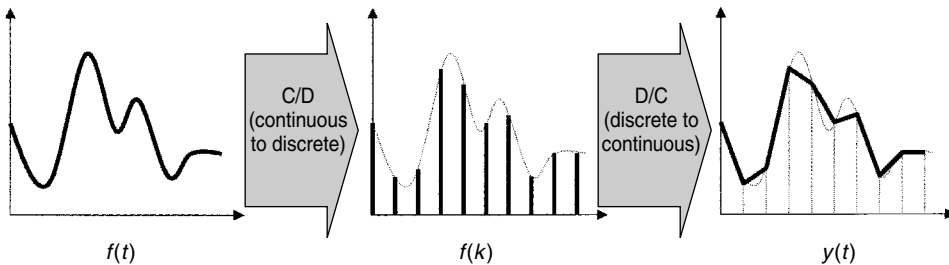


Figure 2.2 Reconstruction of a continuous signal from a discrete signal.

2.1.3 The sampling theorem

The **sampling theorem** states that in order to process a continuous time signal with frequency components comprised between 0 and F_{\max} , the sampling rate should be at least $2 * F_{\max}$. Intuitively, it can be understood by looking at the quantization of a pure sinusoid. In Figure 2.3 the original signal of frequency 1.1 is sampled at frequency 1. The resulting PAM signal is identical to the sampling result at frequency 1 of a signal at frequency 0.1. This is the **aliasing** phenomenon.

In fact if T is the sampling period (**radial frequency** $\Omega_r = 2\pi/T$):

- All sinusoids of frequency $\omega_r + m\Omega_r$ will have the same PAM representation as the sinusoid of frequency ω_r , since $\cos((\omega_r + m(2\pi/T))t)$ is sampled as

$$\cos((\omega_r + m(2\pi/T)kT) = \cos((\omega_r kT + mk2\pi) = \cos(\omega_r kT)$$

- The sinusoids of frequencies $\Omega_r/2 + \omega_r$ and $\Omega_r/2 - \omega_r$ have the same PAM representation because

$$\begin{aligned} \cos((\Omega_r/2 \pm \omega_r)kT) &= \cos(\pi k \pm \omega_r kT) = \cos(\pi k) \cos(\omega_r kT) \mp \sin(\pi k) \sin(\omega_r kT) \\ &= \cos(\pi k) \cos(\omega_r kT) \end{aligned}$$

This is illustrated on Figure 2.4.

The conclusion is that there is one-to-one mapping between a sinusoid and its PAM representation sampled at frequency Ω only if sinusoids are restricted to the $[0, \Omega/2]$ range. This also applies to any signal composed of mixed sinusoids: the signal should not

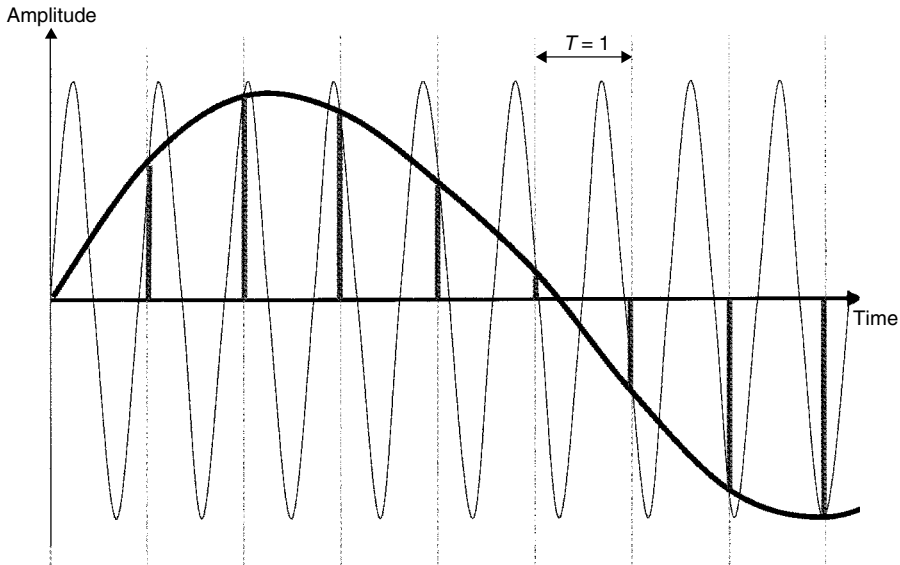


Figure 2.3 Aliasing of frequency 1.1, sampled at frequency 1, wrapped into frequency 0.1.

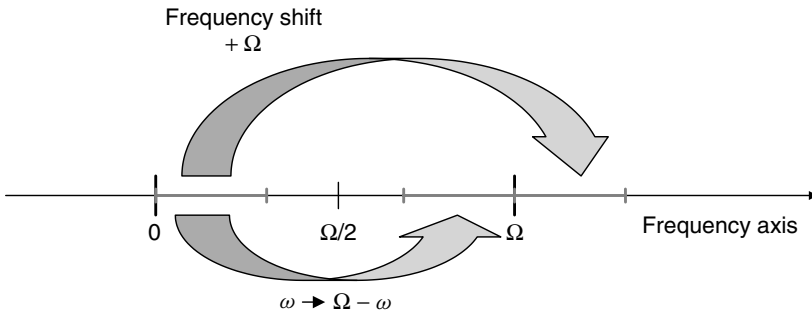


Figure 2.4 The different types of frequency aliasing.

have any frequency component outside the $[0, \Omega/2]$ range. This is known as the **Nyquist theorem**, and $\Omega = 2\omega$ is called the **Nyquist rate** (the minimal required sampling rate for a signal with frequency components in the $[0, \omega]$ range).

The Nyquist (or Shannon) theorem also proves that it is possible to exactly recover the original continuous signal from the PAM representation, if the sampling rate is at or above the Nyquist rate. It can be shown that the frequency spectrum (Fourier transform) of a PAM signal with sampling frequency F_s is similar to the frequency spectrum of the original signal, repeated periodically with a period of F_s and with a scaling factor.

From Figure 2.5 it appears that the original signal spectrum can be recovered by applying an ideal low-pass filter with a cutting frequency of $F_s/2$ to the PAM signal. The unique condition to correctly recover the original analog spectrum is that there is no frequency wrapping in the infinite PAM spectrum. The only way to achieve this

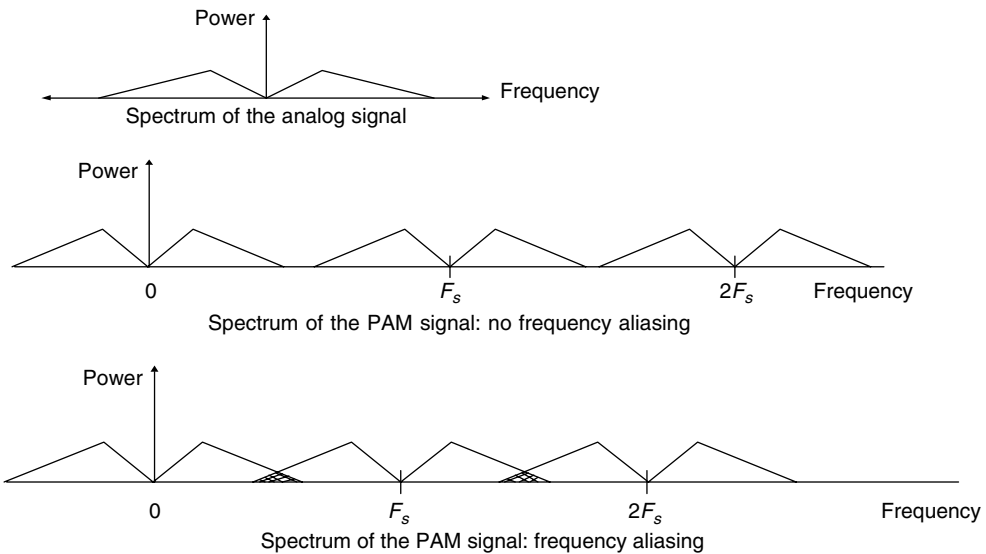


Figure 2.5 The Nyquist rate and frequency wrapping.

is for the bandwidth of the original analog signal to be strictly limited to the frequency band $[0, F_s/2]$. Figure 2.5 shows an ideal situation and a frequency-wrapping situation; in the case of frequency wrapping, the recovered signal is spoiled by frequency aliasing.

The spectrum of real physical signals (such as the electrical signal generated by a microphone) do not have a well-defined frequency limit. Therefore, before the sampling process, it is necessary to cut off any frequency component beyond the Nyquist frequency by using an ‘anti-aliasing’ analog filter. In order to avoid this discrete component (it is not obvious to approximate an ideal low-pass filter with analog technology), modern oversampled noise-shaping analog-to-digital converters (also called sigma delta coders) use a very high-sampling frequency (the input signal is supposed not to have any very high-frequency component) but internally apply digital decimation (subsampling) filters which perform the anti-aliasing task before the sampling rate is reduced.

In the digital-to-analog chain, the reconstruction filter is responsible for transforming the discrete digital signal into a continuous time signal.

The value of the sampling frequency not only determines the transmitted signal bandwidth but also impacts the amount of information to be transmitted: for instance, wide-band, high-quality audio signals must be sampled at high frequencies, but this generates far more information than the regular 8,000-Hz sampling frequency used in the telephone network.

2.1.4 Quantization

With the sampling process discussed in the previous paragraph, we are not yet in the digital world. The PAM signal is essentially an analog signal because the amplitude of each pulse is still a continuous value that we have not attempted to measure with a number. In fact we have lost only part of the information so far (the part of the sampled signal above one-half of the sampling frequency). We will lose even more information when we measure the amplitude of each pulse.

Let’s imagine that a folding rule is used to measure the amplitude of the PAM signal. Depending of the graduation or precision of the scale, the number that represents the PAM signal can be more or less precise ... but it will never be exact. The PAM signal can be represented by the digital signal with pulses corresponding to the measured values, plus a PAM signal with pulses representing the errors of the quantization process. The signal encoding in which each analog sample of the PAM signal is encoded in a binary code word is called a **PCM (pulse code modulation)** representation of the signal. The analog-to-digital conversion is called quantization.

With a more precise quantization process, we minimize the amplitude of the noise, but we cannot avoid introducing some noise in the quantization process (**quantization noise**). Once quantization noise is introduced in a speech or audio transmission chain, there is no chance to improve the quality by any means. This has important consequences: for instance, it is impossible to design a digital echo canceler working on a PCM signal with a signal-to-echo ratio above the PCM signal’s signal-to-noise ratio.

Therefore there are two sources of loss of information when preparing a signal for digital processing:

- The loss of high-frequency components.
- Quantization noise.

The two must be properly balanced in any analog-to-digital (A/D) converter as both influence the volume of information that is generated: it would be meaningless to encode with a 24-bit accuracy a speech signal which is intentionally frequency-limited to the 300–3,400-Hz band; the limitation in frequency is much more perceptible than the ‘gain’ in precision brought by the 24 bits of the A/D chain.

If uniform quantization is applied (‘uniform’ means that the scale of our ‘folding rule’ is linear) the power of the quantization noise can be easily derived. All the step sizes of the quantizer have the same width D ; therefore, the error amplitude spans between $-D/2$ to $+D/2$ and it can be shown [B1] that the power of this error is:

$$E^2 = \frac{D^2}{12}$$

For a uniform quantizer using N bits (N is generally a power of 2) the maximum **signal-to-noise ratio (SNR)** achievable in decibels is given by:

$$SNR(dB) = 6.02N - 1.73$$

For example, a CD player uses a 16-bit linear quantizer and the maximum achievable SNR is 94.6 dB. This impressive figure hides some problems: the maximum value is obtained for a signal having the maximum amplitude (e.g., a sinusoid going from $-32,768$ to $+32,767$). In fact, the SNR is directly proportional to the power of the signal: the curve representing the SNR against the input power of the signal is a straight line. If the power of the input signal is reduced by 10 dB, the SNR is also reduced by 10 dB. For very low-power sequences of music, some experts (golden ears) can be disturbed by the granularity of the sound reproduced by a CD player and prefer the sound of an old vinyl disk.

Because of this problem, the telecom industry generally uses quantizers with a constant SNR ratio regardless of the power of the input signal. This requires nonlinear quantizers (Figure 2.6).

As previously stated, the sampling frequency and the number of bits used in the quantization process both impact the quality of the digitized signal and the resulting information rate: some compromises need to be made. Table 2.1 [A2] gives an overview of the most common set of parameters for transmitting speech and audio signals (assuming a linear quantizer).

Even a relatively low-quality telephone conversation results in a bitrate around 100 kbit/s after A/D conversion. This explains why so much work has been done to reduce this bitrate while preserving the original quality of the digitized signal. Even the well-known A-law or μ -law PCM G.711 coding schemes at 64 kbit/s, used worldwide in all digital-switching machines and in many digital transmission systems, can be viewed as a speech coder.

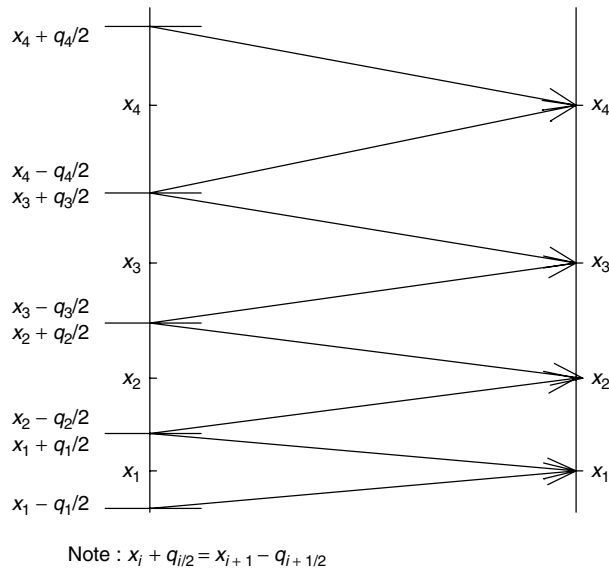


Figure 2.6 Example of a nonlinear quantizer. Any value belonging to $[x_i - q_i/2, x_i + q_i/2]$ is quantized and converted in x_i . The noise value spans in $[-q_i/2, +q_i/2]$.

Table 2.1 Common settings for analog-to-digital conversion of audio signals

Type	Transmitted bandwidth (Hz)	Sampling frequency (kHz)	Number of bits in A/D and D/A converters	Bitrate in kbit/s	Main applications
Telephone speech	300–3,400	8	12 or 13	96 or 104	PSTN, ISDN networks, digital cellular
Wide-band speech (and audio)	50–7,000	16	14 or 15	224 or 240	Video and audio conferencing, FM radio
High-quality speech and audio	30–15,000	32	16	512	Digital sound for analog TV (NICAM)
	20–20,000	44.1	16	706	audio CD player
	10–22,000	48	Up to 24	1,152	Professional audio

2.1.5 ITU G.711 A-law or μ -law, a basic coder at 64 kbit/s

A linear quantizer is not usually optimal. It can be mathematically demonstrated that if the **probability density function (PDF)** of the input signal is known, an optimal

quantizer [B1, B2] can be computed which leads to a maximal SNR for this signal. The resulting quantizer is not linear for most signals. Of course, the main issue is to know the PDF of a given signal; for random speech and audio signal, this is a very difficult task as it may depend on multiple factors (language, speaker, loudness, etc.).

Another approach to finding an optimal quantizer is to look for a quantizer scale which yields an SNR independent of the level of the signal. It can be shown that this requires a logarithmic scale: the step size of the quantizer is doubled each time the input level is doubled. This process is called **companding** (*compress and expanding*): compared with the PAM signal, the digital PCM representation of the signal is ‘compressed’ by the logarithmic scale, and it is necessary to expand each PCM sample to obtain the PAM signal back (with quantization noise).

The ITU telephony experts also noted that the 12–13-bit precision of the linear quantizers discussed above were only useful for very weak signals, and such a precision was not necessary at higher levels. Therefore, a step size equivalent to the step size of a 12-bit linear quantizer would be needed only at the beginning of the logarithmic scale.

The ITU G.711 logarithmic voice coder uses the concept of companding, with a quantization scale for weak signals equivalent to a 12-bit linear scale. Two scales were defined, the A-law (used in Europe and over all international links) and the μ -law (used in North America and Japan). The two laws rely on the same approximation of a logarithmic curve: using segments with a slope increasing by a factor of 2, but the exact length of segments and slopes differ between the A-law and the μ -law. This results in subtle differences between the A-law and the μ -law: the A-law provides a greater dynamic range than the μ -law, but the μ -law provides a slightly better SNR than the A-law for low-level signals (in practice, the least significant bit is often stolen for signaling purposes in μ -law countries, which degrades the theoretical SNR).

G.711 processes a digital, linear, quantized signal (generally, A/D converters are linear) on 12 bits (sign + amplitude; very often A/D outputs are 2’s complements that require to be converted to the sign + amplitude format). From each 12-bit sample, the G.711 converter will output a 8-bit code represented in Figure 2.7:

In Figure 2.7, S is the sign bit, E2E1E0 is the exponent value, and M3M2M1M0 is the mantissa value. A-law or μ -law encoding can be viewed as a floating point representation of the speech samples.

The digital-encoding procedure of the G.711 A-law is represented in Table 2.2 [A1]. The X, Y, Z, T values are come from the code and are transmitted directly as M3, M2, M1, M0 (the mantissa). Note that the dashed area corresponds to quantization noise which is clearly proportional to the input level (constant SNR ratio).

Figure 2.8 represents the seven-segment A-law characteristic (note that, even though we have eight segments approximating the log curve, segments 0 and 1 use the same slope).

On the receiving side, the 8-bit A-law code is expanded into 13 bits (sign + amplitude), representing the linear quantization value. In order to minimize decoded quantization noise, an extra bit is set to ‘1’ for the first two segments (see Table 2.3)



Figure 2.7 The G.711 8-bit code.

Table 2.2 Amplitude encoding in G.711

Segment number (sign bit omitted)			Amplitude coded with 11 bits (sign + amplitude, sign bit omitted)										
			B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
0	0	0	0	0	0	0	0	0	0	X	Y	Z	T
0	0	1	0	0	0	0	0	0	1	X	Y	Z	T
0	1	0	0	0	0	0	0	1	X	Y	Z	T	N
0	1	1	0	0	0	0	1	X	Y	Z	T	N	N
1	0	0	0	0	0	1	X	Y	Z	T	N	N	N
1	0	1	0	0	1	X	Y	Z	T	N	N	N	N
1	1	0	0	1	X	Y	Z	T	N	N	N	N	N
1	1	1	1	X	Y	Z	T	N	N	N	N	N	N

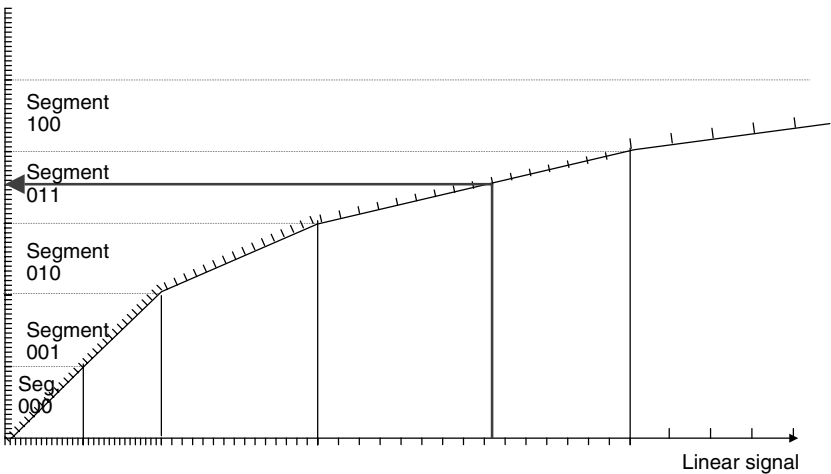


Figure 2.8 Logarithmic approximation used by G.711 A-law.

Table 2.3 Decoding table for G.711 8-bit codes

Exponent	Sign bit	Decoded amplitude using $\frac{1}{2}$ quantization steps (12 bits)											
		B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0	B - 1
0	S	0	0	0	0	0	0	0	M3	M2	M1	M0	1
1	S	0	0	0	0	0	0	1	M3	M2	M1	M0	1
2	S	0	0	0	0	0	1	M3	M2	M1	M0	1	0
3	S	0	0	0	0	1	M3	M2	M1	M0	1	0	0
4	S	0	0	0	1	M3	M2	M1	M0	1	0	0	0
5	S	0	0	1	M3	M2	M1	M0	1	0	0	0	0
6	S	0	1	M3	M2	M1	M0	1	0	0	0	0	0
7	S	1	M3	M2	M1	M0	1	0	0	0	0	0	0

Clearly, the gain of using G.711 is not in quality but in the resulting bitrate: G.711 encodes a 12-bit, linearly quantized signal into 8 bits. If the sampling frequency is 8 kHz (the standard for telecom networks), the resulting bitrate is 64 kbit/s.

The only drawback of G.711 is to reduce the SNR for high-powered input signals (see Figure 2.9) compared with linear quantization. However, experience shows that the overall perceived (and subjective) quality is not dramatically impacted by the reduction of the SNR at high levels (listeners perceive some signal-independent noise).

In fact, most of the information is lost during initial sampling and 12-bit linear quantization. If listeners compare a CD quality sample recorded at a sample rate of 44.1 kHz at 16 bits, the critical loss of perceived quality occurs after subsampling at 8 kHz on 16 bits: there is a net loss of clarity and introduction of extra loudness, especially for the female voice. The reduction of quantization from 16 to 12 bits also introduces some granular noise. The final A- or μ -law logarithmic compression is relatively unimportant in this 'degradation' chain.

The A- or (μ)-law compression scheme is naturally a lossy compression: some noise is introduced and the input signal (on 12 bits) can never be recovered. This is true for all coders. All voice coders are designed for a given signal degradation target. The best coders for a given target are those that manage to use the smallest bitrate while still fulfilling the quality target.

Beyond the degradations mentioned above, the audio signal is low-pass-filtered (the conventional transmitted band is 300 Hz to 3,400 Hz in Europe and 200 Hz to 3,200 Hz in the US and Japan). This band limitation for the low frequencies of the speech signal throws out some essential spectral components of speech. It goes beyond the Nyquist requirements and was initially set for compatibility with analog modulation schemes for telephone multiplex links; it also takes into account the non-ideal frequency response of real filters.

Today, with the entire digital network going directly to customers' premises (ISDN, cellular, and of course VoIP), this limitation is not mandatory and becomes obsolete.

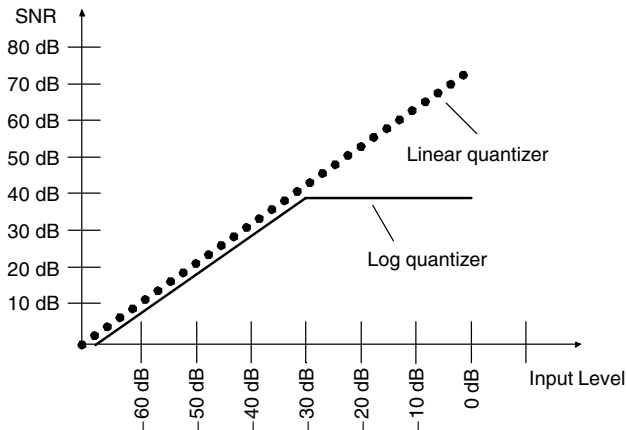


Figure 2.9 G.711 signal-to-noise ratio.

- SNR for linear quantizer: max = 74 dB (..);
- SNR for log type (A- or μ -law) quantizer: max = 38 dB (—).

The G.711 encoding process can be built very easily from off-the-shelf integrated circuits (priority encoders, etc.). G.711 encoding and decoding requires a very low processing power (hundreds of channels can be decoded in real time on a simple PC). In the early days of digital telecommunications, this was mandatory.

We will see that new coders are designed to give the same degradation for a lower bit rate:

- The required processing power increases (mainly for the coding part).
- The coding process introduces more delay (this is because coders need to look at more than one sample of the original signal before being able to produce a reduced bitrate version of the signal).

2.2 The basic tools of digital signal processing

2.2.1 Why digital technology simplifies signal processing

2.2.1.1 Common signal-processing operations

Signal-processing circuits apply a number of operations to the input signal(s):

- Sum.
- Difference.
- Multiplication (modulation of one signal by another).
- Differentiation (derivative).
- Integration.
- Frequency analysis.
- Frequency filtering.
- Delay.

It is obvious that the sum and difference operations are easy to perform with discrete time digitized signals, but they are also very easy to perform with analog systems. On the other hand, all other operations are much simpler to perform with digital systems.

The differentiation of a signal $f(t)$, for instance, typically requires an inductance or a capacitor in an analogue system, both of which are very difficult to miniaturize. But the derivative $f'(t) = \lim_{d \rightarrow 0} \frac{f(t+d) - f(t)}{d}$ can be approximated very easily by $(f(k) - f(k-1))/T$, where $f(k)$ is the discrete time digitized version of $f(t)$ with a sampling period T .

Similarly, the primitive F of a function f can be approximated on the digitized version of f summing all samples $f(k) * T$ (Figure 2.10).

All audio filters realizable using discrete components can today be emulated digitally. With the ever-increasing frequency of modern processors, even radio frequency signals are now accessible to digital signal processing.

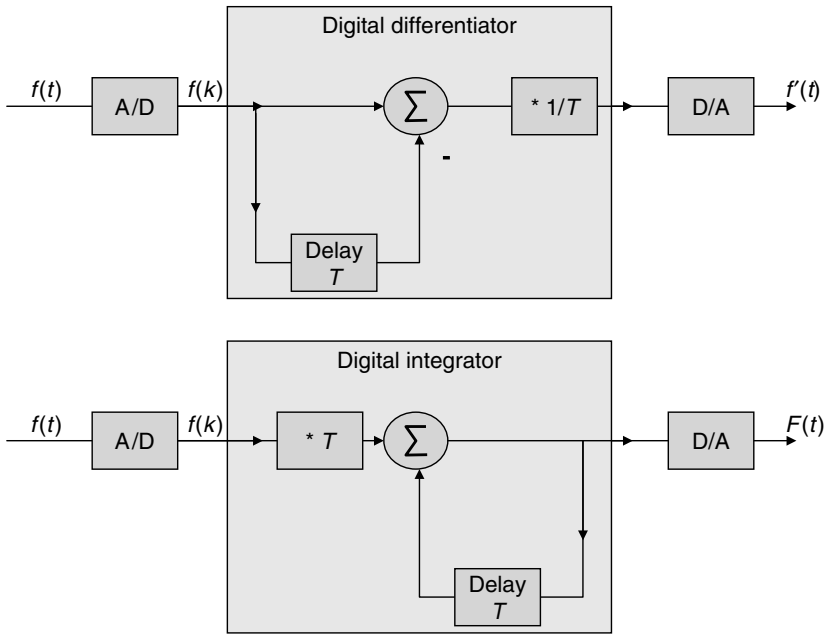


Figure 2.10 Differentiation and integration with digital filters.

The tools presented below allow engineers to synthesize digital filters that implement a desired behavior or predict the behavior of a given digital filter.

2.2.1.2 Example of an integro-differential filter

Most filters can be represented as a set of integro-differential equations between input signals and output signals. For instance, in the following circuit (Figure 2.11) the input voltage and the resulting current are linked by the following equation:

$$y''(t) + 3y'(t) + 2y(t) = f'(t)$$

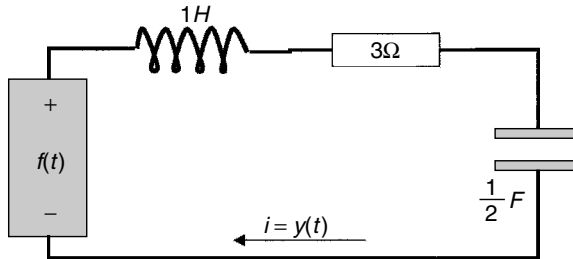


Figure 2.11 Simple circuit that can be modeled by an integro-differential equation.

or if D is the symbol of the differentiation operator:

$$(D^2 + 3D + 2)y(t) = Df(t)$$

The $D^2 + 3D + 2$ part is also called the characteristic polynomial of the system. The solutions of $x^2 + 3x + 2 = 0$ also give the value of the exponents of the pure exponential solutions of the equation when the input signal $f(t)$ is null ('zero input solution'). The reader can check that -1 and -2 are the roots of $x^2 + 3x + 2$ and that e^{-t} and e^{-2t} are solutions of the $y''(t) + 3y'(t) + 2y(t) = 0$ equation. The solutions are complex in general, but should occur as pairs of conjugates for real systems (otherwise the coefficients of the characteristic polynomial are not real), so that real solutions can be obtained by combining exponentials obtained from conjugate roots. Repeated roots r yield solutions of the form $t^{i-1}e^{rt}$ if the root is repeated i times.

Let's assume a sampling period of 1. The system equation can readily be transformed in a discrete time form (linear difference equation):

$$\frac{y[k+2] - 2y[k+1] + y[k]}{T^2} + 3\frac{y[k+1] - y[k]}{T} + 2y[k] = f[k+1] - f[k]$$

or if E denotes the 'advance operator' $E(f(k)) = f(k+1)$:

$$\left(\frac{E^2}{T^2} + \left(\frac{-2}{T^2} + \frac{3}{T}\right)E + \left(\frac{1}{T^2} - \frac{3}{T} + 2\right)\right)y[k] = (E - 1)f[k]$$

The left-hand side of this equation accepts solutions of the form $c\gamma^k$, where γ is a solution of the $\frac{x^2}{T^2} + \left(\frac{-2}{T^2} + \frac{3}{T}\right)x + \left(\frac{1}{T^2} - \frac{3}{T} + 2\right)$ polynomial. In the case of repeated roots, there are also solutions of the form $k^n\gamma^k$ (γ is generally a complex number).

2.2.2 The Z transform and the transfer function

2.2.2.1 Definition

The unilateral¹ Z transform of a discrete time function $f(k)$ is defined as $F(z) = \sum_{k=0}^{\infty} f(k)z^{-k}$. It is only defined on a certain domain of convergence of the complex variable z . The Z transform can be inverted:

$$f(k) = \frac{1}{2\pi j} \oint F(z)z^{k-1}dz$$

(the integral is performed on a closed path within the convergence domain in the complex plane).

¹ The bilateral Z transform also exists but is useful only for the analysis of non-causal systems. For the bilateral Z transform the sum starts at $-\infty$.

Table 2.4 Short extract of a Z transform table

$f(k)$	$F(z)$
$u(k)$ (step function $u(k) = 0, k < 0; u(k) = 1, k \geq 0$)	$\frac{z}{z-1}$
$ku(k)$	$\frac{z}{(z-1)^2}$
$\gamma^{k-1}u(k-1)$	$\frac{1}{z-\gamma}$
$k\gamma^k u(k)$	$\frac{\gamma z}{(z-\gamma)^2}$
$k^2\gamma^k u(k)$	$\frac{\gamma z(z+\gamma)}{(z-\gamma)^3}$

The Z transform is a linear operator: any linear combination of functions is transformed into the same linear combination of their respective Z transforms.

In practice these complex calculations are simplified by the use of transform tables that cover most useful signal forms. A small extract is presented in Table 2.4.

2.2.2.2 Properties

The Z transform has important properties. If $u(k)$ designates the step function ($u(k) = 0, k < 0; u(k) = 1, k \geq 0$) and $F(z)$ is the Z transform of $f(k)u(k)$, then:

- The Z transform of $f(k-1)u(k-1)$ is $1/z \cdot F(z)$. This is the delay property.
- The Z transform of $f(k-m)u(k-m)$ is $1/(z^m) \cdot F(z)$.
- The Z transform of $f(k-1)u(k)$ is $1/z \cdot F(z) + f(-1)$.
- The Z transform of $f(k+1)u(k)$ is $zF(z) - zf(0)$. This is the advance property.
- The Z transform of $f(k+2)u(k)$ is $z^2F(z) - z^2f(0) - zf(1)$.

The Z transform is a powerful tool to solve linear difference equations with constant coefficients.

2.2.2.3 Notation

Note that the Z transform of a unit delay is '1/z' and the z transform of a unit advance is 'z'. Both expressions will appear in diagrams in the following subsections.

In the following subsections, some figures will show boxes with an input, one or more outputs, adders, and multipliers, similar to Figure 2.12.

The meaning is the following: the sampled signal E is filtered by $H_1(z)$ resulting in response signal Y . Then, signal T is obtained by subtracting the previous output S (one sample delay) from signal Y . Finally, signal S is obtained by filtering signal T by filter $H_2(z)$.

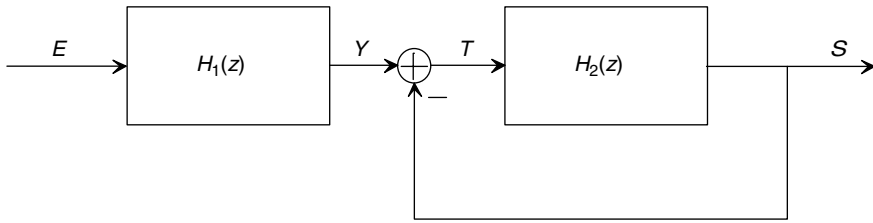


Figure 2.12 Typical digital filter representation.

2.2.2.4 Using the Z transform. Properties of the transfer function

With $T = \frac{1}{4}$ in the discrete difference equation above, for instance, we have:

$$(16E^2 - 20E + 6)y(k) = (E - 1)f(k)$$

If the Z transform of $y(k)u(k)$ is $Y(z)$ and the Z transform of $f(k)u(k)$ is $F(z)$, we have:

$$\begin{aligned} 16y(k+2) &\longrightarrow 16z^2Y(z) - 16z^2y(0) - 16zy(1) \\ -20y(k+1) &\longrightarrow -20zY(z) + 20zy(0) \\ 6y(k) &\longrightarrow 6Y(z) \\ f(k+1) &\longrightarrow zF(z) - zf(0) \\ -f(k) &\longrightarrow -F(z) \end{aligned}$$

We see that we have terms using $y(0)$ and $y(1)$ which are usually not known (but could be found by solving the equation iteratively for a given $f(k)$).

Let's try another approach and rewrite the equation as:

$$16y(k) - 20y(k-1) + 6y(k-2) = f(k-1) - f(k-2)$$

We get:

$$\begin{aligned} 16y(k) &\longrightarrow 16Y(z) \\ -20y(k-1) &= -20(y(k-1)u(k)) \\ &\longrightarrow -20(1/z \cdot Y(z) + y(-1)) = -20Y(z)/z \quad \text{if } y(-1) = 0 \\ 6y(k-2) &\longrightarrow 6(Y(z)/z^2 + y(-1)/z + y(-2)) = 6Y(z)/z^2 \quad \text{if } y(-2) = 0 \\ f(k-1)u(k) &\longrightarrow F(z)/z + f(-1) = F(z)/z \quad \text{if } f(p < 0) = 0 \text{ (causal input)} \\ f(k-2)u(k) &\longrightarrow F(z)/z^2 + f(-1)/z + f(-2) = F(z)/z^2 \quad \text{(causal input)} \end{aligned}$$

We obtain:

$$Y(z) = \frac{F(z)(1/z - 1/z^2)}{(16 - 20/z + 6/z^2)}$$

If we assume we want to find a solution with $f(k) = (2)^{-k}u(k) = (0.5)^k u(k)$ (the transform table tells us that $F(z) = z/(z - 0.5)$) we obtain:

$$Y(z) = \frac{z(z - 0.5)(1/z - 1/z^2)}{16 - 20/z + 6/z^2}$$

Decomposing the rational fraction into simpler components and using the transform table would give us $y(k)$.

The expression $Y(z)/F(z)$ is called the **transfer function** $H(z)$ of the system. We notice that the coefficients of $H(z)$ look familiar:

$$H(z) = \frac{z - 1}{16z^2 - 20z + 6}$$

When considering the equation using the advance operator form $(16E^2 - 20E + 6)y(k) = (E - 1)f(k)$, the numerator coefficients are the same as the coefficients for $f(k)$ and the denominator coefficients are the same as the coefficients for $y(k)$.

This is generally the case: the transfer function $H(z)$ can be obtained very simply from the coefficients of the difference equation using the advance operator (which will be shown in Subsection 2.2.2.5). This is one of the reasons the Z transform is so useful, even without complex calculations!

Another interesting property is that the Z transform of the **impulse response** of the system $h(k)$ is $H(z)$. The impulse function $\delta(0)$ is the input signal with $e(0) = 1$ and $e(k) = 0$ everywhere else. The impulse response is the response $s(k)$ of the system when the input is $\delta(0)$. The proof goes beyond the scope of this book.

2.2.2.5 Application for FIR and IIR filters

The impulse response $h(k)$ of a linear, time-invariant, discrete time filter determines its response to any signal:

- Because of linearity, the response to an impulse of amplitude a is $ah(k)$.
- Because of time invariance, the response to a delayed impulse $\delta(k - x)$ is $h(k - x)$.

Any input signal $e(k)$ can be decomposed into a sum of delayed impulses, and because of the linearity of the filter we can calculate the response. Each impulse $e(x)$ creates a response $e(x)h(k - x)$, where k is the discrete time variable. The response $s(n)$ at instant n is $e(x)h(n - x)$. The sum of all the response components at instant n for all $e(x)$ is:

$$s(n) = \sum_{x=-\infty}^{\infty} e(x)h(n - x)$$

This is a convolution of e and h in the discrete time domain. This relation is usually rewritten by taking $m = n - x$:

$$s(n) = \sum_{m=-\infty}^{m=+\infty} e(n - m)h(m)$$

For a physically realizable system (which cannot guess a future input signal and therefore cannot react to $\delta(0)$ before time 0), we must have $h(x) = 0$ for $x < 0$. Physically realizable systems are also called **causal systems**.

Filters that only have a finite impulse response are called **finite impulse response (FIR) filters**. The equation for an FIR filter is:

$$s(n) = \sum_{k=0}^{k=N} e(n-k)h(k)$$

where the $h(k)$ for $k = 0$ to N are constants that characterize the system. In voice-coding filters these constants are sometimes dynamically adapted to the signal, but with a timescale that is much lower than the variance of the signal itself: they are in fact a succession of FIR filters with varying coefficients.

Filters that have an infinite impulse response are called **infinite impulse response (IIR) filters**. In many filters the response is infinite because recursivity has been introduced in the equation of the filter. The equation for a recursive IIR filter is:

$$s(n) = \sum_{k=0}^{k=N} e(n-k)a(k) - \sum_{k=1}^{k=L} s(n-k)b(k)$$

Note that we have introduced the past values of the output ($s(n-k)$) in the formula and that the values $a(k)$ and $b(k)$ characterize the system. When we compute the Z transform of the time domain equation of an IIR filter:

$$s(n) = \sum_{k=0}^{k=N} e(n-k)a(k) - \sum_{k=1}^{k=L} s(n-k)b(k)$$

we obtain²:

$$S(z) = E(z) \sum_{k=0}^{k=N} z^{-k}a(k) - S(z) \sum_{k=1}^{k=L} z^{-k}b(k)$$

or

$$S(z) \left(1 + \sum_{k=1}^{k=L} z^{-k}b(k) \right) = E(z) \sum_{k=0}^{k=N} z^{-k}a(k)$$

$$\begin{aligned} 2 \sum_{n=-\infty}^{\infty} \left(\sum_{k=0}^N e(n-k)a(k) \right) z^{-n} &= \sum_{k=0}^N \sum_{n=-\infty}^{\infty} e(n-k)a(k)z^{-n} = \sum_{k=0}^N a(k) \sum_{n=-\infty}^{\infty} e(n-k)z^{-n} \\ &= \sum_{k=0}^N a(k)z^{-k}E(z) \end{aligned}$$

We can also calculate the output-to-input ratio in the Z domain:

$$S(z) = E(z)H(z) \quad \text{with} \quad H(z) = \frac{\sum_{k=0}^{k=N} z^{-k} a(k)}{1 + \sum_{k=1}^{k=L} z^{-k} b(k)}$$

We see that the transfer function in the Z domain has an immediate expression from the coefficients of the filter equation.

2.2.2.6 System realization

A given transfer function $H(z)$ is easily realizable by a discrete time filter. For instance, if:

$$H(z) = \frac{b_3 z^3 + b_2 z^2 + b_1 z + b_0}{z^3 + a_2 z^2 + a_1 z + a_0}$$

then a first step would be to obtain:

$$X(z) = \frac{1}{z^3 + a_2 z^2 + a_1 z + a_0} * F(z)$$

which is easy by considering the corresponding difference equation:

$$x(k+3) = -a_2 x(k+2) - a_1 x(k+1) - a_0 x(k) + f(k)$$

which is realized by a system like that in Figure 2.13.

A second step is to obtain $Y(z)$ by a linear combination of the $z^i X(z)$, as in Figure 2.14.

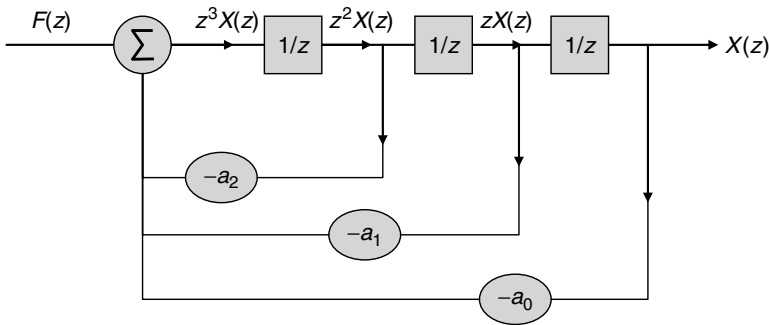


Figure 2.13 Realization of $H(z)$ denominator.

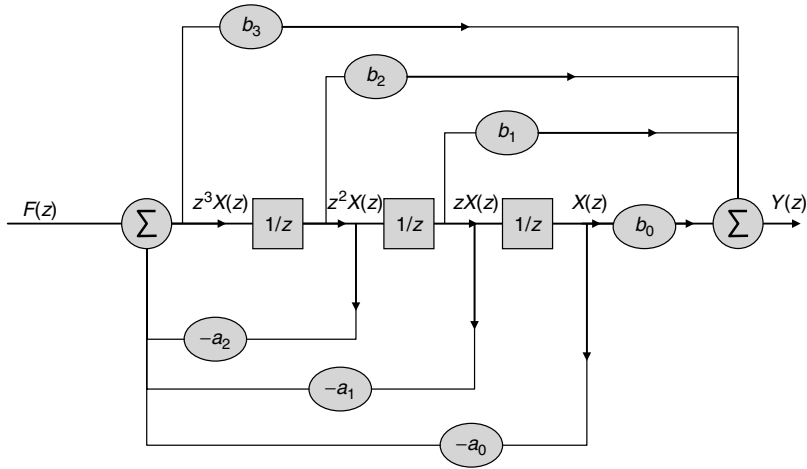


Figure 2.14 Full realization of $H(z)$.

2.2.2.7 Realization of frequency filters

The fact the transfer function $H(z) = Y(z)/F(z)$ is also the Z transform of the impulse response makes it very useful to determine the frequency response of a discrete time filter. If $h(k)$ is the impulse response of a system, the system response $y(k)$ to input z^k is:

$$y(k) = \text{convolution}(h(k), z^k) = \sum_{m=-\infty}^{\infty} h(m)z^{k-m} = z^k \sum_{m=-\infty}^{\infty} h(m)z^{-m} = H(z)z^k$$

where $H(z)$ is the Z transform of the filter impulse response and the transfer function as well. A sampled continuous time sinusoid $\cos(\omega t)$ is of the form $\cos(\omega T k) = \text{Re}(e^{j\omega T k})$ where T is the sampling period. A sample sinusoid respecting the Nyquist limit must have $\omega < \pi/T$. If we take $z = e^{j\omega T}$, the above result tells us that the frequency response of the filter to the discrete time sinusoid is:

$$y(k) = H(e^{j\omega T}) \cdot e^{j\omega T k}$$

Therefore, we can predict the frequency response of a system by studying $H(e^{j\omega T})$. $H(z)$ can be rewritten as a function of its zeros z_i and its poles p_i :

$$H(z) = b_n \frac{(z - z_1)(z - z_2) \cdots (z - z_n)}{(z - p_1)(z - p_2) \cdots (z - p_m)}$$

for stable systems the poles must be inside the unit complex circle, and for physically realizable systems we must have $n < m$ and poles and zeros should occur as pairs of conjugates.

A graphical representation of the transfer function (Figure 2.15) for two zeros and two poles makes it simple to understand how H behaves as a function of ω .

The amplitude of the original sinusoid is multiplied by:

$$|H(e^{j\omega T})| = b_n \frac{d_{z_1} d_{z_2} \cdots d_{z_n}}{d_{p_1} d_{p_2} \cdots d_{p_m}}$$

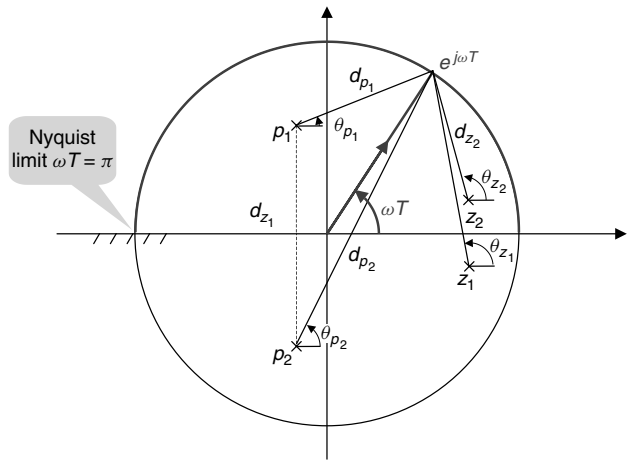


Figure 2.15 Graphical interpretation of the transfer function.

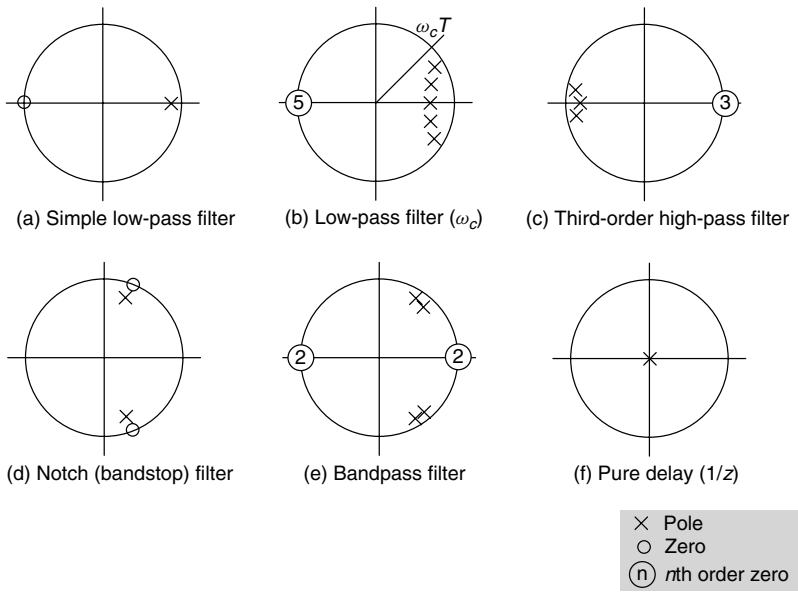


Figure 2.16 Graphical representation of common filters.

and the phase of the original sinusoid is changed by the angle:

$$\angle H(e^{j\omega T}) = (\theta_{z_1} + \theta_{z_2} + \dots + \theta_{z_n}) - (\theta_{p_1} + \theta_{p_2} + \dots + \theta_{p_n})$$

Frequency filters can be realized by placing poles near the frequencies that need to be amplified and zeros near the frequencies that need to be attenuated. Figure 2.16 gives a few examples.

In a simple low-pass filter (case A), a pole is placed near point 1 (it needs to be inside the unit circle for a stable system), and a zero at point -1 (zeros can be anywhere). The cut-off frequency for such a filter is at $\omega T = \pi/2$.

In order to ensure that the gain is sustained on a specific band $[0 - \omega_c]$, more poles must be accumulated near the unit circle in the band where the gain must be close to unity (case B).

The principle of the high-pass filter (C) is similar, but the roles of the zeros and poles are inverted. A higher order filter will have a sharper transition at the cut-off frequency and therefore will better approach an ideal filter. Note that a realizable system (where the future is not known in advance) requires more poles than zeroes or an equal number of poles and zeros.

A notch (bandstop) filter (D) is obtained by placing a zero at the frequency that must be blocked. A zero must be placed at the conjugate position for a realizable system (all the coefficients of the polynomials of the transfer function fraction must be real). Poles can be placed close to the zeros to quickly recover unity gain on both sides of the blocked frequency.

A bandpass filter (E), can be obtained by enhancing the frequencies in the transmission band with poles and attenuating frequencies outside this band by placing zeros at points 1 and -1 .

Note that a pole placed at the origin (F) does not change the amplitude response of the filter, and therefore a pole can always be added there to obtain a physically realizable system (more poles than zeros). A filter with a single pole at the origin is in fact a pure delay of period T (linear phase response of $-\omega T$). This is logical: filters cannot be realized if they need to know a future sample \dots and can be made realizable by delaying the response of the filter in order to accumulate the required sample before computing the response. Similarly a zero at the origin is a pure advance of T .

The ease with which arbitrary digital filters can be realized using the results of this section and the method of the previous section sharply contrasts with the complexity of analog filters, especially for high-order filters. This is the reason discrete time signal processing has become so prevalent.

2.2.3 Linear prediction for speech-coding schemes

2.2.3.1 Linear prediction

Linear prediction is used intensively in speech-coding schemes; it uses a linear combination of previous samples to construct a predicted value that attempts to approach the next input sample:

$$s_p(n) = \sum_{k=1}^{k=p} a_k s(n-k)$$

gives the predicted value at time n .

The coefficients a_k must be chosen to approach the $s(n)$ value. If $s_p(n)$ is indeed similar to $s(n)$, then the error signal $e(n) = s(n) - s_p(n)$ can be viewed as a residual signal resembling a white noise.

With this remark, we can decompose the issue of transmitting speech information (the waveform) into two separate problems:

- The transmission of the set of coefficients a_k (or some coded representations).
- The transmission of information related to the error signal $e(n)$.

Ideally, if $e(n)$ was **white noise**, then only its power should be sent. In reality, $e(n)$ is not white noise and the challenge to speech coder experts is to model this error signal correctly and to transmit it with a minimal number of bits.

The a_k coefficients are called **linear prediction coefficients (LPCs)** and p is the order of the model. Each LPC vocoder uses its own methods for computing the optimal a_k coefficients. One common method is to compute the a_k that minimize the quadratic error on the samples to predict, which leads to a linear system (the equation of Yule–Walker) that can be solved using the Levinson–Shur method. Usually, these coefficients are computed on a frame basis of 10–30 ms during which the speech spectrum can be considered as stationary.

2.2.3.2 The LPC modeling filter

In the previous subsection, we showed that a speech signal s could be approached by a linearly predicted signal s_p obtained by an LPC filter L . Another way to view this is to say that the speech signal, filtered by the $(1 - L)$ filter, is a residual error signal ideally resembling white noise.

At this point, it is interesting to wonder whether the inverse filter can approach the original speech spectrum by filtering an input composed of white noise. To find the expression of the inverse filter, we can use the previous equation, replacing $s_p(n)$ by its expression as a function of s :

$$e(n) = s(n) - \sum_{k=1}^{k=p} a_k s(n-k)$$

or in the Z domain:

$$E(z) = S(z) \left(1 - \sum_{k=1}^{k=p} a_k z^{-k} \right)$$

So we have:

$$S(z) = \frac{E(z)}{\left(1 - \sum_{k=1}^{k=p} a_k z^{-k} \right)}$$

which gives the ‘speech’ signal by filtering an input composed of white noise.

The digital filter:

$$H(z) = \frac{1}{\left(1 - \sum_{k=1}^{k=p} a_k z^{-k} \right)} = \frac{1}{A(z)}$$

is called the LPC modeling filter. It is an all pole filter (no zero) that models the source (speech). If we want to evaluate the residual error signal we only need to filter the speech signal ($s(n)$) by the filter $A(z)$ because we have $E(z) = S(z)A(z)$. $A(z)$ is often called

the LPC analysis filter (giving the residual signal) and $H(z) = 1/A(z)$ the LPC synthesis filter (giving the speech signal from the residual signal). These concepts are intensively used in the low-bitrate speech coder schemes discussed in the following section.

Note that we are only trying to approach the frequency spectrum of the original speech signal, not the exact time representation: this is because human hearing is not sensitive to the exact phase or time representation of a signal, but only to its frequency components.

2.3 Overview of speech signals

2.3.1 Narrow-band and wide-band encoding of audio signals

Audio engineers distinguish five categories of audio quality:

- The telephony band from 300 Hz to 3,400 Hz. An audio signal restricted to this band remains very clear and understandable, but does alter the natural sound of the speaker voice. This bandwidth is not sufficient to provide good music quality.
- The audio wide-band from 30 Hz to 7,000 Hz. Speech is reproduced with an excellent quality and fidelity, but this is still not good enough for music.
- The hi-fi band from 20 Hz to 15 kHz. Excellent quality for both voice and music. Hi-fi signals can be recorded on one or multiple tracks (stereo, 5.1, etc.) for spatialized sound reproduction.
- The CD quality band from 20 Hz to 20 kHz.
- Professional quality sound from 20 Hz to 48 kHz

Table 2.5 shows the bitrate that needs to be used for each level of audio quality, without compression.

2.3.2 Speech production: voiced, unvoiced, and plosive sounds

Speech sounds are characterized by the shape of the vocal tract which consists of the vocal cords, the lips, and the nose [B1]. The overall frequency spectrum of a speech sound is determined by the shape of the vocal tract and the lips (Figure 2.17). The vocal

Table 2.5 Uncompressed bitrate requirements according to audio quality

	Sampling frequency (kHz)	Quantization (bits)	Nominal bitrate (kbit/s)
Telephony	8	13	104
Wide-band	16	14	224
Hi-fi	32	16	512 mono (1,024 stereo)
CD	44.1	16	705.6 mono (1,411 stereo)
Professional	96	24	13,824 (5.1 channels)

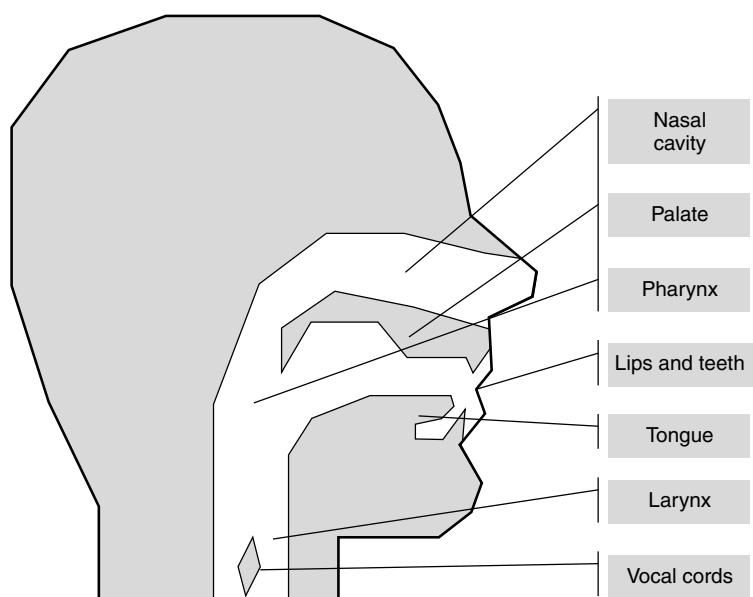


Figure 2.17 Human voice production.

tract introduces resonance at certain frequencies called formants. This resonance pattern carries a lot of information.

There are mainly three types of speech sounds: **voiced**, **unvoiced**, and **plosive**.

Periodically closing and opening the vocal cords produces voiced speech. The period of this closing and opening cycle determines the frequency at which the cords vibrate; this frequency is known as the pitch of voiced speech. The pitch frequency is in the range of 50–400 Hz and is generally lower for male speakers than for female or child speakers. The spectrum of a voiced speech sample presents periodic peaks at the resonance frequency and its odd harmonics (the formants). The voiced speech spectrum can be easily modeled by an all-pole filter with five poles or ten real coefficients computed on a frame length of 10–30 ms.

During unvoiced speech, such as ‘s’, ‘f’, ‘sh’, the air is forced through a constriction of the vocal cords; unvoiced speech samples have a noise-like characteristic and consequently their spectrum is flat and almost unpredictable.

Speech is produced by the varying state of the vocal cords, and by the movement of the tongue and the mouth. Not all speech sounds can be classified as voiced or unvoiced. For instance, ‘p’ in ‘puff’ is neither a voiced nor an unvoiced sound: it is of the plosive type.

Many speech sounds are complex and based on superimposing modes of production, which makes it very difficult to correctly model the speech production process and consequently to encode speech efficiently at a low bitrate.

Figures 2.18–2.23 give some samples of voiced, unvoiced, and mixed speech segments, and their corresponding frequency spectrum associated with a 10th order LPC modeling filter frequency response.

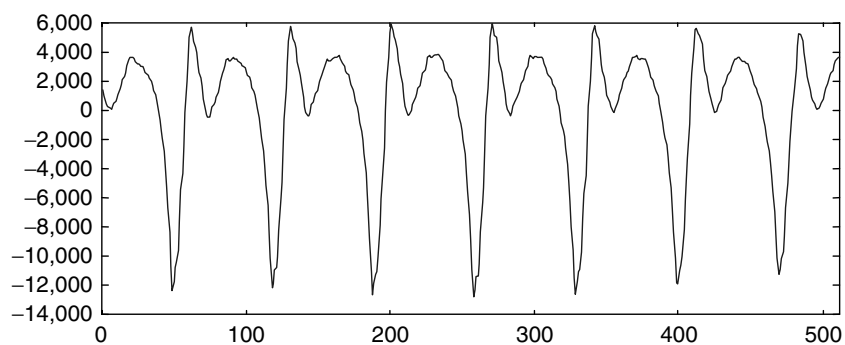


Figure 2.18 Time representation of a voiced speech sequence (in samples).

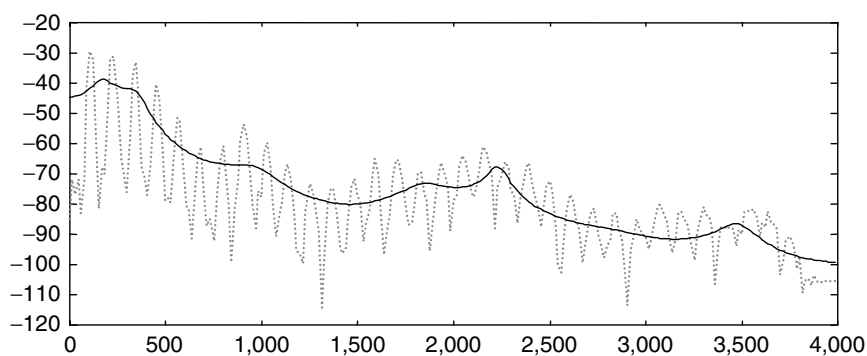


Figure 2.19 Frequency spectrum of the voiced speech segment (dotted line) and the 10th order LPC modelling filter response.

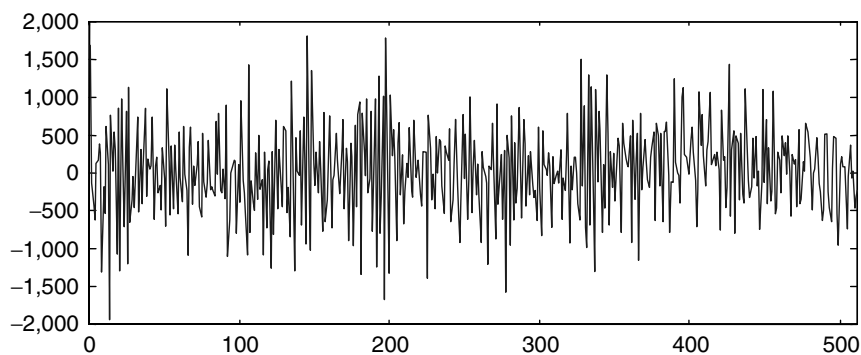


Figure 2.20 Time representation of an unvoiced speech sequence (in samples).

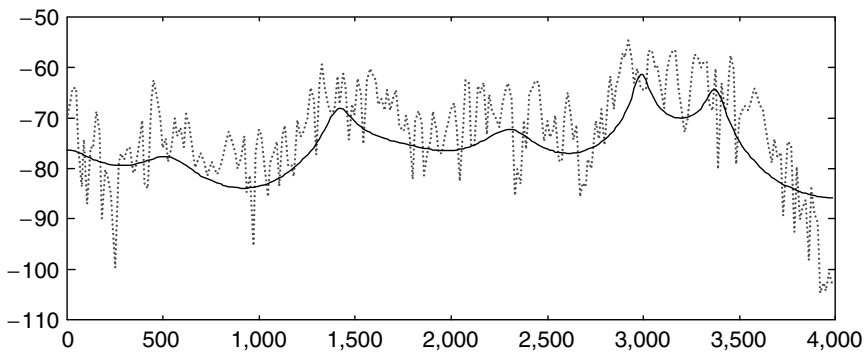


Figure 2.21 Frequency spectrum of the unvoiced speech segment (dotted line) and the 10th order LPC modelling filter response.

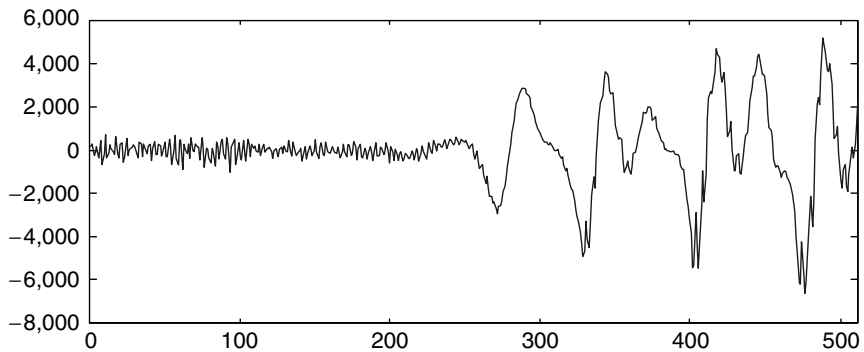


Figure 2.22 Time representation of a mixed speech sequence (in samples).

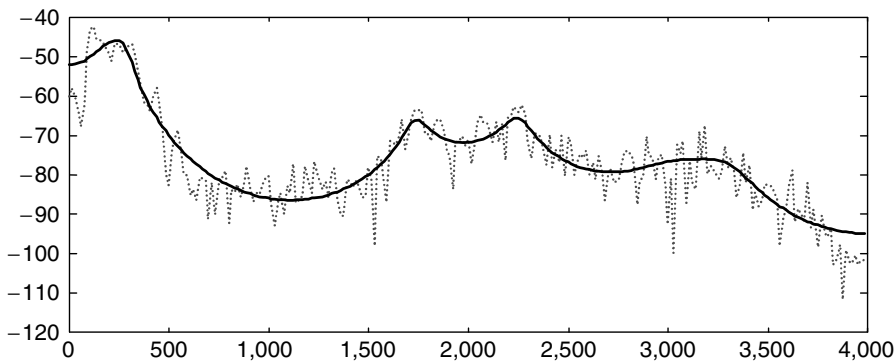


Figure 2.23 Frequency spectrum of the mixed speech segment (dotted line) and the 10th order LPC modelling filter response.

2.3.3 A basic LPC vocoder: DOD LPC 10

By being able to distinguish voiced and unvoiced speech segments, it is possible to build a simple source filter model of speech (Figure 2.24) and a corresponding source speech coder, also called a **vocoder** (Figure 2.25). The detection of voiced segments is based on the autocorrelation of the processed frame after filtering through the LPC analysis filter. If the autocorrelation is rather flat and there is no obvious pitch that can be detected, then the frame is assumed to be unvoiced; otherwise, the frame is voiced and we have computed the pitch.

The DOD 2,400-bit/s LPC 10 [A8] speech coder (called LPC 10 because it has ten LP coefficients) was used as a standard 2,400-bit/s coder from 1978 to 1995 (it was subsequently replaced by the mixed excitation linear predictor, or MELP, coder). This vocoder has parameters as shown in (Table 2.6).

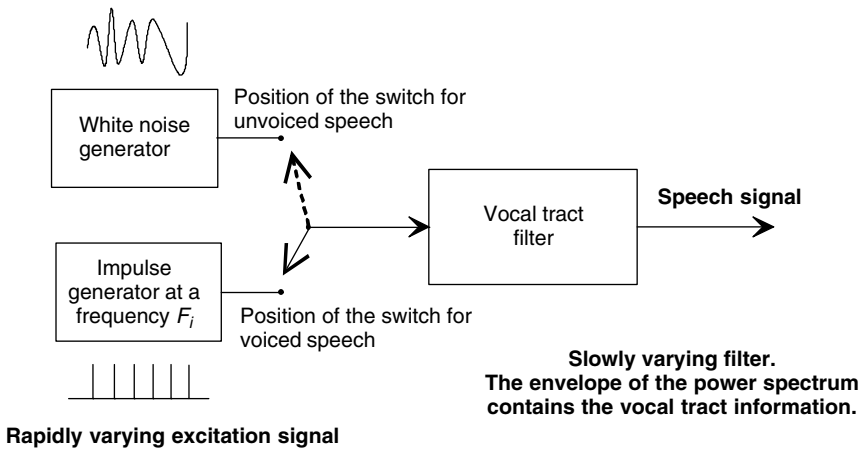


Figure 2.24 DOD LPC 10 voice synthesis for voiced and unvoiced segments (a source filter model of speech).

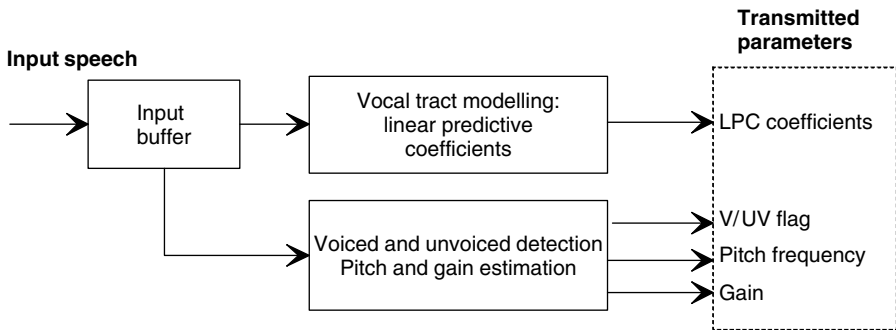


Figure 2.25 Basic principle of a source speech coder called a vocoder.

Table 2.6 DOD LPC 10 frame size, bit allocation and bitrate

Sampling frequency	8 kHz
Frame length	180 samples = 22.5 ms
Linear predictive filter	10 coefficients = 42 bits
Pitch and voicing information	7 bits
Gain information	5 bits
Total information	54 bits per frame = 2,400 bit/s

The main disadvantage of source coders, based on this simple voiced/unvoiced speech production model, is that they generally give a very low speech quality (synthetic speech). Such coders cannot reproduce toll-quality speech and are not suitable for commercial telephony applications. The MELP coder made some progress by being able to model voice segments as a mix of voiced and unvoiced sounds, as opposed to a binary choice.

2.3.4 Auditory perception used for speech and audio bitrate reduction

The coders described previously attempt to approach the exact frequency spectrum of the source speech signal. This assumes that human hearing can perceive all frequencies produced by the speaker. This may seem logical, but human hearing cannot in fact perceive any speech frequency at any level. All acoustical events are not audible: there is a curve giving the perception threshold, depending on the sound pressure level and the frequency of the sound [A4, A9, A14]. Weak signals under this threshold cannot be perceived. The maximum of human hearing sensitivity is reached between 1,000 Hz and 5,000 Hz. In addition some sounds also affect the sensitivity of human hearing for a certain time. In order to reduce the amount of information used to encode speech, one idea is to study the sensitivity of human hearing in order to remove the information related to signals that cannot be perceived. This is called ‘perceptual coding’ and applies to music as well as voice signals.

The human ear is very complex, but it is possible to build a model based on **critical band** analysis. There are 24 to 26 critical bands that overlap bandpass filters with increasing bandwidth, ranging from 100 Hz for signals below 500 Hz to 5,000 Hz for signals at high frequency.

In addition a low-level signal can be inaudible when masked by a stronger signal. There is a predictable time zone, almost centered on the masker signal, that makes all the signals inside this area inaudible, even if they are above their normal perception threshold. This is called **simultaneous frequency domain masking**, which is used intensively in perceptual audio-coding schemes and includes pre- and post-masking effects.

Although these methods are not commonly used in low-bitrate (4–16 kbit/s) speech coders, they are included in all the modern audio coders (ISO MPEG-1 Layer I, II, III,³ MPEG-2 AAC, AC3, or Dolby Digital). These coders rely on temporal to frequency domain transformation (analysis filter bank) coupled to an auditory system-modeling

³ The MPEG-1 Layer III audio coder is also known as MP3 for Web users.

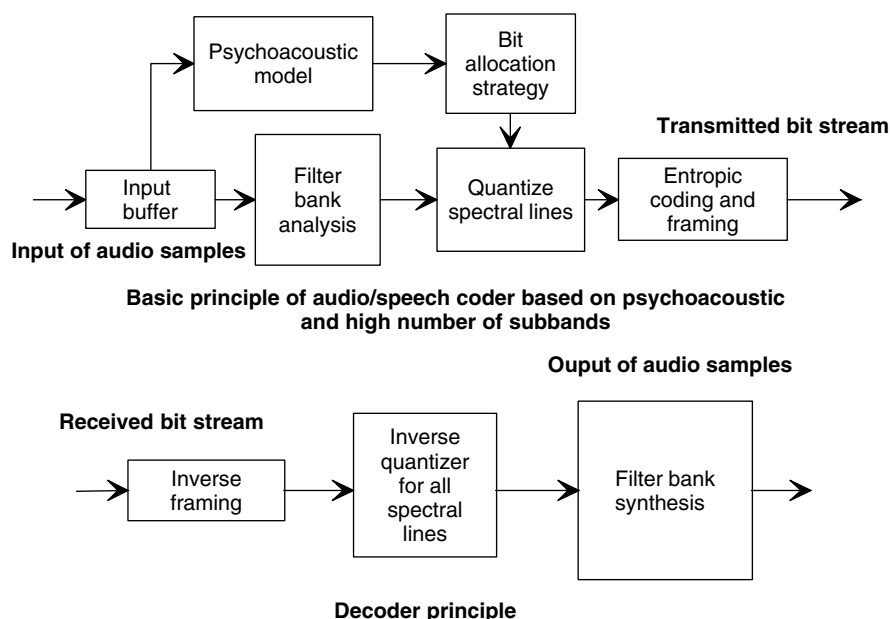


Figure 2.26 Usage of filter banks for audio signal analysis and synthesis.

procedure that calculates masking thresholds and drives a dynamic bit allocation function. Bits are allocated to each band in order to fit the overall bitrate and masking threshold (see Figure 2.26) requirements.

Today, audio signals can be efficiently encoded (almost CD-like quality [A9, A10, A11, A12]) in about 64 kbit/s for a single monophonic channel with the most advanced audio-coding (AAC) schemes. Wideband (20–7,000 Hz) speech and audio coders can use the same scheme to encode in only 24 kbit/s or 32 kbit/s (although there are some issues related to the analysis filter bank—overlap and add procedure in the decoder—that result in annoying pre-echo phenomena. This is mainly due to the nonstationary characteristic of the speech signal and is very perceptible when onset appears).

Some low-bitrate speech coders do not use the perceptual model for speech coding itself, but rather to better evaluate the residual error signal. **Analysis by synthesis (ABS)** speech coders (addressed later) ponder the error signal used in the closed-loop search procedure by a perceptual weighting filter derived from the global spectrum of speech. The function of this perceptual weighting filter is to redistribute the quantizing noise into regions where it will be masked by the signal. This filter significantly improves subjective coding quality by properly shaping the spectrum of the error: error noise is constrained to remain below the audible threshold when the correlated signal is present. In ABS decoders, a post-filter may also be used to reduce noise between the maxima of the spectrum (formants) by reducing the signal strength in these regions and boosting the power of formants. This significantly improves the perceived quality on the MOS scale, but there is a price to pay: post-filters do alter the naturalness (fidelity) of the decoded speech. An example of such a filter is given in the introduction to Section 2.7.

2.4 Advanced voice coder algorithms

2.4.1 Adaptive quantizers. NICAM and ADPCM coders

We have already mentioned that if the **probability density function (PDF)** of the input is known one optimal quantizer can be computed for the signal. Linear or logarithmic quantizers are time-unvarying systems: their step sizes are fixed for the entire duration of the signal. Logarithmic quantizers (such as G.711) are an optimization that provides an SNR independent of the level of the signal.

It is also possible to adapt the quantizer dynamically to best match the instantaneous characteristics of the signal.

Many voice coders use dynamic quantization algorithms. The rules and types of adaptation used to encode the signal can be transmitted with the encoded signal (forward adaptive quantizers), but this is not required: backward adaptive quantizers use only the characteristics of the previously transmitted encoded signal to optimize the processing of the current sample(s), enabling the receiver to also compute the optimal adaptation that will be used for the next received encoded signal.

In addition, the optimal characteristics of the adaptive quantizer can be selected (or computed) for each sample, based on the characteristics of a group of contiguous samples (such a group is called a **frame**). A frame-based adaptation procedure is more efficient in terms of transmitted bitrate, especially when forward quantizer selection is used. The size of the frame must be selected carefully: if the size is too small there may be a large overhead for transmitting the scaling information, but if the block size is too large the quantizing steps may become inadequate for some portions of the frame, leading to large errors in the quantization process.

Figure 2.27 shows the principle of a forward adaptive quantizer and Figure 2.28 shows the principle of an inverse forward adaptive quantizer.

NICAM is an example of a coder using a forward adaptive quantizer (Figure 2.29). The **NICAM (near-instantaneous companding and multiplexing)** system is used to transmit the audio stereo signal digitally on analog TV channels using the PAL and SECAM TV color systems. The NICAM system transmits two stereo audio channels sampled at

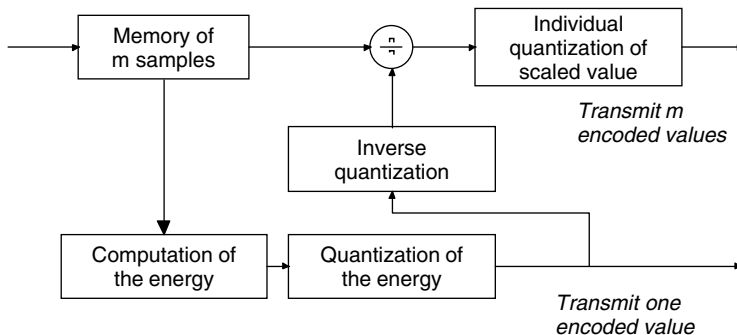


Figure 2.27 Principle of a forward adaptive quantizer.

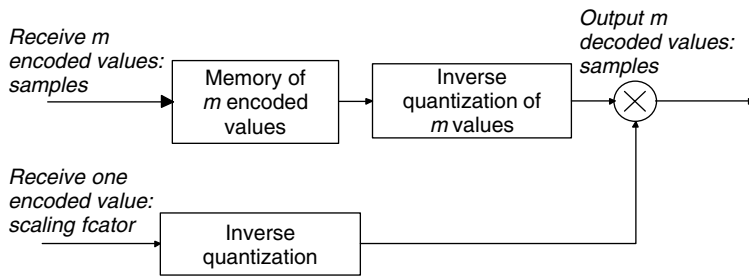


Figure 2.28 Principle of a forward adaptive inverse quantizer.

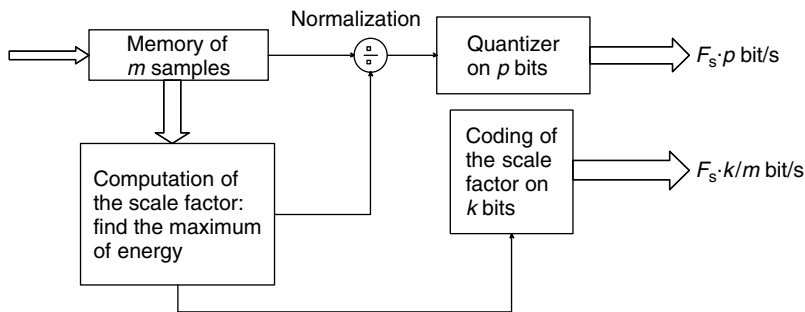


Figure 2.29 Near-instantaneous quantizer using: $F_s \cdot (p + k/m)$ bit/s.

32 kHz in a bitrate of 728 kbit/s. NICAM memorizes a buffer (the near-instantaneous characteristic . . .) of 32 samples and evaluates the mean power during this period of time, which is used to normalize the input samples. A fixed 10-bit logarithmic quantizer is then used on the normalized signal. The transmitted frame comprises the individual quantized samples, the scaling factor information, framing information, and some parity bits that protect the compressed audio signal against transmission errors.

It has been shown that, for the same subjective quality, the use of the quasi-instantaneous (32 samples) system requires 10.1 bits per sample compared with 11 bits per sample using a classical sample by sample logarithmic quantizer. There is a gain of 10% resulting from the use of block analysis and the forward ‘adaptive’ quantizer.

For backward adaptive quantizers, there is no need to transmit any information related to the scaling procedure; the mean power is estimated on the quantized signal and, therefore, the inverse quantizer can reconstruct this information exactly (Figures 2.30 and 2.31).

A very simple but efficient backward adaptive quantizer called ‘one-word memory’ is used in the ADPCM G.726 and G.727 ITU-T speech coders [A13]. A simple coefficient M_i depending only on the previous quantized sample determines the compression or expansion of the quantization steps for the next sample. If the quantizer has 4 bits (1 sign bit and 8 ranges of quantization), there are 8 M_i fixed coefficients (each implicitly associated with a quantizing range) insuring the compression or expansion of the quantizer. When large values are input to the quantizer, the multiplier value is greater than 1 and for small previous values the multiplier value is less than 1. This tends to force the

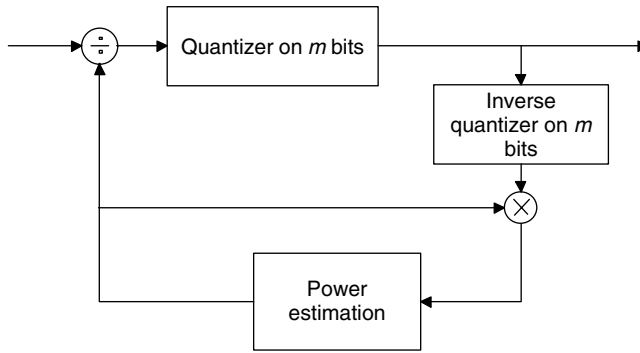


Figure 2.30 Principle of a backward adaptive quantizer.

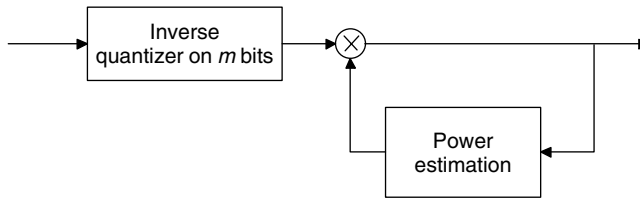


Figure 2.31 Principle of an m bit backward inverse quantizer.

adaptive quantizer to track the dynamics of the input signal (we can also consider that the previous measurement gave us some information on the probability density for the next sample, which we use to optimize the quantification). A fixed quantizer can be used and there is no need to transmit any scaling information to the decoder side (see Figures 2.32 and 2.33). Transmission errors will cause desynchronization of the coder and the decoder for a single sample.

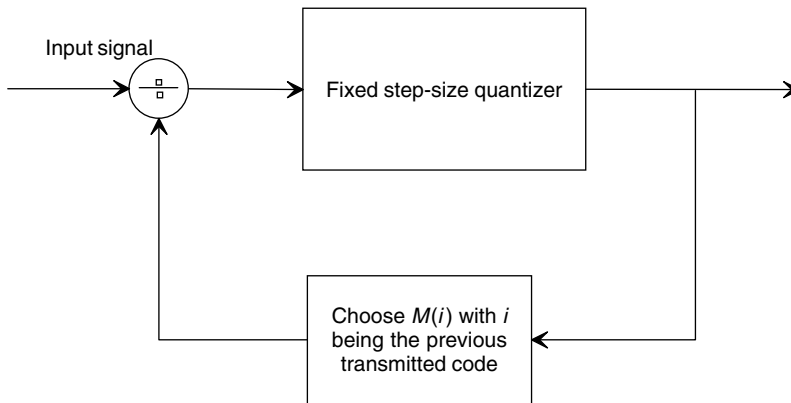


Figure 2.32 One-word memory adaptive quantizer.

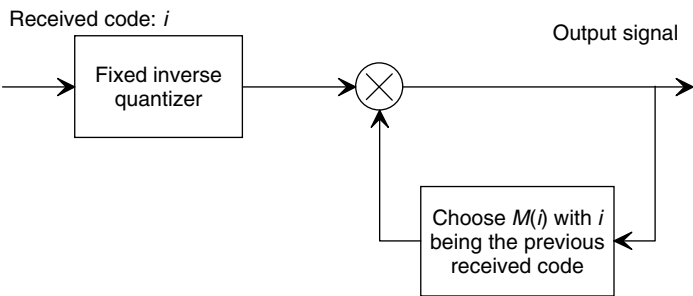


Figure 2.33 One-word memory adaptive inverse quantizer.

Table 2.7 Eight expansion coefficients attached to eight quantization ranges

M_0	0.969
M_1	0.974
M_2	0.985
M_3	1.006
M_4	1.042
M_5	1.101
M_6	1.208
M_7	1.449

Table 2.7 gives the M_i values for an eight-level quantizer (for each sign) optimized for exponential distribution.

The G.726 quantizer only needs to send 4 bits per sample (32 kbit/s), instead of 8 for G.711. G.726 is commonly used on many PSTN communication links when there is a need to reduce the transmitted bitrate.

2.4.2 Differential predictive quantization

In speech and audio signals, there is a strong correlation between the present sample and the previous one. The consequence is that if we subtract the previous sample from the present one, the variance of the difference signal will be lower than the variance of the original signal: it will require less bits to be quantized.

Unfortunately, we cannot directly use the exact previous sample value because it is inaccessible to the decoder. Instead, we must use the value of the previous sample as decoded by the receiver. In order to do this, the encoder relies on a ‘local decoder’ feedback loop which is common in speech and audio compression schemes. We have:

$$E(n) = X(n) - X_d(n - 1)$$

where $X_d(n - 1)$ is the decoded value at time $n - 1$, and we transmit the quantized version of $E(n)$, which is $Q[E(n)]$.

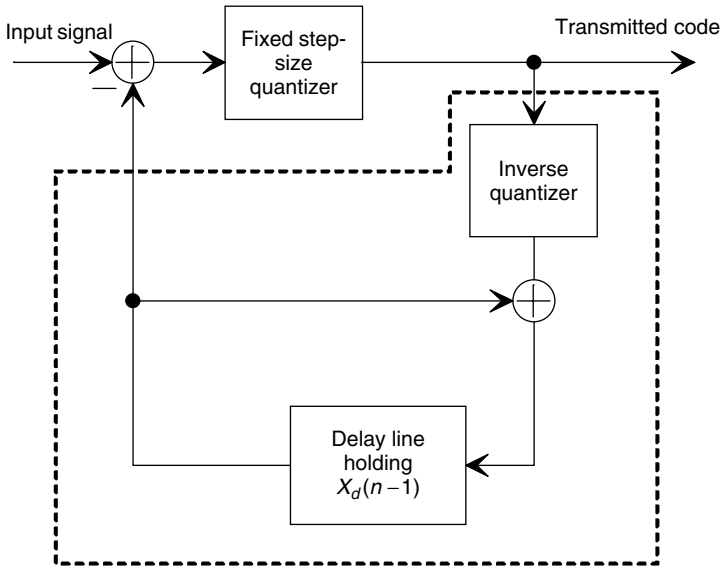


Figure 2.34 Principle of a differential quantizer (one-word memory prediction). The local decoder is in the dotted box and is identical to the distant decoder.

At the decoder side, we can compute the decoded value at time n :

$$X_d(n) = X_d(n-1) + Q^{-1}[Q[E(n)]] = X(n) + (Q^{-1}[Q[E(n)]] - E(n))$$

$X_d(n)$ approaches $X(n)$, but has the small difference introduced by quantization noise: $(Q^{-1}[Q[E(n)]] - E(n))$. If Q was ideal, then the noise signal would be zeroed.

Figure 2.34 illustrates the basic principle of a waveform speech or audio coder (called waveform because it tracks the temporal shape of the signal, as opposed to its frequency spectrum): all the concepts such as prediction or differential encoding are present.

The previous scheme is not a realistic one due to its sensibility to transmission errors: any transmission error will permanently desynchronize the decoder.

A more robust solution is to use a correlation coefficient:

$$E(n) = X(n) - X_d(n-1)$$

is replaced by:

$$E(n) = X(n) - C_1 * X_d(n-1)$$

where C_1 is the correlation coefficient. A value below unity will decrease the influence of a transmission error over time.

Like all linear prediction schemes, this works only if there is some correlation in the input signal (i.e., it does not exhibit a flat frequency spectrum (white noise)). In the case of noise, there is no correlation between adjacent input samples. There is no chance to predict the future sample knowing the previous one. By contrast, speech and audio

signals, due to their production mode, exhibit a non-flat spectrum and consequently high correlation exists between samples.

This differential encoding method can be generalized by using more than one previous sample to build the predicted term and by using a dynamically computed correlation factor:

- In waveform or temporal coders working on a sample-by-sample basis, a temporal prediction of the signal is built from a linear combination of previous (decoded) samples. The coefficients are not transmitted; they are computed by a symmetrical procedure in the decoder.
- Vocoder or ABS speech coders filter the input signal by an inverse model based on correlation coefficients. It is the residual signal (output of the filter) which is encoded and transmitted with the modeling filter coefficients (called linear prediction coefficients, LPCs). LPC analysis is typically performed on a time frame of 10–30 ms at a sampling frequency of 8 kHz. This is a period of time where the speech signal can be considered as quasi-stationary.

In these algorithms, based on a history of more than one sample, the term $X_d(n)$ is replaced by $X_p(n)$, which is the value of the predicted signal based on previous samples:

$$X_p(n) = \sum_{i=1}^{i=N} A_i X(n-i)$$

As indicated in Figure 2.35, coefficients A_i can be fixed or ‘adaptive’ (i.e., computed for each new sample). When fixed, they are nonoptimal and derived from the average (if it really exists) frequency spectrum of the signal.

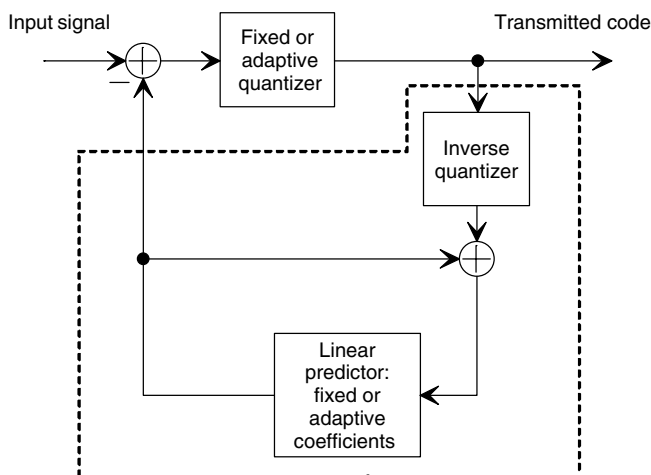


Figure 2.35 General principle of a differential (fixed or adaptive) coder. The local decoder is in the dotted box and is identical to the distant decoder.

Computation of the set of coefficients A_i in order to minimize quadratic error requires solving a set of linear equations [B2]. Even for a frame-by-frame analysis (such as for vocoder or ABS coders), this is a complex computational task which is out of reach of most real-time implementations. Many approximation algorithms have been developed to reduce computational complexity:

- For waveform coders, the set A_i , which is generally not transmitted, is continuously (on a sample-by-sample basis) adapted by the ‘stochastic gradient algorithm’ or by a simple ‘sign’ algorithm, where the absolute value of coefficients with the same sign as the error are reduced, and vice versa.
- For frequency or analysis by synthesis speech-coding schemes, the set A_i must be quantized and transmitted to the decoder side. The set of coefficients A_i or similar quantities modeling the short-term (10–30 ms) spectrum of the speech signal have to be computed. The direct inversion of the matrix obtained by expressing the minimization of quadratic errors is not used. More efficient algorithms have been studied and tuned to efficiently compute LPCs and to quantize them. Among them, the Levinson–Durbin algorithm and the Schur recursion are the most frequently used iterative methods to compute the A_i (Levinson–Durbin algorithm) or some partial coefficients called *parcours* (Schur recursion).

2.4.3 Long-term prediction for speech signal

Once a linear predictor (LPC, [B1]) has been used to filter the original speech signal, the correlation between adjacent samples is removed: the LPC filter $1/A(z)$ models the average (short-term) spectrum of speech.

However, for voiced speech, the pitch introduces a long-term correlation. The fine structure of the speech spectrum is present in this residual signal. Due to pitch-induced quasi-periodicity, the residual signal still exhibits large variations. A pitch predictor can be used to remove the long-term correlation remaining in the residual signal. The simplest form of this pitch predictor filter (called the **long term predictor (LTP) filter**) is $B(z) = 1 - \beta z^{-M}$, where M is the pitch period and β a scalar gain. This filter subtracts from the current speech sample the value of a previous sample (at a distance of M samples) with a scaling factor of β . This procedure reduces the quasi-periodic behavior of the residual signal. A more generalized form of this LTP filter is $B(z) = 1 - \sum_i \beta_i z^{-M-i}$, called a multi-tap LTP filter.

In speech processing and coding, one of the main issues is to find the parameters of this LTP filter: the gain and lag values (β and M). These coefficients can be computed by evaluating the inter-correlation between frames of speech with different lag values and to find the maximum of these inter-correlation values; each maximum determines a lag value. Then the gain can be obtained by the normalization procedure (division of the power of the frame by the maximum inter-correlation found; sometimes an LTP gain of greater than unity can be found). This procedure is known as an open-loop search procedure as opposed to the closed-loop search found in some advanced CELP coders (adaptive codebook for long-term prediction).

Very often, since the frame length of speech coders is generally in the range 160–240 samples and the number of samples between two pitch periods is between 20 and 140, an LTP analysis is done on a subframe basis; this is also due to the fact that the pitch lag varies faster than the vocal tract (LPC filter). Moreover, the pitch lag may be not exactly equal to an entire number of samples, leading to the concept of fractional lags used in the LTP filter. The procedure to find this fractional lag must upsample the signal to be analyzed in order to find a fractional lag; for example, upsampling by a factor of 8 allows us to find a lag with a precision equal to one-eighth of the sampling period (generally for speech, 125 μ s). This fractional lag LTP is much more time-consuming, but it significantly improves the quality of decoded speech.

2.4.4 Vector quantization

Up to now, we have focused on sample-by-sample quantizers. With sample-by-sample, or scalar, quantization, each sample is mapped or rounded off to one discrete element of the codebook. This can be optimized by forming vectors of samples (or other quantities such as LPC or LSP coefficients) which can be quantized jointly in a single operation. Vector quantization is one of the most powerful tools used in modern speech and audio coders. In vector quantization, a block of M samples (or other items such as linear predictive coefficients) forms a vector that is mapped at predetermined points in M -dimensional space and portioned into cells; Figure 2.36 shows the case of bidimensional space.

For scalar quantization, quantization noise is added to each sample to be encoded and decoded; on the other hand, for vector quantization, the noise is concentrated around

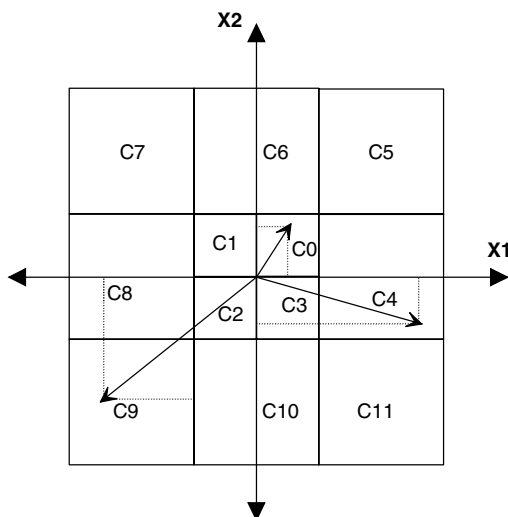


Figure 2.36 Vector quantization. Two-dimensional space for a vector quantizer. Vectors of components X_1 and X_2 are localized in cells C_0 to C_{11} ; the index of the cell is transmitted at the decoder.

the selected vector and correlated for all components. Generally, vector quantization is more efficient than scalar quantization because the codebook can be optimized to use this correlation. For example, in vocoders or source speech coders, such as the LPC 10, independent scalar quantization of the ten LPC coefficients requires about 50 bits per frame (20–30 ms), but vector quantization needs only 25 bits per frame for the same subjective and perceived quality.

This is a significant improvement, but the counterpart is that vector quantization requires much more processing power and is also more sensitive to transmission errors than scalar quantization: an error on one decoded vector impacts all the individual elements of the vector.

There are several types of vector quantization procedures, such as binary, gain shape, split, etc.: in each case the design and optimization of the codebook is of prime importance. Optimizing space partitioning and finding the best vector representatives requires a very large database so that the codebook can be optimized. Distortion measures correlated with human perception and some subjective tests are sometimes required to choose the best codebook.

2.4.5 Entropy coding

This technique is not specific to speech and audio coders, it is also used for most video coders and fax, as well as many file compression tools. The principle of entropy encoding is to map the parameters to be transmitted (e.g., a bit pattern) to code words of variable length, and to use shorter (with a minimum number of bits) **code words** to represent more frequently transmitted parameter values and longer code words for the least used. Huffman codes and RLC (running length code) are some representatives of such codes.

Huffman coding [A21] represents an object with a number of bits that is smaller for objects with larger probabilities. The algorithm builds a binary tree iteratively by first assembling the two objects with the lowest probabilities ω_1 and ω_2 in a node associated with weight $\omega_1 + \omega_2$. The object with the smallest probability ω_1 is located to the left of the node. The new node with weight $\omega_1 + \omega_2$ is added to the collection of objects and the algorithm is restarted (Figure 2.37).

Such an entropy-coding scheme can be placed after a classical speech or audio coder on the bitstream to be transmitted. No additional framing information is required in the encoded bitstream (prefix condition code).

2.5 Waveform coders. ADPCM ITU-T G.726

Waveforms coders are also called temporal speech coders; they rely on a time domain and sample-by-sample approach. Such coders use the correlation between continuous samples of speech and are based on adaptive quantizers and adaptive (generally backward) predictors. They are very efficient in the range 40–24 kbit/s, but quality degrades quickly (around 16 kbit/s).

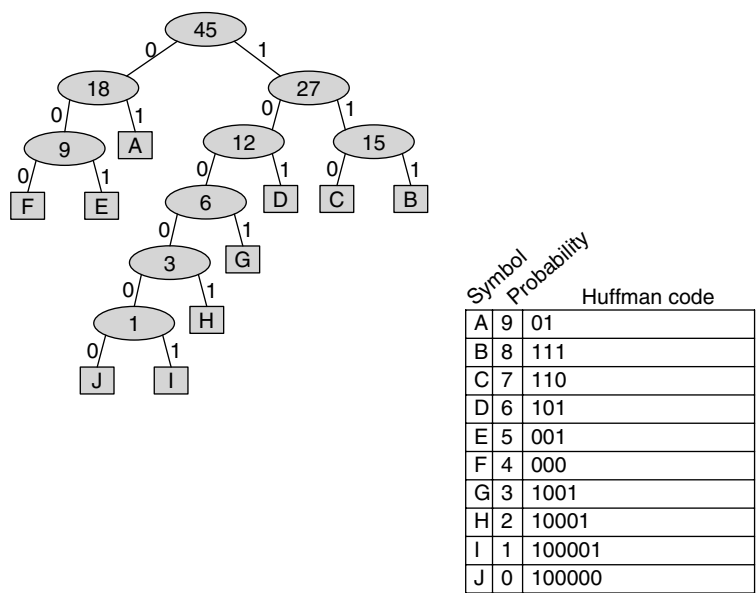


Figure 2.37 Principle of Huffman encoding.

The most widely used standardized waveform coder (excluding ITU-T G.711) is the ADPCM ITU-T G.726 [A13] speech coder which operates at 16, 24, 32,⁴ or 40 kbit/s. The 32-kbit version is used in DECT (digital enhanced cordless telecommunication) wireless phones in Europe, in PHS (personal handy-phone system) phones in Japan, or in **DCME** (**digital circuit multiplication equipment**) device on submarine cables.

ADPCM stands for adaptive differential pulse code modulation; the name itself explains the basic principle of the G.726 speech coder (see Figure 2.38).

The adaptive quantizer is a one-word memory type (or Jayant type) as described in Section 2.2. The adaptive predictor is a mixed structure with six zeros and two poles; it processes the reconstructed signal using a two-coefficient adaptive filter (the poles) and the decoded difference signal using six-coefficient adaptive filter (the zeros).

The basic scheme (Figure 2.38) does not include some useful features such as a dynamic switch for selecting alternative strategies when voice band modem signals are detected in order to allow the ADPCM coder to adapt to modem signals. One of the major drawbacks of coding schemes that reduce the bitrate and rely on the speech characteristics is that they fail for non-speech signals: voice band modem signals are completely synthetic and do not fit the prediction and adaptation procedures tailored for speech signals. The dynamic strategy switch allows transmission of a 9,600-bit/s modem signal for 32 kbit/s ADPCM and a 144,00-bit/s signal (V.33) for 40 kbit/s ADPCM.

The G.726 and its predecessor G.721, standardized in 1984, were the first bit reduction schemes used for civilian telecommunications. It is still one of the most widely used coders

⁴ The old ITU-T G.721 speech coder used in voice storage systems is equivalent to G.726 at 32 kbit/s.

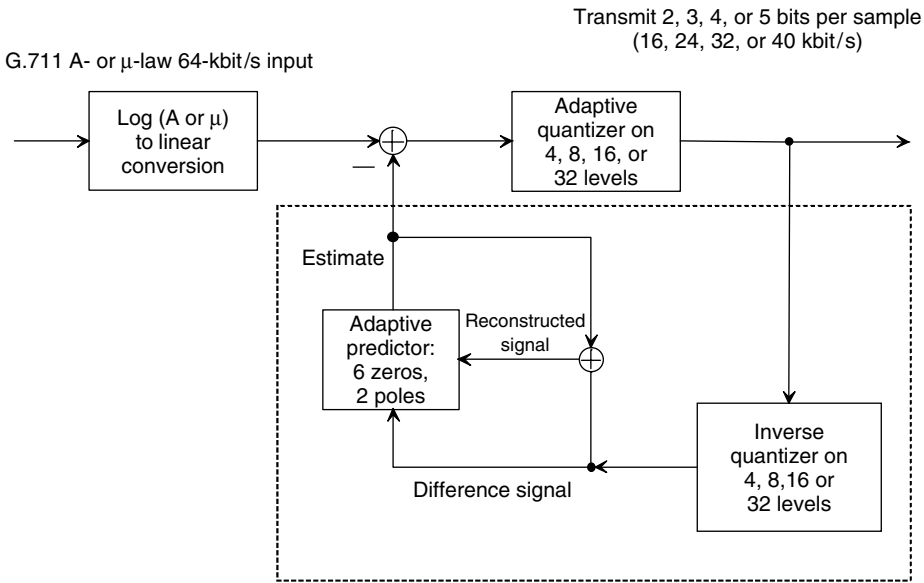


Figure 2.38 ITU-T G.726 ADPCM (16, 24, 32, or 40 kbit/s) basic scheme. The distant decoder is equivalent to the local decoder inside the dotted box.

over terrestrial and submarine cables, in combination with speech interpolation. Speech interpolation relies in the statistical distribution of speech activity on a large number of affluent speech links. In a conventional conversation, each speaker is active less than 50% of the time on each side of the transmission link; the corresponding bandwidth can be used to transmit another voice channel. This becomes even easier with VoIP by using discontinuous speech transmission. Using speech interpolation and ADPCM (G.726 ADPCM at 32 kbit/s) a DCME can achieve a compression gain of 4 to 5.

Due to the symmetrical form of the encoder and decoder (they only differ by their quantizer procedures) of ADPCM, both use a similar processing power of approximately 5 MIPS (16-bit fixed point). Despite this low complexity, the speech quality of G.726 is very good (above 24 kbit/s), as indicated in Figure 2.39.

One interesting feature of the ADPCM coder is its relative immunity to bit errors compared with PCM. As shown in Figure 2.40, there is a significant difference for a **BER (bit error rate)** of 10^{-3} in favor of the ADPCM coder. There are two main reasons: PCM is very sensitive to an error on the sign bit, and ADPCM combines the state variables of the algorithm and consequently, it becomes more robust. This is a typical difference that disappears in VoIP, as errors do not occur as isolated bit errors, but result in complete frame loss (as a packet is rejected if the checksum is wrong).

Although ADPCM coders are not based on a frame-by-frame analysis and speech-coding procedure, in some circumstances (e.g., for voice over IP), ADPCM codes may be transmitted in a packet form. One packet assembles several codes (typically 10–30 ms), each corresponding to one unique sample. In the case of packet loss or ‘frame’ errors, the situation with PCM or ADPCM can be disastrous compared with hybrid or ABS (analysis

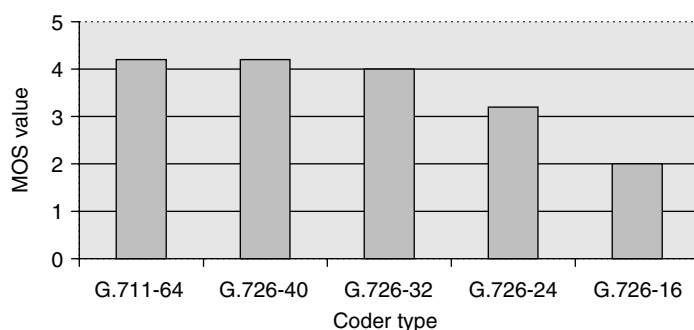


Figure 2.39 Typical MOS scores of common voice coders.

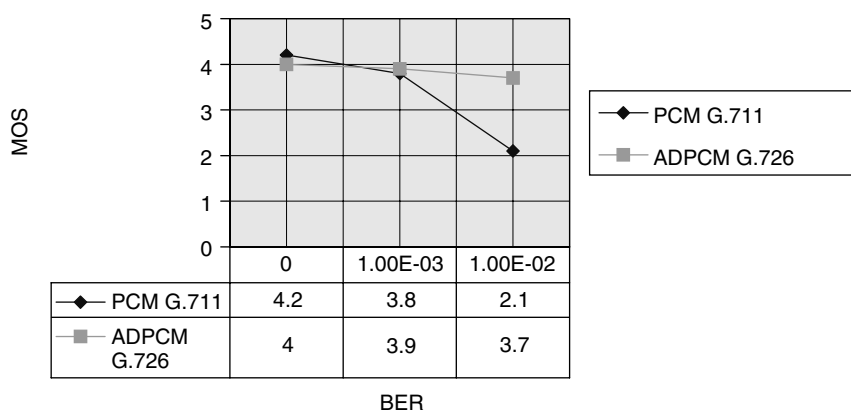


Figure 2.40 Comparison between the BER sensitivity of ADPCM and that of PCM.

by Synthesis) speech coders which can rely on the last valid received parameters (such as LPC and LTP coefficients) to rebuild an approximation of the complete form of the signal for the lost frame. For ADPCM, the loss of many code words breaks the pursuit of the distant decoder against the local decoder, and a long time (250–500 ms) is needed to recover a stable state.

2.5.1 Coder specification . . . from digital test sequences to C code

The G.726 (or more exactly its predecessor G.721) was the first speech coder whose specification included an exhaustive set of digital test vectors. This is required to insure interoperability between equipment built by different manufacturers.

The set of test vectors was required because G.726 did not include a C reference, but an extensive documentation on a fixed point implementation. The fixed point implementation is a strong requirement for economical implementations in DSPs (digital signal processors) or for dedicated VLSI. The ITU-T recommendation includes the exact format (fixed

point) of the variables, constants, state variables, and tables used in the algorithm. It also describes most of the operations required by the algorithm, such as addition, subtraction, fixed point multiplication, and control of possible saturation (which may happen frequently in fixed point arithmetic).

The lack of a reference code was a problem, and later ITU-T introduced reference fixed point ANSI C code for new coders, where all mathematical operations (add, multiply, etc.) are fully specified (this reference implementation is called basic op for 'basic operation'). Today, an ANSI C reference code is the main part of the recommendation of many speech coders, such as ITU-T G.723.1 or G.729. Test vectors are also provided to facilitate the verification of compliance to the standard. These test vectors are designed to provide an extensive coverage of the algorithms used in the implementation for both coding and decoding.

Floating point versions of some algorithms are also useful to improve the quality of implementations in PCs and workstations, and eliminate interoperability issues between fixed point and floating point implementations (e.g., a VoIP gateway using fixed point DSPs and a client PC software using native floating point arithmetic for efficiency). Specific test vectors also help verify the interoperability between different floating point implementations, due to the variety of floating point number representations.

2.5.2 Embedded version of the G.726 ADPCM coder G.727

One desirable feature of a coder is the ability to dynamically adjust coder properties to the instantaneous conditions of transmission channels. This requires some synchronization between the encoder and the decoder when the encoding properties change.

ADPCM can dynamically switch between one of the multiple encoding rates. In this case **embedded** means that a core quantizer is used for the fundamental operations of the coder, and additional quantification bits are allocated to an 'enhancement' quantizer. The scale used by the core quantizer is subdivided to form the scale of the enhancement quantizer. In order to ensure that synchronization is not lost even if some 'enhancement' bits are changed or even not transmitted, the decoder synchronization state is based only on the bits from the 'core' quantizer. This makes it possible to steal or remove some bits in the transmitted code words without desynchronizing the distant decoder, allowing a 'graceful' degradation in the decoded speech without requiring external signaling transmission means. This feature is very useful in applications, such as DCME or PCME (packet circuit multiplication equipment), in overload situations (too many active channels present at the same time) or for 'in band' signaling or 'in band' data transmission.

This concept is used in the embedded version of the G.726 (ITU-T, G.727 recommendation [A1]). In order to insure that the distant decoder tracks the local decoder correctly and due to the fact that this distant decoder may receive code words with robbed bits, the inner loop of prediction relies on the inverse core version of the quantizer:

- On the encoder side, the difference signal is encoded with the full number of steps of the enhanced quantizer, but bits in excess in the enhanced version are masked before feeding the inverse core quantizer.

- On the decoder side, the excess bits of the received code word are masked in order to feed the core inverse quantizer which is used in the prediction and reconstruction inner loop, but the entire received code word enters the enhanced adaptive quantizer, whose output is used to build the final output.

If there are no robbed bits, the output quality is enhanced, but is not as good as if the enhanced version of the quantizer had been used in the inner loop of the encoder and decoder, using all available quantization bits: that is the price to pay for the ‘embedded’ feature.

Figures 2.41 and 2.42 illustrate the G.727 concept.

2.5.3 Wide-band speech coding using a waveform-type coder

2.5.3.1 G.722

In the world of telephony, G.711 is frequently used as ‘the’ reference of voice quality, ignoring the fact that G.711 encodes only the 300–3,400-Hz band. The truth is that it is very difficult to go beyond G.711 quality in traditional telephone networks, because most of the components, from switches to transmission links, assume a G.711 signal (with the exception of transparent ISDN, which is available in some countries).

This is no longer true with voice over IP, where virtually any encoding scheme can be used end to end on the IP network. There are strong requirements to offer a better speech

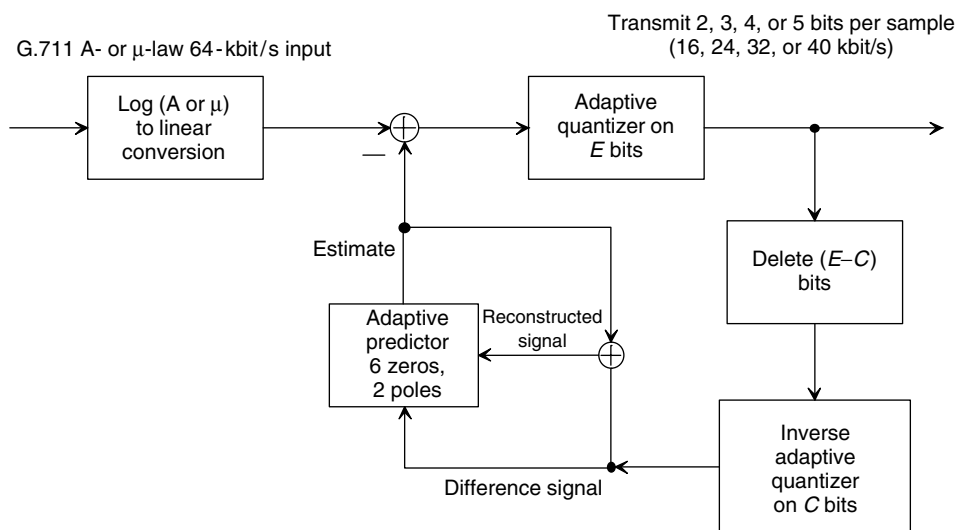


Figure 2.41 G.727 encoder. ITU-T G.727 embedded ADPCM (16, 24, 32, or 40 kbit/s) basic scheme. G.727 is characterized by the enhance and core pairs (E , C) values for quantizers. C can have 2, 3, or 4 as values and E 2, 3, 4, or 5.

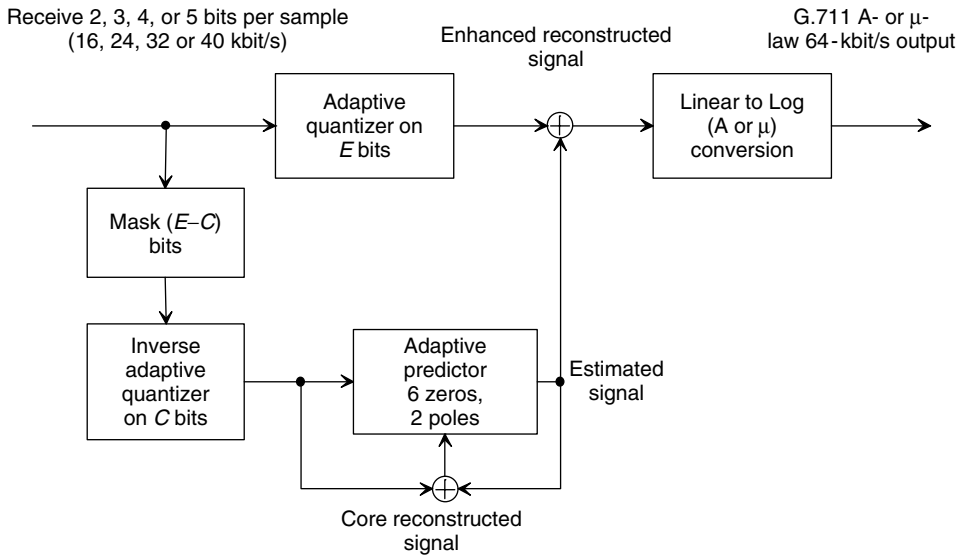


Figure 2.42 G.727 decoder. ITU-T G.727 embedded ADPCM (16, 24, 32, or 40 kbit/s) decoder basic scheme.

and audio quality for videoconference and audioconference systems [A4, A14]. While most coders focus on providing an acceptable voice quality for the lowest possible bitrate, it is also possible to increase the audio quality as much as possible for a given bitrate.

Scientists and engineers were well aware of the possibilities of waveform ADPCM speech coders to reduce the bitrate by a factor of about 0.5 and naturally tried to use a similar technique to encode wide-band speech. Wide band refers to a transmitted frequency band of 50 Hz up to 7,000 Hz compared with the traditional telephony bandwidth (300 Hz to 3,400 Hz).

G.722 was proposed by France Telecom and NTT, and adopted by ITU in 1988. The fundamental idea is to split the band to be transmitted in two subbands: a lower subband spanning from 0 Hz to 4,000 Hz and a higher subband spanning from 4,000 Hz to 8,000 Hz. Then, after a subsampling procedure reducing the sampling frequency from the original 16 kHz down to 8 kHz, two ‘classical’ ADPCM encoders can be applied to reduce the bitrate. Subsampling is possible because subband frequency filtering has eliminated the aliasing effect.

Subband separation uses a pair of quadratic mirror filters. QMF filters are the precursors of the filter bank theory used for psychoacoustic coders.⁵ In many ways the wide-band ITU-T G.722 speech and audio coder is a precursor of the more recent psychoacoustic audio coders: the splitting of the original band into two subbands and the allocation of more bits in the lower subband optimizes the efficiency of the prediction that the most sensitive frequency band performs noise quantization masking. The energy of speech

⁵ These filter banks (with a number of bands from 32 up to 1,024) are intensively used in audio bitrate reduction (ISO-MPEG, AAC, Dolby Digital, etc. [A14]).

signals is more concentrated in the lower subband, and allocating more bits in this subband increases the quality of decoded speech.

G.722 encodes a wide-band signal into a bitstream of 64 kbit/s (the basic PCM bitrate). In the lower subband, 6 bits are used for the adaptive quantizer with an embedded characteristic: the core quantizer uses 4 bits and the enhanced version uses 6 bits. This scheme is very similar to the one found in the embedded version (G.727). This allows the system to steal some bits for signaling purposes (framing with H.221) and to transmit some ancillary data. The decoder should be signaled the mode of operation (64, 56, or 48 kbit/s), although some realizations do not signal the mode and permanently use the full 6 bits. In the higher subband, a 2-bit adaptive quantizer (nonembedded) is used producing a 16-kbit/s bitrate (much lower than the 48 kbit/s used for the lower subband which is perceptually more important).

The coding scheme of G.722 is illustrated in Figure 2.43, and the decoding principle of G.722 is shown on Figure 2.44.

The ITU-T G.722 wide-band speech coder is commonly used in teleconference systems adhering to the H.320 recommendation. The quality is quite good for speech and music at 64 kbit/s and 56 kbit/s (MOS of 4.3 and 4 compared with an original with the same bandwidth rated at 4.3). As there is no specific ‘production model’ (e.g., for speech) in that waveform coder, samples of music are correctly encoded.⁶ When used at 48 kbit/s, reproduced speech becomes more noisy (due to the 4-bit quantizer in the lower subband).

G.722 shares with other waveform ADPCM coder types a relative immunity to bit errors and is more robust than a direct PCM stream. The low-delay characteristic of the G722 is also a major advantage compared with more recent frame-based audio coding schemes. All

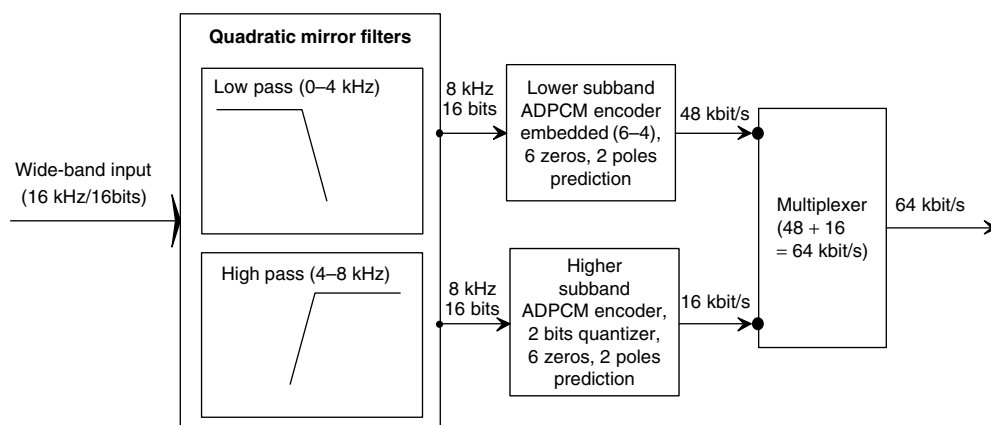


Figure 2.43 G.722 encoder. ITU-T G.722 wide-band encoder, subband ADPCM with QMF filter (48-kbit/s embedded ADPCM in lower subband and 16-kbit/s ADPCM in higher subband).

⁶ Although a bit allocation more favorable to the upper subband, such as 5 bits in the lower band and 3 bits in the higher band, has performed better on many music samples. As the main applications were for teleconference systems, preference was given to the fixed bit allocation strategy that favors speech quality.

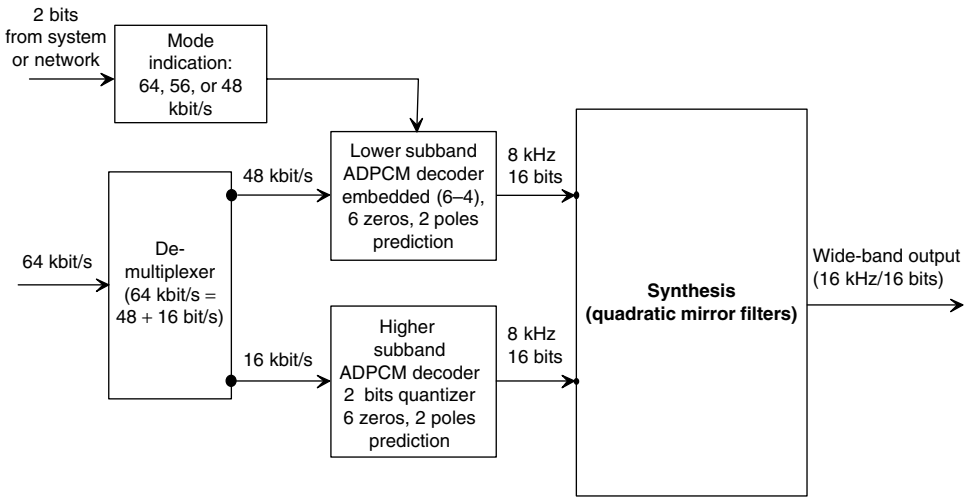


Figure 2.44 G.722 decoder. ITU-T G.722 wide-band decoder, subband ADPCM with QMF filter (48-kbit/s embedded ADPCM in lower subband and 16-kbit/s ADPCM in higher subband).

the waveform coders, such as ADPCM and PCM, have very low algorithmic delay ranging from three to four samples (300–500 μ s with an 8-kHz sampling frequency). In the case of G.722, QMF analysis and synthesis filters add a delay of about 3 ms. The resulting total delay remains excellent and ensures good interactivity for teleconference systems.

G.722 is one of the coders recommended for use in H.323 systems and is available in several commercial implementations.

2.5.3.2 G.722.1

One of the limitations of G.722 is that it cannot be used below 48 kbit/s. The more recent G.722.1 (September 1999) can encode a wide-band signal with a bitrate of 24 kbit/s or 32 kbit/s (a proprietary Pictoretel version exists at 16 kbit/s, called Siren™).

G.722.1 works on frames of 40 ms (640 samples sampled at 16 kHz) with an overlap of 20 ms. On each frame of 40 ms, it multiplies the signal by a sinusoid (therefore the amplitude of the signal at both ends of the frame converges to 0), then performs a **discrete cosine transform (DCT)**. The whole operation is called the **modulated lapped transform (MLT)**; it is illustrated in Figure 2.45.

The result is the encoding of a 20-ms frame using 480 bits at 24 kbit/s and 640 bits at 32 kbit/s. Each frame is encoded independently; there is no state at the receiver. This interesting property prevents frame de-synchronization in the case of frame erasures, typically on VoIP systems. The resulting spectrum is analysed in 16 regions, in order to determine which region is more important (perception model) for the listener. Each frequency region is then quantized and vector-encoded using a Huffman encoding. The more important frequency regions (from a perception point of view) are allocated more bits than the less important frequency regions.

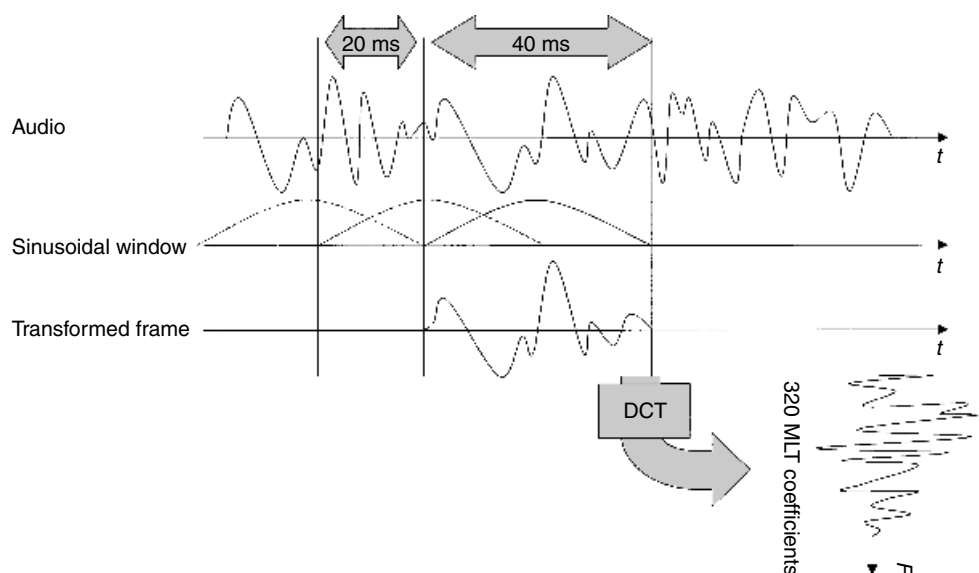


Figure 2.45 Modulated lapped transform used in G.722.1.

This coder uses about 14 MIPS (3% of a Pentium PIII-600) and is supported in the Windows XP® Messenger softphone under the proprietary 16-kbit/s version (Siren™).

2.6 Hybrids and analysis by synthesis (ABS) speech coders

2.6.1 Principle

In previous sections we have studied two types of coders:

- Waveform coders that remove the inter-sample correlation by using linear prediction. The differential coding scheme used with adaptive quantizers gives good performances with a bitrate between 32 kbit/s and 24 kbit/s.
- Linear predictive coders (or vocoders) use a simple model of speech production (voiced or unvoiced types), modeled by a slowly variable filter (updated on a 20–30-ms frame basis) which shapes the spectrum of the decoded speech. LPC coders are used for very low-bitrate speech coders (1,200–2,400 bit/s), but speech quality is low ('synthetic' quality).

Hybrids and analysis by synthesis (ABS) coders combine the best of the two approaches in order to build efficient coding schemes using a bitrate between 6 kbit/s and 16 kbit/s.

ABS coders use a frame of samples to compute the LPC filter coefficients modeling the vocal tract, as well as a long-term predictive (LTP) filter that removes the 'pitch'

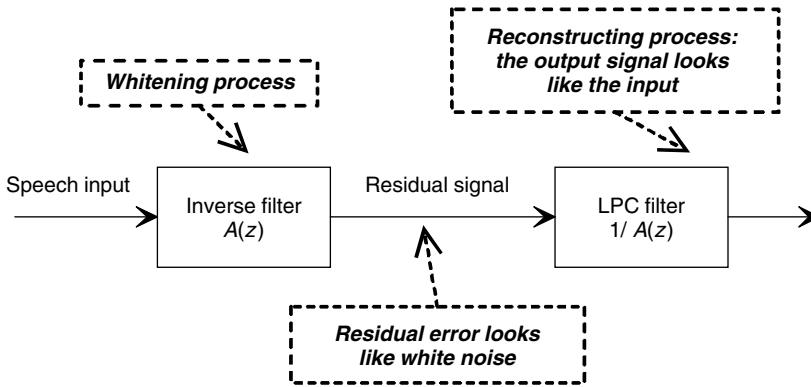


Figure 2.46 The residual error signal after filtering speech by the inverse filter.

correlation. Both LPC and LTP coefficients are encoded (vector quantization is frequently used) and transmitted. But, unlike LPC coders that need to classify the speech type between ‘voiced’ or ‘unvoiced’ and transmit this information, hybrids and ABS speech coders avoid such classification by finding some means of encoding the residual error signal between the inverse LPC/LTP filter (see Figure 2.46) and the original signal.

In **residual excited linear predictive (RELP)** speech coders, the residual signal is fed to a low-pass filter and the resulting signal is classically encoded in PCM form. RELP coders give good results around 10 kbit/s by transmitting the LPC/LTP coefficients and the encoded residual signal. RELP speech coders do not attempt to remove the pitch contribution (they do not apply a dedicated, long-term predictive filter).

Analysis by synthesis (ABS) speech coders use a slightly different method. Instead of encoding the residual error signal (a method focused on the ‘output’), they attempt to compute which excitation input signal to the inverse LPC/LTP filter will result in a decoded speech signal as close as possible to the original signal. The excitation parameters are transmitted to the decoder.

The ABS principle is shown in Figure 2.47.

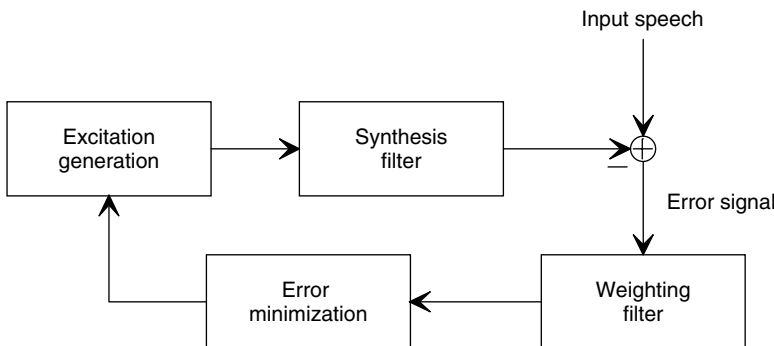


Figure 2.47 Analysis by ABS encoder principle.

The ABS speech coder optimization algorithm finds the ‘best’ vector of configuration parameters for the excitation generator. This best vector is searched by using an error minimization loop based on the perceptual error between the original speech and the synthesized signal. The synthesis filter is a cascade of the inverse LPC filter and inverse LTP filter. ABS coders can be considered both as synthesis filters (LPC/LTP approach) and waveform coders (minimization of a waveform error); they are also called hybrid waveform speech coders. An ABS decoder is very simple as shown on Figure 2.48.

2.6.2 The GSM full-rate RPE-LTP speech coder (GSM 06.10)

The most widely used ABS speech coder is the GSM full-rate codec, standardized by the ETSI in 1988 for the cellular digital mobile system. This coding scheme was proposed by PKI, IBM France, and France Telecom. It uses **regular pulse excitation (RPE)** with **long-term prediction (LTP)**, or **RPE-LTP**, at a bitrate of 13 kbit/s [A16]. The GSM coder feeds the inverse ABS filter with an excitation signal that is optimized to minimize the error signal. GSM uses a series of regular pulses, special cases of ‘multi-pulse’ excitation signals that will be studied later. The choice of RPE to ‘encode’ the residual signal allows for lower complexity implementation compared with general multi-pulse optimization.

In the GSM full-rate coder, the signal is first buffered into a frame of 20 ms (160 samples), then classical LPC analysis finds the eight coefficients that model the vocal tract. These coefficients (also called *parcours* for *partial correlation*) are encoded and transmitted in the bitstream. The entire input buffer is inverse-filtered by the inverse LPC filter, resulting in 160 residual (LPC) samples.

These 160 residual samples are subdivided in four subframes of 40 samples. In each subframe, the algorithm seeks the optimal LTP filter gain and delay. The LTP filter was described in Section 2.4.3. The use of subframes reflects the fact that pitch (which is between 75 Hz and 400 Hz depending on the age and gender of the speaker) varies more rapidly than vocal tract characteristics. The LTP lag and gain are encoded and transmitted for each subframe.

The LTP contribution is then subtracted from the residual signal for each subframe of 40 samples.

This difference signal is then encoded using the RPE procedure, which splits the original 40 samples of the difference signal into four subseries of samples:

- The first starts with the value of sample index 0, then picks one sample value out of 4, from index 3 up to index 36.
- The second starts with index 1, then picks one sample value out of 4, from index 4 up to 37.

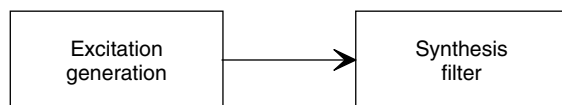


Figure 2.48 ABS decoder principle.

- The third starts with index 2, then picks one sample value out of 4, from index 5 up to 38.
- The last starts with index 3, then picks one sample value out of 4, from index 6 up to the last index of the subframe.

Of the four series, the one that best approaches the original 40 residual samples is chosen; two bits per subframe are required to indicate the choice to the receiver. The maximum energy of the samples in the selected subsequences is also encoded, using 6 bits. All the samples of the subsequence are normalized by this quantized energy, then scalar-quantized with 3 bits. Each series consists of a subsampled process which is a hard low-pass filter with a frequency cutting around 1,300 Hz. This privileges the male voice over female or child voices.

The bit allocation for one frame of the GSM RPE-LTP speech coder is given in Table 2.8. The GSM RPE-LTP encoder principle is shown in Figure 2.49 and the decoder on Figure 2.50.

Although the RPE-LTP yields a speech quality slightly lower than standard telephony it is well suited for mobile communications systems because it resists transmission errors rather well. The MOS figure of the RPE-LTP is around 3.8 compared with 4.2 of the G.711 PCM.

The ETSI 06–10 GSM RPE-LTP recommendation includes a detailed description in fixed point arithmetics relying on the use of ‘basic operators’. Digital test sequences are also given to check conformity to the standard. Although some floating versions of this standard exist and are used in VoIP software, some subtle issues may arise in interoperability with the genuine fixed point version.

In addition to basic speech encoding, a **VAD (voice activity detection)**, **DTX (discontinuous transmission)**, and **CNG (comfort noise generation)** scheme was added to the coder. VAD detects whether valid speech is present and otherwise transmits (less frequently) parameters containing the noise information. In the case of GSM, these parameters are based on the LPC parameters and on the energy of the noise. They are packed in a SID (silence description) frame which is sent every 80 ms (four frames compared with

Table 2.8 GSM full-rate bit allocation

<i>RPE-LTP frame length = 160 samples = 20 ms</i>	
Vocal tract: LPC coefficients; 8 parcors = 36 bits	36
<i>Subframe length = 40 samples = 5 ms (4 subframes)</i>	
Grid selection = 2 bits	8
Maximum of energy of selected series = 6 bits	24
Scalar quantization of 13 samples = $13 * 3 = 39$ bits	156
LTP lag = 7 bits	28
LTP gain = 2 bits	8
Total	260
<i>Bit rate = $260 / 20 \text{ ms} = 13 \text{ kbit/s}$</i>	

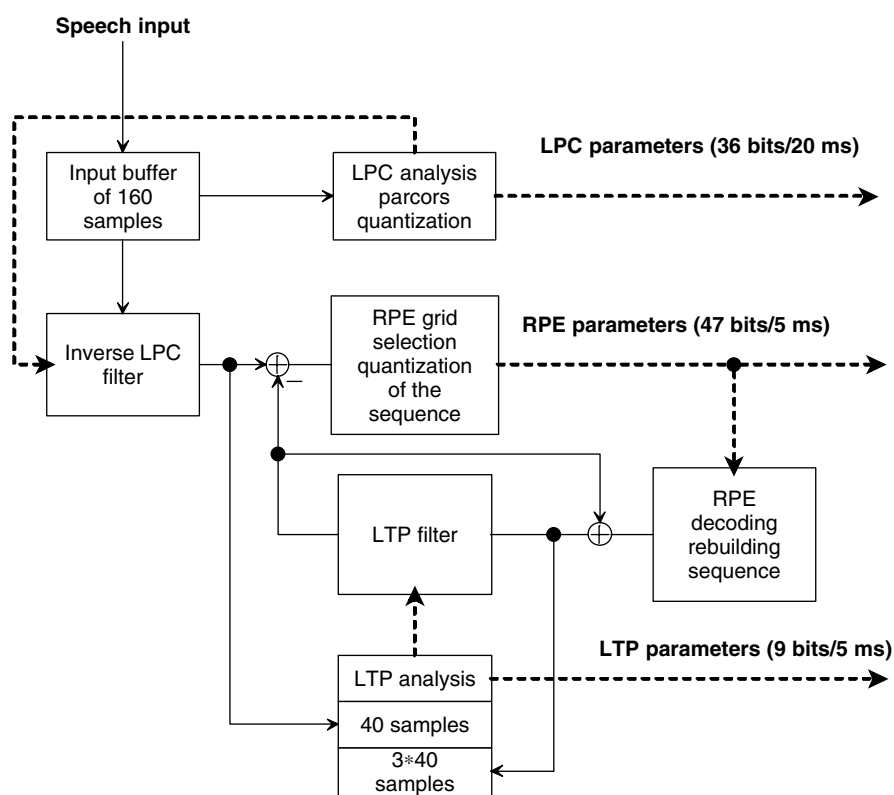


Figure 2.49 Basic principle of the RPE-LTP full-rate (13-kbit/s) GSM speech coder.

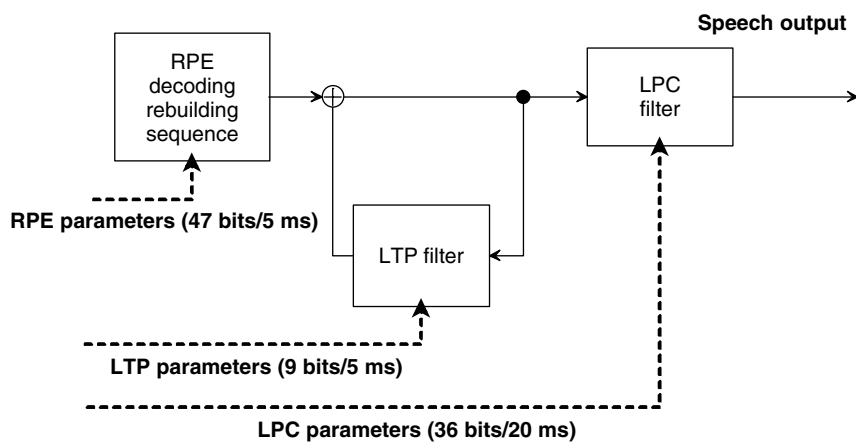


Figure 2.50 Basic principle of the RPE-LTP full-rate (13-kbit/s) GSM speech decoder.

the 20-ms speech frame). It must be pointed out that the design of a good and efficient VAD algorithm is almost as complex as the design of good speech coder.

The GSM 6.10 coder reflects the constraints of the processing power commonly available in 1988; it is being progressively replaced by GSM 6.60.

The GSM 6.60 coder is based on the ACELP technology proposed by Nokia and the University of Sherbrooke. It only uses 12.2 kbits/s (less than the 13 kbit/s of GSM 6.10, leaving some capacity for error protection). When there are no errors on the transmission channel, the voice quality is equivalent to G.726 at 32 kbit/s (toll quality).

2.7 Codebook-excited linear predictive (CELP) coders

CELP coders are in essence linear predictive coders equipped with an ABS search procedure. They were invented in the 1980s by Bell Labs (under the supervision of B.S. Atal and M.R. Schroöder). As we have already seen, once the short-term correlation in the signal has been removed by the LPC filter and the long-term correlation (or pitch contribution) has been removed by the LTP filter, the quality of reproduction depends essentially on the selection of an optimal excitation signal.

A possible choice is a multi-pulse excitation signal. The position and amplitude of each pulse are searched iteratively using an ABS algorithm. The main pulse position is searched first, then the algorithm locates the optimal second pulse, and so on. The coder bitstream must encode the position and amplitude of each pulse modeling the excitation. Note that the regular pulse solution used in the GSM full rate is a particular case of the multi-pulse excitation signal, which significantly decreases the computing power required for computation of a general multi-pulse excitation signal.

The optimization of a multi-pulse signal is very complex in general, because the number of candidate vectors is very large. In CELP coders, a codebook based on vector quantization is built, trained, and optimized off-line on a large ‘speech’ database. Only these vectors are used as candidates for the excitation generator that feeds the LTP and LPC synthesis filters. The excitation signal (index in the codebook and value of gain) that best approximates the original speech input signal is selected according to a perceptual error criterion.

The role of the perceptual filter is to redistribute noise in frequency ranges where it will be less audible due to the higher energy of the main signal: the noise will be masked by the signal itself. Significant improvements of the subjective quality [A3] are observed when using this **perceptual weighting filter**. The filter $W(z) = A(z)/A(z/\gamma)$, with a bandwidth expansion coefficient γ less than 1, forces the noise to be reinforced in the neighborhood of the formants and to be lowered in the region where the signal is weak. Although absolute noise power is generally increased, listeners generally prefer this situation.

One big issue with CELP coders is the difficulty of finding the best index and associated gain in the codebook, as the codebook is very large. For a long time, this has been a barrier to practical implementation in real time. Algorithmic simplifications brought to the initial design (efficient codebook search or algebraic codebooks) and the growth

of available MIPS (million instructions per second) in modern DSPs have finally made it possible to implement CELP coders in real time.

The basic scheme of a CELP coder is shown on Figure 2.51.

LPCs are first computed and quantized for an entire frame of speech (10–30 ms). Vector quantization and line spectrum pairs are increasingly used due to their efficiency. LTP lag and gain are searched and quantized on a subframe basis as well as the codebook index and associated gain G_i .

The decoder is much less complex than the encoder (there is no ABS search procedure) and can include an optional post-filter as shown in Figure 2.52.

In order to improve perceived quality, the post-filter aims at reducing the noise level in frequency bands located between the maxima of the spectrum (located near the harmonics). A typical implementation is a short-term post-filter which is derived from LPCs in a similar way as the perceptual weighting filter in the encoder. Modern post-filters can also include a long-term prediction post-filter and a tilt compensation post-filter. The introduction of the post-filter can significantly increase the MOS rating of CELP decoders; nevertheless, it may affect the fidelity of decoded speech if its action is exaggerated.

The basic scheme for a CELP encoder relies on an open-loop search for the long-term correlation coefficients of the LTP filter. A more advanced implementation refines this procedure by first conducting an open-loop search for an LTP lag, then testing fractional lags in the neighborhood of this initial lag in an adaptive codebook. The chosen value is selected by an **ABS-MSE (mean square error)** procedure.

The remaining components (called innovations) of the residual signal are nonpredictable, and a best matching excitation vector is searched in another codebook, called the stochastic codebook. The design of the stochastic codebook, which models samples that more or less resemble noise, is complex. There are two main approaches. The first

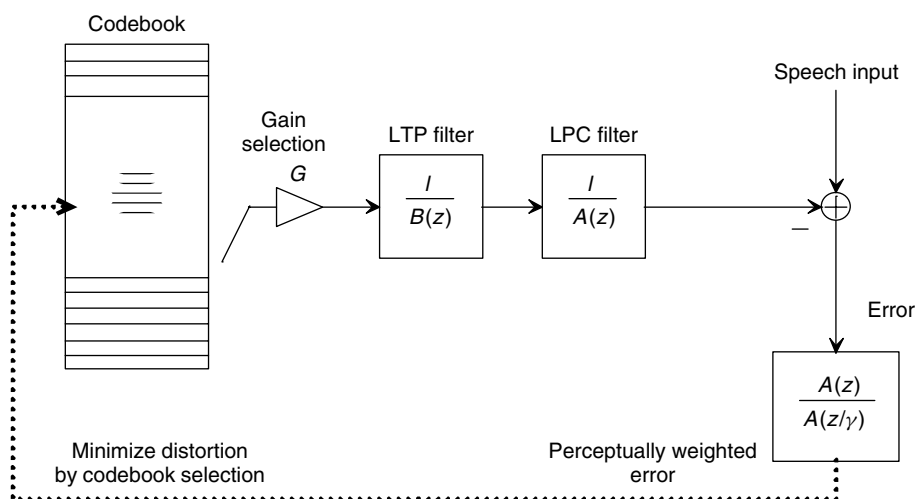


Figure 2.51 Basic concept of a CELP coding algorithm. The quantized LTP and LPC parameters are transmitted on a frame basis. The quantized gain G and the codebook index are transmitted (sometimes on a subframe basis).

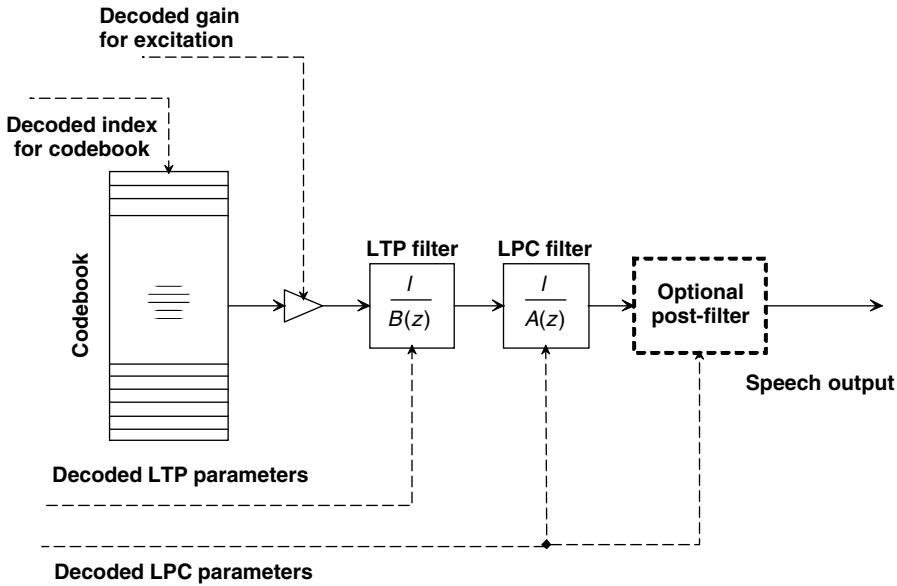


Figure 2.52 Basic concept of a CELP decoding algorithm.

is to build the codebook before the execution phase of the encoder by using training and optimization on large speech databases. The second is based on a predetermined set of patterns, which are combined, resulting in the optimal excitation vector (see Section 2.7.1 on G.729 for an example). The optimal combination is computed during the ABS mean square error procedure (e.g., selection of the pulse location and associated gain). The latter method is used, for example, in ACELP (algebraic CELP) or MP-MLQ (multipulse maximum likelihood quantization).

The algorithm is therefore based on a closed-loop search in two codebooks:

- The adaptive codebook which is devoted to long-term prediction.
- The stochastic codebook which deals with those components in the residual signal that are nonpredictable.

The closed-loop search selects four parameters:

- (1) An index in the stochastic codebook.
- (2) An optimal gain corresponding to the index selected in the stochastic codebook.
- (3) A lag (integer or fractional) in the adaptive codebook.
- (4) An optimal gain corresponding to the selected lag value.

The optimal excitation search for the LPC synthesis filter is therefore modified as shown in Figure 2.53.

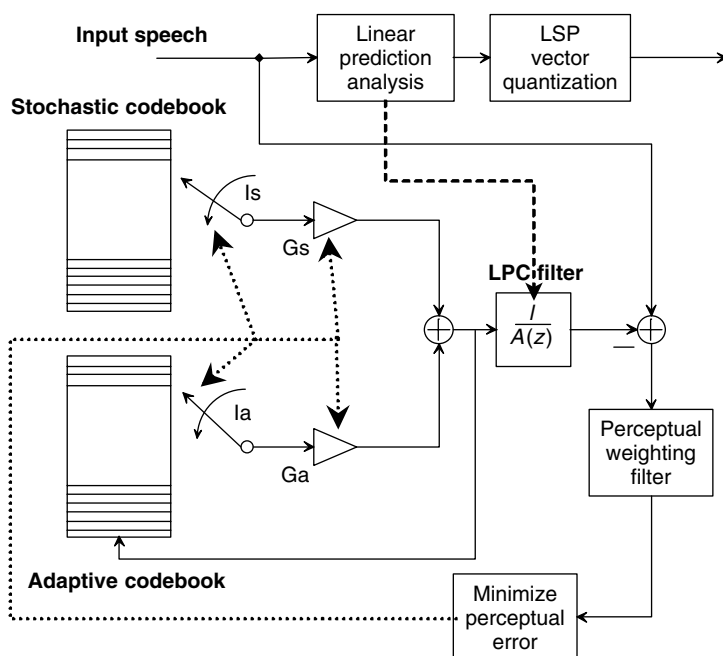


Figure 2.53 Advanced concept of a CELP encoding algorithm. I_s , G_s , I_a , G_a , and the LPC (LSP) parameters are transmitted.

Using such algorithms (ABS with stochastic and adaptive codebooks, and LSP vector quantization), CELP speech coders excel in the range 4.8–16 kbit/s. Many international standards in that range of bitrates are CELP or derivative CELP speech coders:

- Federal Standard 1016 4800 bit/s CELP [A17].
- ITU-T 8-kbit/s G.729 CS-ACELP and dual-rate multimedia ITU-T G.723.1 (5.3 kbit/s and 6.3 kbit/s, ACELP, MP-MLQ).
- ITU-T low-delay CELP ITU-T 16-kbit/s G.728. In order to fulfill the stringent requirement of low delay, a long LPC backward-adaptive filter is used in place of the LPC and LTP classical filters; no LPCs are transmitted to the decoder side and only the index vector and associated gain is transmitted.
- ETSI enhanced full-rate GSM speech coder and the half-rate GSM speech coder, as well as the AMR and WB-AMR coders.

2.7.1 ITU-T 8-kbit/s CS-ACELP G.729

The ITU-T G.729 [A18] (Conjugate Structure Algebraic CELP) was proposed by the University of Sherbrooke, France Telecom, NTT, and ATT. It has a frame length of 10 ms with two subframes of 5 ms. The short-term analysis and synthesis are based on

tenth-order linear prediction filters. Due to the short frame length of 10 ms, LSPs (line spectral pairs) are quantized by using a fourth-order moving average (MA) prediction. The residue of linear prediction is quantized by an efficient two-stage vector quantization procedure (the CS used in the coder name refers to this). An open-loop search for the lag of the LTP analysis is made to select the initialization value for the closed-loop search in each subframe. Pitch predictor gain is close to unity, but the fixed codebook gain varies much more. This gain is estimated by a fourth-order MA gain predictor with fixed coefficients, from the sequence of previous, fixed codebook excitation vectors. This is the main difference between the G.729 encoder scheme and the one described on Figure 2.53; this gain predictor appears in the decoder scheme in Figure 2.54.

The lag and gain of the LTP filter, the optimal algebraic codebook and the fixed algebraic excitations are jointly vector-quantized using 7 bits.

The innovation codebook is built by combining four pulses of amplitudes +1 or -1. The locations of the four pulses are picked from a predetermined set as shown in Table 2.9.

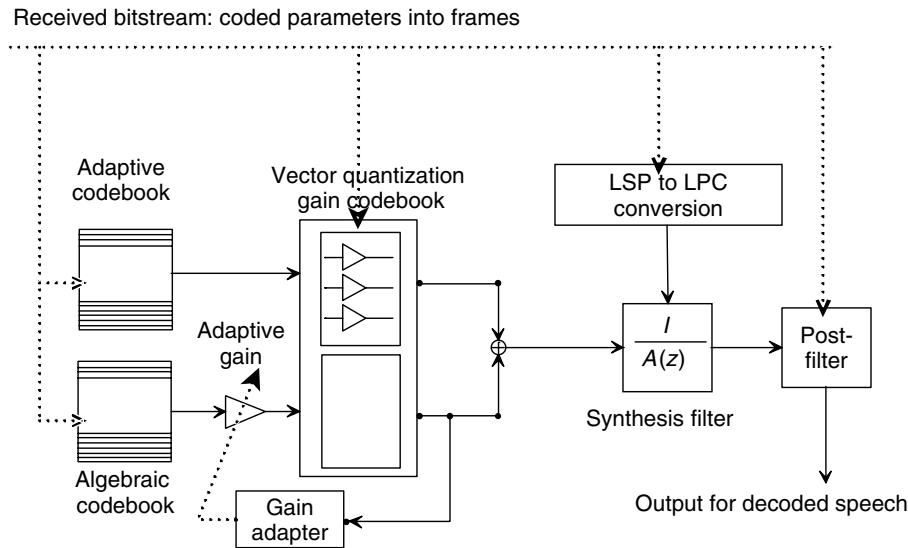


Figure 2.54 Basic principle of the ITU-T G.729 CS-ACELP 8-kbit/s speech decoder.

Table 2.9 Predetermined set of pulses of the innovation codebook used by G.729

Amplitude	Positions of pulses
±1	0, 5, 10, 15, 20, 25, 30, 35
±1	1, 6, 11, 21, 26, 31, 36
±1	2, 7, 12, 17, 22, 27, 32, 37
±1	3, 8, 13, 18, 23, 28, 33, 38
	4, 9, 14, 19, 24, 29, 34, 39

Table 2.10 G.729 bit allocation

Parameter	Subframe of 40 samples		Frame of 80 samples
	1st	2nd	
LSP	—	—	18
Pitch delay	8	5	13
Pitch parity	1	—	1
Algebraic code	13 + 4	13 + 4	34
Gain codebook	4 + 3	4 + 3	14
Total	—	—	80

The pulse positions of the first three pulses are encoded with 3 bits (eight possibilities) and the position of the fourth pulse is encoded with 4 bits (16 possibilities). Each pulse also requires 1 bit to encode the amplitude (± 1). This gives a total of 17 bits for the algebraic codebook in each subframe. Since only four nonzero pulses are in the innovation vector, very fast search procedures are made possible. Four nested loops corresponding to each pulse are used.

The structure of the final bitstream at 8 kbit/s is given in Table 2.10.

The G.729 decoder includes a post-filter consisting of three filters: a long-term post-filter, a short-term post-filter and a **tilt** compensation post-filter. The structure of the G.729 decoder is shown in Figure 2.54.

The ITU-T G.729 includes a detailed description in both fixed and floating point (annex C) with associated digital test vectors. Annex B describes a VAD/DTX/CNG scheme similar to G.723.1 (which was designed before G.729).

G.729 is recommended for use in voice over frame relay systems under the name clear voice. G.729 uses 16 MIPS. G.729 annex A is a lower complexity version (10 MIPS for the encoder compared with 18 MIPS) which was initially designed and recommended for **DSVD (digital simultaneous voice and data systems)**, but is now widely used in VoIP systems. G.729 also defines extensions at 6.4 kbit/s (annex D) and 11.8 kbit/s (annex E) which target **DCME** and **PCME** applications.

2.7.2 ITU-T G.723.1: dual-rate speech coder for multimedia communications transmitting at 5.3 kbit/s and 6.3 kbit/s

2.7.2.1 Speech encoding

G.723.1 is the result of an ITU competition for an efficient speech-coding scheme at a low bitrate for videoconferencing applications using a 28.8-kbit/s or 33.4-kbit/s V.34 voice band modem; this resulted in a compromise between the two best candidates (Audiocodes and DSP Group on one side and France Telecom on the other). This explains the two models of innovation codebooks found in the standard: the MP-MLQ (Audiocodes) for the higher bitrate and the ACELP (University of Sherbrooke) for the lower bitrate. There are some subtle differences between the general, advanced, CELP speech-coding scheme presented previously and the G.723.1 general structure, but the

basic principles and algorithmic tools are the same. The excitation signal for the high-rate coder is multi-pulse maximum likelihood quantization (MP-MLQ) and for the low-rate coder it is algebraic code-excited linear prediction (ACELP, the principle used in G.729 and GSM EFR).

The frame size is 30 ms and there is an additional look-ahead of 7.5 ms, resulting in a total algorithmic delay of 37.5 ms. Subframe duration is 7.5 ms. The MP-MLQ block vector quantization resembles the algebraic vector quantization procedure: six pulses with sign ± 1 for even subframes and five pulses with sign ± 1 for odd subframes are searched with an ABS MSE procedure. There is also a restriction on pulse positions: the positions can either be all odd or all even (indicated by a 'grid bit'). For the lower bitrate, the ACELP codebook was tuned to fit 5.3 kbit/s.

Tables 2.11 and 2.12 give the bit allocation for the two bitrates. The 189 bits of the higher bitrate are packed in 24 bytes and the 158 bits of the lower bitrate are packed in 20 bytes. Depending on the selected rate, either 24 or 20 bytes must be sent every 30 ms. Two bits in the first byte are used for signaling the bitrate and for the VAD/DTX/CNG operations described in Section 2.7.2.2.

The ITU-T recommendation includes a 16-bit, fixed point, detailed description and a floating point reference program (annex B). Both are provided as ANSI C programs. For the floating point version, software tools were designed to allow implementers to check their realizations. Conformance to the standard can be checked by undertaking all the digital test sequences. The complexity in fixed point for the encoder and both bitrates is around 16 MIPS. Annex C, devoted to mobile application, includes some mobile channel error-coding schemes.

G.723.1 is—together with G.729—one of the most well known coders used in VoIP networks and is predominantly used in PC-based systems. While most embedded systems (such as network gateways) support both G.729 and G.723.1, some of the leading IP phone vendors unfortunately recently decided to stop supporting G.723.1. This situation makes the lives of network administrators difficult, since many PC to IP phone calls can only negotiate G.711 as the common coder.

Table 2.11 Bit allocation table for the 6.3-kbit/s G.723.1 encoder (MP-MLQ)

Parameters coded	Subframe 0	Subframe 1	Subframe 2	Subframe 3	Total
LPC indices					24
Adaptive codebook lags	7	2	7	2	18
All the gains combined	12	12	12	12	48
Pulse positions	20	18	20	18	73(Note)
Pulse signs	6	5	6	5	22
Grid index	1	1	1	1	4
Total:					189

Note: By using the fact that the number of code words in the fixed codebook is not a power of 2, three additional bits are saved by combining the four MSBs of each pulse position index into a single 13-bit word.

Table 2.12 Bit allocation table for the 5.3-kbit/s G.723.1 (ACELP)

Parameters coded	Subframe 0	Subframe 1	Subframe 2	Subframe 3	Total
LPC indices					24
Adaptive codebook lags	7	2	7	2	18
All the gains combined	12	12	12	12	48
Pulse positions	12	12	12	12	48
Pulse signs	4	4	4	4	16
Grid index	1	1	1	1	4
Total:					158

2.7.2.2 Discontinuous transmission and comfort noise generation (annex A)

In order to reduce the transmitted bitrate during silent periods in-between speech, silence compression schemes have to be designed. They are typically based on the voice activity detection (VAD) algorithm and a comfort noise generator (CNG) that reproduces an artificial noise at the decoder side. The VAD must precisely detect the presence of speech and send this information to the decoder side. The G.723.1 VAD operates on a speech frame of 30 ms, and includes some spectral and energy computations.

One interesting feature of the VAD/DTX/CNG scheme of the G.723.1 coding scheme is that, when the characteristics of environmental noise do not change, nothing at all is transmitted. When needed, only the spectral shape and the energy of the comfort noise to be reproduced at the decoder side are sent. The spectral shape of the noise is encoded by LSP coefficients quantized with 24 bits and its energy with 6 bits. With the two mode-signaling bits, this fits in 4 bytes. The two signaling bits in each packet of 24, 20, or 4 bytes indicates either a 24-byte 6.3-kbit/s speech frame, a 20-byte 5.3-kbit/s speech frame, or a 4-byte CNG frame. The G.723.1 can switch from one bitrate to the other on a frame-by-frame basis (each 30 ms). At the decoder side, four situations can appear:

- (1) Receiving a 6.3-kbit/s frame (24 bytes).
- (2) Receiving a 5.3-kbit/s frame (20 bytes).
- (3) Receiving a CNG frame (4 bytes).
- (4) Receiving nothing at all (untransmitted frame).

In situations (1) to (3), the decoder reproduces the speech frame or generates the comfort noise signal with parameters indicated in the CNG frame. In situation (4), the decoder incorporates some special procedures to reproduce a comfort noise based on previously received CNG parameters. Similar VAD/DTX/CNG schemes have been included in G.729 and its annexes.

2.7.3 The low-delay CELP coding scheme: ITU-T G.728

In general, CELP coders cannot be used when there is a requirement for a low-encoding algorithmic delay. This is due to the LPC modeling principle, which requires a frame length of 10–30 ms (average stationary period of the speech signal) to compute the LPC.

Traditional low-delay encoders, such as PCM and ADPCM waveform speech coders, introduce a very low delay and do not significantly impact network planning (introduction of electrical echo cancellers). Unfortunately, they do not work at low bitrates.

The ITU was looking for a relatively low-bitrate encoder (16 kbit/s), with a low algorithmic delay (maximum 5 ms).

The G.728 low-delay coding scheme was designed by AT&T [A20], which efficiently merged the two concepts of stochastic codebook excitation (CELP) and backward prediction. In that scheme, there is no need to transmit the LPCs' which are computed in both the encoder and decoder, in a backward loop. Since backward prediction works on the current frame of samples from data of the previous samples, a relatively long set of samples can be analyzed to optimize the LPC filter without requiring a long frame to be accumulated before transmission.

The synthesis filter used in the ABS-MSE loop procedure does not include any LTP filter, but, in order to correctly represent high pitch values (and to efficiently encode generic signals such as music), its length is extended to 50 backward coefficients, updated every 20 samples. The coefficients are not transmitted but adapted (computed) in a backward manner by using the reconstructed signal in the encoder and decoder.

The frame length for the innovative codebooks is equal to only 5 samples (0.625 ms). For each set of 5 samples, an index found in the stochastic codebook of 128 entries is transmitted with a sign bit and a gain coded on 2 bits. In order to obtain an optimized codebook structure (the vectors), a very long and time-consuming training sequence on a large speech signal database was necessary. The gain is not directly encoded on 2 bits: a linear predictor is used to predict the gain, and the error of the optimal gain versus the predicted gain is encoded and transmitted. This leads to Table 2.13, the bit allocation table for the LD-CELP G.728. The LD-CELP speech encoder principle is shown on Figure 2.55.

In order to increase resistance to transmission errors, the index of the codebooks is transmitted using Gray encoding. Unlike a normal binary system, Gray encoding ensures that adjacent integers only have a single bit of difference: while a bit error can result in a large error on the integer value, a bit error in Gray encoding minimizes the error in the

Table 2.13 G.728 bit allocation

	Bit allocation per frame		Bitrate (bit/s)
	Parameters	Numbers of bits	
Excitation	Index	7	11,200
	Gain	2	3,200
	Sign	1	1,600
Frame length: 0.625 ms (5 samples)			16,000

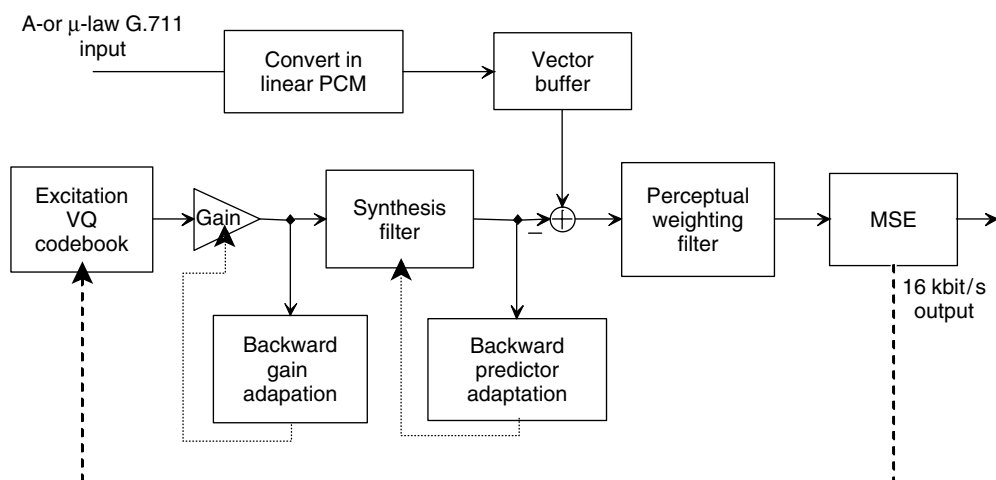


Figure 2.55 Low-delay CELP ITU-T G.728 encoder principle.

encoded value. For instance, integers 0 to 15 are encoded as 00, 01, 11, 10, 110, 111, 101, 100, 1100, 1101, 1111, 1110, 1010, 1011, 1001, 1000.

The introduction of the post-filter in the decoder shown in Figure 2.56 significantly improves the quality of decoded speech (this has allowed the AT&T proposal to fulfill the ITU-T requirements). G.728 has a very good score on the MOS scale (around 4) and is used in the H.320 videoconference system to replace the G.711 64 kbit/s with an identical quality bitstream of 16 kbit/s, leaving almost 48 kbit/s for the video on a single ISDN B channel. G.728 is also used in some modern DCME (digital circuit multiplication equipment), with extensions to 9.6 kbit/s and 12.8 kbit/s (replacing G.726 at 16 kbit/s, 24 kbit/s, and 32 kbit/s).

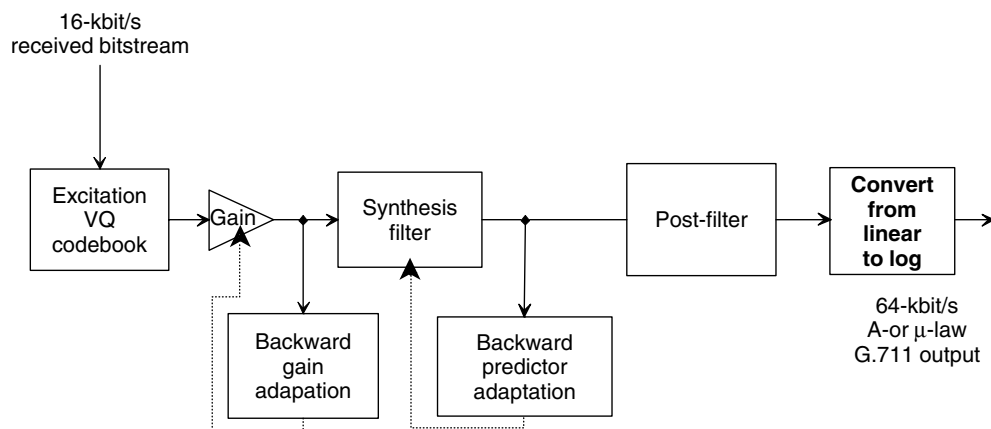


Figure 2.56 Low-delay CELP ITU-T G.728 decoder principle.

The major weaknesses of the original LD-CELP coding scheme are the difficulty to handle voice-band modem signals (an extension to 40 kbit/s is defined in annex I to solve this problem) and the high sensitivity to frame erasure due to the very long backward LPC filter and the use of a gain adaptation predictor. Recent work has significantly improved robustness and led to a new annex in the ITU-T G.728 suite of recommendations.

Another issue that G.728 shares with the G.729 coder (as opposed to the G.723.1 coder) is that there is no framing information in the transmitted bitstream. G.723.1 uses 2 bits in the first transmitted byte to indicate the type of packet. G.728 produces a 10-bit code for each 5-sample frame, but the decoder must precisely know which is the first, second, third, and fourth packet of 10 bits in order to synchronize the backward LPC filter adaptation procedure (although speech remains intelligible with G.728 even if desynchronization occurs). Strictly speaking, the use of G.728 requires a delay of 4 frames ($4 \times 0.625 \text{ ms} = 2.5 \text{ ms}$).

In the H.320 suite of recommendations, the H.221 framing procedure specifies a positioning mechanism for four packets of 10 bits of G.728 (2 bits per byte of a 64-kbit/s stream) or 80 bits of the G.729 8-kbit/s stream (1 bit per byte of a 64-kbit/s stream).

The first detailed description introduced in 1992 was for a floating point DSP and two additional years of work were needed to finalize a fixed point (16-bit) description in annex G. Unfortunately, the description is not in the form of ANSI C code, but extensive documentation.

The complexity of G.728 in fixed point is around 20 MIPS for the encoder and 13 MIPS for the decoder.

2.7.4 The AMR and AMR-WB coders

The adaptive multi-rate (AMR) coder is the result of ongoing work by ETSI (European Telecommunications Standards Institute) and 3GPP (3rd Generation Partnership Project, founded in December 1998), in collaboration with T1 in the US, TTC (Telecommunication Technologies Committee) and ARIB (Association of Radio Industries and Businesses) in Japan, TTA (Telecommunication Technologies Association) in Korea, and CWTS (China Wireless Telecommunication Standard group) in China, for the third generation of cellular telephony systems. In the current generation of cellular systems, three voice coders are used:

- GSM-FR, standardized in 1987, produces a 13-kbit/s bitstream and provides relatively good quality, with good immunity to background noise.
- GSM-HR, standardized in 1994, reduces the bitrate to 5.6 kbit/s, but is much more sensitive to background noise, which prevented any significant deployment.
- GSM-EFR, standardized in 1996, enhances the voice quality of GSM-FR in the presence of background noise with a similar bitrate (12.2 kbit/s), but the enhancement is perceptible only on error-free transmission channels.

While most voice coders seek to optimize the bitrate for a given quality of transmission channel for a desired voice quality level, so far little work has been done to take into

account the variable quality of a transmission channel. On most wire lines, it is true that the quality of transmission lines does not vary significantly, but obviously this is not the case for wireless transmission channels. With VoIP, the network conditions experienced by a PC-based phone, for instance, may also vary widely depending on whether the connection is via Ethernet, WiFi, at the office, or at a hotel. When the quality of the transmission channel varies, the optimal allocation of bits between source encoding and channel encoding varies: as the quality of the transmission link decreases, it becomes more efficient to allocate more bits to error protection schemes and fewer bits to the source encoding algorithm. Instead of optimizing the next generation coder for a given bitrate or transmission quality, it was decided that the new coder should be able to adapt to variable conditions (a 'multi-rate' coder) and provide optimal behavior under all these conditions ('adaptive'). The goals were:

- To improve the quality of GSM-FR on a channel with transmission errors, for a similar bitrate.
- To provide acceptable quality even on half-rate transmissions, in order to enhance transmission density in case of congestion.
- To adapt dynamically to the conditions of the radio channel.

The narrow-band AMR coder was standardized in March 1999; in addition, it was decided to study a version of the AMR coder for wide-band audio encoding (**AMR-WB**), which was finally standardized in March 2001.

2.7.4.1.1 Narrow-band AMR (GSM 6.90 ACELP AMR)

Narrow-band AMR provides eight bitrates (kbit/s):

- 7.95; 7.4 (IS 136); 6.7 (PDC-EFR); 5.9; 5.15 and 4.75 for half-rate transmission (similar to GSM-HR).
- 12.2 (GSM-EFR) and 10.2 for full-rate transmission (similar to GSM-FR and for UMTS).

Three of these modes interwork with existing equipment:

- GSM-EFR in 12.2-kbit/s mode.
- DAMPS in 7.4-kbit/s mode.
- PDC-EFR in 6.7-kbit/s mode.

Each mode is associated with a channel encoder which adds redundancy and interlacing in order to fill the available channel capacity (22.8 kbit/s for full rate, 11.2 for half-rate). On GSM networks, only four modes can be signaled (which can change dynamically at each even frame) and each service provider must select which modes are optimal for his network. While the network decides which mode to use depending on the conditions, the mobile terminal can signal its preferences. On UMTS networks, mobile terminals have to implement all eight modes.

The AMR coder is a CELP coder using ten LPCs. The various bitrate modes differ essentially in the number of bits allocated to quantization of the post-LPC residual signal: 38 bits for 12.2-kbit/s mode, 26 bits for 7.4-, 6.7-, and 5.9-kbit/s modes, and 23 bits for 5.15- and 4.75-kbit/s modes. All modes use an LTP filter to remove the pitch contribution, with some small precision differences depending on the mode (one-third precision for most modes). All AMR modes also use a post-filter to enhance the perceptual quality of the reproduced signal. Manufacturers of AMR devices have a choice of two algorithms for the VAD (one from Ericsson and Nokia, the other from Motorola); both reformed similarly during testing. The algorithm for the correction of erased frames was left out of the normative standard, although one example algorithm is provided. This provides some room for implementers to improve the quality of their algorithms and differentiate.

Besides the dynamic mode switching that optimizes bit allocation between source coding and channel encoding, the AMR also supports **unequal bit error detection and protection (UED/UEP)**. UED/UEP allows the loss of fewer frames over a network with a high bit error rate. Obviously, this has no impact on VoIP, since all errors are frame erasures.

2.7.4.1.2 AMR-WB (ITU G.722.2, UMTS 26171)

AMR-WB has been selected by 3GPP (TS 26.171) for UMTS phase 5 and was standardized by ITU as G.722.2 in January 2002. (A coder G.722.1 proposed by Picturitel was also standardized with similar characteristics, but it did not meet all the desired criteria for a 3G wideband codec). The AMR-WB algorithm was proposed by VoiceAge, Ericsson, and Nokia. It mainly targets three types of applications:

- GSM with a full-rate channel with a source-encoding rate limited to 14.4 kbit/s (TRAU frame).
- GSM-FR and EDGE with a full-rate channel with a source + channel-encoding rate limited to 22.8 kbit/s.
- UMTS with a source rate limited to 32 kbit/s.

The design goals of AMR-WB included:

- A voice quality at 16 kbit/s equal or superior to G.722 at 48 kbit/s.
- A voice quality at 24 kbit/s equal or superior to G.722 at 56 kbit/s.

AMR-WB provides nine bitrates (kbit/s):

- 14.25, 12.65, 8.85, and 6.6 for GSM-TRAU applications.
- 19.85, 18.25, and 15.85 are also available for GSM-FR applications.
- 23.85 and 23.05 are also available for EDGE and UMTS applications.

The AMR-WB coder jointly encodes the 0–6,400-Hz subband and the 6,400–7,000-Hz subband. The lower subband is processed by a CELP algorithm using a 16-coefficient LPC

filter, with the residual signal encoded using 46 bits for all modes except 6.6-kbit/s mode (36 bits). LTP filter analysis is extended to the full band or limited to the lower subband depending on the mode. The higher subband of the signal is regenerated by filtering a white noise signal with an LPC filter deduced from transmitted LPCs. One VAD algorithm has been standardized (annex A). As in the case of the AMR narrow-band coder, a frame erasure correction algorithm is provided, but is not part of the normative standard.

While AMR is mandatory for all terminals, the AMR-WB coder is mandatory only for terminals capable of sampling voice at 16 kHz; this will be introduced in UTMS phase 5. In multimedia communications, only AMR can be used during circuit switching, while both AMR and AMR-WB can be used for packet-switched communications (phase 5).

2.8 Quality of speech coders

Most speech coders have been designed to achieve the best possible level of speech reproduction quality, within the constraints of a given source-encoding bitrate. For narrow-band coders, the reference is ‘toll quality’, or the quality of speech encoded by the G.711 coder. For wide-band coders (transmitting the full 50–7,000-Hz band), the reference is the G.722 coder.

In fact, assessing the quality of a speech coder is a complex task which depends on multiple parameters:

- The absolute quality of the reproduced speech signal. This is the most used figure, but does not take into account interactivity (i.e., the delay introduced by the speech coder in a conversation). Several methods exist to assess the absolute, noninteractive speech quality of a coder. We will describe the MOSs which are the result of the ACR (absolute category rating) method and the DMOSs obtained with the CCR (comparative category rating) method. Several environmental conditions may influence speech degradation and need to be taken into account, such as speech input level, the type and level of background noise (bubble noise, hall noise, etc.).
- The delay introduced by the coder algorithm (algorithmic delay). This delay is due to the size of the speech signal frames that are encoded and to the additional signal samples that the coder needs to accumulate before encoding the current frame (look-ahead). Obviously, delay is only relevant for interactive communications, not for voice storage applications or noninteractive streaming applications.
- The complexity of the coder, which will result in additional processing delay on a given processor.
- The behavior of the coder for music signals, modem signals (maximum transmission speed that can be obtained), and DTMF transmission.
- Tandeming properties (i.e., the number of times voice can be encoded and decoded before voice quality becomes unacceptable). This can be assessed with the same coder used repeatedly or with other well known coders (e.g., the GSM coders used in cellular phones).

- Sensibility to errors (bit errors for cellular or DCME applications, or for VoIP frame erasures).
- The flexibility of the coder to dynamically adapt bit allocation to congestion and degradation of the transmission channel. Some coders provide only a fixed bitrate, while others can switch between bitrates dynamically (**embedded** coders). Hierarchical coders like G.722 can generate several simultaneous streams of encoded speech data: a core stream that needs to be transmitted as reliably as possible through the transmission channel (either on a high QoS level or using an efficient redundancy mechanism), and one or more ‘enhancement’ streams that can be transmitted on lower quality channels.

The importance of these parameters depends on the final application and the target transmission network (fixed network, wireless network, serial transmission links that generate bit errors, packet transmission links that generate frame erasure errors, etc.). For most applications, the shortlist of key parameters includes the bitrate, complexity of the coder, delay, and quality.

When a standard body decides to standardize a new voice coder, the first step is to specify the quality acceptance criteria for the future coder. As an example, Table 2.14 [A19] is a summary of the terms of reference set to specify the ITU 8-kbit/s coder (the future G.729). This new coder was intended to ‘replace’ the G.726 at 32 kbit/s or the G.728 at 16 kbit/s.

2.8.1 Speech quality assessment

In order to assess the level of quality of a speech coder, **objective measurements** (computed from a set of measurements on the original signal and the reproduced signal) are not reliable for new coders. In fact, most objective, automated measurement tools can only be used on well-known coders and well-known networks, and simply perform some form of interpolation using quality scores in known degradation conditions obtained using a subjective method. In the early days of VoIP, people tended to apply the known, objective measurement tools, calibrated on fixed TDM networks, without realizing that transmission link properties were completely different: frame erasure as opposed to random bit errors, correlated packet loss, degradations due to the dynamic adaptation of jitter buffers, etc. Needless to say, many of these ‘objective’ tests were in fact designed as a marketing tool for this or that voice coder.

Subjective measurements are therefore indispensable. What can assess the quality of a voice coder better than a human being? Unfortunately, subjective measurements of speech quality require a substantial effort and are time-consuming. In order to obtain reliable and reproducible results, a number of precise guidelines must be followed:

- Ensure that the total number of listeners is sufficient for statistically reliable results.
- Ensure that the auditory perception of listeners is normal (medical tests may be necessary).
- Instruct the listeners of the methodology of the tests.
- Ensure that the speech material is diversified: gender of talkers, pronunciation, age of the talkers (child).

Table 2.14 Terms of reference for the 8 kbit/s ITU-T speech coder

Items	Parameters	Requirements	Objectives
Quality for speech		No worse than that of ITU-T G.726 at 32 kbit/s	
Performance in the presence of transmission errors (bit error)	Random bit error: BER ≤ 0.1	No worse than that of ITU-T G.726 at 32 kbit/s under similar conditions	Equivalent to ITU-T G.728
Performance in the presence of frame erasure	Indication of frame erasure (random and burst)	Less than 0.5 MOS when there are 3% missing frames	As small as possible
Input-level dependence	- 36 dB, -16 dB	No worse than that of ITU-T G.726 at 32 kbit/s	As small as possible
Algorithmic delay		≤ 16 ms	≤ 5 ms
Total codec delay		≤ 32 ms	≤ 10 ms
Cascading		2 asynchronous coding ≤ 4 asynchronous ITU-T G.726 at 32 kbit/s	3 asynchronous coding ≤ 4 asynchronous ITU-T G.726 at 32 kbit/s
Tandeming with other ITU-T standards		≤ 4 asynchronous ITU-T G.726 at 32 kbit/s	3 asynchronous coding ≤ 4 asynchronous ITU-T G.726 at 32 kbit/s
Sensibility to background noise	Car noise	No worse than that of ITU-T G.726 at 32 kbit/s	
	Bubble noise		
	Multiple speakers		

- Ensure that the test is performed in several languages by a number of experienced organizations (problems may occur with languages other than English, Japanese, German, Italian, Spanish, or French even on a well-standardized speech coder),
- Ensure that all the environmental conditions of use of the candidate coder are tested (such as level dependencies, sensibility to ambient noise and type of noise, error conditions, etc.).
- Appropriate choice of pertinent listening conditions: choice of equipment (headphones, telephone handsets, loudspeakers) and loudness of the samples.

These tests are fully specified in ITU-T recommendations ITU-T P.800 and P.830 [A1], [A4]. Obviously, these tests are very time-consuming, expensive (dedicated rooms and studios, high-quality audio equipment), and require well-trained and experienced organizations.

The following subsections provide an overview of these methods. We will focus on listening opinion tests, although other tests, such as conversation opinion tests, also exist.

2.8.2 ACR subjective test, mean opinion score (MOS)

For low-bitrate telephone speech coders (between 4 kbit/s and 32 kbit/s), the **absolute category rating (ACR)** is the most commonly used subjective measurement method. It is the method that produces the well-known **MOS** figure.

In ACR subjective tests, listeners are asked to rate the ‘absolute’ quality of speech samples, without knowing what the reference audio sample is. Listening quality is generally assessed using the scale in Table 2.15.

An MOS is an absolute judgment without references, but in order to insure coherence and calibration between successive tests, some reference is needed. For this purpose, a reference audio sample is inserted among the samples given to listeners (without any notice). Very often, the **modulated noise reference unit (MNRU)** is used: this device simulates voice degradation and noise level equivalent to that produced by the A- or μ -law PCM coding scheme. It is still common to read press articles or conference presentations that present ‘the’ MOS of a new coder without also presenting the MOS obtained in the test by a reference coder. Such values should be considered with skepticism: some vendors choose to give an MOS of ‘5’ to G.711, shifting all MOSs up by almost one full MOS point, while others do not even have such a reference coder as part of their test.

Table 2.15 Listening quality scale for absolute category rating

Excellent	5
Good	4
Fair	3
Poor	2
Bad	1

The MOS figure is calculated statistically from the marks given to each audio sample by listeners. The relevance of MOS and the confidence interval of the results must be determined by statistical analysis, requiring a lot of experiments. Generally, an ACR subjective test requires an average of 24 listeners (3 groups of 8). The typical test sample consists in a double sentence: 0.5 s of silence, 2 s for sentence #1, 0.5 s of silence, 2 s for sentence #2.

Figure 2.57 provides an overview of typical MOS values for various categories of speech coders as a function of bitrate [A6]. More precisely, Table 2.16 gives the MOS figure and type of well-known, ITU-T standardized speech coders. For mobile standards see Table 2.17 and for DOD standards see Table 2.18.

Table 2.18 clearly shows the magnitude of the improvements in speech coders in ten years: the speech quality that can be obtained at 2.4 kbit/s goes from synthetic to 3.2 (fair)!

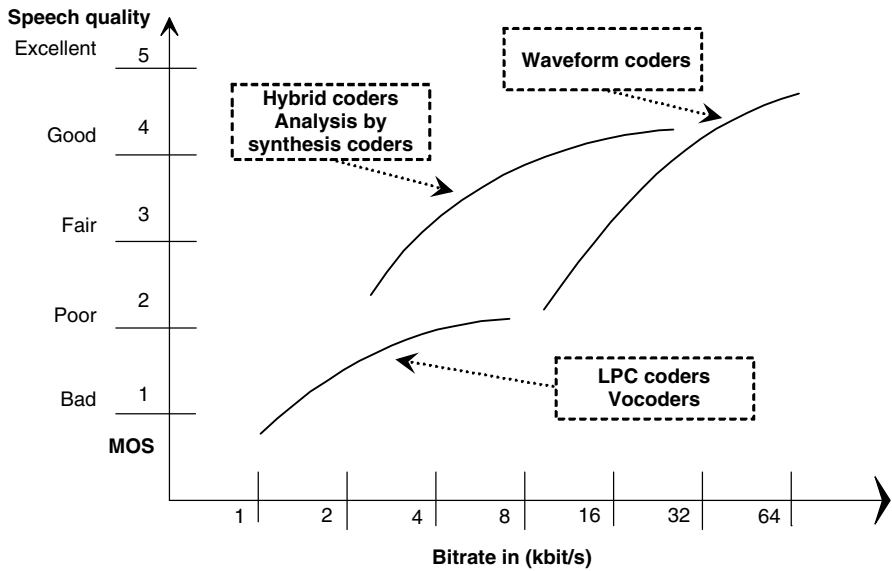


Figure 2.57 MOSs as a function of the bitrate and coder technology.

Table 2.16 MOSs of some ITU coders

Standard	G.711	G.726 or G.721	G.728	G.729	G.723.1
Date of approbation	1972	1990 (1984)	1992	1995	1995
Bitrate (kbit/s)	64	16/24/32/40	16	8	6.3–5.3
Type of coder	Waveform: PCM	Waveform: ADPCM	ABS: LD-CELP	ABS: CS-ACELP	ABS: MP-MLQ, CS-ACELP
Speech quality (MOS)	4.2	2/3.2/4/4.2	4.0	4.0	3.9/3.7

Table 2.17 MOSs of coders used in mobile telephony

Standard	ETSI-GSM 06.10	ETSI-GSM 06.20	ETSI-GSM-EFR	ETSI-TETRA	USA IS54 TDMA	USA IS96 CDMA	JAPAN JDC 1	JAPAN JDC 2
Date of approbation	1988	1994	1995	1994	1989	1992	1990	1994
Bitrate (kbit/s)	13	5.6	13	4.56	7.95	8/4/2/1	6.7	3.6
Type of coder	ABS: RPE-LTP	ABS: VSELP	ABS: ACELP	ABS: ACELP	ABS: VSELP	ABS: Qualcomm CELP	ABS: VSELP	ABS: PSI-CELP
Speech quality (MOS)	3.6–3.8	3.5–3.7	4	3.3–3.5	3.5–3.7	3.3–3.5	3.4–3.6	3.4–3.6

Table 2.18 MOS scores of military coders

Standard	American DOD FS1015	American DOD FS1016	American DOD
Date of approbation	1984	1990	1995
Bitrate (kbit/s)	2.4	4.8	2.4
Type of coder	Vocoder: LPC 10	ABS: CELP	ABS: MELP
Speech quality (MOS)	Synthetic quality	3	3.2

2.8.3 Other methods of assessing speech quality

ACR is not the only method available for speech quality assessments. The **degradation category rating (DCR)** and the **comparison category rating (CCR)** are also used, mostly for high-quality coders.

The DCR method is preferred when good-quality speech samples are to be compared. The DCR method produces a **degradation mean opinion score (DMOS)**. The range of degradation is presented Table 2.19.

DCR methodology is similar to ACR, except that the reference sample is known to the listener and presented first: pairs of samples (A–B) or repeated pairs (A–B, A–B) are presented with A being the quality reference.

CCR is similar to DCR, but the order of the reference sample and the evaluated coder sample is chosen at random: this method is interesting mostly for speech enhancement systems. The result is a **comparison mean opinion score (CMOS)**.

For all interactive communication systems, especially VoIP, conversational tests are also very instructive because they try to reproduce the real service conditions experienced by final users. The degradations introduced by echo and delays, not present in MOS tests, can also be taken into account. The test panel is asked to communicate using the system under test (e.g., DCME or VoIP) and is instructed to follow some scenario or to play some game and finally give their opinion on the communication quality and on other parameters, such as clarity, level of noise, perception of echoes, delays, interactivity, etc. Once again, each participant gives a score from 1 to 5 (as described in the ITU-T P.800 recommendation), and statistical methods are used to compute the test result (**MOS_c**, ‘c’ for communication) and the confidence interval. Interactive tests are very difficult to control, and consistency and repeatability are very hard to obtain.

An example of the sample conditions used in international experiments conducted by ITU-T when selecting a 8-kbit/s candidate is given in Table 2.20.

Table 2.19 DMOS table

Degradation is inaudible	5
Degradation is audible but not annoying	4
Degradation is slightly annoying	3
Degradation is annoying	2
Degradation is very annoying	1

Table 2.20 Typical ITU experiments for coder selection

Experiment #	Description
Experiment 1	Clean speech quality and random bit error performance
Experiment 2	Tandem connection and input-level dependence
Experiment 3	Frame erasure: random and burst
Experiment 4	Car noise, bubble noise, multiple speakers, and music
Experiment 5	Signaling tones: DTMF, tones, etc.
Experiment 6	Speaker dependence: male, female, child

2.8.4 Usage of MOS

As MOSs represent a mean value, extreme care must be taken to select or promote a speech coder for a specific application. It must be checked that all the candidate coders are evaluated under the same conditions (clean speech, level dependence, background noise of several types and level of noise, sensibility to bit errors, frame erasure, etc.) and that the test conditions actually represent the real conditions of the communication channels used by the application. International bodies, such as the ITU-T, TIA, ETSI, JDC, are well aware of the situation and evaluate each coder according to a rigorous and thorough methodology. Too often, manufacturers publish and promote MOS results that have no scientific value. A few examples of common tricks are:

- Publishing good MOS results with a high network loss rate (10%!), but with a carefully engineered loss pattern that does not represent the real situation (e.g., exactly one packet out of 33 is lost, as opposed to the correlated packet loss in a real network).
- Taking a higher MOS value for the reference coder, but omitting this detail in the final test documentation.
- Using test samples free from background noise.
- Using listening equipment of low quality that smooths the perception of all coders and therefore boosts the results of the tested coder after normalization.

2.9 Conclusion on speech-coding techniques and their near future

2.9.1 The race for low-bitrate coders

Many coding schemes have not been described in this chapter:

- The MELP (mixed excitation LPC) coder, retained in the new 2,400-bit/s US Federal standard.
- The VSELP (vector sum excited LP) coder, used in the half-rate 5.6-kbit/s GSM system.

- The multi-rate Q-CELP (Qualcomm CELP) at 1, 2, 4 and 8 kbit/s, used in the cellular US IS96 CDMA system.
- Multi-band excitation (MBE) coders.
- Sinusoidal transform coders (STCs).

The number of coding schemes reflects the constant progress of speech-coding technology. This progress has been driven by major telecommunication applications.

The first application of voice coding was the optimization of submarine cables and expensive long-distance links. The focus was on reducing bitrate while preserving good voice quality, and on providing reasonable support for modem and fax transmission. This led to relatively simple voice coders like the ITU G.726 at 32 kbit/s (1990).

Since 1990 the bitrate required to reach toll quality has decreased to about 8 kbit/s, or one bit per sample!

2.9.2 Optimization of source encoding and channel encoding

After 1999, the priority was no longer the absolute reduction of the bitrate, because the price of bandwidth continuously decreased on fixed lines. The driving application for voice-coding technology became wireless telephony. Wireless telephony offers a limited transmission bandwidth, which can be addressed by existing algorithms, but more importantly the transmission quality of the transport channel varies continuously. The best voice quality depends not only on how good the source encoding of the voice coder is, but also and, just as importantly, on how well channel encoding can correct transmission errors.

The priority of coder research became the optimal combination of source-encoding and channel-encoding methods in a given envelope. Both compete for the available bitrate on the channel:

- If the number of errors is low, the channel-encoding algorithm is not necessary and does not generate any redundancy information, and the full available bitrate can be used for the voice coder (source encoding).
- If the number of errors is high, the channel-encoding algorithm will generate a lot of redundancy information to protect voice coder information. As a consequence, the source-encoding algorithm needs to reduce its bitrate.

Dynamic optimization of the source-encoding and channel-encoding allocation within the available bitrate is a complex problem. The AMR and AMR-WB coders are the result of research carried out on this problem: both use multiple source-encoding algorithms, each combined with a channel-encoding algorithm, and the optimal mode is switched dynamically as transmission conditions change.

This new generation of voice coders provides a much more homogenous experience over a varying quality radio channel: voice quality does degrade as the radio conditions of the transmission channel get worse, but does so progressively, without the catastrophic

degradation experienced with single-mode codecs. Dynamic selection of the optimal source coder and channel coder makes the best possible use of the transport link under any conditions.

To a large extent, the enhancements of voice coders that were originally designed for radio channels are also valid for VoIP. The only significant difference, on is that radio channels create bit errors in the data stream (characterized by a **bit error rate, or BER**), while IP networks create frame-level (packet-level) errors. For a given bitrate, VoIP can also benefit from an optimal combination of source encoding and channel encoding, but the optimal channel-encoding method for VoIP differs from the optimal channel-encoding method for wireless applications, as it must protect against frame erasures.

2.9.3 The future

2.9.3.1 VoIP

What should we expect next? Perhaps the most important feedback from early VoIP trials was that there was no market for sub-toll-quality voice. Users are not only not prepared to pay less for such voice quality, they are not prepared to pay at all. As a consequence there are no big incentives to continue to decrease the bitrate of a pure voice coder, and IP overheads would make such progress irrelevant anyway. Although there is still some progress for voice coders to make at 4 kbit/s and below, none of these coders achieves toll quality, and therefore they can only be used in degraded conditions, in combination with a high-redundancy channel-encoding method, or in military applications.

One of the issues about current coders is their poor performance for the transport of music, another is the degradation of voice encoding when there are multiple speakers or background noise. It seems that most of the efforts in the coming years will be to improve these weaknesses, while keeping a bitrate of 8 kbit/s or even above.

Unlike wireless networks, which will always have a tight bandwidth constraint (shared medium), VoIP applications benefit from the constant progress of wired transmission links. As the cost of bandwidth decreases, it becomes more interesting to provide users with a better telephony experience. Some VoIP systems already support wide-band voice coders, such as G.722, which make it easier to recognize the speaker and provide a more natural sound. Beyond wide band, multichannel coders (stereo, 5.1) can provide spatialized sound, which can be useful for audio- or videoconferences. Since 2002, it seems the focus of voice encoding for VoIP systems has shifted from low-bitrate encoders to these high-quality wide-band encoders.

We believe that in the coming years wideband encoders will become increasingly common in VoIP systems.

2.9.3.2 Broadcast systems

Both wireless telephony and VoIP are interactive, one-to-one systems, where there is only one transmission channel. Audio broadcast systems on the Internet pose a different problem. For such systems there are many transmission channels, each with different degradation levels. If a separate information stream is sent on each channel (multi-unicast),

then dynamic mode selection works, but for multicast systems, where everyone receives the same information, it would not be optimal to use the bit allocation between source encoding and channel encoding that is optimal for the worst channel, as even listeners using the best transmission channels would experience poor audio quality.

For such links, the focus is on hierarchical coders, which produce several streams of information: a core stream, providing low quality, is transmitted with the highest possible redundancy (and an above-average QoS level is available), one or more enhancement streams that provide additional information on top of the core stream information, allowing receivers to improve playback quality. ISO-MPEG 4 is an example of a hierarchical encoder. These systems are mainly useful for broadcast of multicast systems, and their use for VoIP (e.g., with H.332) mainly depends on the deployment of multicast-capable IP networks.

2.10 References

2.10.1 Articles

- [A1] ITU-T. G.7xx, P.3xx, H.3xx Recommendations, ETS ETSI xxxx, IS ISO-MPEG yy.
- [A2] P. Combescure and M. Mathieu. Codage des signaux sonores. *L'Écho des recherches*, **121**(3), 1985, pp. 13–22.
- [A3] P. Combescure, A. Le Guyader, D. Jouvét, and C. Sorin. Le traitement du signal vocal. *Annales des Télécommunications*, **50**(1), 1995, pp. 142–164.
- [A4] A. Gersho. Advances in speech and audio compression. *Proceedings of the IEEE*, June 1994, pp. 900–918.
- [A5] S. Dimoultsas. Standardizing speech coding technology for network application. *IEEE Communications Magazine*, November 1993, pp. 26–33.
- [A6] R.V. Cox. Speech coders: From idea to product. *AT&T Technical Journal*, March/April 1995, pp. 14–21.
- [A7] L.M. Supplee, R.P. Cohn, and J.S. Collura. MELP: The new federal standard at 2400 BPS. *Proceedings of ICASSP Conference*, 1997, pp. 1591–1594.
- [A8] T.E. Tremain. The government standard linear predictive coding algorithm: LPC 10. *Speech Technology*, April 1982, pp. 40–49.
- [A9] Y. Mahieux, J-P. Petit, and A. Charbonnier. Codage pour le transport du son de haute qualité sur le réseau de télécommunications. *L'Écho des recherches*, **146**(4), 1991, pp. 25–35.
- [A10] K.H. Brandenburg, G. Stoll, Y.F. Dehéry, J.D. Johnston, L.D. Kerkof, and E.F. Schröder. ISO-MPEG-1 audio: A generic standard for coding of high quality digital audio. *Journal of the Audio Engineering Society*, **42**, October 1994, pp. 780–792.
- [A11] M. Bosi, K.H. Brandenburg, S. Quackenbush, L. Fielder, K. Akagiri, H. Fuchs, M. Dietz, J. Herre, G. Davidson, and Y. Oikawa. ISO/IEC MPEG-2 advanced audio coding. *Journal of Audio Engineering Society*, **45**(10), October 1997, pp. 789–814.
- [A12] L.D. Fielder, M. Bossi, G. Davidson, M. Davis, C. Todd, and S. Vernon. AC-2 and AC-3: Low-complexity transformed-based audio coding. *Collected Papers on Digital Audio Bit-Rate Reduction*, pp. 54–72. Editors N. Gilchrist and C. Grewin. Audio Engineering Society.
- [A13] W.R. Daumer, P. Mermelstein, X. Maître, and I. Tokizawa. Overview of the ADPCM coding algorithm. *Proceedings of GLOBECOM*, 1984, pp. 23.1.1–23.1.4.

- [A14] P. Noll. Wide band speech and audio coding. *IEEE Communications Magazine*, November 1993.
- [A15] X. Maitre. 7 kHz audio coding within 64 kbit/s. *IEEE Journal on Selected Areas on Communications*, **6**(2), February 1988, pp. 283–298.
- [A16] K. Hellwig, P. Vary, D. Massaloux, J.P. Petit, C. Galand, and M. Rosso. Speech codec for the European mobile radio system. *GLOBECOM Conference*, 1989, pp. 1065–1069.
- [A17] J.P. Campbell, T.E. Tremain, and V.C. Welsh. The federal standard 1016 4800 bps CELP voice coder. *Digital Signal Processing*, **1**, 1991, pp. 145–155.
- [A18] R. Salami, C. Laflamme, J.P. Adoul, A. Kataoka, S. Hayashi, T. Moriya, et al. Design and description of CS-ACELP: A toll quality 8 kbit/s speech coder. *IEEE Transactions on Speech and Audio Processing*, **6**(2), March 1998, pp. 116–130.
- [A19] Special features on ITU-T standard algorithm for 8 kbit/s speech coding. *NTT Review*, **8**(4), July 1996.
- [A20] J.H. Chen, R.V. Cox, Y.C. Lin, N. Jayant, and M.J. Melchner. A low-delay CELP coder for the CCITT 16 kb/s speech coding standard. *IEEE Journal on Selected Areas on Communications*, **10**(5), June 1992, pp. 830–849.
- [A21] D.A. Huffman. A method for the construction of minimum-redundancy codes. *Proc. Inst. Radio Eng.*, **40**, 1952, pp. 1098–1101.

2.10.2 Books

- [B1] A.M. Kowdoz. *Digital Speech: Coding for Low Bit Rate Communications Systems*. Wiley.
- [B2] N. Moreau. *Techniques de compression des signaux*. Masson, CNET-ENST, 1995.

2.11 Annexes

2.11.1 Main characteristics of ITU-T standardized speech coders

Standard	G.711	G.721	G.726	G.727	G.728	G.729	G.723.1	G.722	G.722	G.722.2 (AMR-WB)
Approval date	1972	1984	1990	1990	1992	1995/99/99	1995	1988	1999	2002
Bitrate (kbit/s)	64	32	16/24/32/40	16/24/32/40	16	8/6.4/11.8	6.3/5.3	48/56/64	32/24	23.85; 23.05; 19.85; 18.25; 15.85; 14.25; 12.65; 8.85; 6.6
Transmitted bandwidth	300 Hz–3.4 kHz	300 Hz–3.4 kHz	300 Hz–3.4 kHz	300 Hz–3.4 kHz	300 Hz–3.4 kHz	300 Hz–3.4 kHz	300 Hz–3.4 kHz	100 Hz–7 kHz	100 Hz–7 kHz	100 Hz–7 kHz
Complexity										
MIPS	0.1	10	12	12	33	22 (12 G.729A)	16/18	10	14	40
RAM (words)	2	256	256	256	3,400	2,500	2,100	256	3,600	6.5 kw
ROM (words)	50	4,000	5,000	5,000	8,000	9,500	7,000	4,000	? kw	16 kw
Frame length (ms)	0.125	0.125	0.125	0.125	0.625	10	30	0.125	20	20
Look-ahead (ms)	0	0	0	0	0	5	7.5	1.5	20	5
Speech quality (MOS)	4.2	4.0	4.0'	4.0'	4.0	4.0	3.9/3.7	+64, 56 kbit/s	+ (32 kbit/s)	+ (23.85 kbit/s)

2.11.2 Main characteristics of cellular mobile standardized speech coders

Country	ETSI EUROPE				TIA USA			CRC JAPAN	
	GSM 06.10 (GSM-FR)	GSM 06.20 (GSM-FR)	GSM 06.71 (AMR)	TETRA	GSM 06.60 (GSM-FR)	IS54 TDMA	IS96 CDMA	IS136 TDMA (PCS)	JDC 1 JDC 2
Approval date	1988	1994	1999	1994	1996	1990	1992	1996	1993
Bitrate (kb/s)	13	5.6	4.75, 5.15, 5.9, 6.7, 7.4, 7.95, 10.2, 12.2	4.56	12.2	7.9	8.4/2/1	7.4	6.7 3.6
Bitrate for channels existing	9.8	5.8	Up to total of 22.8 (HR) or 11.2 (LR)	2.633	10.6	5.1		5.6	4.5 2
Transmitted bandwidth	300 Hz	300 Hz	300 Hz	300 Hz	300 Hz	300 Hz	300 Hz	300 Hz	300 Hz
Complexity	3.4	3.4	3.4	3.4	3.4	3.4	3.4	3.4	3.4
MIPS	2.5	1.5	15	15	15.1	14	10-12	16.1	30-40
RAM (words)	800	3,200	4,800	4,000	4,700	?	?	2,400	?
ROM (words)	2,000	18,000	15,000	7,000	5,900	?	?	5,400	?
Frame length (ms)	20	20	20	20	20	20	20	20	40
Overhead (ms)	0	?	5	?	0	5	5	5	10
Speech quality	3.6-3.8	3.5-3.7	Var	3.3-3.5	4.1	3.5-3.7	3.3-3.5	4	3.4/3.6
Noise << quality >>	-	-	Var	-	-	-	-	-	-
Capability to encode music	-	-	-	-	-	-	-	-	-
Robustness to errors	+/-	+/-	+/+	-/+	-/+	-/+	-/+	-/+	+/+
Maximum speed for modems (kb/s)									
Acceptable number of tandem	2	2	2/1	2/1	2	2	2/1	2	2/1
Coder type (all AAS)	RPE LTP	VSELP	ACELP	ACELP	ACELP	VSELP	CELP	ACELP	VSELP PS