CHAPTER

22

Audio Processing

Audio processing covers many diverse fields, all involved in presenting sound to human listeners. Three areas are prominent: (1) *high fidelity music reproduction*, such as in audio compact discs, (2) *voice telecommunications*, another name for telephone networks, and (3) *synthetic speech*, where computers generate and recognize human voice patterns. While these applications have different goals and problems, they are linked by a common umpire: the human ear. Digital Signal Processing has produced revolutionary changes in these and other areas of audio processing.

Human Hearing

The human ear is an exceedingly complex organ. To make matters even more difficult, the information from *two* ears is combined in a perplexing neural network, the human brain. Keep in mind that the following is only a brief overview; there are many subtle effects and poorly understood phenomena related to human hearing.

Figure 22-1 illustrates the major structures and processes that comprise the human ear. The *outer ear* is composed of two parts, the visible flap of skin and cartilage attached to the side of the head, and the *ear canal*, a tube about 0.5 cm in diameter extending about 3 cm into the head. These structures direct environmental sounds to the sensitive *middle and inner ear* organs located safely inside of the skull bones. Stretched across the end of the ear canal is a thin sheet of tissue called the *tympanic membrane* or *ear drum*. Sound waves striking the tympanic membrane cause it to vibrate. The middle ear is a set of small bones that transfer this vibration to the *cochlea* (inner ear) where it is converted to neural impulses. The cochlea is a liquid filled tube roughly 2 mm in diameter and 3 cm in length. Although shown straight in Fig. 22-1, the cochlea is curled up and looks like a small snail shell. In fact, *cochlea* is derived from the Greek word for *snail*.

When a sound wave tries to pass from air into liquid, only a small fraction of the sound is transmitted through the interface, while the remainder of the energy is reflected. This is because air has a *low* mechanical impedance (low acoustic pressure and high particle velocity resulting from low density and high compressibility), while liquid has a *high* mechanical impedance. In less technical terms, it requires more effort to wave your hand in water than it does to wave it in air. This difference in mechanical impedance results in most of the sound being reflected at an air/liquid interface.

The middle ear is an *impedance matching* network that increases the fraction of sound energy entering the liquid of the inner ear. For example, fish do not have an ear drum or middle ear, because they have no need to hear in air. Most of the impedance conversion results from the difference in *area* between the ear drum (receiving sound from the air) and the *oval window* (transmitting sound into the liquid, see Fig. 22-1). The ear drum has an area of about 60 $(mm)^2$, while the oval window has an area of roughly 4 $(mm)^2$. Since pressure is equal to force divided by area, this difference in area increases the sound wave pressure by about 15 times.

Contained within the cochlea is the *basilar membrane*, the supporting structure for about 12,000 sensory cells forming the *cochlear nerve*. The basilar membrane is stiffest near the oval window, and becomes more flexible toward the opposite end, allowing it to act as a *frequency spectrum analyzer*. When exposed to a high frequency signal, the basilar membrane resonates where it is stiff, resulting in the excitation of nerve cells close to the oval window. Likewise, low frequency sounds excite nerve cells at the far end of the basilar membrane. This makes specific fibers in the cochlear nerve respond to specific frequencies. This organization is called the **place principle**, and is preserved throughout the auditory pathway into the brain.

Another information encoding scheme is also used in human hearing, called the **volley principle**. Nerve cells transmit information by generating brief electrical pulses called *action potentials*. A nerve cell on the basilar membrane can encode audio information by producing an action potential in response to each cycle of the vibration. For example, a 200 hertz sound wave can be represented by a neuron producing 200 action potentials per second. However, this only works at frequencies below about 500 hertz, the maximum rate that neurons can produce action potentials. The human ear overcomes this problem by allowing several nerve cells to take turns performing this single task. For example, a 3000 hertz tone might be represented by *ten* nerve cells alternately firing at 300 times per second. This extends the range of the volley principle to about 4 kHz, above which the place principle is exclusively used.

Table 22-1 shows the relationship between sound intensity and perceived loudness. It is common to express sound intensity on a logarithmic scale, called **decibel SPL** (Sound Power Level). On this scale, 0 dB SPL is a sound wave power of 10^{-16} watts/cm^2, about the weakest sound detectable by the human ear. Normal speech is at about 60 dB SPL, while painful damage to the ear occurs at about 140 dB SPL.

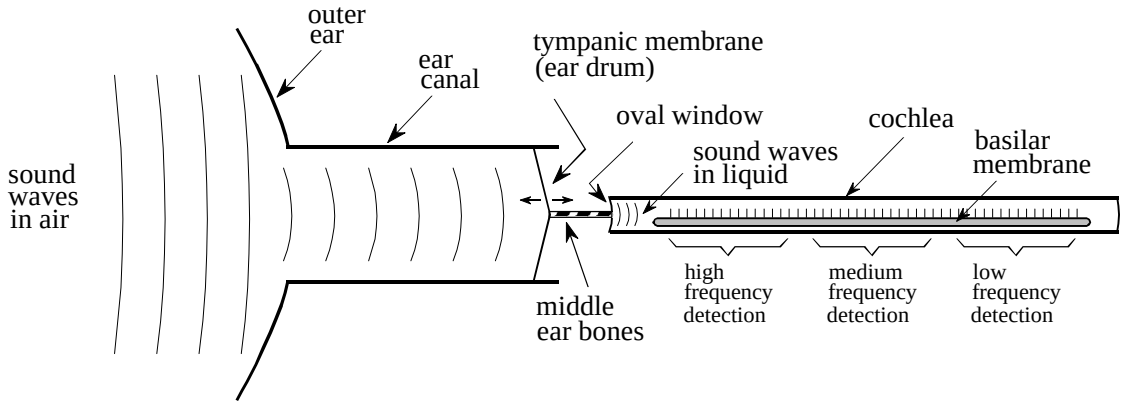

FIGURE 22-1
Functional diagram of the human ear. The outer ear collects sound waves from the environment and channels them to the tympanic membrane (ear drum), a thin sheet of tissue that vibrates in synchronization with the air waveform. The middle ear bones (hammer, anvil and stirrup) transmit these vibrations to the oval window, a flexible membrane in the fluid filled cochlea. Contained within the cochlea is the basilar membrane, the supporting structure for about 12,000 nerve cells that form the cochlear nerve. Due to the varying stiffness of the basilar membrane, each nerve cell only responses to a narrow range of audio frequencies, making the ear a frequency spectrum analyzer.

The difference between the loudest and faintest sounds that humans can hear is about 120 dB, a range of one-million in amplitude. Listeners can detect a *change* in loudness when the signal is altered by about 1 dB (a 12% change in amplitude). In other words, there are only about 120 levels of loudness that can be perceived from the faintest whisper to the loudest thunder. The sensitivity of the ear is amazing; when listening to very weak sounds, the ear drum vibrates less than the diameter of a single molecule!

The perception of loudness relates roughly to the sound power to an exponent of 1/3. For example, if you increase the sound power by a factor of *ten*, listeners will report that the loudness has increased by a factor of about *two* ($10^{1/3} \approx 2$). This is a major problem for eliminating undesirable environmental sounds, for instance, the beefed-up stereo in the next door apartment. Suppose you diligently cover 99% of your wall with a perfect soundproof material, missing only 1% of the surface area due to doors, corners, vents, etc. Even though the sound power has been reduced to only 1% of its former value, the perceived loudness has only dropped to about $0.01^{1/3} \approx 0.2$, or 20%.

The range of human hearing is generally considered to be 20 Hz to 20 kHz, but it is far more sensitive to sounds between 1 kHz and 4 kHz. For example, listeners can detect sounds as low as 0 dB SPL at 3 kHz, but require 40 dB SPL at 100 hertz (an amplitude increase of 100). Listeners can tell that two tones are different if their frequencies differ by more than about 0.3% at 3 kHz. This increases to 3% at 100 hertz. For comparison, adjacent keys on a piano differ by about 6% in frequency.

	Watts/cm^2	Decibels SPL	Example sound
	10^{-2}	140 dB	Pain
	10^{-3}	130 dB	
↑ Louder	10^{-4}	120 dB	Discomfort
	10^{-5}	110 dB	Jack hammers and rock concerts
	10^{-6}	100 dB	
	10^{-7}	90 dB	OSHA limit for industrial noise
	10^{-8}	80 dB	
	10^{-9}	70 dB	
	10^{-10}	60 dB	Normal conversation
	10^{-11}	50 dB	
	10^{-12}	40 dB	Weakest audible at 100 hertz
	10^{-13}	30 dB	
↓ Softer	10^{-14}	20 dB	Weakest audible at 10kHz
	10^{-15}	10 dB	
	10^{-16}	0 dB	Weakest audible at 3 kHz
	10^{-17}	-10 dB	
	10^{-18}	-20 dB	

TABLE 22-1
Units of sound intensity. Sound intensity is expressed as power per unit area (such as watts/cm^2), or more commonly on a logarithmic scale called *decibels SPL*. As this table shows, human hearing is the most sensitive between 1 kHz and 4 kHz.

The primary advantage of having *two* ears is the ability to identify the *direction* of the sound. Human listeners can detect the difference between two sound sources that are placed as little as three degrees apart, about the width of a person at 10 meters. This directional information is obtained in two separate ways. First, frequencies above about 1 kHz are strongly *shadowed* by the head. In other words, the ear nearest the sound receives a stronger signal than the ear on the opposite side of the head. The second clue to directionality is that the ear on the far side of the head hears the sound slightly *later* than the near ear, due to its greater distance from the source. Based on a typical head size (about 22 cm) and the speed of sound (about 340 meters per second), an angular discrimination of three degrees requires a timing precision of about 30 microseconds. Since this timing requires the volley principle, this clue to directionality is predominately used for sounds less than about 1 kHz.

Both these sources of directional information are greatly aided by the ability to turn the head and observe the change in the signals. An interesting sensation occurs when a listener is presented with exactly the same sounds to both ears, such as listening to monaural sound through headphones. The brain concludes that the sound is coming from the center of the listener's head!

While human hearing can determine the *direction* a sound is from, it does poorly in identifying the *distance* to the sound source. This is because there are few clues available in a sound wave that can provide this information. Human hearing weakly perceives that high frequency sounds are nearby, while low frequency sounds are distant. This is because sound waves dissipate their higher frequencies as they propagate long distances. Echo content is another weak clue to distance, providing a perception of the room size. For example,

sounds in a large auditorium will contain echoes at about 100 millisecond intervals, while 10 milliseconds is typical for a small office. Some species have solved this ranging problem by using *active sonar*. For example, bats and dolphins produce clicks and squeaks that reflect from nearby objects. By measuring the interval between transmission and echo, these animals can locate objects with about 1 cm resolution. Experiments have shown that some humans, particularly the blind, can also use active echo localization to a small extent.

Timbre

The perception of a continuous sound, such as a note from a musical instrument, is often divided into three parts: **loudness**, **pitch**, and **timbre** (pronounced "timber"). *Loudness* is a measure of sound wave intensity, as previously described. *Pitch* is the frequency of the fundamental component in the sound, that is, the frequency with which the waveform repeats itself. While there are subtle effects in both these perceptions, they are a straightforward match with easily characterized physical quantities.

Timbre is more complicated, being determined by the *harmonic content* of the signal. Figure 22-2 illustrates two waveforms, each formed by adding a 1 kHz sine wave with an amplitude of *one*, to a 3 kHz sine wave with an amplitude of *one-half*. The difference between the two waveforms is that the one shown in (b) has the higher frequency *inverted* before the addition. Put another way, the third harmonic (3 kHz) is phase shifted by 180 degrees compared to the first harmonic (1 kHz). In spite of the very different time domain waveforms, these two signals sound *identical*. This is because hearing is based on the *amplitude* of the frequencies, and is very insensitive to their *phase*. The *shape* of the time domain waveform is only indirectly related to hearing, and usually not considered in audio systems.

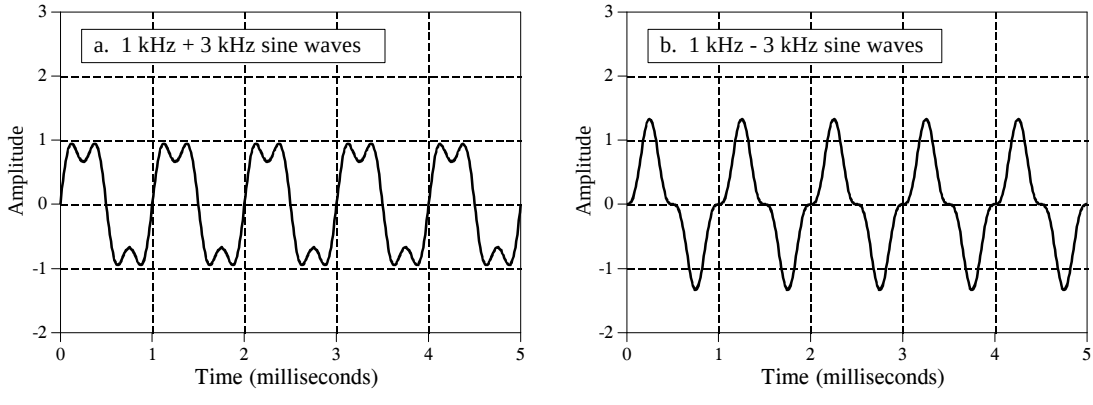

FIGURE 22-2
Phase detection of the human ear. The human ear is very insensitive to the relative phase of the component sinusoids. For example, these two waveforms would sound identical, because the *amplitudes* of their components are the same, even though their relative *phases* are different.

The ear's insensitivity to phase can be understood by examining how sound propagates through the environment. Suppose you are listening to a person speaking across a small room. Much of the sound reaching your ears is reflected from the walls, ceiling and floor. Since sound propagation depends on frequency (such as: attenuation, reflection, and resonance), different frequencies will reach your ear through different paths. This means that the relative phase of each frequency will change as you move about the room. Since the ear disregards these phase variations, you perceive the voice as *unchanging* as you move position. From a physics standpoint, the phase of an audio signal becomes randomized as it propagates through a complex environment. Put another way, the ear is insensitive to phase because it contains little useful information.

However, it cannot be said that the ear is completely deaf to the phase. This is because a phase change can rearrange the *time sequence* of an audio signal. An example is the chirp system (Chapter 11) that changes an impulse into a much longer duration signal. Although they differ only in their phase, the ear can distinguish between the two sounds because of their difference in duration. For the most part, this is just a curiosity, not something that happens in the normal listening environment.

Suppose that we ask a violinist to play a note, say, the *A* below middle *C*. When the waveform is displayed on an oscilloscope, it appear much as the sawtooth shown in Fig. 22-3a. This is a result of the sticky rosin applied to the fibers of the violinist's bow. As the bow is drawn across the string, the waveform is formed as the string sticks to the bow, is pulled back, and eventually breaks free. This cycle repeats itself over and over resulting in the sawtooth waveform.

Figure 22-3b shows how this sound is perceived by the ear, a frequency of 220 hertz, plus harmonics at 440, 660, 880 hertz, etc. If this note were played on another instrument, the waveform would *look* different; however, the ear would still hear a frequency of 220 hertz plus the harmonics. Since the two instruments produce the same fundamental frequency for this note, they sound similar, and are said to have identical *pitch*. Since the relative amplitude of the *harmonics* is different, they will not sound identical, and will be said to have different *timbre*.

It is often said that timbre is determined by the shape of the waveform. This is true, but slightly misleading. The perception of timbre results from the ear detecting harmonics. While harmonic content is determined by the shape of the waveform, the insensitivity of the ear to phase makes the relationship very one-sided. That is, a particular waveform will have only one timbre, while a particular timbre has an infinite number of possible waveforms.

The ear is very accustomed to hearing a fundamental plus harmonics. If a listener is presented with the combination of a 1 kHz and 3 kHz sine wave, they will report that it sounds natural and pleasant. If sine waves of 1 kHz and 3.1 kHz are used, it will sound objectionable.

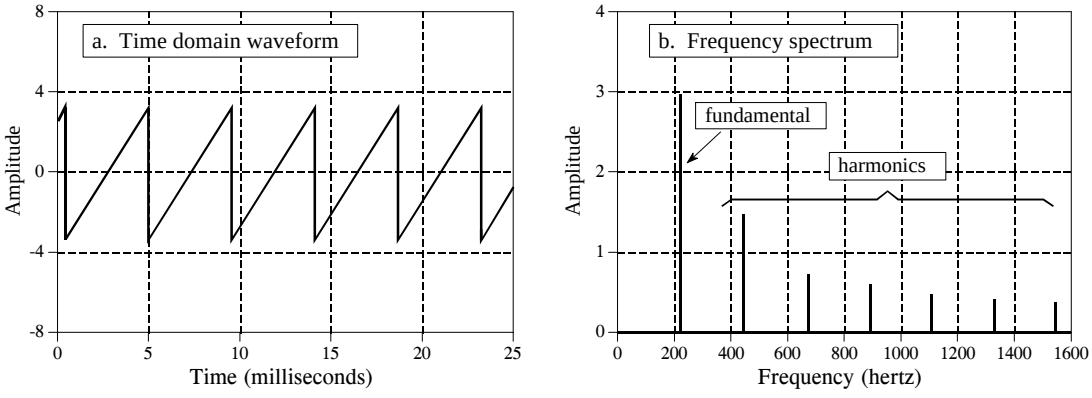

FIGURE 22-3
Violin waveform. A bowed violin produces a sawtooth waveform, as illustrated in (a). The sound *heard by the ear* is shown in (b), the fundamental frequency plus harmonics.

This is the basis of the standard musical scale, as illustrated by the piano keyboard in Fig. 22-4. Striking the farthest left key on the piano produces a fundamental frequency of 27.5 hertz, plus harmonics at 55, 110, 220, 440, 880 hertz, etc. (there are also harmonics between these frequencies, but they aren't important for this discussion). These harmonics correspond to the fundamental frequency produced by other keys on the keyboard. Specifically, every *seventh* white key is a harmonic of the far left key. That is, the eighth key from the left has a fundamental frequency of 55 hertz, the 15th key has a fundamental frequency of 110 hertz, etc. Being harmonics of each other, these keys sound similar when played, and are harmonious when played in unison. For this reason, they are *all* called the note, *A*. In this same manner, the white key immediate right of each *A* is called a *B*, and *they* are all harmonics of each other. This pattern repeats for the seven notes: *A*, *B*, *C*, *D*, *E*, *F*, and *G*.

The term **octave** means a *factor of two in frequency*. On the piano, one octave comprises eight white keys, accounting for the name (*octo* is Latin for *eight*). In other words, the piano's frequency doubles after every seven white keys, and the entire keyboard spans a little over seven octaves. The range of human hearing is generally quoted as 20 hertz to 20 kHz,

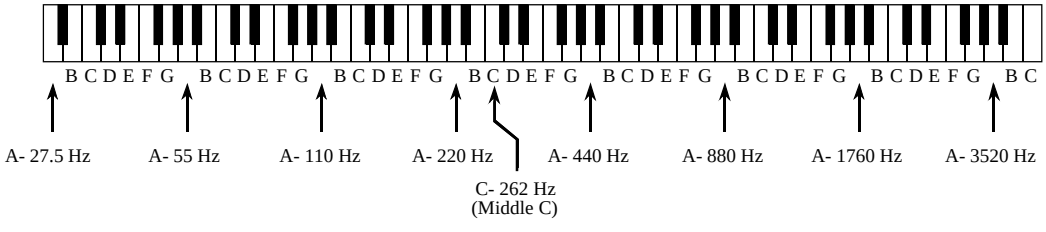

FIGURE 22-4
The Piano keyboard. The keyboard of the piano is a *logarithmic* frequency scale, with the fundamental frequency doubling after every seven white keys. These white keys are the notes: *A, B, C, D, E, F and G*.

corresponding to about ½ octave to the left, and two octaves to the right of the piano keyboard. Since octaves are based on doubling the frequency every fixed number of keys, they are a *logarithmic* representation of frequency. This is important because audio information is generally distributed in this same way. For example, as much audio information is carried in the octave between 50 hertz and 100 hertz, as in the octave between 10 kHz and 20 kHz. Even though the piano only covers about 20% of the frequencies that humans can hear (4 kHz out of 20 kHz), it can produce more than 70% of the audio information that humans can perceive (7 out of 10 octaves). Likewise, the highest frequency a human can detect drops from about 20 kHz to 10 kHz over the course of an adult's lifetime. However, this is only a loss of about 10% of the hearing ability (one octave out of ten). As shown next, this logarithmic distribution of information directly affects the required *sampling rate* of audio signals.

Sound Quality vs. Data Rate

When designing a digital audio system there are two questions that need to be asked: (1) how good does it need to sound? and (2) what data rate can be tolerated? The answer to these questions usually results in one of three categories. First, **high fidelity music,** where sound quality is of the greatest importance, and almost any data rate will be acceptable. Second, **telephone communication**, requiring natural sounding speech *and* a low data rate to reduce the system cost. Third, **compressed speech**, where reducing the data rate is very important and some unnaturalness in the sound quality can be tolerated. This includes military communication, cellular telephones, and digitally stored speech for voice mail and multimedia.

Table 22-2 shows the tradeoff between sound quality and data rate for these three categories. High fidelity music systems sample fast enough (44.1 kHz), and with enough precision (16 bits), that they can capture virtually all of the sounds that humans are capable of hearing. This magnificent sound quality comes at the price of a high data rate, 44.1 kHz × 16 bits = 706k bits/sec. This is pure brute force.

Whereas music requires a bandwidth of 20 kHz, natural sounding speech only requires about 3.2 kHz. Even though the frequency range has been reduced to only 16% (3.2 kHz out of 20 kHz), the signal still contains 80% of the original sound information (8 out of 10 octaves). Telecommunication systems typically operate with a sampling rate of about 8 kHz, allowing natural sounding speech, but greatly reduced music quality. You are probably already familiar with this difference in sound quality: FM radio stations broadcast with a bandwidth of almost 20 kHz, while AM radio stations are limited to about 3.2 kHz. Voices sound normal on the AM stations, but the music is weak and unsatisfying.

Voice-only systems also reduce the precision from 16 bits to 12 bits per sample, with little noticeable change in the sound quality. This can be reduced to only 8 bits per sample if the quantization step size is made unequal. This is a widespread procedure called **companding**, and will be

Sound Quality Required	Bandwidth	Sampling rate	Number of bits	Data rate (bits/sec)	Comments
High fidelity music (compact disc)	5 Hz to 20 kHz	44.1 kHz	16 bit	706k	Satisfies even the most picky audiophile. Better than human hearing.
Telephone quality speech	200 Hz to 3.2 kHz	8 kHz	12 bit	96k	Good speech quality, but very poor for music.
(with companding)	200 Hz to 3.2 kHz	8 kHz	8 bit	64k	Nonlinear ADC reduces the data rate by 50%. A very common technique.
Speech encoded by Linear Predictive Coding	200 Hz to 3.2 kHz	8 kHz	12 bit	4k	DSP speech compression technique. Very low data rates, poor voice quality.

TABLE 22-2
Audio data rate vs. sound quality. The sound quality of a digitized audio signal depends on its *data rate*, the product of its sampling rate and number of bits per sample. This can be broken into three categories, high fidelity music (706 kbits/sec), telephone quality speech (64 kbits/sec), and compressed speech (4 kbits/sec).

discussed later in this chapter. An 8 kHz sampling rate, with an ADC precision of 8 bits per sample, results in a data rate of 64k bits/sec. This is the *brute force* data rate for natural sounding speech. Notice that speech requires less than 10% of the data rate of high fidelity music.

The data rate of 64k bits/sec represents the straightforward application of sampling and quantization theory to audio signals. Techniques for lowering the data rate further are based on *compressing* the data stream by removing the inherent redundancies in speech signals. Data compression is the topic of Chapter 27. One of the most efficient ways of compressing an audio signal is **Linear Predictive Coding (LPC)**, of which there are several variations and subgroups. Depending on the speech quality required, LPC can reduce the data rate to as little as 2-6k bits/sec. We will revisit LPC later in this chapter with *speech synthesis*.

High Fidelity Audio

Audiophiles demand the utmost sound quality, and all other factors are treated as secondary. If you had to describe the mindset in one word, it would be: *overkill*. Rather than just matching the abilities of the human ear, these systems are designed to *exceed* the limits of hearing. It's the only way to be sure that the reproduced music is pristine. Digital audio was brought to the world by the **compact laser disc**, or **CD**. This was a revolution in music; the sound quality of the CD system far exceeds older systems, such as records and tapes. DSP has been at the forefront of this technology.

Figure 22-5 illustrates the surface of a compact laser disc, such as viewed through a high power microscope. The main surface is shiny (reflective of light), with the digital information stored as a series of dark pits burned on the surface with a laser. The information is arranged in a single track that spirals from the outside to the inside, the same as a phonograph record. The rotation of the CD is changed from about 210 to 480 rpm as the information is read from the outside to the inside of the spiral, making the scanning velocity a constant 1.2 meters per second. (In comparison, phonograph records spin at a fixed rate, such as 33, 45 or 78 rpm). During playback, an optical sensor detects if the surface is reflective or nonreflective, generating the corresponding binary information.

As shown by the geometry in Fig. 22-5, the CD stores about 1 bit per $(\mu m)^2$, corresponding to 1 million bits per $(mm)^2$, and 15 billion bits per disk. This is about the same feature size used in integrated circuit manufacturing, and for a good reason. One of the properties of light is that it cannot be focused to smaller than about one-half wavelength, or 0.3 µm. Since both integrated circuits and laser disks are created by optical means, the fuzziness of light below 0.3 µm limits how small of features can be used.

Figure 22-6 shows a block diagram of a typical compact disc playback system. The raw data rate is 4.3 million bits per second, corresponding to 1 bit each 0.28 µm of track length. However, this is in conflict with the specified geometry of the CD; each pit must be no shorter than 0.8 µm, and no longer than 3.5 µm. In other words, each binary *one* must be part of a group of 3 to 13 *ones*. This has the advantage of reducing the error rate due to the optical pickup, but how do you force the binary data to comply with this strange bunching?

The answer is an encoding scheme called **eight-to-fourteen modulation (EFM)**. Instead of directly storing a byte of data on the disc, the 8 bits are passed through a look-up table that pops out 14 bits. These 14 bits have the desired bunching characteristics, and are stored on the laser disc. Upon playback, the binary values read from the disc are passed through the inverse of the EFM look-up table, resulting in each 14 bit group being turned back into the correct 8 bits.

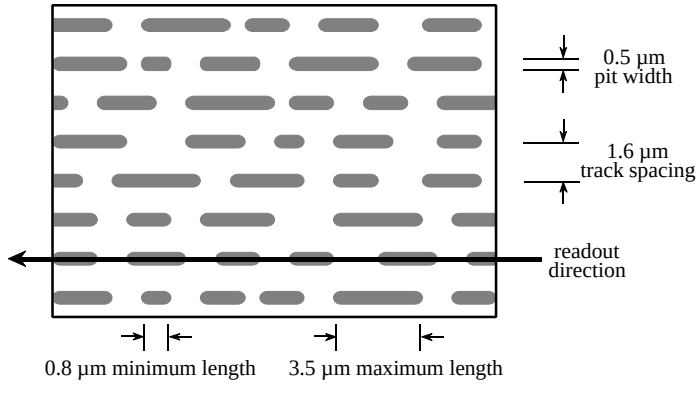

FIGURE 22-5
Compact disc surface. Micron size pits are burned into the surface of the CD to represent ones and zeros. This results in a data density of 1 bit per μm^2, or one million bits per mm^2. The pit depth is 0.16 µm.

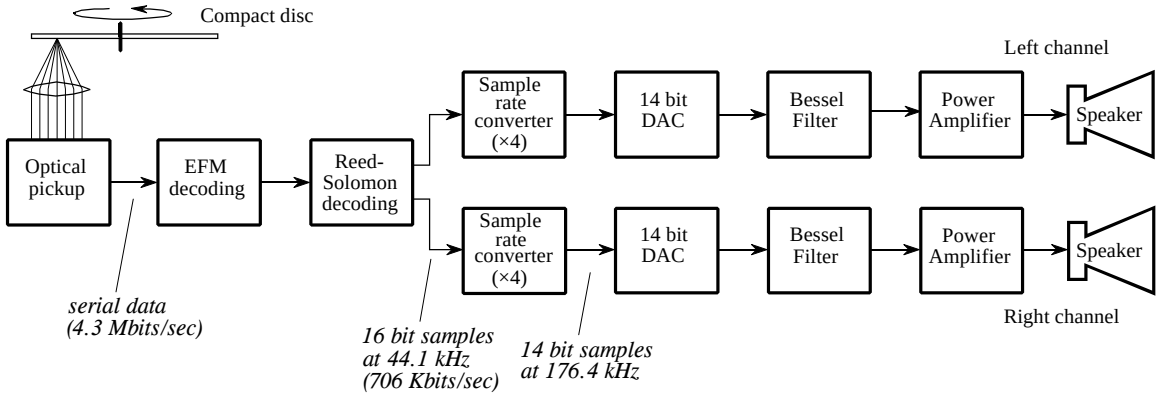

FIGURE 22-6
Compact disc playback block diagram. The digital information is retrieved from the disc with an optical sensor, corrected for EFM and Reed-Solomon encoding, and converted to stereo analog signals.

In addition to EFM, the data are encoded in a format called **two-level Reed-Solomon coding**. This involves combining the left and right stereo channels along with data for error detection and correction. Digital errors detected during playback are either: *corrected* by using the redundant data in the encoding scheme, *concealed* by interpolating between adjacent samples, or *muted* by setting the sample value to zero. These encoding schemes result in the data rate being *tripled*, i.e., 1.4 Mbits/sec for the stereo audio signals versus 4.3 Mbits/sec stored on the disc.

After decoding and error correction, the audio signals are represented as 16 bit samples at a 44.1 kHz sampling rate. In the simplest system, these signals could be run through a 16 bit DAC, followed by a low-pass analog filter. However, this would require high performance analog electronics to pass frequencies below 20 kHz, while rejecting all frequencies above 22.05 kHz, ½ of the sampling rate. A more common method is to use a **multirate** technique, that is, convert the digital data to a higher sampling rate before the DAC. A factor of four is commonly used, converting from 44.1 kHz to 176.4 kHz. This is called **interpolation**, and can be explained as a two step process (although it may not actually be carried out this way). First, three samples with a value of zero are placed between the original samples, producing the higher sampling rate. In the frequency domain, this has the effect of duplicating the 0 to 22.05 kHz spectrum three times, at 22.05 to 44.1 kHz, 41 to 66.15 kHz, and 66.15 to 88.2 kHz. In the second step, an efficient *digital* filter is used to remove the newly added frequencies.

The sample rate increase makes the sampling interval smaller, resulting in a smoother signal being generated by the DAC. The signal still contains frequencies between 20 Hz and 20 kHz; however, the Nyquist frequency has been increased by a factor of four. This means that the analog filter only needs to pass frequencies below 20 kHz, while blocking frequencies above 88.2 kHz. This is usually done with a three pole Bessel filter. Why use a *Bessel* filter if the ear is insensitive to phase? Overkill, remember?

Since there are four times as many samples, the number of bits per sample can be reduced from 16 bits to 14 bits, without degrading the sound quality. The $\sin(x)/x$ correction needed to compensate for the zeroth order hold of the DAC can be part of either the analog or digital filter.

Audio systems with more than one channel are said to be in **stereo** (from the Greek word for *solid*, or *three-dimensional*). Multiple channels send sound to the listener from different directions, providing a more accurate reproduction of the original music. Music played through a monaural (one channel) system often sounds artificial and bland. In comparison, a good stereo reproduction makes the listener feel as if the musicians are only a few feet away. Since the 1960s, high fidelity music has used two channels (left and right), while motion pictures have used four channels (left, right, center, and surround). In early stereo recordings (say, the Beatles or the Mamas And The Papas), individual singers can often be heard in only one channel or the other. This rapidly progressed into a more sophisticated **mix-down**, where the sound from many microphones in the recording studio is combined into the two channels. Mix-down is an art, aimed at providing the listener with the perception of *being there*.

The four channel sound used in motion pictures is called **Dolby Stereo**, with the home version called **Dolby Surround Pro Logic**. ("Dolby" and "Pro Logic" are trademarks of Dolby Laboratories Licensing Corp.). The four channels are encoded into the standard left and right channels, allowing regular two-channel stereo systems to reproduce the music. A Dolby decoder is used during playback to recreate the four channels of sound. The left and right channels, from speakers placed on each side of the movie or television screen, is similar to that of a regular two-channel stereo system. The speaker for the center channel is usually placed directly above or below the screen. Its purpose is to reproduce speech and other visually connected sounds, keeping them firmly centered on the screen, regardless of the seating position of the viewer/listener. The surround speakers are placed to the left and right of the listener, and may involve as many as twenty speakers in a large auditorium. The surround channel only contains midrange frequencies (say, 100 Hz to 7 kHz), and is *delayed* by 15 to 30 milliseconds. This delay makes the listener perceive that speech is coming from the screen, and not the sides. That is, the listener hears the speech coming from the front, followed by a delayed version of the speech coming from the sides. The listener's mind interprets the delayed signal as a reflection from the walls, and ignores it.

Companding

The data rate is important in telecommunication because it is directly proportional to the *cost* of transmitting the signal. Saving bits is the same as saving money. **Companding** is a common technique for reducing the data rate of audio signals by making the quantization levels *unequal*. As previously mentioned, the loudest sound that can be tolerated (120 dB SPL) is about one-million times the amplitude of the weakest sound that can be detected (0 dB

SPL). However, the ear cannot distinguish between sounds that are closer than about 1 dB (12% in amplitude) apart. In other words, there are only about 120 different loudness levels that can be detected, spaced logarithmically over an amplitude range of one-million.

This is important for digitizing audio signals. If the quantization levels are equally spaced, 12 bits must be used to obtain telephone quality speech. However, only 8 bits are required if the quantization levels are made *unequal*, matching the characteristics of human hearing. This is quite intuitive: if the signal is small, the levels need to be very close together; if the signal is large, a larger spacing can be used.

Companding can be carried out in three ways: (1) run the analog signal through a nonlinear circuit before reaching a linear 8 bit ADC, (2) use an 8 bit ADC that internally has unequally spaced steps, or (3) use a linear 12 bit ADC followed by a digital look-up table (12 bits in, 8 bits out). Each of these three options requires the same nonlinearity, just in a different place: an analog circuit, an ADC, or a digital circuit.

Two nearly identical standards are used for companding curves: **μ255 law** (also called **mu law**), used in North America, and **"A" law**, used in Europe. Both use a logarithmic nonlinearity, since this is what converts the spacing detectable by the human ear into a linear spacing. In equation form, the curves used in μ255 law and "A" law are given by:

EQUATION 22-1
Mu law companding. This equation provides the nonlinearity for μ255 law companding. The constant, μ, has a value of 255, accounting for the name of this standard.

$$y = \frac{\ln(1+\mu x)}{\ln(1+\mu)} \qquad \text{for } 0 \le x \le 1$$

EQUATION 22-2
"A" law companding. The constant, *A*, has a value of 87.6.

$$y = \frac{1+\ln(Ax)}{1+\ln(A)} \qquad \text{for } 1/A \le x \le 1$$

$$y = \frac{Ax}{1+\ln(A)} \qquad \text{for } 0 \le x \le 1/A$$

Figure 22-7 graphs these equations for the input variable, *x*, being between -1 and +1, resulting in the output variable also assuming values between -1 and +1. Equations 22-1 and 22-2 only handle positive input values; portions of the curves for negative input values are found from symmetry. As shown in (a), the curves for μ255 law and "A" law are nearly identical. The only significant difference is near the origin, shown in (b), where μ255 law is a smooth curve, and "A" law switches to a straight line.

Producing a stable nonlinearity is a difficult task for analog electronics. One method is to use the logarithmic relationship between current and

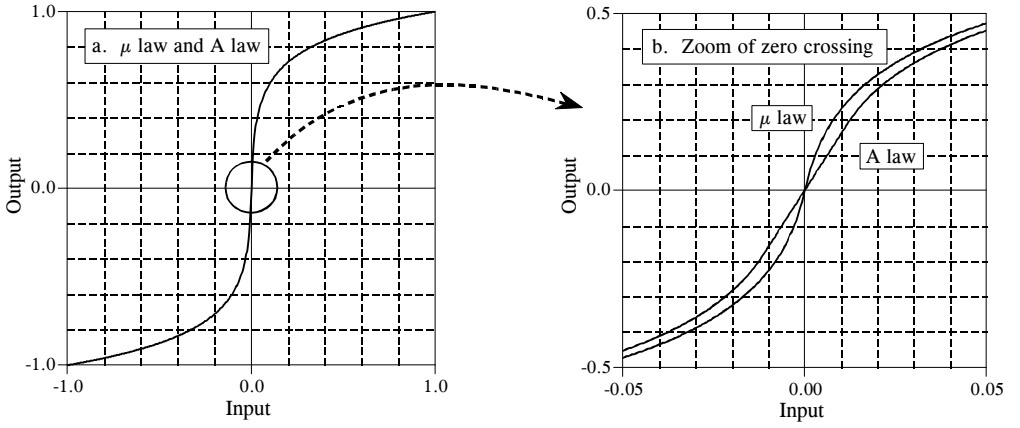

FIGURE 22-7
Companding curves. The μ255 law and "A" law companding curves are nearly identical, differing only near the origin. Companding increases the amplitude when the signal is small, and decreases it when it is large.

voltage across a *pn* diode junction, and then add circuitry to correct for the ghastly temperature drift. Most companding circuits take another strategy: approximate the nonlinearity with a group of straight lines. A typical scheme is to approximate the logarithmic curve with a group of 16 straight segments, called **cords**. The first bit of the 8 bit output indicates if the input is positive or negative. The next three bits identify which of the 8 positive or 8 negative cords is used. The last four bits break each cord into 16 equally spaced increments. As with most integrated circuits, companding chips have sophisticated and proprietary internal designs. Rather than worrying about what goes on inside of the chip, pay the most attention to the pinout and the specification sheet.

Speech Synthesis and Recognition

Computer generation and recognition of speech are formidable problems; many approaches have been tried, with only mild success. This is an active area of DSP research, and will undoubtedly remain so for many years to come. You will be very disappointed if you are expecting this section to describe how to build speech synthesis and recognition circuits. Only a brief introduction to the typical approaches can be presented here. Before starting, it should be pointed out that most commercial products that produce human sounding speech do not *synthesize* it, but merely play back a digitally recorded segment from a human speaker. This approach has great sound quality, but it is limited to the prerecorded words and phrases.

Nearly all techniques for speech synthesis and recognition are based on the model of human speech production shown in Fig. 22-8. Most human speech sounds can be classified as either **voiced** or **fricative**. Voiced sounds occur when air is forced from the lungs, through the vocal cords, and out of the mouth and/or nose. The vocal cords are two thin flaps of tissue stretched across

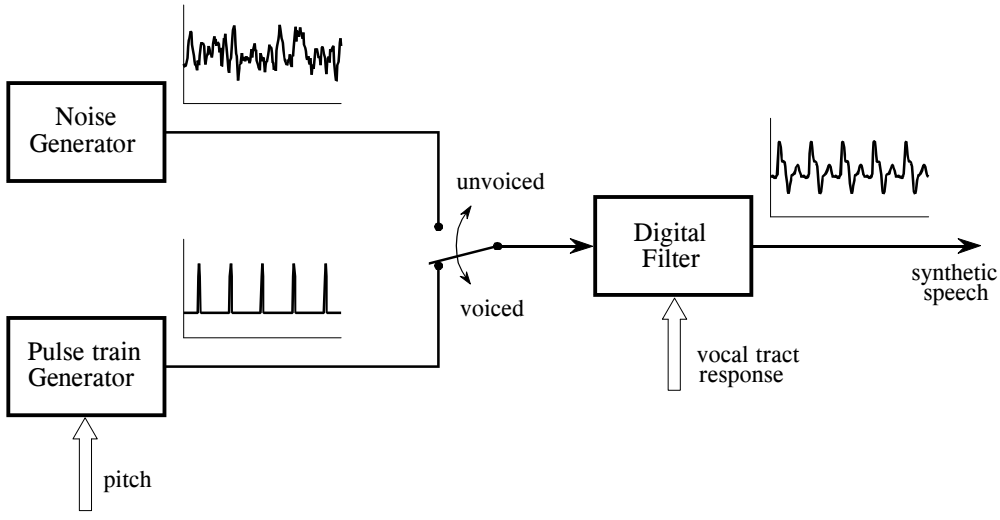

FIGURE 22-8
Human speech model. Over a short segment of time, about 2 to 40 milliseconds, speech can be modeled by three parameters: (1) the selection of either a periodic or a noise excitation, (2) the pitch of the periodic excitation, and (3) the coefficients of a recursive linear filter mimicking the vocal tract response.

the air flow, just behind the Adam's apple. In response to varying muscle tension, the vocal cords vibrate at frequencies between 50 and 1000 Hz, resulting in periodic puffs of air being injected into the throat. Vowels are an example of voiced sounds. In Fig. 22-8, voiced sounds are represented by the pulse train generator, with the pitch (i.e., the fundamental frequency of the waveform) being an adjustable parameter.

In comparison, *fricative* sounds originate as random noise, not from vibration of the vocal cords. This occurs when the air flow is nearly blocked by the tongue, lips, and/or teeth, resulting in air turbulence near the constriction. Fricative sounds include: *s*, *f*, *sh*, *z*, *v*, and *th*. In the model of Fig. 22-8, fricatives are represented by a *noise generator*.

Both these sound sources are modified by the acoustic cavities formed from the tongue, lips, mouth, throat, and nasal passages. Since sound propagation through these structures is a linear process, it can be represented as a linear filter with an appropriately chosen impulse response. In most cases, a *recursive* filter is used in the model, with the recursion coefficients specifying the filter's characteristics. Because the acoustic cavities have dimensions of several centimeters, the frequency response is primarily a series of resonances in the kilohertz range. In the jargon of audio processing, these resonance peaks are called the **format frequencies**. By changing the relative position of the tongue and lips, the format frequencies can be changed in both frequency and amplitude.

Figure 22-9 shows a common way to display speech signals, the **voice spectrogram**, or **voiceprint**. The audio signal is broken into short segments,

say 2 to 40 milliseconds, and the FFT used to find the frequency spectrum of each segment. These spectra are placed side-by-side, and converted into a grayscale image (low amplitude becomes light, and high amplitude becomes dark). This provides a graphical way of observing how the frequency content of speech changes with time. The segment length is chosen as a tradeoff between *frequency resolution* (favored by longer segments) and *time resolution* (favored by shorter segments).

As demonstrated by the *a* in *rain*, voiced sounds have a periodic time domain waveform, shown in (a), and a frequency spectrum that is a series of regularly spaced harmonics, shown in (b). In comparison, the *s* in *storm*, shows that fricatives have a noisy time domain signal, as in (c), and a noisy spectrum, displayed in (d). These spectra also show the shaping by the format frequencies for both sounds. Also notice that the time-frequency display of the word *rain* looks similar both times it is spoken.

Over a short period, say 25 milliseconds, a speech signal can be approximated by specifying three parameters: (1) the selection of either a periodic or random noise excitation, (2) the frequency of the periodic wave (if used), and (3) the coefficients of the digital filter used to mimic the vocal tract response. Continuous speech can then be synthesized by continually updating these three parameters about 40 times a second. This approach was responsible for one the early commercial successes of DSP: the *Speak & Spell*, a widely marketed electronic learning aid for children. The sound quality of this type of speech synthesis is poor, sounding very mechanical and not quite human. However, it requires a very low data rate, typically only a few kbits/sec.

This is also the basis for the **linear predictive coding** (**LPC**) method of speech compression. Digitally recorded human speech is broken into short segments, and each is characterized according to the three parameters of the model. This typically requires about a dozen bytes per segment, or 2 to 6 kbytes/sec. The segment information is transmitted or stored as needed, and then reconstructed with the speech synthesizer.

Speech recognition algorithms take this a step further by trying to recognize patterns in the extracted parameters. This typically involves comparing the segment information with templates of previously stored sounds, in an attempt to identify the spoken words. The problem is, this method does not work very well. It is useful for some applications, but is far below the capabilities of human listeners. To understand why speech recognition is so difficult for computers, imagine someone unexpectedly speaking the following sentence:

Larger run medical buy dogs fortunate almost when.

Of course, you will not understand the meaning of this sentence, because it has none. More important, you will probably not even understand all of the individual words that were spoken. This is basic to the way that humans

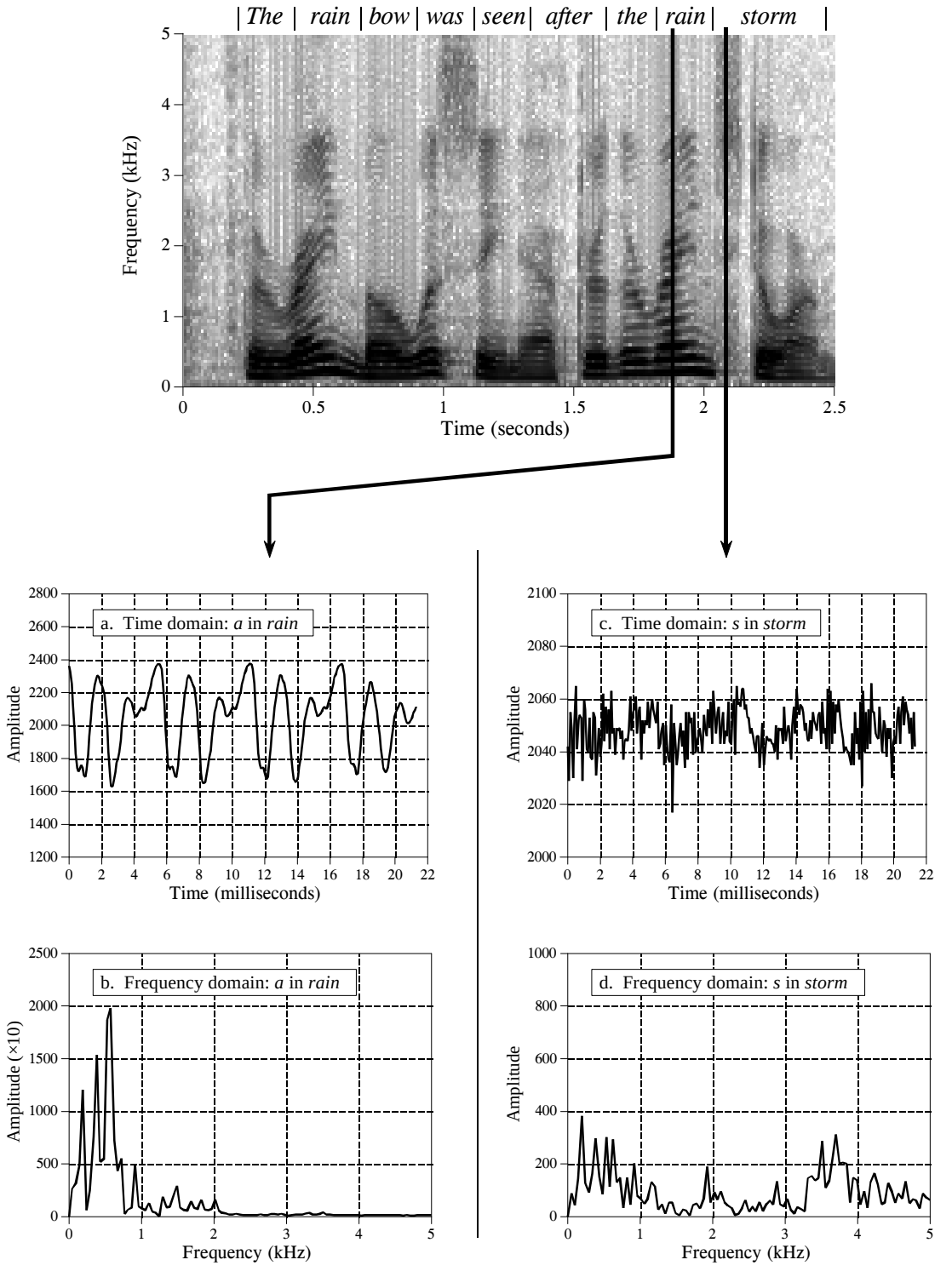

FIGURE 22-9
Voice spectrogram. The spectrogram of the phrase: "*The rainbow was seen after the rain storm.*" Figures (a) and (b) shows the time and frequency signals for the voiced *a* in *rain*. Figures (c) and (d) show the time and frequency signals for the fricative *s* in *storm*.

perceive and understand speech. Words are recognized by their sounds, but also by the *context* of the sentence, and the *expectations* of the listener. For example, imagine hearing the two sentences:

The child wore a spider ring on Halloween.

He was an American spy during the war.

Even if exactly the same sounds were produced to convey the underlined words, listeners *hear* the correct words for the context. From your accumulated knowledge about the world, you know that children don't wear secret agents, and people don't become spooky jewelry during wartime. This usually isn't a conscious act, but an inherent part of human hearing.

Most speech recognition algorithms rely only on the sound of the individual words, and not on their context. They attempt to *recognize words*, but not to *understand speech*. This places them at a tremendous disadvantage compared to human listeners. Three annoyances are common in speech recognition systems: (1) The recognized speech must have distinct pauses between the words. This eliminates the need for the algorithm to deal with phrases that sound alike, but are composed of different words (i.e., *spider ring* and *spy during*). This is slow and awkward for people accustomed to speaking in an overlapping flow. (2) The vocabulary is often limited to only a few hundred words. This means that the algorithm only has to search a limited set to find the best match. As the vocabulary is made larger, the recognition time and error rate both increase. (3) The algorithm must be *trained* on each speaker. This requires each person using the system to speak each word to be recognized, often needing to be repeated five to ten times. This personalized database greatly increases the accuracy of the word recognition, but it is inconvenient and time consuming.

The prize for developing a successful speech recognition technology is enormous. Speech is the quickest and most efficient way for humans to communicate. Speech recognition has the potential of replacing writing, typing, keyboard entry, and the electronic control provided by switches and knobs. It just needs to work a little better to become accepted by the commercial marketplace. Progress in speech recognition will likely come from the areas of artificial intelligence and neural networks as much as through DSP itself. Don't think of this as a technical *difficulty*; think of it as a technical *opportunity*.

Nonlinear Audio Processing

Digital filtering can improve audio signals in many ways. For instance, *Wiener filtering* can be used to separate frequencies that are mainly signal, from frequencies that are mainly noise (see Chapter 17). Likewise, *deconvolution* can compensate for an undesired convolution, such as in the restoration of old

recordings (also discussed in Chapter 17). These types of linear techniques are the backbone of DSP. Several *nonlinear* techniques are also useful for audio processing. Two will be briefly described here.

The first nonlinear technique is used for reducing wideband noise in speech signals. This type of noise includes: magnetic tape hiss, electronic noise in analog circuits, wind blowing by microphones, cheering crowds, etc. Linear filtering is of little use, because the frequencies in the noise completely overlap the frequencies in the voice signal, both covering the range from 200 hertz to 3.2 kHz. How can two signals be separated when they overlap in both the time domain *and* the frequency domain?

Here's how it is done. In a short segment of speech, the amplitude of the frequency components are greatly *unequal*. As an example, Fig. 22-10a illustrates the frequency spectrum of a 16 millisecond segment of speech (i.e., 128 samples at an 8 kHz sampling rate). Most of the signal is contained in a few large amplitude frequencies. In contrast, (b) illustrates the spectrum when only random noise is present; it is very irregular, but more uniformly distributed at a low amplitude.

Now the key concept: if both signal and noise are present, the two can be partially separated by looking at the *amplitude* of each frequency. If the amplitude is large, it is probably mostly signal, and should therefore be retained. If the amplitude is small, it can be attributed to mostly noise, and should therefore be discarded, i.e., set to zero. Mid-size frequency components are adjusted in some smooth manner between the two extremes.

Another way to view this technique is as a *time varying Wiener filter*. As you recall, the frequency response of the Wiener filter passes frequencies that are mostly signal, and rejects frequencies that are mostly noise. This

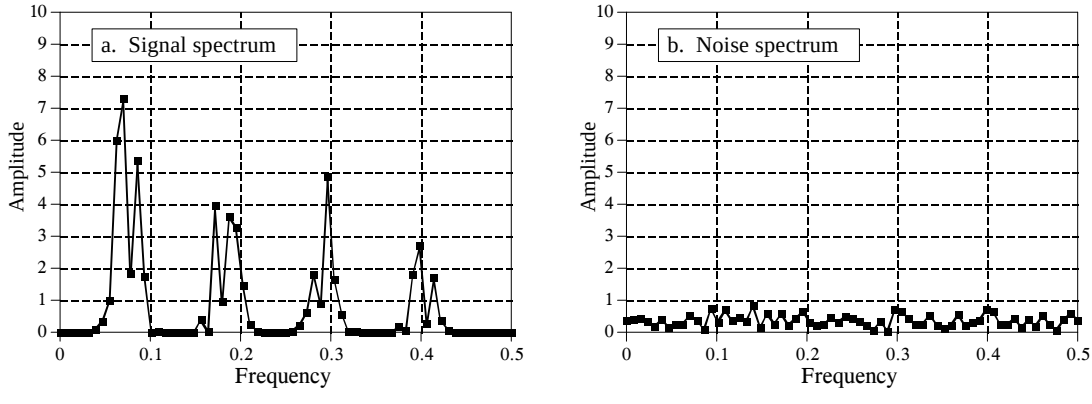

FIGURE 22-10
Spectra of speech and noise. While the frequency spectra of speech and noise generally overlap, there is some separation if the signal segment is made short enough. Figure (a) illustrates the spectrum of a 16 millisecond speech segment, showing that many frequencies carry little speech information, *in this particular segment*. Figure (b) illustrates the spectrum of a random noise source; all the components have a small amplitude. (These graphs are not of real signals, but illustrations to show the noise reduction technique).

requires a knowledge of the signal and noise spectra *beforehand*, so that the filter's frequency response can be determined. This nonlinear technique uses the same idea, except that the Wiener filter's frequency response is recalculated for each segment, based on the spectrum *of that segment*. In other words, the filter's frequency response changes from segment-to-segment, as determined by the characteristics of the signal itself.

One of the difficulties in implementing this (and other) nonlinear techniques is that the overlap-add method for filtering long signals is not valid. Since the frequency response changes, the time domain waveform of each segment will no longer align with the neighboring segments. This can be overcome by remembering that audio information is encoded in frequency patterns that change over time, and not in the shape of the time domain waveform. A typical approach is to divide the original time domain signal into *overlapping* segments. After processing, a smooth window is applied to each of the overlapping segments before they are recombined. This provides a smooth transition of the frequency spectrum from one segment to the next.

The second nonlinear technique is called **homomorphic** signal processing. This term literally means: *the same structure*. Addition is not the only way that noise and interference can be combined with a signal of interest; multiplication and convolution are also common means of mixing signals together. If signals are combined in a nonlinear way (i.e., anything other than addition), they cannot be separated by linear filtering. Homomorphic techniques attempt to separate signals combined in a nonlinear way by making the problem *become* linear. That is, the problem is converted to the *same structure* as a linear system.

For example, consider an audio signal transmitted via an AM radio wave. As atmospheric conditions change, the received amplitude of the signal increases and decreases, resulting in the loudness of the received audio signal slowly changing over time. This can be modeled as the audio signal, represented by $a[\ ]$, being *multiplied* by a slowly varying signal, $g[\ ]$, that represents the changing gain. This problem is usually handled in an electronic circuit called an *automatic gain control* (AGC), but it can also be corrected with nonlinear DSP.

As shown in Fig. 22-11, the input signal, $a[\ ] \times g[\ ]$, is passed through the logarithm function. From the identity, $\log(x+y) = \log x + \log y$, this results in two signals that are combined by addition, i.e., $\log a[\ ] + \log g[\ ]$. In other words, the *logarithm* is the homomorphic transform that turns the nonlinear problem of *multiplication* into the linear problem of *addition*.

Next, the added signals are separated by a conventional linear filter, that is, some frequencies are passed, while others are rejected. For the AGC, the gain signal, $g[\ ]$, will be composed of very low frequencies, far below the 200 hertz to 3.2 kHz band of the voice signal. The logarithm of these signals will have more complicated spectra, but the idea is the same: a high-pass filter is used to eliminate the varying gain component from the signal.

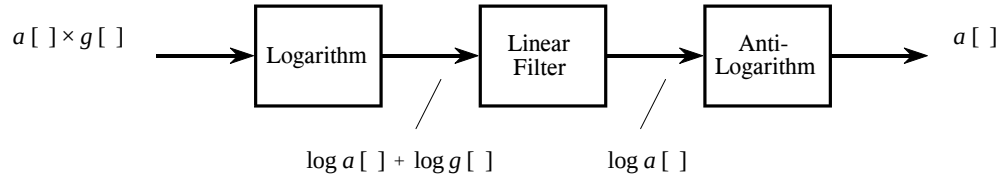

FIGURE 22-11
Homomorphic separation of multiplied signals. Taking the logarithm of the input signal transforms components that are *multiplied* into components that are *added*. These components can then be separated by linear filtering, and the effect of the logarithm undone.

In effect, $\log a[\,] + \log g[\,]$ is converted into $\log a[\,]$. In the last step, the logarithm is undone by using the exponential function (the anti-logarithm, or e^x), producing the desired output signal, $a[\,]$

Figure 22-12 shows a homomorphic system for separating signals that have been *convolved*. An application where this has proven useful is in removing echoes from audio signals. That is, the audio signal is convolved with an impulse response consisting of a delta function plus a shifted and scaled delta function. The homomorphic transform for convolution is composed of two stages, the *Fourier transform*, changing the convolution into a multi-plication, followed by the *logarithm*, turning the multiplication into an addition. As before, the signals are then separated by linear filtering, and the homomorphic transform undone.

An interesting twist in Fig. 22-12 is that the linear filtering is dealing with frequency domain signals in the same way that time domain signals are usually processed. In other words, the time and frequency domains have been swapped from their normal use. For example, if FFT convolution were used to carry out the linear filtering stage, the "spectra" being multiplied would be in the *time domain*. This role reversal has given birth to a strange jargon. For instance, *cepstrum* (a rearrangment of *spectrum*) is the Fourier transform of the logarithm of the Fourier transform. Likewise, there are *long-pass* and *short-pass* filters, rather than low-pass and high-pass filters. Some authors even use *Quefrency Alanysis* and *liftering*.

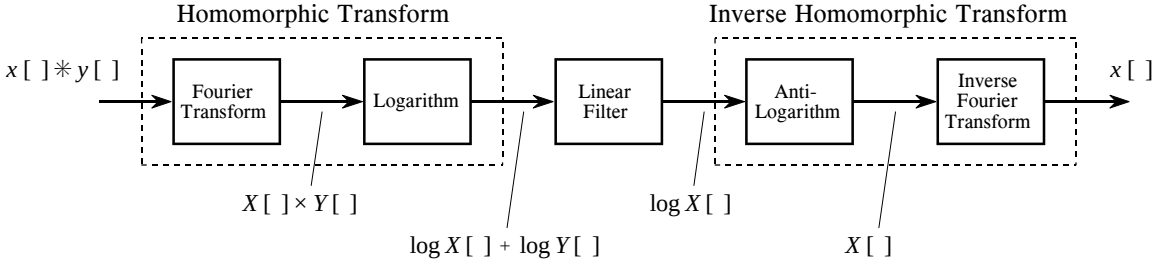

FIGURE 22-12
Homomorphic separation of convolved signals. Components that have been *convolved* are converted into components that are *added* by taking the Fourier transform followed by the logarithm. After linear filtering to separate the added components, the original steps are undone.

Keep in mind that these are simplified descriptions of sophisticated DSP algorithms; homomorphic processing is filled with subtle details. For example, the logarithm must be able to handle both negative and positive values in the input signal, since this is a characteristic of audio signals. This requires the use of the *complex logarithm*, a more advanced concept than the logarithm used in everyday science and engineering. When the linear filtering is restricted to be a *zero phase* filter, the complex log is found by taking the simple logarithm of the absolute value of the signal. After passing through the zero phase filter, the sign of the original signal is reapplied to the filtered signal.

Another problem is *aliasing* that occurs when the logarithm is taken. For example, imagine digitizing a continuous *sine wave*. In accordance with the sampling theorem, two or more samples per cycle is sufficient. Now consider digitizing the logarithm of this continuous sine wave. The sharp corners require many more samples per cycle to capture the waveform, i.e., to prevent aliasing. The required sampling rate can easily be 100 times as great after the log, as before. Further, it doesn't matter if the logarithm is applied to the continuous signal, or to its digital representation; the result is the same. Aliasing will result unless the sampling rate is high enough to capture the sharp corners produced by the nonlinearity. The result is that audio signals may need to be sampled at 100 kHz or more, instead of only the standard 8 kHz.

Even if these details are handled, there is no guarantee that the linearized signals *can* be separated by the linear filter. This is because the spectra of the linearized signals can overlap, even if the spectra of the original signals do not. For instance, imagine adding two sine waves, one at 1 kHz, and one at 2 kHz. Since these signals do not overlap in the frequency domain, they can be completely separated by linear filtering. Now imagine that these two sine waves are multiplied. Using homomorphic processing, the log is taken of the combined signal, resulting in the log of one sine wave plus the log of the other sine wave. The problem is, the logarithm of a sine wave contains many harmonics. Since the harmonics from the two signals overlap, their complete separation is not possible.

In spite of these obstacles, homomorphic processing teaches an important lesson: signals should be processed in a manner *consistent* with how they are formed. Put another way, the first step in any DSP task is to understand how information is represented in the signals being processed.