1

Introduction and Fundamentals

In digital audio signal processing and digital audio effects, we are primarily concerned with systems that take a discrete, uniformly sampled audio signal, process it, and produce a discrete, uniformly sampled output audio signal. Therefore, we start by introducing some fundamental properties of sound that are used over and over again, then how we represent it as a digital signal, and then we move on to how we describe the systems that act on and modify such signals. This is not meant to give a detailed overview of digital signal processing, which would involve discussion of continuous time signals, infinite signals, and mathematical relationships. Rather, we intend to focus on just the type of signals and systems that are encountered in audio effects, and on the most useful properties and representations. Having said that, this is also intended to be self-contained. Very little prior knowledge is assumed, and it should not be necessary to refer to more detailed discussions in other texts in order to understand these concepts.

Understanding Sound and Digital Audio

Fundamentally, all audio is composed of waveforms. Vibrating objects create pressure waves in the air; when these waves reach our ears, we perceive them as sound. With the invention of the telephone in the 19th century, audio was first encoded as an electric signal, with the changes in electric voltage representing the changes in pressure over time. Until the late 20th century, electric recording and transmission was all analog: sound was represented by a continuous waveform over time.

In this book, we will work almost exclusively with digital audio. Rather than representing audio as a continuous voltage, as in analog, the waveform will be composed of discrete samples over time. These samples can be stored, processed, and ultimately reconstructed as sound we can hear. Digital audio systems generally begin with an *analog-to-digital converter* (ADC), which captures periodic snapshots of the electrical voltage on an audio transmission line and represents these snapshots as discrete numbers. By capturing the voltage many thousands of times per second, one can achieve a very close approximation of the original audio signal. This encoding method is known

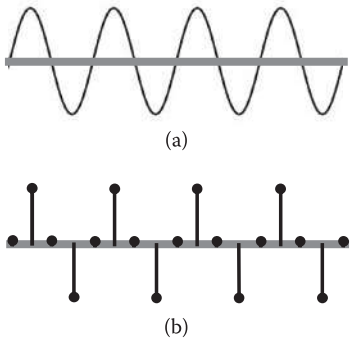

FIGURE 1.1
A continuous time signal (a), and its digital representation, found by sampling the signal uniformly in time (b).

as *pulse code modulation*, and is the encoding format used in the WAV and AIFF audio formats. Pulse code modulation is also one of the most popular forms of ADC, and certainly one of the simplest to explain.

Thus, a continuous time audio signal, such as captured from a microphone, is represented as a digital signal with uniform timing between samples (see Figure 1.1). But digital audio signals need not be derived from analog, nor even represent any physical sound. They can be completely synthetic, and generated using digital signal processing techniques. We will touch on this later in the text when discussing low-frequency oscillators (Chapter 2), phase vocoders (Chapter 8), and other concepts. It is important to note that unless additional information is stored, there is no distinction between those digital audio signals that were generated from conversion of analog signals and those that were generated from digital sound synthesis techniques (though, of course, real-world signals are likely to have more noise and more complex phenomena).

There are three important characteristics of almost any digital audio data: sample rate, bit depth, and number of channels.

Sample rate is the rate at which the samples are captured or played back. It is typically measured in Hertz (Hz), or cycles per second. In this case, one cycle represents one sample. An audio CD has a sample rate of 44,100 Hz, or 44.1 kHz. Higher sampling rates allow a digital recording to accurately record higher frequencies of sound, or to provide a safety margin in case of additional noise or artifacts introduced in the recording, processing, or playback; 48 kHz is often used in audiovisual production, and sample rates of 96 or 192 kHz are used in high-resolution audio, such as in DVD-Audio, or in professional audio production.

The *bit depth* specifies how many bits are used to represent each audio sample. The most common choices in audio are 16 bit and 24 bit. The bit depth also determines the theoretical dynamic range of the audio signal. In digital audio, amplitude is often expressed as a unitless number, representing a ratio between the current intensity and the highest (or lowest) possible intensity that can be represented. The maximum absolute value for this ratio is known as the *dynamic range.* In an ideal ADC, the dynamic range, in decibels (see below), is very roughly 6.02 times the number of bits. Thus, 16-bit audio could represent signals whose loudness ranges over 96 dB, e.g., from a quiet whisper to a loud rock concert.

The *number of channels* actually refers to the fact that audio content will often be composed of several different channels, each one representing its own signal. This is most often the case in stereo or surround sound, where each channel may represent the sound sent to each loudspeaker. Monaural audio, however, is typically encoded as a single channel. We will return to these concepts in Chapter 9.

Digital audio may be encoded with or without *data compression*. When data compression is used, sophisticated algorithms are used to encode and re-represent the data such that they take up much less space. Hence, a decoder must be used to convert the data back into time domain samples before playback. The compression can be either *lossless* (the decoded data are identical to the original data before compression) or *lossy*. Modern lossy audio compression techniques use knowledge of psychoacoustics to minimize the perceived degradation of audio that occurs when a substantial amount of the information contained in the original signal is discarded.

Data compression also introduces one more characteristic of audio data, the *bit rate*. This is the number of bits per unit of time. For lossless signals, this is simply the bit depth times the sample rate times the number of channels. For instance, CD audio would typically have a bit rate of 1,411.2 kbps (kilobits per second):

$$16\frac{\text{bits}}{\text{sample}} \cdot 44100\frac{\text{samples}}{\text{second}} \cdot 2\,(\#\,\text{channels}) = 1411200\frac{\text{bits}}{\text{second}} \tag{1.1}$$

For audio signals that have undergone lossy compression, the bit rate is usually greatly reduced. Most compression schemes, including mp3 and aac, transmit audio with a bit rate between 30 and 500 kbps.

It should be noted that there is a lot of fine detail regarding quantization, sampling, dynamic range, and lossy compression of audio data that has been omitted here. For the purpose of this text, it is sufficient to know the format and general meaning of these concepts, but the reader is also encouraged to refer to signal processing texts for more detailed discussion [1–5].

WHY 44.1 KHZ?

Perhaps the most popular sample rate used in digital audio, especially for music content, is 44.1 kHz, or 44,100 samples per second. The short answer as to why it is so popular is simple; it was the sample rate chosen for the Compact Disc and, thus, is the sample rate of much audio taken from CDs, and the default sample rate of much audio workstation software.

As to why it was chosen as the sample rate for the Compact Disc, the answer is a bit more interesting. In the 1970s, when digital recording was still in its infancy, many different sample rates were used, including 37kHz and 50 kHz in Soundstream's recordings [6]. In the late 70s, Philips and Sony collaborated on the Compact Disc, and there was much debate between the two companies regarding sample rate. In the end, 44.1 kHz was chosen for a number of reasons.

According to the Nyquist theorem, 44.1 kHz allows reproduction of all frequency content below 22.05 kHz. This covers all frequencies heard by a normal person. Though there is still debate about perception of high frequency content, it is generally agreed that few people can hear tones above 20 kHz.

This 44.1 kHz also allowed the creators of the CD format to fit at least 80 minutes of music (more than on a vinyl LP record) on a 120 millimeter disc, which was considered a strong selling point.

But 44,100 is a rather special number: $44{,}100 = 2^2 \times 3^2 \times 5^2 \times 7^2$, and hence, 44.1kHz is actually an easy number to work with for many calculations.

Working with Decibels

We often deal with quantities that can cover a very wide range of values, from very large to very small. The *decibel scale* is a useful way to represent such quantities. The *decibel* (dB) is a logarithmic representation of the ratio between two values. Typically, both values represent power, and hence, the decibel is unitless. One of these values is usually a reference, so that the decibel scale can represent absolute levels. The decibel representation of a level is then 10 times the logarithm to base 10 of the ratio of the two power quantities. Since power is usually the square of a magnitude, we can write a value in decibels in terms of the magnitudes or powers as

$$x_{dB} = 10\log_{10}\left(x^2/x_0^2\right) = 20\log_{10}\left(|x|/|x_0|\right) \tag{1.2}$$

If not specified, x_0 is usually assumed to be 1. So, for example, 1 million is 60 dB, and 0.001 is –30 dB. Whether a decibel or linear scale is used often depends just on which one best conveys the relevant information.

Level Measurements

The sound pressure measured from a source is inversely proportional to the distance from the source. Suppose a sound pressure p_1 is measured at distance r_1 from the source; then the sound pressure p_2 at distance r_2 can be calculated as

$$p_2 = p_1 r_1 / r_2 \tag{1.3}$$

Not all sources radiate uniformly in every direction. For example, a violin radiates more sound upward from the top of the instrument than from the sides or back. Measurements at different angles may therefore give different results.

The intensity I of a sound is given by the sound pressure times the particle velocity. It gives the sound power per unit area, and is measured in watts per square meter, W/m^2. Whereas pressure is proportional to the distance to the sound source, the intensity is proportional to the square of the distance to the sound source, giving a $1/r^2$ relationship.

A decibel scale is used to represent the very wide range of sound intensities that can be perceived. The sound intensity level, L_I, given in dB, is the log ratio of a given intensity I to a reference. The reference level is usually set to $I_0 = 10^{-12}$ W/m^2, which is considered to be roughly the threshold of hearing at 1 kHz.

$$L_I = 10\log_{10}(I/I_0) = 120 + 10\log_{10} I \tag{1.4}$$

Exact measurement of intensity is difficult, and the intensity values will fluctuate over time. So sound pressure level (SPL) is often used instead. The sound pressure level or sound level L_p is also given in decibels above a standard reference level, p_{ref}, of 2×10^{-5} N/m^2 = 20 μPa, the sound pressure threshold of human hearing.

$$L_p = 10\log_{10}\left(p_{rms}^2 / p_{ref}^2\right) = 20\log_{10}\left(p_{rms}/p_{ref}\right) \tag{1.5}$$

where p_{ref} is the reference sound pressure and p_{rms} is the RMS (root mean square, or square root of the average value of the squared signal) sound pressure being measured. In this text, we do not often refer to SPLs, since we will deal mostly with the processing of digital signals, where the physical sound level is not known.

For digital signals, level measurements are also given in a decibel representation. But now, decibels are measured relative to full scale, denoted dBFS. This is possible since most digital systems have a defined maximum available peak level.

Zero dBFS represents the maximum possible digital level. For example, if the maximum signal amplitude on a linear scale is 1 and the actual amplitude

is 0.5, then signal level would be defined as 20 $\log_{10}(0.5/1) = -3.01$ dBFS, or 3 dB below peak level. However, if RMS measurements are used, then the definition may be ambiguous. Different conventions are used for RMS measurements. Some RMS-based level measurements set the reference level so that peak and RMS measurements of a square wave will produce the same result, all dBFS measurements will be negative, and the maximum sine wave that can be produced without clipping will have value –3.01 dBFS.

An alternative (though much less common) definition gives the reference level so that peak and RMS measurements of a sine wave will produce the same result. A full-scale sine wave would be at 0 dBFS, but a full-scale square wave would exceed this, at +3 dBFS. In audio production (see Chapter 12), meters are provided so that the user will know if maximum levels are exceeded and clipping will occur.

In Chapter 6, we will discuss some methods of estimating the levels of digital signals for use in dynamics processing. Though given on a decibel scale, these estimates are tailored to the audio effect and may be different from the dBFS value described here.

Representing and Understanding Digital Signals

The time between samples may be given as T_s so that the sampling frequency is given as $f_s = 1/T_s$. Therefore, the digital input signal may be represented as discrete sampling of a continuous signal, $x(0)$, $x(T_s)$, $x(2T_s)$, … . If we consider only the sample number, then a finite signal consisting of N samples can be represented as $x[0]$, $x[1]$, …, $x[N-1]$.*

Representing Complex Numbers

Almost always, we will have real-valued signals, and our audio effects produce real-valued results. But a lot of the math and analysis of signals and effects is actually easier to do when generalizing the discussion to complex numbers. So it is extremely useful to keep a few properties of complex numbers in mind. First, any complex number can be written in several ways:

$$x = a + bj = re^{j\theta} \tag{1.6}$$

Here, a is the real part and b is the imaginary part, and j is defined to be $\sqrt{-1}$. The complex number can be plotted as a point on a plane where the

* In this text, brackets are generally used when we refer to functions of discrete integer samples, and parentheses are used for functions of continuous time.

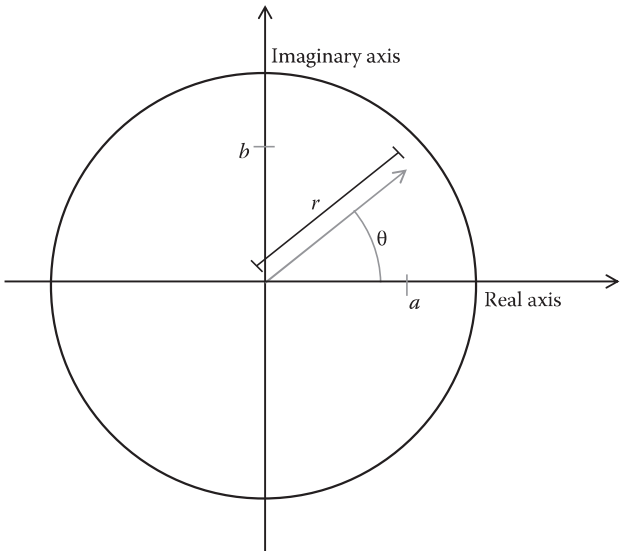

FIGURE 1.2
A complex number $a + jb = re^{j\theta}$ depicted in the complex plane. The unit circle, $\cos\theta + j\sin\theta = e^{j\theta}$ is also depicted.

x-direction is the real component and the *y*-direction is the imaginary one, in which case *r* is the magnitude of the vector from (0, 0) to (*a*, *b*), and θ is the angle between the *x*-axis and the vector, known as the phase. This is depicted in Figure 1.2.

Using a form of Euler's identity, $e^{j\theta} = \cos\theta + j\sin\theta$, we have

$$\begin{aligned} a &= r\cos\theta \\ b &= r\sin\theta \\ r &= \sqrt{a^2 + b^2} \end{aligned} \tag{1.7}$$

It is a bit tricky to derive the phase θ from *a* and *b*. It is not just the arctan(*b*/*a*), since this doesn't distinguish between the case when *b* is positive and *a* negative or when *b* is negative and *a* positive. That is, the arc tangent function has a range of $-\pi/2$ to $\pi/2$, but phase ranges from $-\pi$ to π. So we use the following:

$$\theta = \operatorname{atan2}(b,a) = \begin{cases} \arctan(b/a) & a \geq 0, b \neq 0 \\ \arctan(b/a) + \pi\,\mathrm{sgn}(b) & a < 0 \\ \text{undefined} & a, b = 0 \end{cases} \tag{1.8}$$

The conjugate of a complex number is simply defined as the same number, but with the sign changed on the complex component. Denoting complex conjugation by *, we then have, from (1.6),

$$x^* = a - bj = re^{-j\theta} \tag{1.9}$$

The square magnitude of a number is given by that number times its complex conjugate,

$$|x|^2 = x \cdot x^* = a^2 + b^2 = r^2 \tag{1.10}$$

Frequency and Time–Frequency Representations

Let's return to our time domain digital signal, $x[0]$, $x[1]$, …, $x[N-1]$. There are many other ways to represent this signal. Probably the most important is the discrete Fourier transform (DFT), which is intended to represent a finite, discrete signal in terms of its frequency components.

$$X[k] = \sum_{n=0}^{N-1} x[n]e^{-jnk2\pi/N} \quad 0 \le k, n \le N-1 \tag{1.11}$$

This converts the signal from being represented in terms of a real value at sample number n to representation in terms of a complex value at frequency bin k. In the same sense that time domain sample n corresponds to discrete sample at time nT_s, frequency bin k corresponds to frequency kf_s/N. Now notice that the output involves complex numbers. More precisely, $X[k]$ gives a phase and amplitude for the frequency content from $(k-1/2)f_s/N$ to $(k+1/2)f_s/N$.

This transformation has a large number of properties, but for understanding digital signals, the following are the most important:

$$\begin{aligned} y[n] = x_1[n] + x_2[n] &\leftarrow\rightarrow Y[k] = X_1[k] + X_2[k] \\ y[n] = ax[n] &\leftarrow\rightarrow Y[k] = aX[k] \end{aligned} \tag{1.12}$$

In other words, if we add two signals together, we add their discrete Fourier transforms together, and if we multiply a signal by a constant, we multiply its discrete Fourier transforms by the same constant.

Now suppose our signal x is a complex sinusoid, $x[n] = ae^{jnl2\pi/N}$, $n = 0, 1, \ldots, N-1$, where a is some constant. Then,

$$X[k] = \sum_{n=0}^{N-1} a e^{j2\pi(l-k)n/N} = \begin{cases} a\sum_{n=0}^{N-1} 1 = aN & l = k \\ a\dfrac{1-e^{j2\pi(l-k)}}{1-e^{j2\pi(l-k)/N}} = 0 & l \neq k \end{cases} \tag{1.13}$$

where we used a well-known identity,

$$\sum_{n=0}^{N-1} x^n = \frac{1-x^N}{1-x}.$$

So each frequency bin in a discrete Fourier transform represents the magnitude of a complex sinusoid. That implies that any finite signal can be represented as a sum of weighted sinusoids.

Finally, we can convert from the frequency representation back to the time domain using the inverse discrete Fourier transform (IDFT),

$$x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k] e^{j2\pi nk/N} \tag{1.14}$$

The DFT allows us to represent the signal as complex-valued frequency components. Each of these has a magnitude and phase. Thus, we can plot the magnitude and phase as a function of frequency for any signal. More common than magnitude plots, however, is the power spectrum that is given as the power in each frequency bin,

$$P(k) = |X(k)|^2 / N^2 \tag{1.15}$$

as a function of frequency.

The discrete Fourier transform and its inverse are very powerful tools, though very computationally intensive. But there is a lot of redundancy in the calculation. An implementation known as the fast Fourier transform (FFT) is commonly used. However, in its standard implementation, it requires that the number of samples be a power of 2.

Even with the FFT, it is still quite slow to compute over a large number of samples. Furthermore, the incoming signal may be very long or infinite, yet one would like to know the frequency content at any given time. Thus, the short-time Fourier transform (STFT) is used:

$$\mathrm{STFT}\{x[n]\} \equiv X[m,k] = \sum_{n=mR}^{N+mR-1} x[n]\mathrm{e}^{-j(n-mR)k2\pi/N} \quad 0 \le k \le N-1 \tag{1.16}$$

This provides estimates of the frequency content at times mR, where R is the hop size, in samples, between successive DFTs. And we often plot the *spectrogram*, $S[m,k] \equiv \cdot X[m,k]\cdot^2$, with S plotted as a color intensity as a function of time and frequency. The relationship between spectrogram and STFT is completely analogous to the relationship between power spectrum and DFT.

Figure 1.3 depicts the waveform, power spectrum, and spectrogram of an excerpt of a solo guitar performance.

Aliasing

An important concept, and one that will feature heavily in dealing with nonlinear processing, is aliasing. Suppose $x[n] = \cos(nl2\pi/N) = [e^{jnl2\pi/N} + e^{-jnl2\pi/N}]/2$, $n = 0, 1, \ldots, N-1$. Then,

$$\begin{aligned} X[k] &= \sum_{n=0}^{N-1} \left. e^{jnl2\pi/N} + e^{-jnl2\pi/N} \quad e^{-j2\pi kn/N} \right/ 2 \\ &= \sum_{n=0}^{N-1} \left. e^{jnl2\pi/N} + e^{jn(N-l)2\pi/N} \quad e^{-j2\pi kn/N} \right/ 2 = \begin{array}{ll} N/2 & l = k \\ N/2 & l = N-k \\ 0 & l \ne k, N-k \end{array}. \end{aligned} \tag{1.17}$$

So a real-valued sinusoid will give two nonzero frequency components, one at k and one at $N-k$. This implies that, for real valued input signals, the frequency spectrum from $N/2$ to N is the mirror image of the spectrum from 0 to $N/2$. We can also easily show that the spectrum for an input frequency sinusoid with frequency $f + f_s$ is the same as for a sinusoid with frequency f. So if a continuous signal is sampled at a frequency f_s, then the sampled signal cannot reproduce frequencies above $f_s/2$.

In fact, when sampling at a frequency f_s, any signal at a frequency $Nf_s + f_c$ or $Nf_s - f_c$, for any integer N, will be indistinguishable from a signal at frequency f_c. As an example of this, consider Figure 1.4. A sinusoid of frequency $0.7f_s$ is sampled at frequency f_s. The samples that result could equally well represent a sinusoid of frequency $0.3f_s$. This property is known as *aliasing*, since these signals are aliases of each other.

For this reason, signals should in general, be band limited before sampling, so that they contain no frequency components greater than $f_s/2$. $f_s/2$ is known as the *Nyquist frequency*. This is also a consequence of the Nyquist–Shannon sampling theorem, which states that such band limited signals can be (in theory) completely reconstructed when sampled at a frequency of at least f_s.

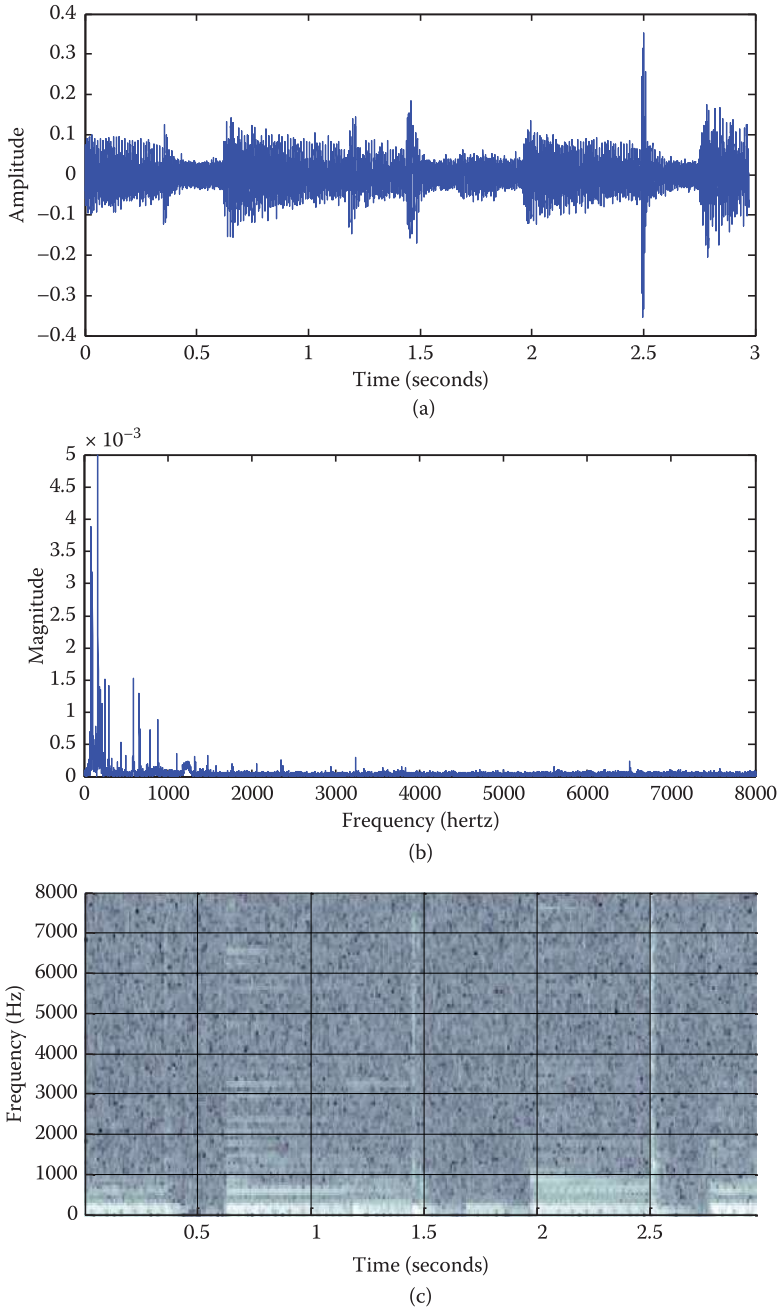

FIGURE 1.3
The time domain waveform (a), frequency domain power spectrum (b), and spectrogram of a 3 s excerpt of a solo guitar performance (c).

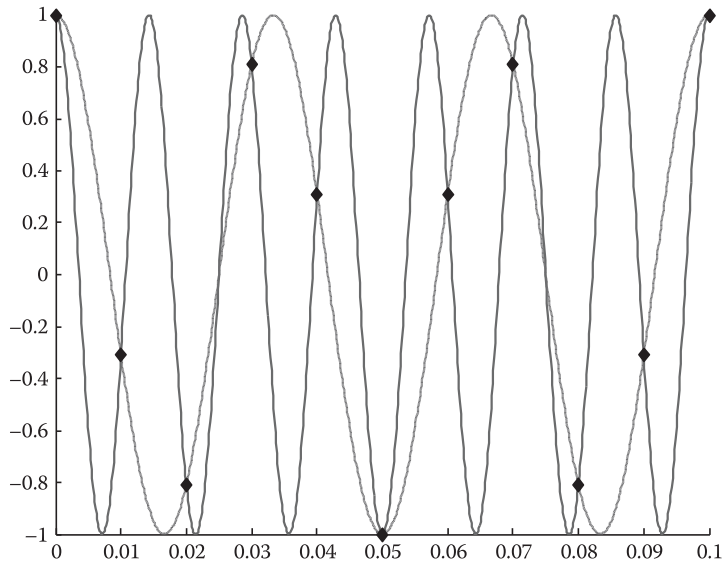

FIGURE 1.4
Two sampled sinusoids, one with frequency 0.3 f_s and the other with frequency 0.7 f_s. They appear identical when sampled at a frequency f_s.

Modifying and Processing Digital Signals

Up to now, we've been looking at how to represent and analyze signals. But how do we modify them? We start by introducing the difference equation, a formula for computing the nth output sample based on current and previous input samples and previous output samples. A linear, time-invariant digital filter may be given as a *difference equation*,

$$y[n] = b_0x[n] + b_1x[n-1] + \ldots b_Nx[n-N] - a_1y[n-1] - \ldots a_My[n-M] \quad (1.18)$$

where x is some input signal and y is the output signal. The constants b_0, …, b_N and a_0, …, a_N are known as coefficients or multipliers. Figure 1.5 shows this as a block diagram, with the input signal entering at the top left and the output signal appearing on the right.

There are many different ways to represent this system. One of the most important is the *impulse response*, which describes the output when the input is just a single pulse:

$$h[n] = y[n], x[n] = \delta[n] \equiv \begin{matrix} 1\, n = 0 \\ 0\, n \neq 0 \end{matrix} \quad (1.19)$$