F. Richard Moore: "The Mathematics of Digital Signal Processing" in Digital Audio Signal Processing, John Strawn, Ed. Kaufmann, 1985.

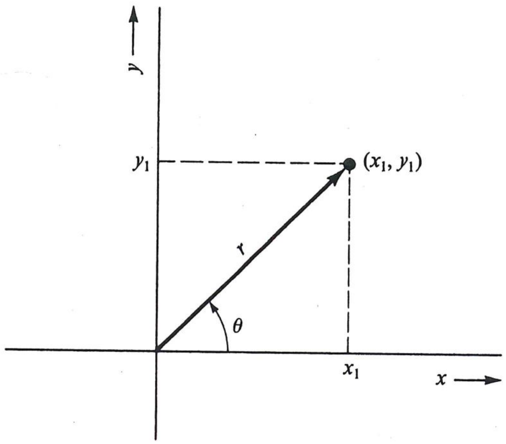

Fig. 1.6. Graph of $c = x_1 + iy_1$

a) 1 000 000 + 100 000 + 10 000 + . . . (infinitely many terms)

b) 100 + 200 + 400 + 800 + . . . to 10 terms

c) $10^6 + 2.5 \times 10^5 + 6.25 \times 10^4 + \ldots$ to n terms

5. If we graph a complex number $c = x_1 + iy_1$ on a plane, we can use x_1 and y_1 for the horizontal and vertical coordinates of a point on that plane (see fig. 1.6). If we draw a straight line from the origin (point (0, 0)) to point (x_1, y_1), we could also use the length r and angle θ of that line to define the locations of the point (x_1, y_1). Find r and θ in terms of x_1 and y_1. (Hint: Pythagoras' theorem states that the square of the length of the hypotenuse of a right triangle is equal to the sum of the squares of the other sides.)

6. Show that $\sin^2(\theta) + \cos^2(\theta) = 1$.

7. Show that $e^{i\theta} = \cos(\theta) + i\sin(\theta)$. (Hint: Use the summation formula, also called the *power series expansion,* for e^x.)

Part II
SAMPLING, TRANSFORMS, AND DIGITAL FILTERING

In part I of this tutorial we discussed some of the basic mathematical ideas relevant to the processing of digital signals. Now we turn to the application of these and other concepts, operating on the assumption that the reader

understands everything in part I thoroughly (although some of this second part can be understood without following the mathematical arguments). Again, it will be impossible to give a detailed account of all the techniques of digital signal processing, because there is simply too much to cover in an article. Thus, the ideas chosen for inclusion here are only the most fundamental, which is to say (hopefully) the most important. Armed with the knowledge presented here, the reader should be able to understand much of the literature in the field, even though we will continue to omit calculus from our mathematical concerns. As in many subjects, notations are often used only as a kind of shorthand for concepts which can be adequately explained without resorting to "higher" mathematics. As we saw in part I, however, the better our mathematical facility, the easier it is to solve certain problems which are otherwise difficult or, at the very least, tedious. Thus we will continue to use extensively the most powerful mathematics at our disposal, that of complex exponentials, in our treatment of sampling and transforms and in our introduction to the concepts of digital filtering.

Before we examine these concepts, however, some words are in order about the general nature of the subject we are studying. Digital signal processing, computer programming, and acoustics all relate to computer music in a similar way: unlike typical subfields such as harmony or counterpoint, which exist as subdivisions of a global realm of study, digital signal processing, computer programming, and acoustics are all *complete fields in themselves,* each with its own motivations, jargon, and subfields. Perhaps the most fascinating aspect of computer music is its attempt to synthesize from such a vast array of knowledge the keys to a rich and expressive sonic art. Since classical times, when mathematics and physics were considered to be subfields of music (recall Pythagoras' investigations of the pitch of vibrating strings of different lengths), music and science have traveled increasingly divergent paths in pursuit of ever-elusive truths and beauty, which indicates, at the very least, that one great difficulty for computer musicians will be to bridge the terminology gaps among several fields at once.

This means that we must be patient and willing to let each field describe itself in its own terms before we can progress to an understanding of how to rephrase these statements in musical terms. We cannot start out by directly asking digital signal processing specialists questions such as "How can we make an oboe sound like a clarinet?" since, for historical reasons, the information is not couched in these terms. We can, however, keep such questions in mind as we study, on the assumption that an understanding of the terminology of the field will allow such questions to be rephrased as problems that can be dealt with. Often the answer will be that the question asks for unknown or poorly understood techniques to be applied, but just as often the search will lead us to other revelations: answers to questions which are begging to be asked! It will be a useful exercise, therefore, to try to imagine what *could* happen to some sound if a particular process were applied to it. Certainly much is already known about the musical effects of digital signal processing, but at this point much more is to be learned.

Also, we must keep in mind that digital signal processing is neither programming nor acoustics; hence even a perfect understanding of it would not necessarily show us how to create satisfying music with a computer. It is, however, a powerful way to think about the manipulation and control of sounds, and as such it will most likely represent an important prerequisite to our understanding of how to create music in new and expressive ways.

SAMPLING

What is a sampled signal? When we watch a movie we are *looking at* a stream of separate, discrete photographs flowing by at a rate of 24 frames (photographs) per second, but we are *seeing* something quite different. The apparent continuous motion on the screen is really the result of *sampling* the position of the various people and objects in the original picture at a sufficient rate to ensure that no important detail of the motion is lost. If the motion were sampled more slowly, say, at 5 times per second, the motion would appear jerky and discontinuous, as it does under strobe lights at discotheques. We can imagine an experiment that must have been executed several times in the history of the motion picture industry: We start out filming various moving scenes at a slow, "flickery" rate and then gradually (or in steps) increase the frame rate until the motion appears smooth and continuous. Of course, we will eventually run into practical difficulties, such as the sensitivity of the film to light, since an increasing frame rate implies a decreasing exposure time for each frame. But, hopefully, before such limits are reached, a smooth rendering of motion will be achieved; indeed, the movie industry has settled on 24 frames per second as a standard which works well enough. But does it work perfectly? The answer is of course not, as anyone who has ever watched wagon wheels in a Western movie can verify. As the wagon starts out from a standstill, the wheels appear to turn slowly forward; as the speed increases, they first appear to go faster, then begin to slow down and go backward, then to stop, then to go forward again, and so on. They never appear to stop completely when the wagon is moving, because their speed blurs their picture, but neither is their motion rendered accurately by the film process. If we were to increase the frame rate of filming, what would be the effect on the image of the turning wheels?

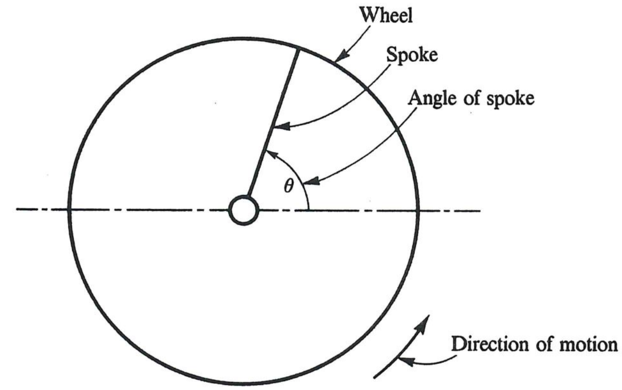

Fig. 1.7. Turning one-spoke wagon wheel

The wagon could go faster before its wheels started to appear to slow down or go backward, but the point is that for any filming speed (sampling rate) there is some upper limit on the rapidity with which motion can take place and still be rendered accurately on the screen. (Since most motions are slow enough, the movie industry deems it unnecessary to increase the frame rate for the sake of rendering chase scenes more believable.)

In order to understand an important aspect of the sampling process, let us imagine that we are filming a documentary on the motion of a one-spoke wagon wheel (see fig. 1.7). Let us arbitrarily say that when the spoke points to the right it is in position zero and that any other position is defined as the angle, measured counterclockwise, from position zero. Hence, at an angle of $\pi/2$ radians the spoke points straight up, at π radians it points to the left, and so on. If the wheel completes exactly one full counterclockwise revolution per second, we can describe its rotational velocity by saying that it is turning at a rate of $+2\pi$ radians per second (clockwise motion would indicate a negative velocity); two counterclockwise revolutions per second would correspond to 4π radians per second, and, in general, F revolutions per second would correspond to $2\pi F$ radians per second. The motion of the wheel is clearly periodic, with a *period* of $T = 1/F$ seconds. F is the *frequency* (repetition rate) with which the wheel turns, so it can be measured in cycles (or repetitions) per second (*hertz,* abbreviated *Hz*), and the quantity $2\pi F$ is called the *radian frequency* at which the wheel rotates (measured in radians per second).

Let us begin filming the rotating wheel at the standard rate of 24 frames per second. Assuming that the wheel turns smoothly from the starting position zero at a rate of $F = 1$ Hz, successive frames of the movie will be

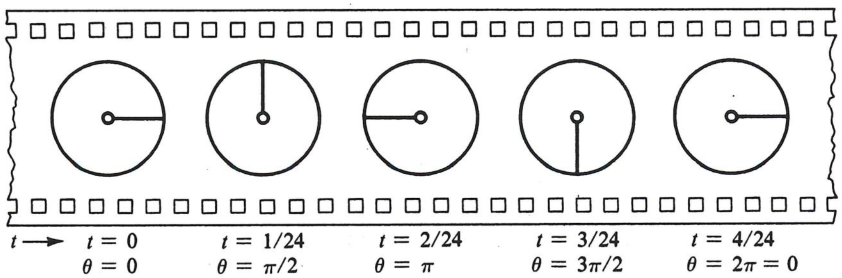

Fig. 1.8. Movie of one-spoke wagon wheel turning (apparently) at $f = 6$ Hz, shot at 24 frames per second

taken at $\theta = 0$, $\theta = (1/24)\cdot 2\pi F = \pi/12$, $\theta = \pi/6$, etc. Since the wheel turns only $1/24^{\text{th}}$ of a revolution at each frame, we can expect our documentary to represent the facts as they are, a good quality for any documentary. Our camera is of course ideal, so it never blurs the picture of the wheel no matter how fast the wheel moves. At $F = 6$ Hz, we would get a series of frames such as those shown in figure 1.8; the wheel turns $6/24 = 1/4$ of a turn at each frame.

At $F = 12$ Hz, the wheel turns halfway around at each frame and its filmed image appears to oscillate back and forth with maximum rapidity (under these conditions). Why maximum? What larger motion could the spoke make on successive frames? If we set F even higher, say, to 18, then the wheel makes 3/4 of a complete revolution at each frame, but this is also indistinguishable from the wheel turning *backward* at a rate of -6 Hz, equivalent to running the film in figure 1.8 in reverse. To make this clearer, consider the case of $F = 23$ Hz. At each frame the wheel turns 23/24 of a revolution, almost all the way from where it starts. To verify that this will appear as a slow backward motion (-1 Hz) one only has to go to a Western movie and watch carefully, taking into account the greater number of spokes on most wagon wheels. Finally, at $F = 24$ Hz, the wheel does not appear to move at all! Thus if we start at $F = 0$ Hz and gradually increase the speed to $F = 24$ Hz, we see the wheel move slowly forward (counterclockwise, i.e., with small positive frequency), go faster to a maximum at $F = 12$ Hz, then slow down while turning in the opposite direction (clockwise motion, negative frequency) and stop completely at $F = 24$ Hz. If we further increase F from 24 to 48 Hz, we see the same sequence of events. This is because all frequencies outside the range 0 to 12 Hz are indistinguishable, except for being positive or negative, from frequencies between 0 and 12 Hz, according to the relationship

$$F_a = \left| F - \frac{(k+1)R}{2} \right|, \qquad \frac{kR}{2} \le F \le \frac{(k+2)R}{2} \tag{1.1}$$

where

F_a is the "apparent" frequency in Hz,
F is the actual frequency in Hz,
R is the sampling rate in Hz (samples per second), and
k is any *odd* integer which satisfies the inequality.

($|F|$ is the "absolute value" or "magnitude" of F without regard to its plus or minus sign.) Thus if the wagon wheel turns at 28 Hz ($=F$) and we film it at 24 frames per second ($=R$), we choose k to be an odd integer which satisfies

$$\frac{kR}{2} \le F \le \frac{(k+2)R}{2} \equiv 12k \le 28 \le 12(k+2)$$

The only odd integer which satisfies this inequality is $k = +1$, since

$$\frac{1\cdot 24}{2} \le 28 \le \frac{3\cdot 24}{2}$$

Therefore the apparent frequency is

$$F_a = \left| 28 - \frac{2\cdot 24}{2} \right| = 4 \text{ Hz}$$

The point is that while F may be any frequency whatsoever, F_a is restricted to a definite range of frequencies which depends on the sampling rate. We can see that F_a and F are the same only if $(k+1)R/2$ is equal to 0, which is true only if $k = -1$. Then

$$F_a = F - 0, \qquad -\frac{R}{2} \le F \le +\frac{R}{2}$$

or simply $F_a = F$ only if $|F| \le R/2$. If $|F| > R/2$, then $F_a \ne F$ and we say that F_a is an *alias* of F. This phenomenon of aliasing, or foldover, is found in all sampled systems, whether they are filmed wagon wheels or digitized waveforms. Equation 1.1 is a statement of the *sampling theorem,* which states that any simple harmonic variation (i.e., a sinusoidal variation of a one-dimensional quantity, a circular motion in a two-dimensional quantity, etc.) which occurs at a rate of F Hz must be sampled at least $2F$ times per second in order to avoid aliasing.

The reader may have already noticed that if the sampling rate is *exactly* twice the frequency being sampled ($R = 2F$), then equation 1.1 is ambiguous, since there are two different values of k which will satisfy the inequality. We will come back to this fine point later on.

QUANTIZATION

In order to process sounds with a computer, we represent their waveforms as sequences of discrete, finite-precision numbers. These are the samples of the instantaneous amplitude of such waveforms taken at brief, regular intervals in time. Any musical waveform can be modeled as a sum of sinusoidal vibrations, each with a particular (though possibly time-varying) amplitude, frequency, and phase. Thus, in order to represent a continuous (analog) waveform accurately with discrete (digital) samples, we must ensure that the sampling frequency is at least two times greater than that of the highest frequency component of the original waveform. The sampling theorem (eq. 1.1) then assures us of the accuracy of our rendition of the waveform if it is *band-limited* to the frequency region below one-half the sampling rate.

The sampling process is achieved by using an *analog-to-digital converter* (ADC), which generates a numerical value in computer-readable form (typically a binary number of 12 to 16 binary digits, or *bits*). The converter's numerical output is proportional to the electrical level (either voltage or current) at its input, which is sampled at a rate ranging from a few hertz (for signals such as seismic waves) to 50 kilohertz (for high-quality audio signals). The analog waveform is typically passed through a low-pass filter to attenuate any components at frequencies greater than half the sampling rate (see fig. 1.9), since these are generally impossible to remove from the digital signal owing to the aliasing effect described above. Whether the distortion due to aliasing produces noticeable effects in musical sounds depends on the relative strengths of the aliased components, but severe aliasing is generally much more noticeable and irritating in sounds than it is in movies of wagon wheels.

The analog-to-digital converter produces a B-bit binary value to represent the instantaneous amplitude of the analog signal at each sample. Since B binary digits may represent at most 2^B different values, this means that the ADC must choose the closest B-bit value available for each sample. Thus, if the band-limited analog signal varies between, say, $+10$ and -10 volts, and $B = 10$ bits, the entire 20 volt (peak-to-peak) range may be represented to an accuracy of $20/2^{10} \approx 0.02$ volts at each sample. In other words, the true voltage amplitude differs from its binary representation by at most ± 10 millivolts, for an accuracy of about $\pm 0.05\%$. Such inaccuracies are often

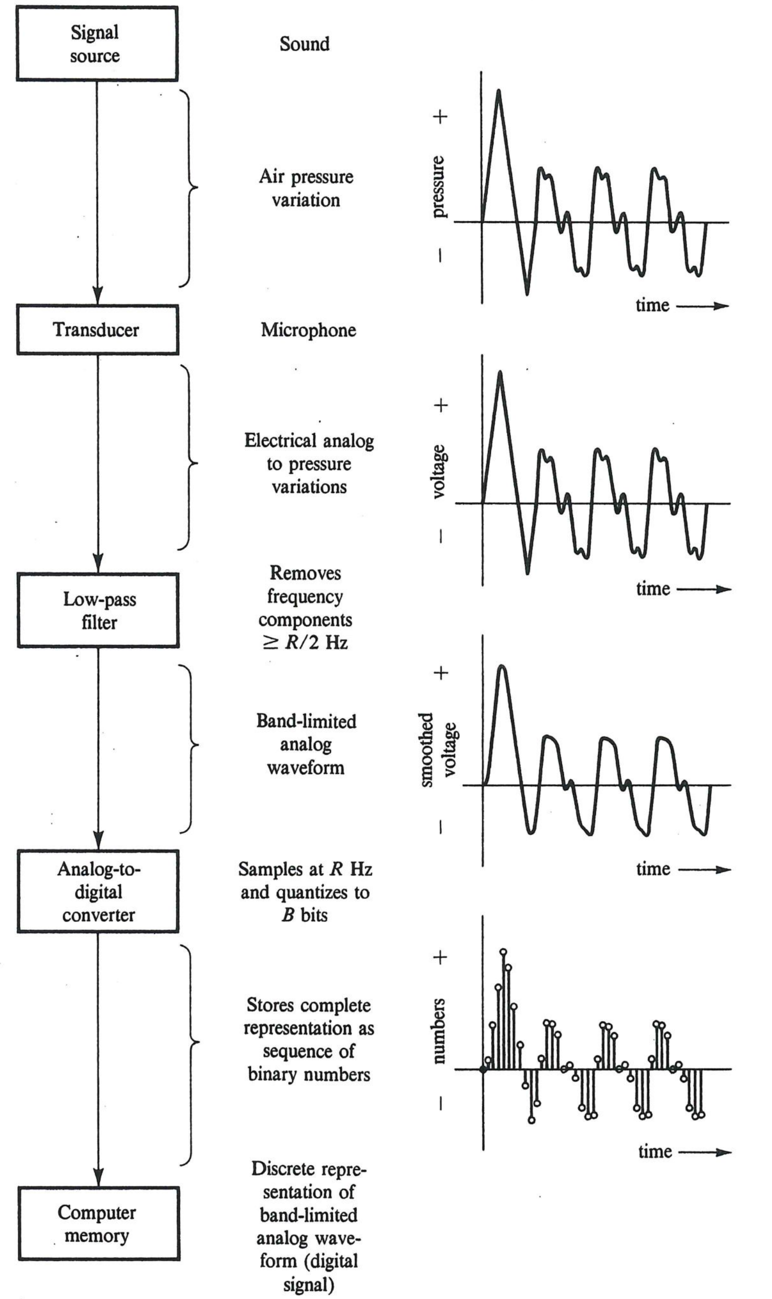

Fig. 1.9. Steps by which a continuous signal is converted into a digital signal for subsequent computer processing

significant, since we can view them as equivalent to a small, constant amount of random noise being added to an otherwise perfectly represented signal. This *quantization* noise, as it is called, is the digital equivalent to tape or amplifier hiss, and it is usually characterized by a *signal-to-quantization noise ratio* (SQNR), expressed in dB (decibels):

$$\text{SQNR in dB} = 20 \log_{10} \frac{\text{signal amplitude}}{\text{noise amplitude}} \tag{1.2}$$

Thus, if a maximum amplitude of 10 volts corresponds to the maximum binary value for a 10-bit ADC, the noise amplitude will be 2^{-10} as great as that of the strongest signal, yielding an SQNR of

$$\begin{aligned} 20 \log_{10} \frac{10}{10 \cdot 2^{-10}} &= 20 \log_{10} 2^{10} \\ &\approx 20 \log_{10} 1000 \\ &= 60 \text{ dB} \end{aligned}$$

The reader may wish to verify that under these conditions and assumptions the SQNR of a B-bit ADC is approximately $6B$ dB. However, two caveats must be kept in mind. First, we are assuming that the quantization error may be treated as a random noise independent of the signal, which is certainly questionable, especially at low sampling rates or for small numbers of bits. Second, if the analog signal amplitude is not maximal, it must be remembered that the noise level remains the same, rendering the quantizing noise more audible and bothersome for very soft sounds than for loud ones. ADCs are available with 8 to 16 bits of resolution, and while the issue has not quite been resolved, it seems that 12-bit ADCs give minimally acceptable sound quality and that improvement beyond 16 bits is probably unnecessary, since at that accuracy the noise levels of transducers and amplifiers become predominant. Special bit-coding techniques may eventually reduce the amount of data in a digital signal, but so far most computer music programs have not dealt with this possibility.

Producing sounds with a computer is just the reverse of the process diagrammed in figure 1.9. A *digital-to-analog converter* (DAC) is used to convert binary numbers to voltage levels; a low-pass filter "smooths" the waveform by passing only those frequencies less than half the sampling rate; and the resulting analog signal is then amplified and transduced by a loudspeaker or earphones.

The numerical version of the signal may be stored in computer memory and processed in a variety of ways. Two of the most important of these processes are transforming the digital signal, in order to *analyze* its frequency spectrum, and filtering, which *alters* its frequency spectrum. The information gained by analyzing the digital signal may be used to understand how such signals might be synthesized—a common objective in computer music—and filtering is a major technique for controlling the quality, or timbre, of sounds produced.

DIGITAL SIGNALS

A digitized signal is not represented as a function of a continuous time variable (t), but rather as a function of discrete values of time (n). In other words, we can think of a digital signal as a *sequence* of numbers, each representing the instantaneous value of a (presumably) continuous time function. Furthermore, we will always assume that the samples are *uniformly spaced* in time. Two basic notations for such discrete-valued functions are commonly used in the digital signal processing literature, either

$$x(n), \qquad N_1 \leq n \leq N_2, \qquad n \in \mathbf{I}$$

or

$$x(nT), \qquad N_1 \leq n \leq N_2, \qquad n \in \mathbf{I}$$

Both of these notations have the same meaning except that in the second the sampling period, T, is shown explicitly. Since it is easy enough to remember that the relationship between successive integer values of n and time depends on the sampling rate, we will use the first notation here. For example, a one-second sine wave at a frequency of 100 Hz is represented as

$$f(t) = \sin(\omega t), \qquad 0 \leq t \leq 1$$

where $\omega = 2\pi F = 2\pi \times 100$. If we sample this waveform at 500 Hz, the discrete form of this equation will be written

$$x(n) = \sin(\omega n), \qquad 0 \leq n \leq 499$$

again with $\omega = 2\pi F$, but n and t are *not* equal to each other. Properly speaking, the quantity nT, where $T = 1/500$ second, is equal to discrete values of t for integer values of n. Note also that we will generally number N samples from 0 to $N - 1$.

Two important special functions which we will need in our discussion are the *impulse*, or *unit sample*, function and the *complex exponential* function. The digital impulse function is defined to be equal to 1 only if its argument is 0 and it has a 0 value otherwise, i.e.:

$$u(n) = \begin{cases} 1 & n = 0 \\ 0 & n \neq 0 \end{cases} \tag{1.3}$$

If we want the impulse to occur on some sample $n_0 \neq 0$, it follows from the definition that the following equation is true:

$$u(n - n_0) = \begin{cases} 1 & n = n_0 \\ 0 & n \neq n_0 \end{cases} \tag{1.4}$$

Figure 1.10 shows a specific example for $n_0 = 4$.

The complex exponential function cannot be graphed quite as easily as the unit sample function, because it has values consisting of both real and imaginary parts:

$$e^{j\omega n} = \cos(\omega n) + j\sin(\omega n) \tag{1.5}$$

where $j^2 = -1$. Figure 1.11 shows *two* graphs of this function, one of its real part and the other of its imaginary part. From looking at the figure we cannot tell the frequency of the sinusoidal waveforms depicted. But we *can* see that there appear to be eight samples in each period of the sinusoidal waves and deduce that the frequency of these waveforms must therefore be

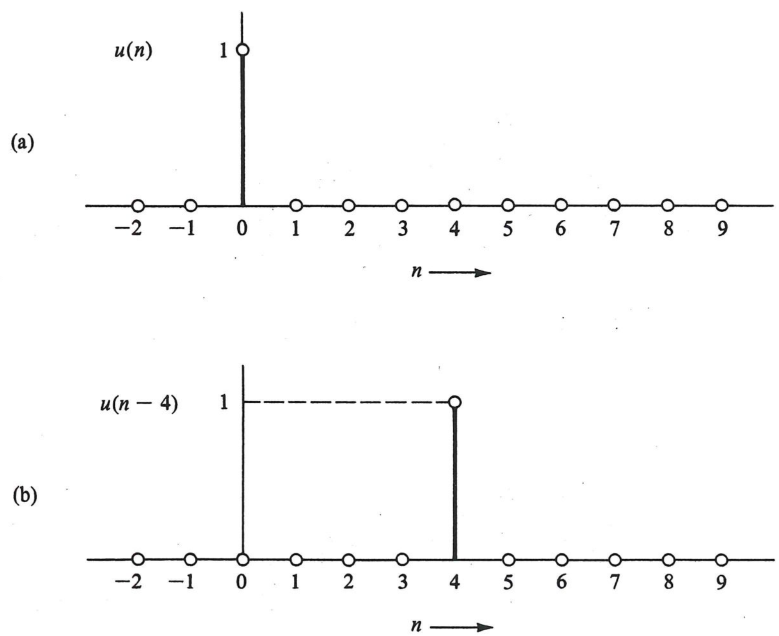

Fig. 1.10. Unit sample (digital impulse) function $u(n)$ and delayed unit sample function $u(n - n_0)$ for the case $n_0 = 4$

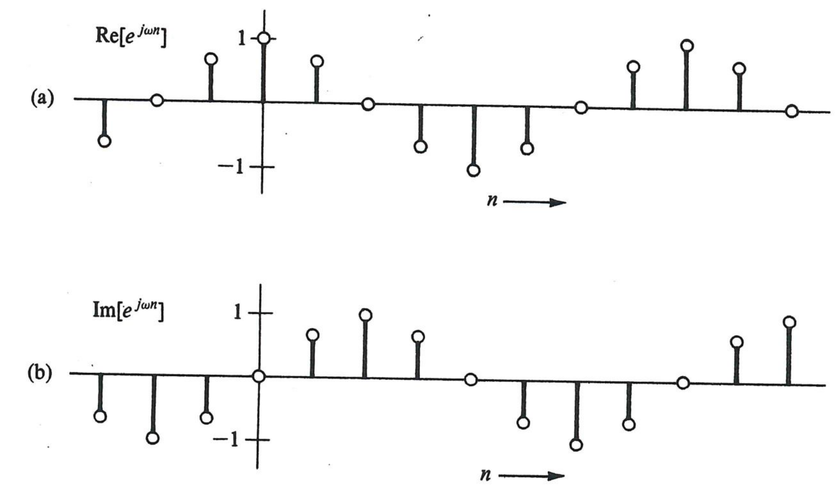

Fig. 1.11. Complex exponential function $e^{j\omega n}$, shown as graphs of its real and imaginary parts, $\omega = 2\pi/N$, $N = 8$

one-eighth the sampling rate. Waveforms obtained by sampling sounds do not have real and imaginary parts, of course; we say that such waveforms are *pure real* or, equivalently, that they are complex with a zero imaginary part.

SPECTRA

As almost everyone knows, two different musical instruments playing the same pitch at the same loudness for the same duration from the same direction still sound different because of what is called their tone color, or timbre. Unfortunately, this subtractive definition of timbre only says what it is not: timbre is that aspect of a sound which is not its pitch (if it has one), loudness, duration, or directionality. What is left is just the microstructure of the sound, and in order to examine it we need a way to literally dissect sounds, i.e., to separate them into their constituent parts. Obviously, a complete description of *all* the constituent parts of a particular sound will include information about its pitch, loudness, and so on, and in a broad sense we might even include these qualities in our definition of timbre. Except on a few electronic instruments such as Theremins, or electronic organs, the tone color does not remain the same when different notes are played, owing to varying string or tube lengths, lip tension, and so on. Since all these variations in tone quality are quite relevant in accounting for the characteristic sounds of musical instruments, we can see that analysis of the microstructure of a sound is likely to yield information only about that particular sound. Even two successive notes played on the same instrument by the same performer in the same manner are likely to have strikingly different microstructures. It is the study of this tonal microstructure, and its relationship to what we hear, that is one of the major concerns in computer music research, for it is here that the complexities of the physics of the instrument, of room acoustics, and of the psychology of the listener enter in.

Our model for describing this microstructure is called the *spectrum* of a sound, by analogy to the spectrum of a beam of light, which may be obtained by passing the light through a prism. The prism has the property that light made up of different frequencies, or colors, is refracted by varying amounts, the index of refraction depending on the component, or primary, color in question. By observing the intensities of the light at these different frequencies, we are able to determine the makeup of the original light beam. If the beam is "pure white" light, we obtain a "full spectrum," proverbially the colors of the rainbow.

The prism for sounds is Fourier analysis. By applying the Fourier transform to the waveform of a sound, we can mathematically determine just which amounts of which frequencies are responsible for that particular waveshape and we can use our analysis as a guide in synthesizing that sound. If the sound consists of *all* audible frequencies in roughly the same amounts, we call the result *white sound,* by analogy to white light. Unfortunately, since a rainbow is considerably more appealing than the steady steamlike hiss of its audible counterpart, we usually refer to this sound as white *noise*. If, however, some frequencies are considerably more predominant than others, the sound becomes *colored;* and if the relationships among these predominant components become roughly *harmonic* (i.e., the frequencies are integer multiples of a single frequency, called the *fundamental* frequency), the tone will acquire a more definite pitch. When the waveform consists entirely of harmonically related frequencies, it will be periodic, with a period equal to the reciprocal of the fundamental frequency (which need not be present for the pitch to be heard at that frequency).

The measurement of sound spectra is complicated by the fact that the spectra of almost all sounds change both rapidly and drastically as time goes by. This situation is worsened by the fact that the accuracy with which we can measure a spectrum inherently decreases as we attempt to measure it over smaller and smaller intervals of time. The spectrum of any instant during the temporal evolution of a waveform does not even exist; for example, we could scarcely tell anything at all about the frequency components of a digital signal by examining a single sample. We can measure what happens

to the spectrum only *on the average* over a short interval of a sound—perhaps a millisecond or so. The longer the interval, the more accurate our measurement of the *average* spectral content during that interval, but the less we know of the variations that occurred during that interval. Thus, the problem of spectral measurement can be seen to be one of finding the best compromise between these opposing goals. Just how much accuracy is needed is still an open question in the realm of musical psychoacoustics: in some cases our ears seem to be much more tolerant of approximations than in others. The historical model of spectra as measured by Hermann von Helmholtz ([1913] 1954) is clearly inadequate for believable resynthesis (Helmholtz was able to determine the average value of spectral components over the duration of entire notes played on individual instruments). A more recent model characterizes a note by attack, steady-state, and decay segments. Such a model is certainly an improvement, but it has limitations when applied to the problems of producing "connected notes" (i.e., to problems of musical *phrasing*). Besides, the "steady state" of any real tone is not "steady" at all.

This discussion is not intended to imply that the situation is hopeless, but only that it is subtle and complex and that it is as important to appreciate the limitations of the spectral measurement techniques presented here as it is to realize their power. There can be little doubt that these techniques, and their relatives and extensions, will be the ones which will eventually yield the basis for a richly expressive computer music.

THE DISCRETE FOURIER TRANSFORM

The two most commonly used transforms in digital signal processing are the *discrete Fourier transform* (DFT) and the *z transform*. The DFT is used to calculate the spectrum of a waveform in terms of a set of harmonically related sinusoids, each with a particular amplitude and phase. It is usually implemented by means of a particularly efficient algorithm known as the FFT (for *fast Fourier transform*), the discovery of which has made spectral computation a much more practical reality than it would be otherwise. Since the DFT is less restrictive (albeit less efficient), and since the FFT is well documented for those with a basic understanding of the DFT (see, for example, Digital Signal Processing Committee [1979]), we will consider only the DFT here. The *z* transform, unlike the FFT, is not something that is typically calculated with a computer, but is rather a mathematical tool used primarily in the theory of digital filters. It is in a sense more general than the DFT, since it includes the DFT as a special case, and it is of considerable interest in the general theory of digital signal processing.

The fundamental operation of the DFT is to decompose an arbitrary waveform into its spectrum. The spectrum of a waveform is a description of that waveform in terms of a number of "basic building blocks" for waveforms, which in the case of the DFT are sinusoids with harmonically related frequencies. By analogy, if we factor an integer into its prime factors, we have in a sense "decomposed" the original number into basic numerical "building blocks"; for example, $340 = 1 \times 2 \times 2 \times 5 \times 17$. The building blocks themselves (the prime numbers) are just those numbers which cannot be further decomposed: they can be expressed only as one times themselves; hence, they are the components of which other numbers are formed and not vice versa. Finally, factoring any integer into primes yields an answer that is *unique:* there is no other set of prime numbers which, when multiplied together, will yield 340 except the set stated above. Perhaps we picked the number 340 in the first place not on the basis of its prime factors but because it was the sum of the ages of everyone in a small room: $340 = 10 + 28 + 32 + 40 + 50 + 50 + 60 + 70$. This is clearly another way to decompose 340, but it is not unique, since an infinite number of sequences sum to 340.

Similarly, the basic building block used by the DFT for waveforms is the sinusoid. The DFT works by treating N samples of a waveform as if it were one N-sample period of an infinitely long waveform composed of a sum of sinusoids which are all harmonics of a fundamental frequency corresponding to the N-sample period. And, like the prime factors discussed above, this set of harmonically related sinusoids, each with a particular amplitude and phase, is unique: no other set of sinusoids could be summed together to obtain the original waveform. Of course there may be other, possibly nonunique ways of decomposing a waveform, just as 340 could be nonuniquely decomposed into nonunique sums of ages rather than a unique product of primes. In fact, other unique decompositions for waveforms exist besides the sum of sinusoids yielded by the DFT, but we will not consider them here.

We should also discuss the concept of *energy at a particular frequency*. A waveform may, in signal processing parlance, have energy at, say, 100 Hz, which means that at least one sinusoidal component with a frequency of 100 Hz and a nonzero amplitude is present in the vibration pattern. Energy in this case designates "that which exists" at 100 Hz which does not exist at, say, 110 Hz. The DFT functions by measuring the amplitudes of sinusoidal components at particular frequencies in a waveform, and since energy can be shown to be proportional to the square of amplitude, we can see that this process measures the energy at such frequencies. We could imagine accomplishing this process in a laboratory with a set of electrical audio filters, each of which would pass energy only at one frequency and block energy at all others. This bank of filters could be used to detect energies at a set of frequencies for an arbitrary input signal. The DFT accomplishes this mathematically in the following way.

Suppose $x(n)$ is a sequence of numbers representing N samples of one period of a waveform with a period of N samples. For example, let $x(n) = A\sin(\omega n)$, with $\omega = 2\pi/N$ and $0 \le n \le N-1$. For $N = 8$ we would have the sequence

$$x(n) = 0, \frac{A}{\sqrt{2}} (\approx 0.707\,A), A, \frac{A}{\sqrt{2}}, 0, -\frac{A}{\sqrt{2}}, -A, -\frac{A}{\sqrt{2}}$$

We can measure the energy at frequency ω by extracting the amplitude, A, of the sinusoid at this frequency. This is accomplished in this case by forming the product of $x(n)$ with $\sin(\omega n)$ and adding up the numbers in the resulting sequence, since

$$\begin{aligned}\sum_{n=0}^{N-1} x(n)\sin(\omega n) &= 0 + \frac{A}{2} + A + \frac{A}{2} + 0 + \frac{A}{2} + A + \frac{A}{2} \\ &= 4A = N\frac{A}{2}\end{aligned}$$

The result is $A/2$, one-half the amplitude of the sinusoid at frequency ω, scaled by N, the number of samples under consideration. We could not simply sum together the numbers in the $x(n)$ sequence to obtain the same result, since summing over any integral number of periods of a sinusoid yields a zero result. This is due to the symmetry of the sine and cosine functions above and below the horizontal axis. However, by multiplying $x(n)$ by $\sin(\omega n)$, we form the sequence $x(n)\sin(\omega n) = A\sin^2(\omega n)$, and all values of $\sin^2$ are nonnegative.

Thus we have "extracted the amplitude" of the sinusoid at frequency ω in $x(n)$ by purely mathematical means. What would happen if we were to "extract the amplitude" of the component of $x(n)$ at frequency 2ω? With $x(n)$ defined as before, we expect that no such component will be detected, i.e., that its amplitude will be 0. In order to verify that this is so, we form the product sequence $x(n)\sin(2\omega n)$ and add up the resulting numbers:

$$\begin{aligned}\sum_{n=0}^{N-1} A\sin(\omega n)\sin(2\omega n) &= \sum_{n=0}^{N-1} \frac{A}{2}[\cos(-\omega n) - \cos(3\omega n)] \\ &= \frac{A}{2}\sum_{n=0}^{N-1}\cos(-\omega n) - \frac{A}{2}\sum_{n=0}^{N-1}\cos(3\omega n) \\ &= 0\end{aligned}$$

We have used a trigonometric identity to show that this sequence is composed of the sum of two cosine waves, one at frequency $-\omega$ and the other at frequency 3ω. Since the cosine wave has the same symmetry above and below the horizontal axis as the sine, both of these components sum to 0 as well, indicating no energy at frequency 2ω. As long as we are summing up the values of a sinusoidal waveform *over any integral numbers of periods*, we get 0. The sinusoids which have an integral number of periods in a duration of N samples are just those corresponding to the harmonics of the frequency with a period of N samples.

So far we have not considered the phase of the sinusoid at frequency ω. We recall from part I that a sinusoid with arbitrary phase and amplitude can be represented as

$$A \sin(\omega n + \phi) = a \cos(\omega n) + b \sin(\omega n) \tag{1.6}$$

where

A is the amplitude,
ϕ is the phase angle,
a is equal to $A \sin(\phi)$, and
b is equal to $A \cos(\phi)$.

Both the amplitude and phase of a sinusoidal component at frequency ω can then be determined by using our multiply-and-sum procedure, first with a $\cos(\omega n)$ multiplier to calculate the a coefficient, then with $\sin(\omega n)$ to calculate the b coefficient. The amplitude and phase of the component are then given by the relations

$$A = \sqrt{a^2 + b^2} \quad \text{and} \quad \phi = \tan^{-1}\left(\frac{a}{b}\right) \tag{1.7}$$

For example, let the sequence $x(n)$ be defined as follows:

$$\begin{aligned} x(n) &= A \sin(\omega n + \phi_1) + B \sin(2\omega n + \phi_2) \\ &= a_1 \cos(\omega n) + b_1 \sin(\omega n) + a_2 \cos(2\omega n) + b_2 \sin(2\omega n) \end{aligned}$$

where $a_1 = A \sin(\phi_1)$, $b_1 = A \cos(\phi_1)$, $a_2 = B \sin(\phi_2)$, and $b_2 = B \cos(\phi_2)$. We "extract" a_1 via our multiply-and-sum procedure, using $\cos(\omega n)$ as a multiplier:

$$\begin{aligned} \sum_{n=0}^{N-1} x(n) \cos(\omega n) &= \sum_{n=0}^{N-1} [a_1 \cos^2(\omega n) + b_1 \sin(\omega n) \cos(\omega n) \\ &\quad + a_2 \cos(2\omega n) \cos(\omega n) + b_2 \sin(2\omega n) \cos(\omega n)] \\ &= a_1 \sum_{n=0}^{N-1} \left[\frac{1}{2} + \frac{1}{2}\cos(2\omega n)\right] \\ &= N\frac{a_1}{2} \end{aligned}$$

Similarly, if we were to use $\sin(\omega n)$ as a multiplier, we could extract b_1; $\cos(2\omega n)$ as a multiplier would extract a_2; and so on. Given both a_1 and b_1, we can then apply equation 1.7 to obtain A and ϕ_1, if desired. This is the principle of the DFT: the multiply-and-sum procedure is applied to determine the amplitudes and phases of each of the harmonics of the waveform.

How many harmonics might be present? According to the sampling theorem, we need at least two samples in each period in order to avoid aliasing; so if $N = 8$, and the sampling rate $R = 8000$ Hz, the only possible frequencies of which our periodic function $x(n)$ could be composed are the harmonics of $8000/8 = 1000$ Hz that "fit," i.e., which have frequencies less than or equal to one-half the sampling rate. However, the sampling theorem is perfectly admissive of negative frequencies, so the complete list of integral multiples of 1000 which have magnitudes ≤ 4000 Hz is

-4000 Hz (harmonic "-4")	
-3000	
-2000	
-1000	
0	(harmonic "0")
$+1000$	(harmonic $+1$, or the fundamental frequency)
$+2000$	
$+3000$	
$+4000$	

The function $x(n)$ is modeled as being composed only of sinusoids at these frequencies, i.e.:

$$x(n) = \sum_{k=-N/2}^{+N/2} a_k \cos(k\omega n) + b_k \sin(k\omega n) \tag{1.8}$$

where

$$\omega = 2\pi/N,$$

$$a_k = \frac{1}{N}\sum_{n=0}^{N-1} x(n)\cos(k\omega n),$$

and

$$b_k = \frac{1}{N}\sum_{n=0}^{N-1} x(n)\sin(k\omega n).$$

Here the $1/N$ factor is included to compensate for the fact that the latter two sums are scaled by N, as derived earlier.

Notice that if $x(n) = A\cos(\omega n)$, and we "extract the amplitude" of $x(n)$ at frequency ω with our multiply-and-sum procedure, we find that the answer is $A/2$:

$$\frac{1}{N}\sum_{n=0}^{N-1} A\cos(\omega n)\cos(\omega n) = \frac{A}{N}\sum_{n=0}^{N-1}\frac{1}{2}[\cos(\omega n - \omega n) + \cos(\omega n + \omega n)]$$
$$= \frac{A}{2}$$

But notice also that if we "extract the amplitude" of $x(n)$ at frequency $-\omega$, the answer is the same:

$$\frac{1}{N}\sum_{n=0}^{N-1} A\cos(\omega n)\cos(-\omega n)$$
$$= \frac{A}{N}\sum_{n=0}^{N-1}\frac{1}{2}\{\cos[\omega n - (-\omega n)] + \cos[\omega n + (-\omega n)]\}$$
$$= \frac{A}{2}$$

Before we proceed, let us consider just what is meant by a *negative frequency*. When we considered wagon wheels, we measured angles in a counterclockwise direction from the right horizontal axis as positive angles and clockwise as negative angles. Clearly the angle + 270° describes the same spoke position as −90°. Similarly, the radian frequency of rotation was positive for counterclockwise motion, and clockwise motion was described as a negative radian frequency. Does it matter whether we use a positive or negative description of a frequency? Not too surprisingly, the answer is yes and no.

Certain mathematical functions have the property known as *evenness*, which means that they are *left-right symmetrical* around 0:

$$f(x) = f(-x) \Rightarrow f(x) \text{ "even"} \tag{1.9}$$

(The symbol $\Rightarrow$ means "implies that.") Other functions have the property of *oddness*, which means that they are *left-right antisymmetrical* around 0:

$$-f(x) = f(-x) \Rightarrow f(x) \text{ "odd"} \tag{1.10}$$

Some functions are even, some are odd, many are neither, and only one is both (it is left as an exercise for the reader to discover the only function that is both even and odd). *Any* function, however, may be thought of as composed of the sum of an even part and an odd part, either (or both) of which may be 0. In other words, any arbitrary function f may be "broken apart." Thus,

$$f(x) = f_e(x) + f_o(x) \tag{1.11}$$

where $f_e(-x) = f_e(x)$ and $f_o(-x) = -f_o(x)$.

Here is the proof that this is so:

$$f(x) = f_e(x) + f_o(x) \Rightarrow f(-x) = f_e(-x) + f_o(-x)$$

But, by the *definitions* of f_e and f_o,

$$f(-x) = f_e(x) - f_o(x)$$

Therefore we can solve for either f_e or f_o by adding (or subtracting) $f(x)$ and $f(-x)$:

$$f(x) + f(-x) = [f_e(x) + f_o(x)] + [f_e(x) - f_o(x)]$$
$$f_e(x) = \frac{1}{2}[f(x) + f(-x)]$$

Similarly,

$$f_o(x) = \frac{1}{2}[f(x) - f(-x)]$$

Clearly,

$$f_e(-x) = \frac{1}{2}[f(-x) + f(x)] = f_e(x)$$

and

$$f_o(-x) = \frac{1}{2}[f(-x) - f(x)] = -f_o(x)$$

Finally,

$$f_e(x) + f_o(x) = \frac{1}{2}[f(x) + f(-x)] + \frac{1}{2}[f(x) - f(-x)]$$
$$= f(x)$$

We have proved that $f(x)$ can be decomposed into a sum of even and odd parts without saying *anything* else at all about $f(x)$, so it is true for all functions.

Getting back to the question of negative frequencies, it is clear that if a function is purely even, such as the cosine function, the sign of the frequency does not matter at all, since

$$\cos(\omega n) = \cos(-\omega n) \tag{1.12}$$

But for a purely odd function, such as sine, it represents a negation of amplitude or, equivalently, a 180° phase shift:

$$\sin(-\omega n) = -\sin(\omega n) = \sin(\omega n \pm \pi) \tag{1.13}$$

Because $A \cos(\omega n)$ is an even function, it is generally meaningless to distinguish between positive and negative frequency cosine *waveforms*. But the *spectrum* of a cosine wave may be considered to contain *both* positive and negative frequency components, both of which have a positive amplitude equal to $A/2$. For a sine waveform $A \sin(\omega n)$, we obtain also positive and negative frequency components, but of opposite amplitude due to the oddness of the sine function: if the positive frequency component has a positive amplitude, then the corresponding negative frequency component will have a negative amplitude, and vice versa. The complete DFT yields the amplitudes and phases of *both* the positive and negative harmonics. The amplitude is split in half at corresponding positive and negative frequencies, with the signs of the amplitude of the odd parts being opposite. This explains the fact that only one-half of the amplitude is measured if we consider only positive frequencies.

One more aspect of cosine and sine waveforms should be mentioned before we proceed to define the DFT. A component at exactly one-half the sampling rate, i.e., with only two samples per period, can only be purely even, since the samples occur at angles 0 and π, and both sin(0) and sin(π) are equal to 0. Thus the "b_k" coefficients of equation 1.8 will always be 0 whenever $k = \pm N/2$. Similarly, at 0 frequency (also called *dc*, for *direct current*, by some engineers and others who talk to these engineers), the "b_0" coefficient is always 0, again since $\sin(0) = 0$.

We are now ready to define the DFT of a sequence $x(n)$. As mentioned before, we *model* $x(n)$ as one N-sample period of a periodic waveform. The DFT will then yield the unique spectrum of $x(n)$ in terms of the amplitudes and phases of sinusoidal components, each with periods harmonically related to N, the number of samples in the transformed signal. While the FFT algorithm generally requires that N be a power of 2, the DFT (which yields exactly the same result, albeit with less computational efficiency) places no restriction on N except, of course, that it be greater than two samples.

Following the usual practice in the literature, we will define the DFT in terms of the complex exponential, which allows us to represent both sine and cosine functions at once. A typical definition for the DFT is then

$$\begin{aligned} \mathrm{DFT}[x(n)] &= X(k) \\ &= \sum_{n=0}^{N-1} x(n)e^{-j\omega nk} \qquad 0 \le k \le N-1 \end{aligned} \tag{1.14}$$

where $\omega = 2\pi/N$ and $e^{-j\omega nk} = \cos(\omega nk) - j \sin(\omega nk)$. The *inverse* DFT is then defined as

$$\begin{aligned} \mathrm{DFT}^{-1}[X(k)] &= x(n) \\ &= \frac{1}{N} \sum_{k=0}^{N-1} X(k)e^{+j\omega kn} \qquad 0 \le n \le N-1 \end{aligned} \tag{1.15}$$

Thus, if $X(k)$ is the DFT of $x(n)$, then $x(n)$ is the inverse DFT of $X(k)$. The mathematical oddness of the imaginary part of the complex exponential necessitates the use of the minus sign in the exponent of the DFT, while the inverse DFT has a positive exponent. Also, since the multiply-and-sum procedure produces values which are scaled by a factor of N, the $1/N$ factor appears in the inverse DFT in order to make the statement $\mathrm{DFT}^{-1}[\mathrm{DFT}[x(n)]] = x(n)$ exactly true. The values of $X(k)$ (the spectrum) are complex, with the real parts corresponding to the a (cosine, even part) coefficients and the imaginary parts corresponding to the b (sine, odd part) coefficients of the spectral components. If we denote $X(k)$ as $a_k + jb_k$, then

$$|X(k)| = \sqrt{a_k^2 + b_k^2} \tag{1.16}$$

is called the *magnitude, modulus,* or *amplitude* of $X(k)$, which is the same as the amplitude of the corresponding spectral component (except for being scaled by N).

$$\arg[X(k)] = \mathrm{pha}[X(k)] = \measuredangle X(k) = \tan^{-1}\left(\frac{b_k}{a_k}\right) \tag{1.17}$$

is called the *argument, phase,* or *angle* of $X(k)$, which is equal to the phase angle of the corresponding spectral component.

The next question is: How do the values of the index k correspond to frequency? In order to understand this correspondence, it is instructive to take the DFT of a specific sequence of numbers and see exactly what we get.

Let us define $x(n)$ as eight samples of a periodic waveform with a period

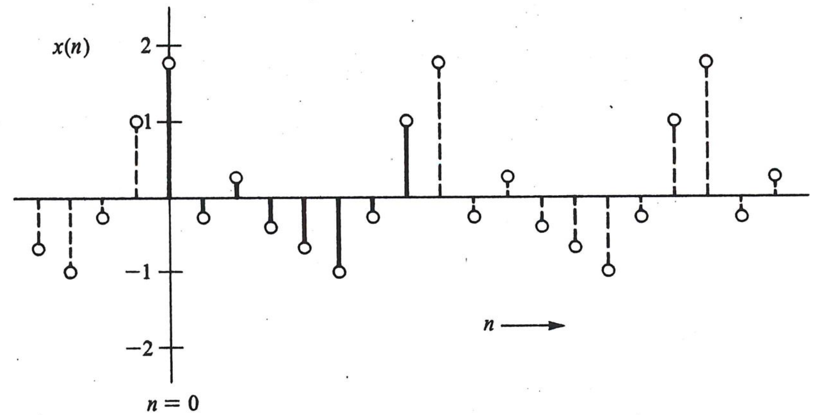

Fig. 1.12. Graph of $x(n)$ as described in table 1.1. The function $x(n)$ is periodic with a period of $N = 8$ samples. Only the period from $n = 0$ to $n = 7$ is considered (solid lines), but presumably the function repeats itself before and after this period (dashed lines).

of eight samples, sampled at $R = 8000$ Hz. The signal $x(n)$ must be composed of harmonics of $8000/8 = 1000$ Hz, so

$$x(n) = \sum_{m=0}^{4} A_m \cos(m\omega n + \phi_m)$$

with $\omega = 2\pi/8 = \pi/4$.

Values for both the amplitudes and the phase angles for each of the five components are given in columns (3) and (4) of table 1.1. Table 1.2 shows actual numerical values for the five components of $x(n)$. The sum of these five components, i.e., the numerical value of the samples of $x(n)$ itself, is shown at the bottom of this table. The so-called analytic (cosine and sine) form of the components of $x(n)$ is shown in table 1.3. By applying the trigonometric identity

$$\begin{aligned}\cos(m\omega n + \phi_m) &= \cos(m\omega n)\cos(\phi_m) - \sin(m\omega n)\sin(\phi_m) \\ &= a_m \cos(m\omega n) - b_m \sin(m\omega n)\end{aligned}$$

with a_m and b_m defined as in columns (5) and (6) of table 1.1, we can see that the form given in table 1.3 yields the same numbers for $x(n)$ as table 1.2.

Figure 1.12 is a graph of $x(n)$. Only those values from $n = 0$ to $n = 7$ are considered (one period); these are shown as solid lines on the graph.

TABLE 1.1 Coefficients for two representations of the sampled waveform shown in figure 1.12

		$\sum_{m=0}^{4} A_m \cos(m\omega n + \phi_m)$		$\sum_{m=0}^{4} [a_m \cos(m\omega n) - b_m \sin(m\omega n)]$	
m (1)	F_m (Corresponding frequency, in hertz) (2)	A_m (Amplitude) (3)	ϕ_m (Phase, in radians) (4)	a_m (5)	b_m (6)
0	0	0.100	0	0.100	0
1	1000	1.000	0	1.000	0
2	2000	0.500	$\pi/3$	0.250	0.433
3	3000	0.333	$\pi/4$	0.236	0.236
4	4000	0.250	$\pi/5$	0.202	0.147

TABLE 1.2 Numerical values of the components of $x(n)$ as described in table 1.1 and their sum, $x(n)$ itself

m \ n	0	1	2	3	4	5	6	7	Component ($\omega = 2\pi/8 = \pi/4$)
0	0.100	0.100	0.100	0.100	0.100	0.100	0.100	0.100	$0.100 \cos(0)$
1	1.000	0.707	0	−0.707	−1.000	−0.707	0	0.707	$1.000 \cos(\frac{n\pi}{4} + 0)$
2	0.250	−0.433	−0.250	0.433	0.250	−0.433	−0.250	0.433	$0.500 \cos(\frac{n\pi}{2} + \frac{\pi}{3})$
3	0.236	−0.333	0.236	0	−0.236	0.333	−0.236	0	$0.333 \cos(\frac{3n\pi}{4} + \frac{\pi}{4})$
4	0.202	−0.202	0.202	−0.202	0.202	−0.202	0.202	−0.202	$0.250 \cos(n\pi + \frac{\pi}{5})$
$\sum_{m=0}^{4}$	1.788	−0.161	0.288	−0.376	−0.684	−0.909	−0.184	1.038	$x(n)$

Note: $x(n)$ is given by $\sum_{m=0}^{4} A_m \cos(m\omega n + \phi_m)$ for $0 \le n \le 7$.

TABLE 1.3 An alternative form of table 1.2 giving the components of $x(n)$ as described in table 1.1

m \ n	0	1	2	3	4	5	6	7	Component ($\omega = 2\pi/8 = \pi/4$)
$a_m \cos(m\omega n)$									
0	0.100	0.100	0.100	0.100	0.100	0.100	0.100	0.100	$0.100 \cos(0)$
1	1.000	0.707	0	−0.707	−1.000	−0.707	0	0.707	$1.000 \cos(\frac{n\pi}{4})$
2	0.250	0	−0.250	0	0.250	0	−0.250	0	$0.250 \cos(\frac{n\pi}{2})$
3	0.236	−0.167	0	0.167	−0.236	0.167	0	−0.167	$0.236 \cos(\frac{3n\pi}{4})$
4	0.202	−0.202	0.202	−0.202	0.202	−0.202	0.202	−0.202	$0.202 \cos(n\pi)$
$b_m \sin(m\omega n)$									
0	0	0	0	0	0	0	0	0	$0 \sin(0)$
1	0	0	0	0	0	0	0	0	$0 \sin(\frac{n\pi}{4})$
2	0	0.433	0	−0.433	0	0.433	0	−0.433	$0.433 \sin(\frac{n\pi}{2})$
3	0	0.167	−0.236	0.167	0	−0.167	0.236	−0.167	$0.236 \sin(\frac{3n\pi}{4})$
4	0	0	0	0	0	0	0	0	$0.147 \sin(n\pi)$

Note: In this representation, $x(n)$ is given by $\sum_{m=0}^{4} [a_m \cos(m\omega n) - b_m \sin(m\omega n)]$ for $0 \le n \le 7$.

TABLE 1.4 Calculation of the DFT

k \ n	0	1	2	3	4	5	6	7
$\cos(2\pi nk/N)$								
0	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
1	1.000	0.707	0	−0.707	−1.000	−0.707	0	0.707
2	1.000	0	−1.000	0	1.000	0	−1.000	0
3	1.000	−0.707	0	0.707	−1.000	0.707	0	−0.707
4	1.000	−1.000	1.000	−1.000	1.000	−1.000	1.000	−1.000
5	1.000	−0.707	0	0.707	−1.000	0.707	0	−0.707
6	1.000	0	−1.000	0	1.000	0	−1.000	0
7	1.000	0.707	0	−0.707	−1.000	−0.707	0	0.707
$-j\sin(2\pi nk/N)$†								
0	0	0	0	0	0	0	0	0
1	0	−0.707	−1.000	−0.707	0	0.707	1.000	0.707
2	0	−1.000	0	1.000	0	−1.000	0	1.000
3	0	−0.707	1.000	−0.707	0	0.707	−1.000	0.707
4	0	0	0	0	0	0	0	0
5	0	0.707	−1.000	0.707	0	−0.707	1.000	−0.707
6	0	1.000	0	−1.000	0	1.000	0	−1.000
7	0	0.707	1.000	0.707	0	−0.707	−1.000	−0.707

Note: This table gives values of the $e^{-j\omega nk}$ multiplier for $\omega = 2\pi/N$, $N = 8$ ($e^{-j\omega nk} = \cos(\omega nk) - j\sin(\omega nk)$).
† All values below are multiplied by j.

The signal $x(n)$ is presumably an eight-sample sequence extracted from a longer sequence with a period of eight samples (other values of this longer sequence are shown on dashed lines). A glance at this figure confirms that the spectral structure of $x(n)$ is not very apparent from observation of its waveform, yet applying the DFT to $x(n)$ will indeed tell us exactly the components of $x(n)$.

In order to calculate the DFT it is useful to make a table of values for $e^{-j\omega nk}$, such as the one shown in table 1.4. We start with $k = 0$:

$$\begin{aligned} X(0) &= \sum_{n=0}^{7} x(n)\, e^{-j0} \\ &= \sum_{n=0}^{7} x(n)\cos(0) - j\sum_{n=0}^{7} x(n)\sin(0) \\ &= \sum_{n=0}^{7} x(n) \\ &= 1.788 + (-0.161) + 0.288 + (-0.376) + (-0.684) \\ &\quad + (-0.909) + (-0.184) + 1.038 \\ &= 0.8 \end{aligned}$$

The real part of the answer (0.8) is supposed to be N times one of the a coefficients in table 1.1, and indeed it is $N \times a_0$. The imaginary part of the answer (0) corresponds to b_0. So $k = 0$ apparently refers to the 0 frequency, direct current, 0^{th} harmonic component of $x(n)$. For $k = 1$:

$$\begin{aligned} X(1) &= \sum_{n=0}^{7} x(n)\, e^{-j\omega n} \\ &= \sum_{n=0}^{7} x(n)\cos(2\pi n/N) - j\sum_{n=0}^{7} x(n)\sin(2\pi n/N) \\ &= [(1.788)(1) + (-0.161)(0.707) + \ldots \text{ etc.}] \\ &\quad + j[(1.788)(0) + (-0.161)(-0.707) + \ldots \text{ etc.}] \\ &= 4 + j0 \end{aligned}$$

which is just $N/2$ times $(a_1 + jb_1)$, the coefficients for the 1^{st} harmonic, or fundamental frequency. Table 1.5 shows the values of $X(k)$ for all k from 0 to 7. Several remarks are in order about $X(k)$. We see that k corresponds to the harmonic number for $k = 0$, 1, 2, and 3. But the dc and half sampling rate components are scaled by N, while the rest are scaled by $N/2$. For $k =$ 5, 6, and 7, we see that the coefficients are the same as $k = 3$, 2, and 1, respectively, except that the sign of the imaginary part i[illegible]ersed. Since

TABLE 1.5 Discrete Fourier transform (DFT) of $x(n)$ as described in table 1.1

k	$X(k)$	Corresponding spectral coefficients	Corresponding frequency (in Hz)
0	$0.8 + j0$	$N(a_0 + jb_0)$	$\pm\ 0$
1	$4 + j0$	$\frac{N}{2}(a_1 + jb_1)$	$+\ 1000$
2	$1 + j1.732$	$\frac{N}{2}(a_2 + jb_2)$	$+\ 2000$
3	$0.944 + j0.944$	$\frac{N}{2}(a_3 + jb_3)$	$+\ 3000$
4	$1.616 + j0$	$N(a_4 + j0)$	$\pm\ 4000$
5	$0.944 - j0.944$	$\frac{N}{2}(a_3 - jb_3)$	$-\ 3000$
6	$1 - j1.732$	$\frac{N}{2}(a_2 - jb_2)$	$-\ 2000$
7	$4 - j0$	$\frac{N}{2}(a_1 - jb_1)$	$-\ 1000$

Note: $X(k) = \text{DFT}[x(n)] = \sum_{n=0}^{N-1} x(n)e^{-j\omega nk}$ for $0 \le k \le N-1$.
Here, $n = 8$ and $R = 8000$ Hz.

the imaginary part corresponds to the sine function component, and since sine is an odd function, it is clear that these are the spectral values of the negative frequencies. That is, $k = 5$ corresponds to the -3^{rd} harmonic, $k = 6$ to the -2^{nd} harmonic, and $k = 7$ to the -1^{st} harmonic. What about $k = 4$? As mentioned above, the sampling process cannot represent an amplitude for a sine component at half the sampling rate, so the imaginary part is 0, even though we used a nonzero value for b_4 (b_4, in fact, has been ignored by the sampling process). Also, the scale factor is N for $k = 4$ instead of $N/2$. This is because both the dc and half sampling rate components perforce have zero imaginary parts; hence it is impossible to split them into distinguishable positive and negative frequencies as can be done with the other components. This says that when sampling at 8000 Hz, we cannot distinguish components at $+0$ Hz from components at -0 Hz, nor can we distinguish components at $+4000$ Hz from components at -4000 Hz. This accounts for the ambiguity in equation 1.1 when $F = \pm R/2$. It also accounts for the different scale factors for $X(0)$ and $X(N/2)$.

In order to completely check our transform definition we should apply the inverse DFT to $X(k)$ and see if $x(n)$ pops out again. This procedure will be left to the reader as a valuable exercise. Let us proceed by examining some of the properties of the DFT and derive some important transforms that will be useful later.

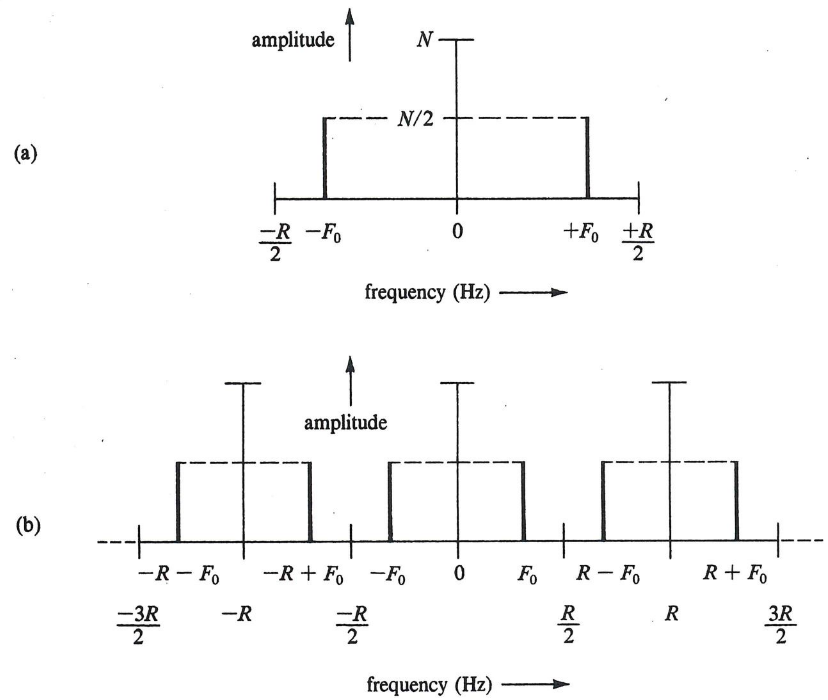

Fig. 1.13. Spectrum of cosine function at frequency $|F_0| < R/2$. In (a) we see just the two principal components, one at $+F_0$ Hz and the other at $-F_0$ Hz, both with an amplitude of $N/2$. In (b) we see three periods of the periodic spectrum of the same function centered around 0 Hz.

First of all, it is important to note that we have defined the DFT in such a way that only the principal values of $X(k)$ are calculated. A more general definition is

$$X(k) = \sum_{n=0}^{N-1} x(n)\, e^{-j\omega nk} \qquad -\infty < k < \infty \tag{1.18}$$

This definition shows explicitly that the spectrum obtained from the DFT is a *periodic function of frequency*. In other words, the principal values of the N-sample DFT of $\cos(\omega n)$ are just what is shown in figure 1.13a, but this is really only a part of the full picture, shown in figure 1.13b. This spectral periodicity is due to the periodicity of the complex exponential function itself, and theoretically it extends over all frequencies for all digital signals. It is easy to see from the graph of the full periodic spectrum of a sampled signal how no frequencies greater in magnitude than $R/2$ can exist. If F_0 in figure 1.13 were instead the frequency $R - F_0$, which is greater than $R/2$, the plots would look exactly the same.

If we (properly) interpret the k index of $X(k) = \mathrm{DFT}[x(n)]$ as negative frequencies for $k > N/2$, and normalize amplitudes that may be scaled by N, we can begin to construct a useful convention for spectral plots. Only the smooth curve need be given (rather than a sequence of dots on the heads of sticks) for functions like cosine or sine—it being understood that this curve is sampled at the sampling rate. When convenient or appropriate, some functions such as the impulse will still be shown with the dot-stick notation.

Transform pairs for some common functions are shown in figure 1.14. It should be remembered that the DFT is really a two-way process: each member of a transform pair transforms into the other member. We see, for example, that the unit sample function (fig. 1.14e) transforms into a constant spectrum (fig. 1.14f). Or we can read this pair in the opposite direction: a constant-valued sampled function has energy only at 0 Hz. The scale markings in this figure correspond to unit values of time or frequency on the horizontal axes and to unit values of amplitude on the vertical axes. It should be recalled that if one of the members of a pair is interpreted as a time function, its amplitude is scaled by 1 and its corresponding spectral amplitudes will be scaled by N.

CONVOLUTION

If $X_1(k) = \mathrm{DFT}[x_1(n)]$ and $X_2(k) = \mathrm{DFT}[x_2(n)]$, then an important property of the DFT known as *linearity* assures us that the following is always true:

$$\mathrm{DFT}[c_1x_1(n) + c_2x_2(n)] = c_1X_1(k) + c_2X_2(k) \qquad (1.19)$$

where c_1 and c_2 are arbitrary constants. Does the same hold true if we multiply $x_1(n)$ and $x_2(n)$? Unfortunately not, since the spectrum of the product function $x_1(n)x_2(n)$ is *not* $X_1(k)X_2(k)$. There is a method for obtaining the spectrum of the product of two different functions known as *convolution*, which we will treat first in a qualitative way. If we *convolve* a function $x(n)$ with $u(n)$, the impulse function (fig. 1.14e), then the result is just the same as $x(n)$. In other words,

$$x(n) * u(n) = x(n) \qquad (1.20)$$

where the asterisk denotes the convolution operation. Note that this is certainly different from multiplying $x(n)$ by $u(n)$, which would result in setting all values of $x(n)$ to 0, except for $x(0)$, which would remain unchanged. The

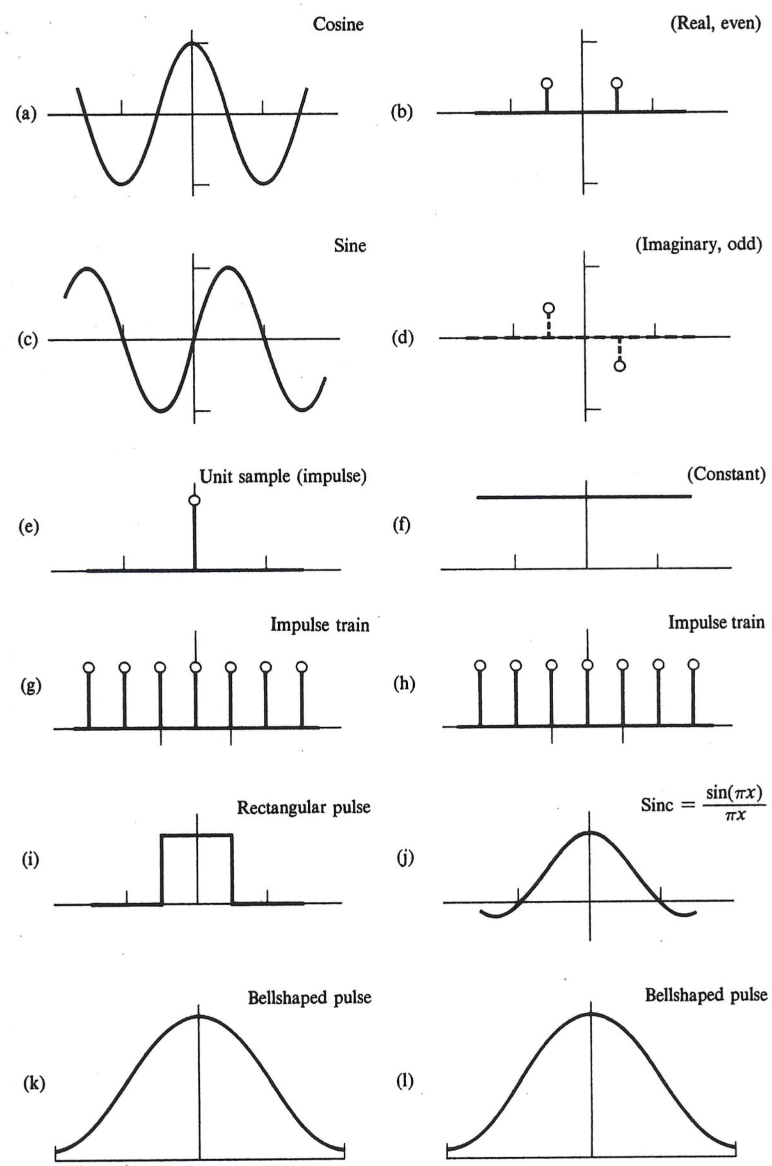

Fig. 1.14. Some transform pairs. Each member of a pair transforms into the other. The tick marks on the horizontal axes represent unit values of time or frequency; on the vertical axes, they represent unit values of amplitude. (After *The Fourier Transform and Its Applications* by Ron Bracewell. Copyright © 1965 by McGraw-Hill, Inc. Used with permission of McGraw-Hill Book Company.)