An Introduction to the Mathematics of Digital Signal Processing

Part II: Sampling, Transforms, and Digital Filtering

F. R. Moore
Bell Laboratories
Murray Hill, New Jersey 07974

INTRODUCTION

In Part I of this tutorial (*Computer Music Journal*, Vol. 2, No. 1), 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 often are 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, transforms, and an introduction to the concepts of digital filtering.

Before we delve into 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 one replete 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 travelled 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 first before we can progress to an understanding of how to rephrase these statements in musical terms. Therefore we cannot initially ask questions such as "How can we make an oboe sound like a clarinet?" directly of digital signal processing, since the information is not couched in these terms for historical reasons. We can, however, keep such questions in mind as we study, on the assumption that an accurate understanding of the terminology of the field will allow such questions to be rephrased into tractable problems. Often, the answer will be that the question asks for unknown or ill-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 often 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 not programming, not acoustics, and 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 AND QUANTIZATION

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 on the screen at a sufficient rate to ensure that no important detail of the motion is lost. Clearly, if the motion were sampled more slowly, say, at 5 times per seond, the motion would appear jerky and discontinuous, as it does under now-familiar strobe lighting 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, and 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, as usual, is: of course not, as anyone who has ever watched a wagon wheel in a western movie can verify. As the wagon starts out from a standing position, the wheels appear to turn slowly forward; then as the speed increases, the wheels first appear to go faster, then begin to slow down and go backwards, then to stop, then to go forwards again, etc. Of course they never appear to stop completely when the wagon is moving since 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? Clearly, the wagon could go faster before its wheels started to appear slowing down or going backwards, 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. (Obviously, 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 Figure 1). Let us arbitrarily say that when the spoke points to the right that 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, etc. If the wheel completes exactly one full counterclockwise revolution per second we can describe its rotational velocity by saying

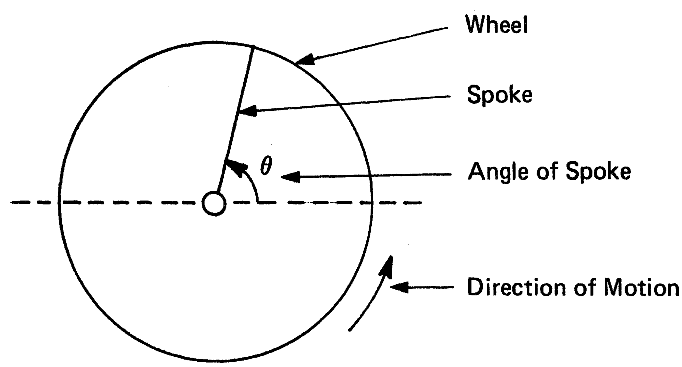


Figure 1. A turning, one-spoke wagon wheel.

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 will 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 smoothly turns from the starting position zero at a rate of $F = 1$ Hz., successive frames of the movie will be taken at $\theta = 0, \theta = (1/24) * 2\pi F = \pi/12, \theta = \pi/6$, etc. Since the wheel turns only $1/24^{th}$ 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 2; 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 the filmed image of it 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 ***backwards*** at a rate of -6 Hz., equivalent to running the film in Fig. 2 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 backwards motion (-1 Hertz), 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 again the same sequence of events. This is

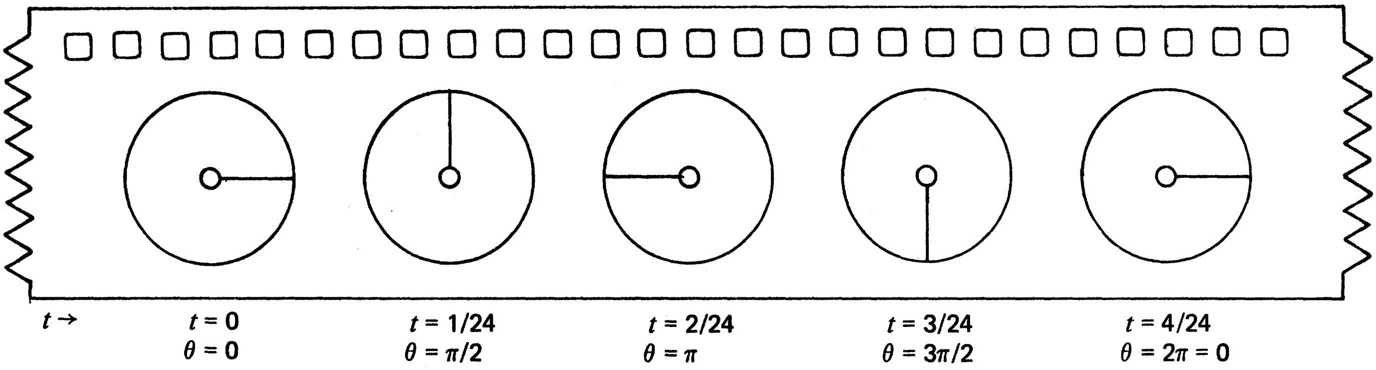


Figure 2. Movie of a one-spoke wagon wheel turning (apparently) at $f = 6$ Hz., shot at 24 frames per second.

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

$$F_a = F - \frac{(k+1)R}{2}, \quad \frac{kR}{2} \leqslant F \leqslant \frac{(k+2)R}{2} \qquad (1)$$

where

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

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

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

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

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

Therefore the apparent frequency is

$$F_a = 28 - \frac{2 \cdot 24}{2} = +4 \text{ Hz.}$$

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 zero, which is true only if $k = -1$. Then

$$F_a = F - 0, \quad -\frac{R}{2} \leqslant F \leqslant +\frac{R}{2},$$

or simply, $F_a = F$ only if $|F| \leqslant R/2$ ($|F|$ is the "absolute value" or "magnitude" of F without regard to its plus or minus sign). If $|F| > R/2$, then $F_a \neq F$, 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) 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) 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.

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 these waveforms taken at brief, regular intervals in time. Any musical waveform can be modelled 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 (Equation (1)) then assures us of the accuracy of our rendition of the waveform if it is *bandlimited* 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 Figure 3), since these are generally impossible to remove from the digital signal due 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 bandlimited analog signal varies between, say, +10 and −10 volts, and $B = 10$ bits, the entire 20 volt (peak-to-peak) range

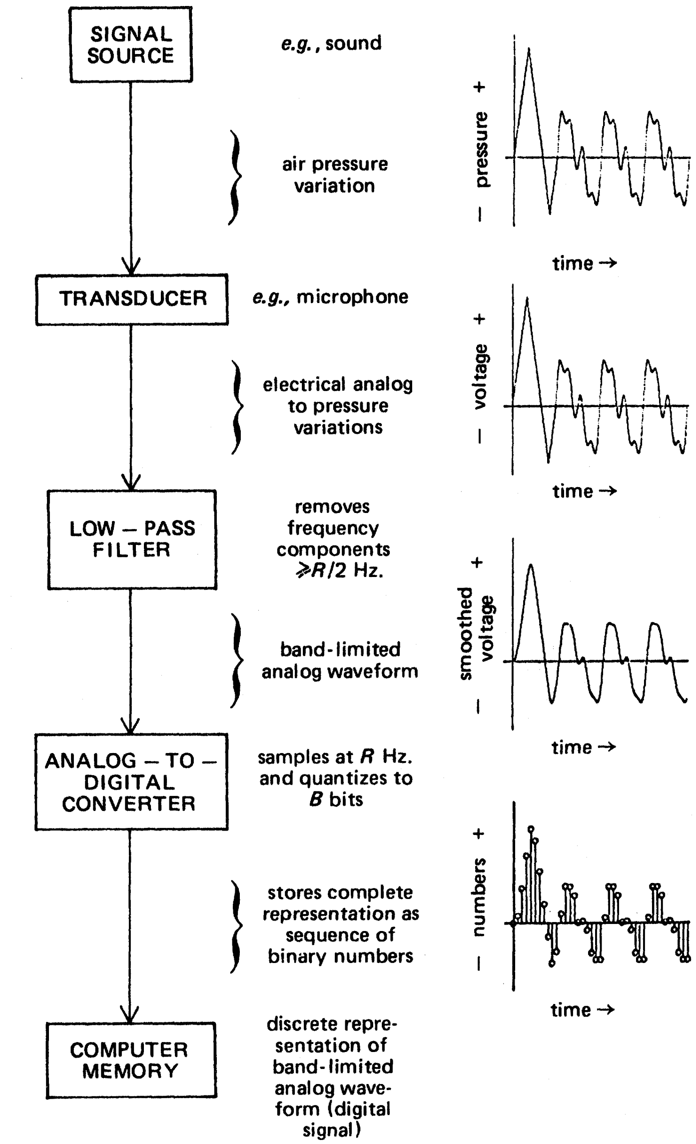


Figure 3. Steps by which a continuous signal is converted into a digital signal for subsequent computer processing.

may be represented to an accuracy of $20/2^{10} \approx .02$ 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 ±.05%. Such inaccuracies are often 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):

$$\text{SQNR in dB} = 20 \log_{10} \frac{\text{signal amplitude}}{\text{noise amplitude}} \qquad (2)$$

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

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

The reader may wish to verify that under these conditions and assumptions, the SQNR of a *B*-bit ADC is approximately 6*B* 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. ADC's are available with 8 to 16 bits of resolution, and while the issue has not quite been resolved, it seems that 12-bit ADC's 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 most computer music programs so far have not dealt with this possibility.

Producing sounds with a computer is just the reverse of the process diagrammed in Figure 3: 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

$$x(n), \quad N_1 \leqslant n \leqslant N_2, \quad n \in \mathbf{I}$$

or

$$x(nT), \quad N_1 \leqslant n \leqslant N_2, \quad n \in \mathbf{I}$$

Both of these notations have the same meaning except that in the second one the sampling period, T, is shown explicitly. Since it is easy enough to remember that the relation 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

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

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

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

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 one only if its argument is zero, and it has a zero value otherwise, *i.e.*

$$u(n) = \begin{cases} 1 & n = 0 \\ 0 & n \neq 0 \end{cases} \qquad (3)$$

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:

$$u(n - n_0) = \begin{cases} 1 & n = n_0 \\ 0 & n \neq n_0 \end{cases} \qquad (4)$$

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

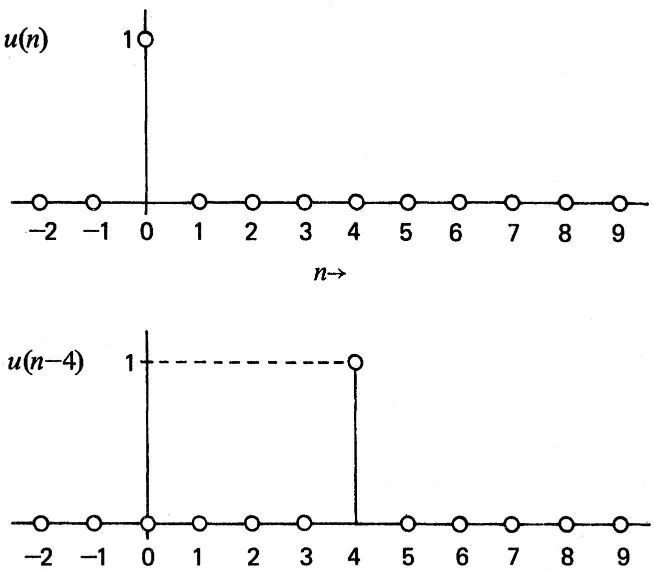


Figure 4. The unit sample (digital impulse) function $u(n)$ and the delayed unit sample function $u(n-n_0)$ for the case $n_0 = 4$.

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:

$$e^{j\omega n} = \cos(\omega n) + j \sin(\omega n) \qquad (5)$$

where $j^2 = -1$.

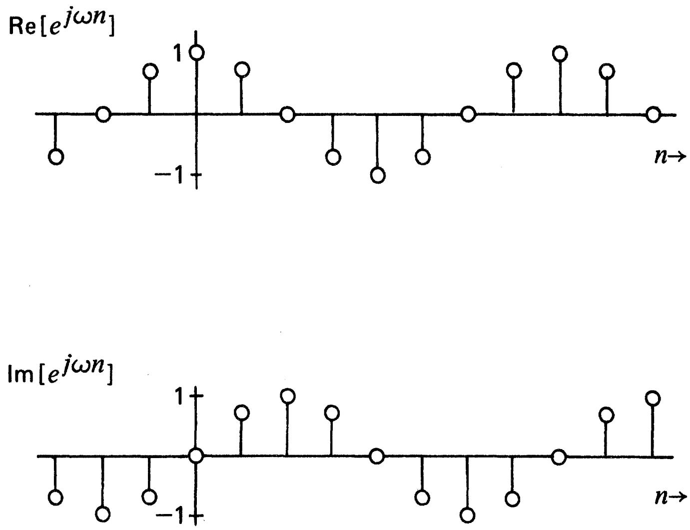


Figure 5. The complex exponential function $e^{j\omega n}$, shown as graphs of its real and imaginary parts, $\omega = 2\pi/N$, $N = 8$.

Figure 5 shows *two* graphs of this function, one of its real part, and the other of its imaginary part. From looking at Figure 5, we cannot tell the frequency of the sinusoidal waveforms depicted. But we *can* see that there appear to be 8 samples in each period of the sinusoidal waves, and deduce that the frequency of these waveforms must therefore be 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 due to 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 analyze them into their constituent parts. Obviously, a complete description of *all* of 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 Theramins, or electronic organs, the tone color does not remain the same when different notes are played, due to varying string or tube lengths, lip tension, and so on. Since all of 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 deepest problems 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 that 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 make-up 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, steam-like 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).

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 Helmholtz (see References) 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 problem of producing "connected notes" (*i.e.,* to problems of musical *phrasing*). Besides, the "steady-state" of any real tone isn't really "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 so-called *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, 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 non-unique ways of decomposing a waveform, just as 340 could be non-uniquely decomposed into non-unique 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, which we won't consider 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 non-zero 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 will 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 $0 \leqslant n \leqslant N-1$. For $N = 8$ we would have the sequence

$$x(n) = \left(0, \frac{A}{\sqrt{2}} \approx .707A,\ A, \frac{A}{\sqrt{2}},\ 0, -\frac{A}{\sqrt{2}}, -A, -\frac{A}{\sqrt{2}} \right)$$

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

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

$$4A = N\frac{A}{2}$$

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 intergral 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 non-negative.

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 zero. 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:

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

$$= \frac{A}{2} \sum_{n=0}^{N-1} \cos(-\omega n) - \frac{A}{2} \sum_{n=0}^{N-1} \cos(3\omega n) = 0$$

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 zero 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 zero. 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 haven't considered the phase of the sinusoid at frequency ω. We recall from Part I of this tutorial that a sinusoid with arbitrary phase and amplitude can be represented as

$$A \sin(\omega n + \phi) = a \cos(\omega n) + b \sin(\omega n) \tag{7}$$

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

$$A = \sqrt{a^2 + b^2} \qquad \text{and} \qquad \phi = \tan^{-1} \frac{b}{a} \tag{8}$$

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

$$x(n) = A \sin(\omega n + \phi_1) + B \sin(2\omega n + \phi_2)$$

$$= a_1 \cos(\omega n) + b_1 \sin(\omega n) + a_2 \cos(2\omega n) + b_2 \sin(2\omega n)$$

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:

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

$$= \sum_{n=0}^{N-1} [a_1 \cos^2(\omega n) + b_1 \sin(\omega n)\cos(\omega n)$$

$$+ a_2 \cos(2\omega n)\cos(\omega n) + b_2 \sin(2\omega n)\cos(\omega n)]$$

$$= a_1 \sum_{n=0}^{N-1} [½ + ½\cos(2\omega n)] = N\frac{a_1}{2}$$

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 Equations (8) 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 2 samples in each period in order to avoid aliasing; so if $N = 8$, and the sampling rate R is = 8000 Hz., the only possible frequencies of which our periodic function $x(n)$ could be composed are the harmonics of 8000/ 8 = 1000 Hz., which "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	

$x(n)$ is modelled as being composed only of sinusoids at these frequencies, *i.e.*,

$$x(n) = \sum_{k=-N/2}^{+N/2} a_k \cos(k\omega n) + b_k \sin(k\omega n) \qquad (9)$$

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

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

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

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 $A/2$:

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

$$= \frac{A}{N}\sum_{n=0}^{N-1} ½[\cos(\omega n - \omega n) + \cos(\omega n + \omega n)] = \frac{A}{2}$$

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

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

$$= \frac{A}{N}\sum_{n=0}^{N-1} ½\left\{\cos[\omega n - (-\omega n)] + \cos[\omega n + (-\omega n)]\right\}$$

$$= \frac{A}{2}$$

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 zero:

$$f(x) = f(-x) \Rightarrow f(x) \text{ "even"} \qquad (10)$$

(the symbol ⇒ means "implies that"). Other functions have the property of *oddness*, which means that they are *left-right antisymmetrical* around zero:

$$-f(x) = f(-x) \Rightarrow f(x) \text{ "odd."} \qquad (11)$$

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 may be thought of, however, as composed of the sum of an even part and an odd part, either (or both) of which may be zero. In other words, any arbitrary function f may be "broken apart." Thus,

$$f(x) = f_e(x) + f_0(x) \qquad (12)$$

where $f_e(-x) = f_e(x)$ and $f_0(-x) = -f_0(x)$.

Here's the proof that this is so:

$$f(x) = f_e(x) + f_0(x) \Rightarrow f(-x) = f_e(-x) + f_0(-x)$$

But, by the *definitions* of f_e and f_0,

$$f(-x) = f_e(x) - f_0(x)$$

Therefore we can solve for either f_e or f_0 by adding (or subtracting) $f(x)$ and $f(-x)$:

$$f(x) + f(-x) = [f_e(x) + f_0(x)] + [f_e(x) - f_0(x)]$$

$$f_e(x) = ½ [f(x) + f(-x)]$$

Similarly,

$$f_0(x) = ½ [f(x) - f(-x)]$$

Clearly,

$$f_e(-x) = ½ [f(-x) + f(x)] = f_e(x) \quad \text{and,}$$

$$f_0(-x) = ½ [f(-x) - f(x)] = -f_0(x).$$

Finally,

$$f_e(x) + f_0(x) = ½ [f(x) + f(-x)] + ½ [f(x) - f(-x)] = f(x).$$

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 doesn't matter at all since

$$\cos(\omega n) = \cos(-\omega n) \quad (13)$$

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

$$\sin(-\omega n) = -\sin(\omega n) = \sin(\omega n \pm \pi) \quad (14)$$

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 2 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 (9) will always be zero whenever $k = \pm N/2$. Similarly, at zero frequency (also called "D.C." for *direct current* by some engineers and others who talk to these engineers), the "b_o" coefficient is always zero, 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 2 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

$$\text{DFT}[x(n)] = X(k)$$

$$= \sum_{n=0}^{N-1} x(n) e^{-jwnk} \qquad 0 \leqslant k \leqslant N-1 \quad (15)$$

where $\omega = 2\pi/N$ and $e^{-jwnk} = \cos(\omega nk) - j \sin(\omega nk)$.

The *inverse* DFT is then defined as

$$\text{DFT}^{-1}[X(k)] = x(n)$$

$$= \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{+jwkn} \qquad 0 \leqslant n \leqslant N-1 \quad (16)$$

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 $\text{DFT}^{-1}[\text{DFT}[x(n)]] = x(n)$ 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 $a_k + j\, b_k$, then

$$|X(k)| = \sqrt{a_k^2 + b_k^2} \quad (17)$$

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).

$$\arg[X(k)] = \text{pha}[X(k)] = \measuredangle X(k) = \tan^{-1}\frac{b_k}{a_k} \qquad (18)$$

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 8 samples of a periodic waveform with a period of 8 samples, sampled at R = 8000 Hz. $x(n)$ 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 $\omega = 2\pi/8 = \pi/4$.

m	F_m (Corresponding frequency in Hz.)	A_m (amplitude)	ϕ_m (phase in radians)	a_m ($A_m \cos\phi_m$)	b_m ($A_m \sin\phi_m$)
0	0	1/10	0	.100	0
1	1000	1	0	1.000	0
2	2000	1/2	$\pi/3$	.250	.433
3	3000	1/3	$\pi/4$	.236	.236
4	4000	1/4	$\pi/5$	.202	.147

Table 1. Coefficients representing one period of a sampled waveform, $x(n) = \sum_{m=0}^{4} A_m \cos(m\omega n + \phi_m)$.

Values for both the amplitudes and the phase angles for each of the five components are given in Table 1. Table 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 are shown at the end of Table 2. The so-called "analytic (cosine and sine) form" of the components of $x(n)$ is shown in Table 3. By applying the trigonometric identity

$$A_m \cos(m\omega n + \phi_m)$$

$$= A_m [\cos(m\omega n)\cos\phi_m - \sin(m\omega n)\sin\phi_m]$$

$$= A_m [a_m \cos(m\omega n) - b_m \sin(m\omega n)]$$

with a_m and b_m defined as in Table 1, we can see that the form given in Table 3 yields the same numbers for $x(n)$ as Table 2.

Figure 6 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. $x(n)$ is presumably an 8-sample sequence extracted from a longer sequence with a period of 8 samples (other values of this longer sequence are shown on dotted lines). A glance at Figure 6 confirms that the spectral structure of $x(n)$ is not very apparent from observation of its waveform,

$$x(n) = \sum_{m=0}^{4} A_m \cos(m\omega n + \phi_m) \qquad 0 \leqslant n \leqslant 7$$

$$\omega = \frac{2\pi}{8} = \frac{\pi}{4}$$

Components:

$$A_m \cos(m\omega n + \phi_m)$$

$m \downarrow$ \ $n \rightarrow$	0	1	2	3	4	5	6	7		
0	.100	.100	.100	.100	.100	.100	.100	.100	=	.100 cos (0)
1	1.000	.707	0	−.707	−1.000	−.707	0	.707	=	$\cos\left(\frac{n\pi}{4} + 0\right)$
2	.250	−.433	−.250	.433	.250	−.433	−.250	.433	=	$.500 \cos\left(\frac{n\pi}{2} + \frac{\pi}{3}\right)$
3	.236	−.333	.236	0	−.236	.333	−.236	0	=	$.333 \cos\left(\frac{3n\pi}{4} + \frac{\pi}{4}\right)$
4	.202	−.202	.202	−.202	.202	−.202	.202	−.202	=	$.250 \cos\ n\pi + \frac{\pi}{5})$
$\sum_{m=0}^{4}$ =	1.788	−.161	.288	−.376	−.684	−.909	−.184	1.038	=	$x(n)$

Table 2. Numerical Values of the Components of $x(n)$ as described in Table 1, and their sum, $x(n)$ itself.

yet applying the DFT to $x(n)$ will indeed tell us exactly the components of which $x(n)$ is composed.

In order to calculate the DFT it is useful to make a table of values for e^{-jwnk} such as the one shown in Table 4. We start with $k = 0$:

$$X(0) = \sum_{n=0}^{7} x(n)\, e^{-j0}$$

$$= \sum_{n=0}^{7} x(n) \cos(0) - j \sum_{n=0}^{7} x(n) \sin(0)$$

$$= \sum_{n=0}^{7} x(n) = 1.788 + (-.161) + .288 + (-.376)$$

$$+ (-.684) + (-.909) + (-.184) + 1.038$$

$$= .8$$

The real part of the answer (.8) is supposed to be N times one of the "a" coefficients in Table 1, and indeed, it is $N \cdot a_0$. The imaginary part of the answer (0) corresponds to b_0. So $k = 0$ apparently refers to the zero frequency, D.C., 0^{th} harmonic component of $x(n)$. For $k = 1$:

$$X(1) = \sum_{n=0}^{7} x(n)\, e^{-jwn}$$

$$= \sum_{n=0}^{7} x(n) \cos(2\pi n/N) - j \sum_{n=0}^{7} x(n) \sin(2\pi n/N)$$

$$= [(1.788)(1) + (-.161)(.707) + \ldots \text{etc.}]$$

$$+ j\,[(1.788)(0) + (-.161)(-.707) + \ldots \text{etc.}]$$

$$= 4 + j0$$

which is just $N/2$ times $(a_1 + jb_1)$, the coefficients for the 1^{st} harmonic, or fundamental frequency. Table 5 shows a

$$x(n) = \sum_{m=0}^{4} [a_m \cos(m\omega n) - b_m \sin(m\omega n)] \qquad 0 \leqslant n \leqslant 7$$

$$\omega = \frac{2\pi}{8} = \frac{\pi}{4}$$

Components:

$a_m \cos(m\omega n)$

$m \downarrow$ \ $n \rightarrow$	0	1	2	3	4	5	6	7		
0	.100	.100	.100	.100	.100	.100	.100	.100	=	$.100 \cos(0)$
1	1.000	.707	0	−.707	−1.000	−.707	0	.707	=	$\cos\left(\frac{n\pi}{4}\right)$
2	.250	0	−.250	0	.250	0	−.250	0	=	$.250 \cos\left(\frac{n\pi}{2}\right)$
3	.236	−.167	0	.167	−.236	.167	0	−.167	=	$.236 \cos\left(\frac{n3\pi}{4}\right)$
4	.202	−.202	.202	−.202	.202	−.202	.202	−.202	=	$.202 \cos(n\pi)$

$b_m \sin(m\omega n)$

$m \downarrow$ \ $n \rightarrow$	0	1	2	3	4	5	6	7		
0	0	0	0	0	0	0	0	0	=	$0 \sin(0)$
1	0	0	0	0	0	0	0	0	=	$0 \sin\left(\frac{n\pi}{4}\right)$
2	0	.433	0	−.433	0	.433	0	−.433	=	$.433 \sin\left(\frac{n\pi}{2}\right)$
3	0	.167	−.236	.167	0	−.167	.236	−.167	=	$.236 \sin\left(\frac{3n\pi}{4}\right)$
4	0	0	0	0	0	0	0	0	=	$.147 \sin(n\pi)$

Table 3. Alternative Form of the Description of the Components of $x(n)$ as Described in Table 1.

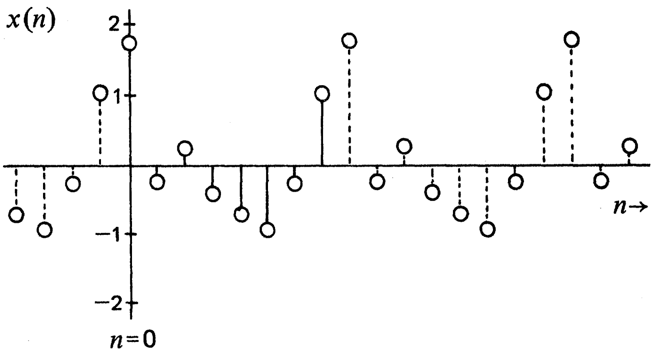


Figure 6. A graph of $x(n)$ as described in Table 1. $x(n)$ is periodic with a period of $N = 8$ samples. Only the period from $n = 0$ to $n = 7$ is considered (solid lines), but presumably the function repeats itself before and after this period (dotted lines).

$$e^{-jwnk} = \cos(\omega nk) - j \sin(\omega nk)$$

component values for $N = 8$, $\omega = 2\pi/N$

$\cos(2\pi nk/N)$

k ↓ \ n →	0	1	2	3	4	5	6	7
0	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
1	1.000	.707	0	−.707	−1.000	−.707	0	.707
2	1.000	0	−1.000	0	1.000	0	−1.000	0
3	1.000	−.707	0	.707	−1.000	.707	0	−.707
4	1.000	−1.000	1.000	−1.000	1.000	−1.000	1.000	−1.000
5	1.000	−.707	0	.707	−1.000	.707	0	−.707
6	1.000	0	−1.000	0	1.000	0	−1.000	0
7	1.000	.707	0	−.707	−1.000	−.707	0	.707

$-j \sin(2\pi nk/N)$

(all values below are multiplied by j)

k ↓ \ n →	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	−.707	−1.000	−.707	0	.707	1.000	.707
2	0	−1.000	0	1.000	0	−1.000	0	1.000
3	0	−.707	1.000	−.707	0	.707	−1.000	.707
4	0	0	0	0	0	0	0	0
5	0	.707	−1.000	.707	0	−.707	1.000	−.707
6	0	1.000	0	−1.000	0	1.000	0	−1.000
7	0	.707	1.000	.707	0	−.707	−1.000	−.707

Table 4. Values of e^{-jwnk} for $\omega = 2\pi/N$, $N = 8$

complete table of $X(k)$ for all values of 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 D.C. 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 is reversed. Since 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. $k = 5$ corresponds

k	$X(k)$	Corresponding Spectral Coefficients	Corresponding Frequency (Hz.)
0	$.8 + j0$	$N(a_0 + jb_0)$	± 0
1	$4 + j0$	$\frac{N}{2}(a_1 + jb_1)$	$+1000$
2	$1 + j\,1.732$	$\frac{N}{2}(a_2 + jb_2)$	$+2000$
3	$.944 + j\,.944$	$\frac{N}{2}(a_3 + jb_3)$	$+3000$
4	$1.616 + j0$	$N(a_4 + j0)$	± 4000
5	$.944 - j.\,944$	$\frac{N}{2}(a_3 - jb_3)$	-3000
6	$1 - j\,1.732$	$\frac{N}{2}(a_2 - jb_2)$	-2000
7	$4 - j0$	$\frac{N}{2}(a_1 - jb_1)$	-1000

$$X(k) = \mathrm{DFT}[x(n)] = \sum_{n=0}^{N-1} x(n)\, e^{-jwnk} \quad 0 \leqslant k \leqslant N-1$$

Table 5: The discrete Fourier transform (DFT) of $x(n)$ as described in Table 1 ($N = 8$, $R = 8000$ Hz.).

to the -3^{rd} harmonic, $k = 6$ corresponds 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 non-zero 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 D.C. and one-half sampling rate components perforce have zero imaginary parts, and hence it is impossible to split them into distinguishable positive and negative frequencies as can be lone with the other components. This says that, when sampling at 8000 Hz., we cannot distinguish components at +0 Hz. from −0 Hz., nor can we distinguish components at +4000 Hz. from components at −4000 Hz. This accounts for the ambiguity in Equation (1) when $F = \pm R/2$. It also accounts for the different scale factors for $X(0)$ and $X(N/2)$.

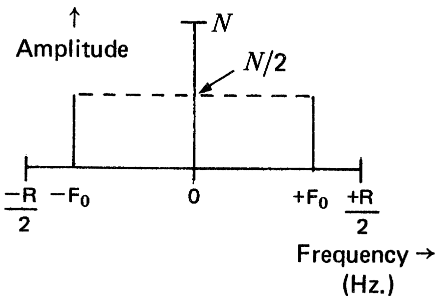
In order to completely check our transform definition we should apply the inverse DFT to $X(k)$ and see if $x(n)$ pops out again. This procedure will be left to the reader as a valuable exercise. Let us proceed by examining some of the properties of the DFT and derive some important transforms that will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

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

$$X(k) = \sum_{n=0}^{N-1} x(n)\, e^{-jwnk} \qquad -\infty < k < \infty \tag{19}$$

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 at the top of Figure 7, but this is really only a part of the full picture, shown at the bottom of Figure 7. This spectral periodicity is due to the periodicity of the complex exponential function itself, and theoretically 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 7 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



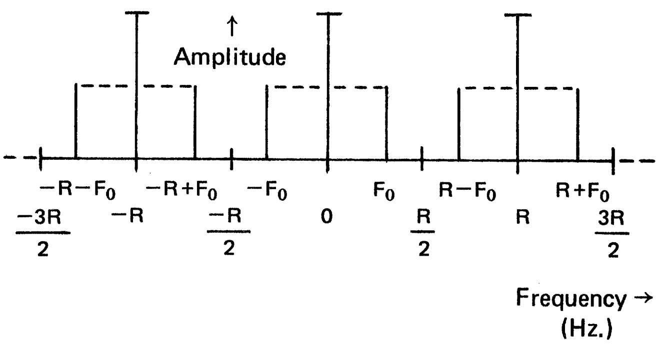


Figure 7. Spectrum of cosine function at frequency $|F_0| < R/2$. At the top we see just the 2 principal components, one at $+F_0$ Hz. and the other at $-F_0$ Hz., both with an amplitude of $N/2$. At the bottom we see three periods of the periodic spectrum of the same functions centered around 0 Hz. Normally only the top figure is used, but the spectrum of all digital signals is actually periodic as shown at the bottom.

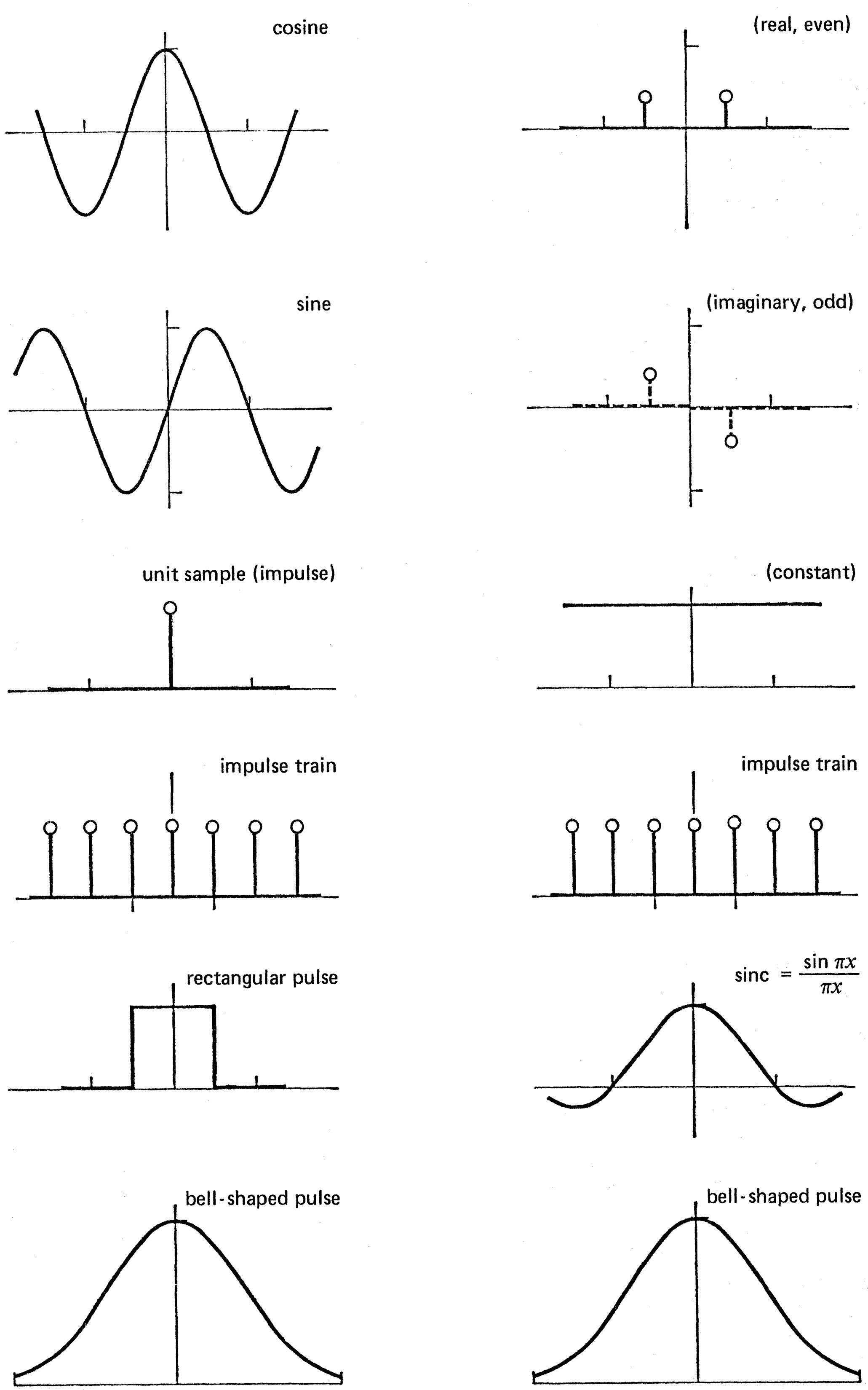


Figure 8. Some transform pairs. Each member of a pair transforms into the other (see text). The tick marks on the horizontal axes represent unit values of time or frequency; on the vertical axes, they represent unit values of amplitude. After *The Fourier Transform and Its Applications* by Ron Bracewell. Copyright © 1965 by McGraw-Hill, Inc. Used with permission of McGraw-Hill Book Company.

curve need be given (rather than a sequence of dots on the heads of sticks) for functions like cosine or sine—it is 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 8. 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 transforms into a constant spectrum. Or we can read this pair in the opposite direction: a constant-valued sampled function has energy only at zero Hz. The scale markings in Figure 8 correspond to unit values of time or frequency on the horizontal axes, and unit values of amplitude on the vertical axes. Remember that if one of the members of a pair is interpreted as a time function, its amplitude is scaled by one, and its corresponding spectral amplitudes will be scaled by N.

CONVOLUTION

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

$$\text{DFT}\,[c_1 x_1(n) + c_2 x_2(n)] = c_1 X_1(k) + c_2 X_2(k) \qquad (20)$$

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 (third from the top in Fig. 8), then the result is just the same as $x(n)$. In other words,

$$x(n) * u(n) = x(n) \qquad (21)$$

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 zero, except for $x(0)$, which would remain unchanged. The impulse is said to be an *identity* function with respect to convolution, since convolving any function with $u(n)$ leaves that function unchanged. If we scale $u(n)$ by a constant c_1, the result is

$$x(n) * c_1 u(n) = c_1 x(n)$$

which is just $x(n)$ again, but scaled by the same constant. If, however, we convolve $x(n)$ with $c_1 u(n - n_0)$, a shifted, scaled impulse function, the result is

$$x(n) * c_1 u(n - n_0) = c_1 x(n - n_0)$$

a shifted, scaled version of $x(n)$ (see Figure 9).

In order to convolve $x(n)$ with a scaled, delayed impulse function, we just scale and delay $x(n)$ by the same amount. But we can view *any* sampled function as a collection, or sequence, of scaled, delayed impulse functions! For example, suppose $x(n)$ is defined by

$$x(n) = \begin{cases} 1 & n = -1 \\ 2 & n = +1 \\ 0 & \text{otherwise} \end{cases}$$

and $y(n)$ is defined as

$$y(n) = \begin{cases} .5 & |n| \leqslant 2 \\ 0 & \text{otherwise} \end{cases}$$

In order to convolve $x(n)$ with $y(n)$ (see Figure 10), we can think of $x(n)$ as being composed of two impulses: one scaled by +1 and delayed by −1 samples, the other scaled by +2 and delayed by +1 samples. Thus we place a copy of $y(n)$ at $n = -1$ (*i.e.*, we form the function $y(n + 1)$), and another copy at $n = +1$, this one scaled by +2. The convolution of $x(n)$ and $y(n)$ is just the sum of these two shifted, scaled versions of $y(n)$. We could also have treated $y(n)$ as a set of five impulses located at $n = -2, -1, 0, 1, 2$, and scaled by .5 each. Placing scaled copies of $x(n)$ at each of these 5 locations and adding up the 5 resulting functions (*i.e.*, $.5\,[x(n+2) + x(n+1) + x(n) + x(n-1) + x(n-2)]$) would have yielded exactly the same result. This indicates that the convolution operation is *commutative*, which is to say $x(n) * y(n) = y(n) * x(n)$.

The mathematical definition of convolution is as follows. Suppose $x(n)$ is a sampled function of duration N_x samples, *i.e.*, $x(n)$ is non-zero only in the range $0 \leqslant n \leqslant N_x - 1$. Let $y(n)$ be a similar function of duration N_y samples. The convolution of $x(n)$ with $y(n)$ defines a new function

$$z(n) = x(n) * y(n) = \sum_{m=0}^{n} x(m)\,y(n - m) \qquad (22)$$

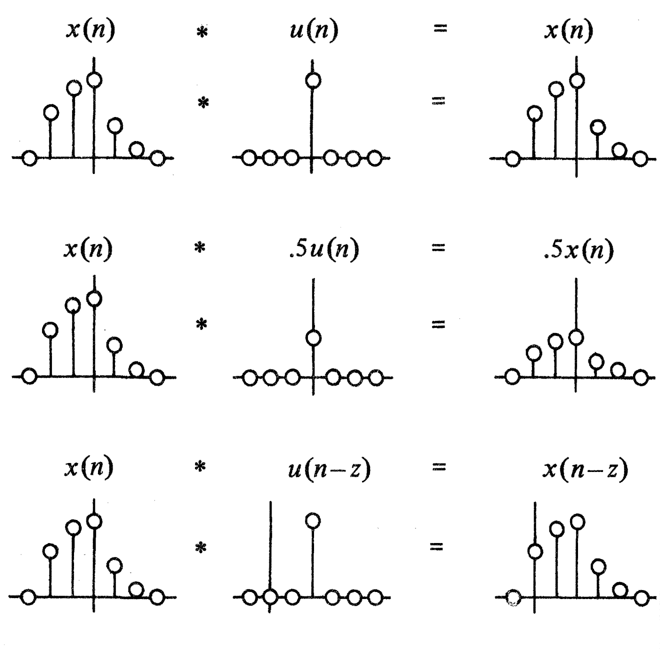


Figure 9. Examples of convolution of an arbitrary sequence $x(n)$ with the unit sample function.

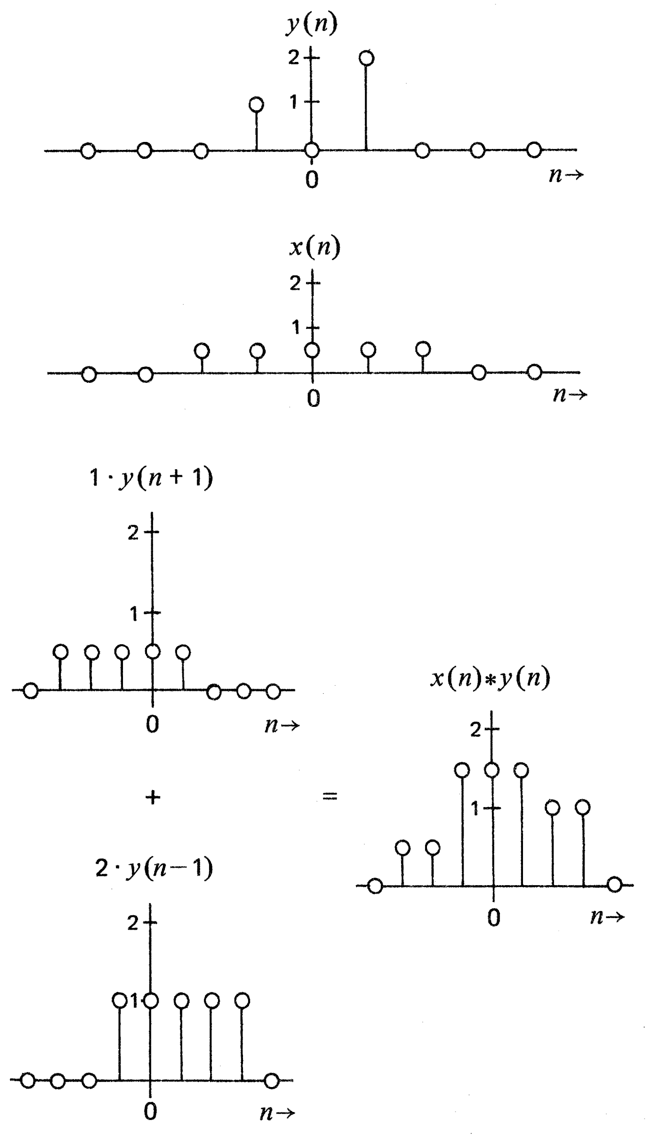


Figure 10. An example of the convolution of two functions, $x(n)$ with $y(n)$.

As is evident in Figure 10, the convolved sequence is generally of a longer duration than either function; the duration of $z(n)$ in Equation (22) is in fact $N_x + N_y - 1$ samples.

When we multiply two waveforms together, their spectra are convolved. Similarly, if two spectra are multiplied, their corresponding waveforms are convolved, as we shall see later when we discuss digital filtering.

Using convolution we can see what happens to the spectrum when we multiply two sampled waveforms together (amplitude modulation). Suppose we have two sampled cosine waves, one at frequency F_1 and the other at frequency F_2 (let both amplitudes be equal to one for simplicity). The sampling rate is R Hz., and is much greater than either F_1 or F_2, so foldover is not of concern. If we were to add the waves together, their spectra would also simply add, and the result is shown at the top of Figure 11. If we multiply the waveforms together, however, we must convolve the spectra of the two separate cosine waveforms. The result is shown at the bottom of Figure 11. We treat the spectrum of frequency F_2 as if it were two impulses with amplitude = ½. We then make two copies of the spectrum of the cosine waveform at frequency F_1, center a copy at each of the two locations indicated by the F_2 impulses, and scale the F_1 copies according to the F_2 impulse strengths. We have just demonstrated yet another trigonometric identity,

$$\cos(A)\cos(B) = \tfrac{1}{2}[\cos(A-B) + \cos(A+B)]$$

since the spectrum of the product of our two cosine waves was just two more cosine waves, one at the frequency $F_1 + F_2$ and the other at the frequency $F_1 - F_2$, both amplitudes scaled by ½. Notice that it is quite possible to get foldover when multiplying waveforms together, since, in this case for example, $F_1 + F_2$ Hz. might well be greater than $R/2$ Hz. even though neither F_1 nor F_2 is. The aliased components would have to obey Equation (1), just like all the well-behaved components.

THE Z-TRANSFORM

The z-transform is widely used in digital signal processing theory, and has a very simple definition. However, like the oriental game of Go, it is much easier to learn what the z-transform is than to master it. We call $x(n)$ a "one-sided" sequence if all of its values are zero for $n < 0$. It is called a finite, one-sided sequence if all values of $x(n)$ are also zero for

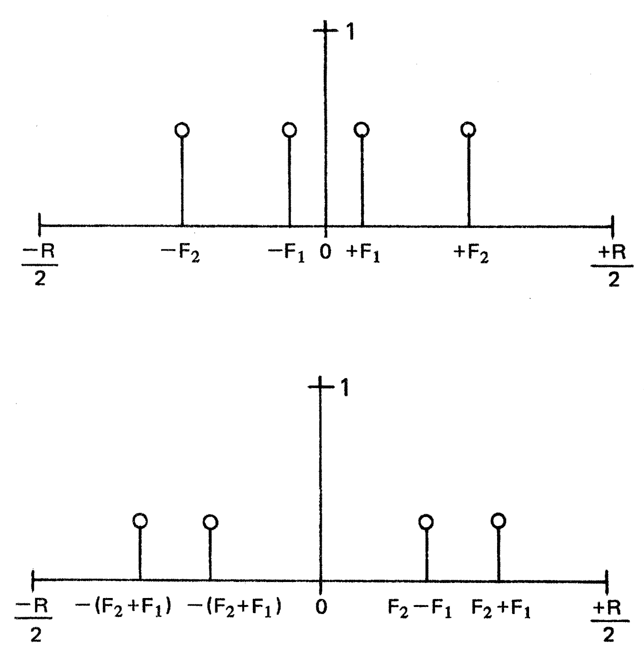


Figure 11. The spectrum of the sum (top) and product (bottom) of two cosine waveforms, one at frequency F_1 and the other at F_2.

$n > N$, for some finite value of N. The z-transform of a one-sided sequence $x(n)$ is then defined to be

$$X(z) = \sum_{n=0}^{\infty} x(n) z^{-n} \quad (23)$$

and for a finite, one-sided sequence of length N, it is

$$X(z) = \sum_{n=0}^{N-1} x(n) z^{-n} \quad (23)$$

where z is a complex variable.

The major difference between the DFT and the z-transform is that while in the DFT we multiply and sum the samples of a waveform with a particular complex value, namely e^{-jw}, in the z-transform, the value of z may be set to *any* complex value whatever. Clearly, if we set z to e^{jw}, then the z-transform is equivalent to the DFT for finite sequences. But the z-transform can also be used to gain an additional kind of insight into the nature of digital signals. For example, if $x(n)$ is the *unit-step* function

$$x(n) = \begin{cases} 1 & n \geq 0 \\ 0 & n < 0 \end{cases}$$

then its one-sided z-transform is just

$$X(z) = \sum_{n=0}^{\infty} 1 \cdot z^{-n} = \sum_{n=0}^{\infty} z^{-n} = \frac{1}{1 - z^{-1}} \quad (25)$$

The closed form of Equation (25) is just a statement of the result of summing an infinitely long geometric series, such as those discussed in Part I of this tutorial. Similarly, we see that the z-transform of the complex exponential

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

is just

$$X(z) = \sum_{n=0}^{\infty} e^{jwn} z^{-n} = \sum_{n=0}^{\infty} (z^{-1} e^{jw})^n$$

$$= \frac{1}{1 - z^{-1} e^{jw}} \quad (26)$$

We recall that such geometric sums *converge* (*i.e.*, sum to a finite value) only for certain restricted values of z. For example, Equation (25) converges only if $|z^{-1}|$ is less than one since otherwise we would be adding together an infinitely long sequence of numbers which do not get smaller as we go along. In order for $|z^{-1}|$ to be less than one, $|z|$ must be greater than one. Since $z = a + jb$ is complex, we can see that $|z| = \sqrt{a^2 + b^2}$ will be greater than one only when $\sqrt{a^2 + b^2} > 1$. If we make a graph of the equation $|z| = \sqrt{a^2 + b^2} = 1$ on the *complex plane* (*i.e.*, graphing the real part along the horizontal axis and the imaginary part along the vertical axis), we find that the graph is simply a circle of radius one centered at the origin, called the *unit circle*. Since each point on the complex plane represents a possible value for z, we see that $|z| > 1$ specifies the set of all points on the complex plane that lie *outside* of the unit circle. Similarly, $|z| = 1$ specifies all points *on* the unit circle, while $|z| < 1$ is the set of all points *inside* it.

The z-transform of any function has a general form:

$$X(z) = A \frac{\prod_{i=1}^{M} (1 - z_i z^{-1})}{\prod_{i=1}^{N} (1 - p_i z^{-1})} \quad (27)$$

where

- A is an arbitrary constant,
- Π is the "product operator," analogous to the "sum operator" Σ except that multiplication rather than addition is specified; for example, $\prod_{c=1}^{4} c = 1 \cdot 2 \cdot 3 \cdot 4 = 24$,
- z_i are M "zeroes" of $X(z)$ and
- p_i are N "poles" of $X(z)$.

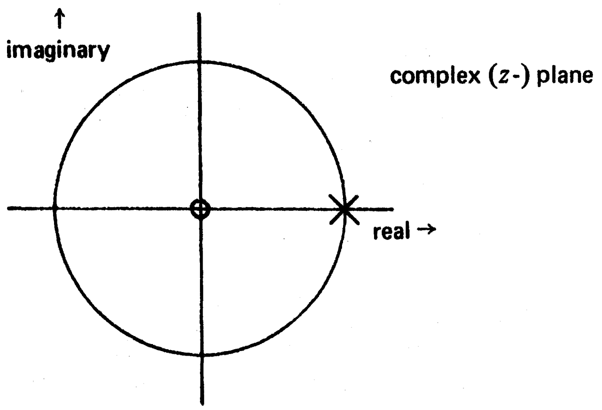
Both the numerator and denominator of Equation (27) are polynomials in the complex variable z, and they are shown here in factored form in order to demonstrate that the z_i and p_i values are really just the roots of the numerator and denominator polynomials, respectively. These roots may be complex, and if they are, we know that as long as the coefficients of the numerator and denominator polynomials are real (as they must be for a realizable system), then the complex roots will always appear in conjugate pairs. We can then represent the z-transform of a sequence by writing the transform in the form given by Equation (27) and plotting the locations of the poles (p_i) and zeros (z_i) as "X's" and "0's" on the complex plane. Such a "pole-zero" plot is just a unique way to represent the sequence $x(n)$ on the complex plane.

For example, Equation (25) is a z-transform with $M = 1$ zero equal to 0 and $N = 1$ pole equal to 1, since

$$\frac{1}{1 - z^{-1}} = \frac{(1 - 0 \cdot z^{-1})}{(1 - 1 \cdot z^{-1})}$$

Similarly, Equation (26) has one (complex) zero with a value of zero, and one (complex) pole with a value of e^{jw}. We can graph these two equations on the complex plane as shown in Figure 12. Poles are graphed as X's, and zeros are graphed as 0's; the unit circle is shown for reference.

The poles of the bottom graph are "mirror images" of each other in relation to the horizontal (real) axis. In other words, the two poles differ only in the sign of their imaginary part, and thus the poles represent a "conjugate pair." The exact location of the e^{jw} pole pair depends on the value of ω, the frequency of the sinusoid. If $\omega = 0$, the pole pair will be coincident, *i.e.*, both X's are placed on top of each other at $z = 1$. As the frequency increases to $\pm R/2$, ω increases to $\pm\pi$,



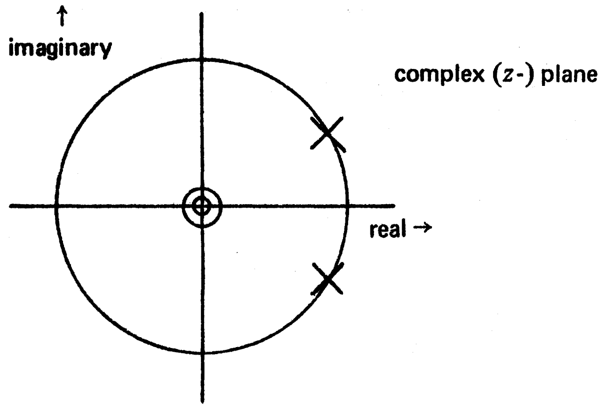


Figure 12. Pole-zero plots of the z-transform of Equation 27. Top: $M = 1, N = 1$. Bottom: $M = 2, N = 2$.

and at $\omega = \pm\pi$, the two poles would be on top of each other at $z = -1$. Intermediate frequencies are represented at intermediate positions along the circumference of the unit circle.

The z-transform, like the DFT, also has an inverse, albeit a more complicated one than the inverse DFT. The only method of performing the inverse z-transform which we will consider here is called partial fraction expansion of $X(z)$. If, by algebraic manipulation, we change the form of $X(z)$ from that of Equation (27) to

$$X(z) = \sum_{i=1}^{N} \frac{c_i}{(1 - p_i z^{-1})} \tag{28}$$

where the p_i are the poles (roots of the denominator polynomial) and the c_i are constants derived in the algebraic manipulation, then the ***inverse*** z-***transform*** is given by:

$$x(n) = \begin{cases} \sum_{i=1}^{N} c_i (p_i)^n & n \geqslant 0 \\ 0 & n < 0 \end{cases} \tag{29}$$

Other methods exist for the inverse z-transform, but unfortunately, none of them is simpler to perform than this one.

The significance of the z-transform is its use in digital filter theory. We will not delve very deeply into this topic due to its vast complexity (no doubt it has imaginary parts in the minds of many theorists), but we can discuss basic elements of some aspects of digital filters.

DIGITAL FILTERING

The basic purpose of a digital filter is to modify the spectrum of digital signals in a desirable way. A digital filter is usually pictured as a "black box" with a signal input and a signal output. The operation of the box is described by what engineers call its *transfer function*, a mathematical formulation of its operation in the transform domain. For example we may wish to create a filter which will pass all frequency components lower in frequency than a certain value, F_c, and to remove all others. Such a filter would be a low-pass filter (LPF), and F_c is its *cutoff frequency.* We may desire to create high pass filters (HPF) or band-pass filters (BPF) to remove or attenuate certain frequencies while passing others. An all-pass filter may be used to modify only the phase on certain components. In general we would like to be able to specify an arbitrary transfer function in terms of a frequency response curve, to be multiplied by the frequency spectrum of any signal which is treated by it, and possibly also a phase response curve, which would modify the phase of an arbitrary signal. The transfer function for an ideal LPF is shown in Figure 13. From now on, we will use "normalized frequency" in our plots, where $\omega = \pm\pi$ corresponds to a frequency of $\pm R/2$. This allows us to speak of frequencies in a way that is independent of any particular sampling rate.

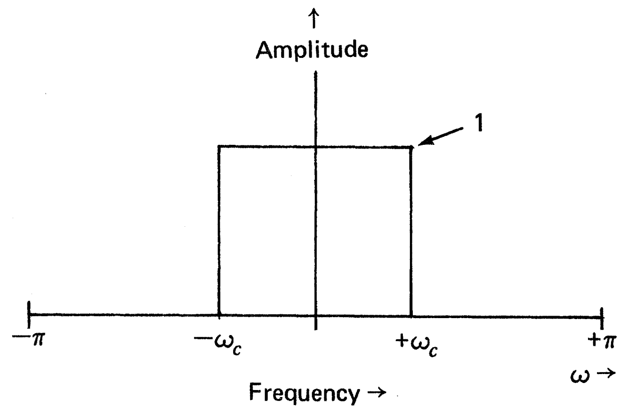


Figure 13. Transfer function for an ideal low pass filter with (normalized) cutoff frequency ω_c.

The filter described in Figure 13 operates by multiplying the spectrum shown by the spectrum of the input signal, thereby setting to zero any frequency components in the signal that are greater than $\pm\,\omega_c$.

If we think of Figure 13 as the spectrum of some signal, call it $h(n)$, then we would be able to implement our ideal LPF by convolving the input signal with $h(n)$, since multiplying spectra corresponds to convolution of waveforms, just as multiplying waveforms corresponds to convolving their spectra. Thus, if $x(n)$ is the input signal and $X(z)$ is its transform, $y(n)$ is the output signal and $Y(z)$ is its transform, and

$h(n)$ is the inverse transform of $H(z)$, then the transfer function of our filter,

$$y(n) = h(n) * x(n) \tag{30}$$

corresponds to

$$Y(z) = H(z)X(z) \tag{31}$$

The function $h(n)$ is called the *impulse response* (or *unit sample response*) of the filter, since $Y(z) = H(z)$ only if $X(z) = 1$, which is the transform of the digital impulse function.

We can uniquely describe any filter by its impulse response $h(n)$, which is just the (inverse) transform of its transfer function $H(z)$. And once we know $H(z)$, which we can obtain from any filter by transforming its response to an impulse function, we know what the filter will do to *any* input signal, since $H(z)$ defines the frequency and phase response of the filter completely.

We calculate the frequency and phase response of a filter with the following steps:

1. Find the impulse response $h(n)$ of the filter. This can be done empirically by exciting the filter with $u(n)$ as an input signal; $h(n)$ is then "what comes out."
2. Calculate the z-transform of the impulse response: $H(z) = \sum_n h(n)z^{-n}$. This sum may be either finite or infinite, depending on whether $h(n)$ is of finite duration.
3. Set z equal to e^{jw}. This makes the transfer function $H(z)$ yield the spectrum of the impulse response, $H(e^{jw})$.
4. The frequency response of the filter is then defined to be the magnitude of the spectrum of its impulse response: $|H(e^{jw})|$.
5. The phase response is similarly pha $[H(e^{jw})]$.

Many subtle problems are encountered in digital filter design. For example, the ideal LPF described above is an example of a filter which is said to be *unrealizable*. If we examine the transform of the "rectangular box" function in Figure 8, we see that it looks something like a cosine wave that dies down in amplitude on both sides of the origin (this function is called the "sinc" function). If the rectangular box were a digital waveform (a sort of single half period of a square wave), then this says that the spectrum of this signal gradually dies away as the magnitude of the frequency increases, both in positive and negative directions. But if the rectangular box is the spectrum, as for the ideal LPF, this says that the impulse response of the filter begins *before* $n = 0$, the time at which the impulse occurs. In other words, the ideal LPF begins to respond to its input *before it has occurred*. No wonder this filter is called unrealizable! It would need some sort of "crystal ball" with which it could look into the future, see an impulse coming from the distant future, and begin its response before its input arrives. The moral is: if anyone offers to sell you an ideal LPF, don't buy it (unless you check it *very* carefully)!

We will end our discussion of digital filters with *the* general formula for all digital filters and two simple examples: a first-order *realizable* (but hardly ideal) LPF, and a second order BPF.

Any digital filter may be described by an equation of the form:

$$y(n) = \sum_{i=0}^{m} b_i x(n-i) - \sum_{i=1}^{N} a_i y(n-i) \tag{32}$$

where

- $x(n)$ is an input signal or sequence,
- $y(n)$ is the output signal,
- b_i are a set of M coefficients describing how $y(n)$ depends on the current input sample and the previous M input samples, and
- a_i are a set of N coefficients describing how $y(n)$ depends on the previous N output samples.

The b_i coefficients of Equation (32) determine the zeros of the transfer function, and the a_i coefficients determine the poles. Some filters have only zeros and no poles (*i.e.*, $N = 0$), and hence their output depends only on the current input sample and the past M samples. Such filters are called *transversal*, or *finite impulse response* (FIR) filters, since their response to an impulse cannot last any longer than M samples, as defined in Equation (32). If a filter contains poles, it is called a *recursive*, or *infinite impulse response* (IIR) filter.

Many filters contain both poles and zeroes in their transfer function, the simplest of which is a first-order LPF, defined by the difference equation:

$$y(n) = x(n) + Ky(n-1) \tag{33}$$

where K is an arbitrary constant less than one. In terms of Equation (32), our first-order filter has $b_0 = 1$ and $a_1 = K$. We can find its transfer function by transforming the impulse response, $h(n)$; the frequency response will then be $|H(e^{jw})|$ and the phase response will be given by pha $[H(e^{jw})]$. If we start with the *initial condition* that $y(-1) = 0$, then we can list the values of $h(n)$ as follows:

$x(n) = u(n)$, $y(-1) = 0$

$y(n) = u(n) + Ky(n-1)$

n	$x(n) = u(n)$	$y(n)$	
0	1	$1 + Ky(-1) =$	$1 = K^0$
1	0	$0 + Ky(0) =$	$K = K^1$
2	0	$0 + Ky(1) =$	$K \cdot K = K^2$
3	0	$0 + Ky(2) =$	$K \cdot K^2 = K^3$
.	.	.	
.	.	.	
.	.	.	

etc.

Since $h(n)$ is infinitely long, our filter is of type IIR with impulse response

$$h(n) = \begin{cases} K^n & n \geq 0 \\ 0 & n < 0 \end{cases}$$

The one-sided z-transform of $h(n)$ is then

$$H(z) = \sum_{n=0}^{\infty} h(n) z^{-n} = \sum_{n=0}^{\infty} K^n z^{-n}$$

$$= \sum_{n=0}^{\infty} (Kz^{-1})^n = \frac{1}{1-Kz^{-1}}$$

Setting z equal to e^{jw} now yields the spectrum for our filter

$$H(e^{jw}) = \frac{1}{1-Ke^{-jw}}$$

We now have only to find the magnitude and phase of the function $H(e^{jw})$ in order to make the frequency and phase responses explicit. A useful property of complex numbers is that the product of a complex number with its conjugate is equal to the squared magnitude of the complex number; *i.e.*, if $z = a + jb$, then

$$zz^* = (a+jb)(a-jb) = a^2 + b^2 = |z|^2$$

(where z^* = the conjugate of $z = a - jb$) It is also easy to verify the following relations:

$$\text{Re } z = \text{ the real part of } z = \frac{1}{2}(z + z^*) = a$$

$$\text{Im } z = \text{ the imaginary part of } z = \frac{1}{2j}(z - z^*) = b$$

$$\text{pha } z = \tan^{-1}\frac{\text{Im } z}{\text{Re } z} = \tan^{-1}\frac{z - z^*}{j(z + z^*)}$$

Thus, in order to find the frequency response we calculate the magnitude of $H(e^{jw})$:

$$|H(e^{jw})| = [H(e^{jw})H(e^{-jw})]^{1/2}$$

$$= \left[\frac{1}{1-Ke^{-jw}} \cdot \frac{1}{1-Ke^{+jw}}\right]^{1/2}$$

$$= \left[\frac{1}{1-K(e^{jw}+e^{-jw})+K^2}\right]^{1/2}$$

Noting from Euler's relation that $e^{jw} + e^{-jw} = 2\cos\omega$, we obtain

$$|H(e^{jw})| = \left[\frac{1}{\sqrt{1-2K\cos\omega + K^2}}\right]$$

The phase response is equal to

$$\text{pha } H(e^{jw}) = \tan^{-1}\left[\frac{\text{Im } H(e^{jw})}{\text{Re } H(e^{jw})}\right]$$

$$= \tan^{-1}\left[\frac{\dfrac{1}{1-Ke^{-jw}} - \dfrac{1}{1-Ke^{+jw}}}{j\left(\dfrac{1}{1-Ke^{-jw}} + \dfrac{1}{1-Ke^{+jw}}\right)}\right]$$

$$= \tan^{-1}\left[\frac{1-Ke^{jw}-(1-Ke^{-jw})}{j(1-Ke^{jw}+1-Ke^{-jw})}\right]$$

$$= \tan^{-1}\left\{\frac{K(e^{-jw}-e^{jw})}{j[2-K(e^{jw}+e^{-jw})]}\right\}$$

Again we make use of Euler's relation in two forms: $e^{jw} + e^{-jw} = 2\cos\omega$ and $e^{jw} - e^{-jw} = 2j\sin\omega$. Also, note that the inverse tangent is an odd function, *i.e.*, $\tan^{-1}(-\theta) = -\tan^{-1}(\theta)$.

$$\tan^{-1}\left\{\frac{K(e^{-jw}-e^{jw})}{j[2-K(e^{jw}+e^{-jw})]}\right\}$$

$$= -\tan^{-1}\left\{\frac{K(e^{jw}-e^{-jw})}{j[2-K(e^{jw}+e^{-jw})]}\right\}$$

$$= -\tan^{-1}\left[\frac{K2j\sin\omega}{j(2-K2\cos\omega)}\right]$$

$$= -\tan^{-1}\left(\frac{K\sin\omega}{1-K\cos\omega}\right)$$

$$= \text{pha } H(e^{jw})$$

We can now make graphs of $|H(e^{jw})|$ and pha $H(e^{jw})$ from $-\pi \leqslant \omega \leqslant +\pi$ to get a complete picture of the transfer function as a frequency response and a phase response. Since the frequency response of most filters varies over such a great numerical range, we typically plot it on a logarithmic vertical scale, such as is done in Figure 14. If we plot $20\log_{10}|H(e^{jw})|$, we can even read the vertical scale directly in dB. A value of 0 dB corresponds to $|H(e^{jw})| = 1$, +6 dB corresponds to $|H(e^{jw})| = 2$, −6 dB corresponds to $|H(e^{jw})| = .5$, and so on.

The frequency response curves in Figure 14 indicate clearly that our first-order system is a low pass filter, but hardly an ideal one.

A simple second-order filter might have the equation

$$y(n) = x(n) + K_1 y(n-1) + K_2 y(n-2) \tag{34}$$

where K_1 and K_2 are constants. In attempting to derive the impulse response of this filter, $h(n)$, we find that the mathematical tools which we have developed so far are unfortunately not adequate to the task. In order to "solve" Equation (34) for $h(n)$ we would need to develop the theory of *difference equations*, which are the digital correspondents of differential equations in the analog world. If we are *given* $h(n)$, however, we could then carry out the necessary calculations to obtain the frequency and phase response. The impulse response corresponding to Equation (34) will be given here without derivation, since it is possible to make use of this filter without being able to solve difference equations. With the following gift from difference equation theory, then, we can proceed.

The impulse response of Equation (34) is either the sum of two exponentially decaying functions, or a sinusoid whose amplitude decays exponentially, depending on the relationship

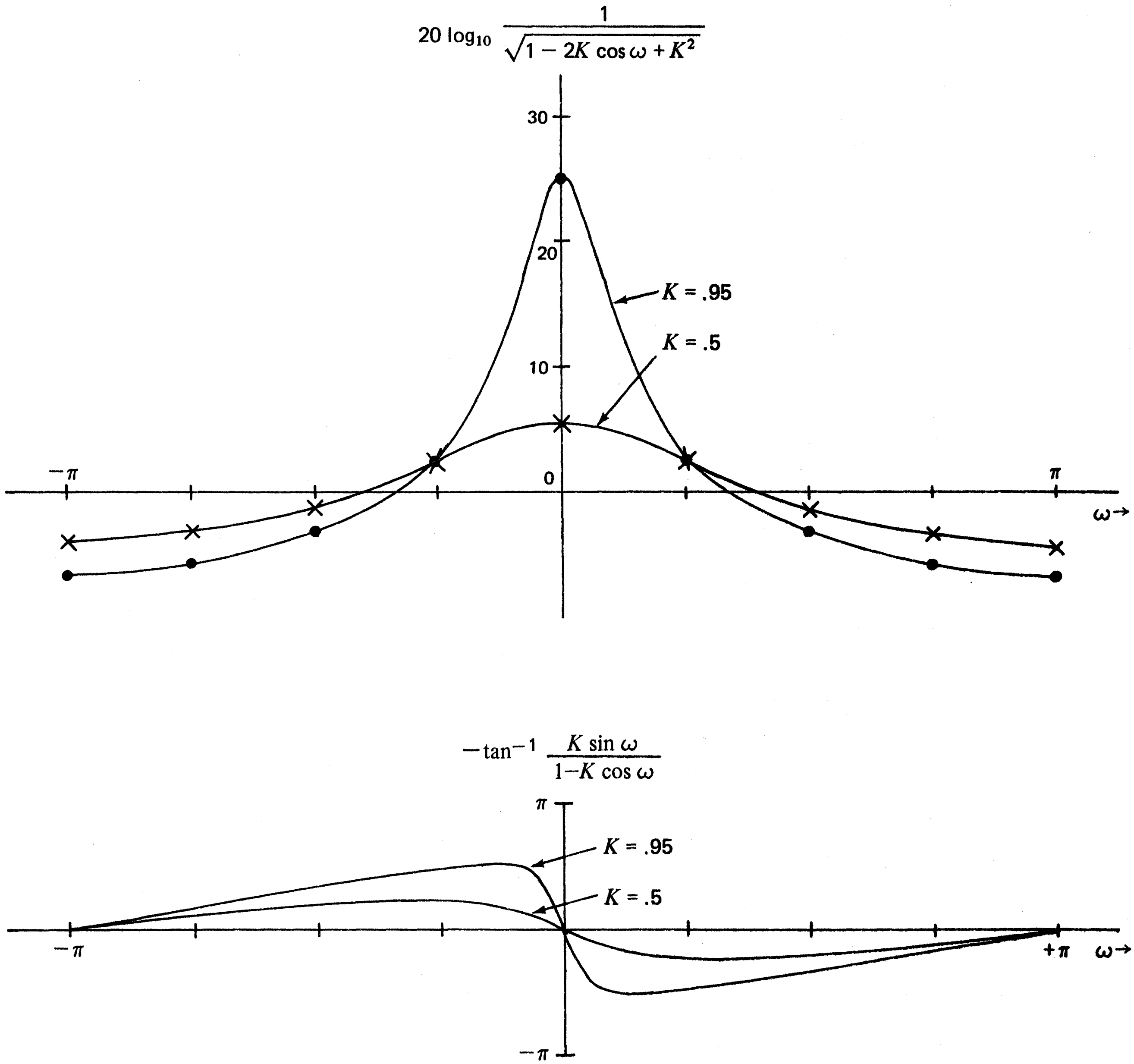


Figure 14. Frequency and phase response of a first-order system (see text) for two values of K (.95 and .5).

between the K_1 and K_2 coefficients. If $K_1 \geqslant -K_2^2/4$, the impulse response is of the form

$$h(n) = \alpha_1 (\rho_1)^n + \alpha_2 (\rho_2)^n \tag{35a}$$

(sum of exponentials). If $K_1 < -K_2^2/4$, then

$$h(n) = \alpha_1 r^n \sin(b_n + \theta) \tag{35b}$$

where

$$r = \sqrt{-K_2} \quad ,$$

$$b = \cos^{-1} \frac{K_1}{2\sqrt{-K_2}} \quad ,$$

$$\theta = b \text{ , and}$$

$$\alpha_1 = \frac{1}{\sin b}$$

The frequency and phase response of this filter can be calculated from the z-transform of Equation (35b), evaluated at $z = e^{jw}$:

$$H(e^{jw}) = \frac{1}{1 - 2r(\cos b)\, e^{-jw} + r^2 e^{-2jw}} \tag{36}$$

The interested reader will find it an exhilerating exercise to plot curves for $|H(e^{jw})|$ and pha $H(e^{jw})$ based on Equation (36).

WHERE TO GO FROM HERE

We have only scratched the surface of mathematics and digital signal processing, but we have scratched it pretty well. Many of the concepts we have discussed are subtle and not easily understood, and it should come as no surprise if the uninitiated reader is a bit confused at this point. Confusion is the natural companion of learning. When Jean Baptiste Joseph Fourier first established the fact that an arbitrary mathematical function could be represented as a sum of sinusoids (in 1807), leading mathematicians of the day refused to believe it, so new and startling was the idea. No doubt the idea confused them, too. People have been confused by this idea ever since, of course, but we now know it to be one of the most useful basic facts about mathematics, science, engineering, and sound, including music. The truth and beauty of the idea makes all the initial confusion worthwhile.

If the confusion persists, however, after a few careful readings and problem solutions, the time has come to attack it before it turns into frustration. Some recommended references for further study are given after the problems, organized by topic. An hour or two with a good teacher can be worth months of self study in certain cases, but skill always comes from practice, and no number of hours watching a teacher solve problems will make a student skilled at this.

Finally, a word about scope. Computer music is a highly interdisciplinary field, containing portions of computer science, music, science, mathematics, and engineering. Just which parts of all of these fields are part of computer music and which are not is unclear at this time, though it *is* clear that no individual is likely to have the utopian qualification of expertise in all of these fields at once. Consequently, it is likely that computer music will continue to be fruitfully practiced by small groups of cooperating individuals rather than by single persons working alone. Also, it means that individuals must be willing to learn as much as possible, and to rely on good sources for that which is unknown. The trick is to be able to discern a good source from a not-so-good source. It would help a great deal if computer musicians, with their various specialities, were able to speak a common language whereby their individual specialities could be described. One such language exists in the form of computer programming principles. It is possible for a composer to explain the rules of harmony in computer programming terms to a mathematician, and for the mathematician to explain integrations in computer programming terms to a composer, if both can program. Thus the core of computer music must be in programming and music composition or performance. Beyond a good working knowledge of these two fields, mathematics seems to be a unifying element for the remaining areas of acoustics, psychology and signal processing. An overview of each of these areas would be useful for work in computer music. Tutorials in the fields of acoustics and psychological acoustics similar to this one in mathematics and signal processing would provide an even more solid ground for discussion. We have covered about as much signal processing as the non-specialist practitioner of computer music is likely to need, at least in the way of basics. We are now in a position to build on this foundation.

Some More Problems

1. (Sampling). Non-bandlimited waveforms are characterized by large jumps (discontinuities) in their waveforms; such a waveform is depicted in analog form in Figure P1. The Fourier representation of the sawtooth as it is shown is

$$F(t) = 2 \sum_{k=1}^{\infty} \frac{(-1)^{k-1}}{k} \sin kt$$

   a) Derive an expression for the equivalent waveform sampled at R samples per second with a frequency of F Hz.
   b) Suppose $R = 10$ kHz. and $F = 440$ Hz. What is the lowest frequency component of this waveform which will be folded (aliased) to an incorrect frequency? What is the amplitude of this component?
   c) Suppose we cannot hear components which are 40 dB weaker in amplitude than the fundamental of the sawtooth. What is the highest frequency sawtooth which will have aliased components 40 dB less strong than the fundamental (again, assume $R = 10000$ Hz.)?

2. (Sampling). Suppose we program a computer to produce a sine wave of constantly increasing frequency according to the relation

   Frequency = 1000 (time in seconds).

   The sampling rate of the digital-to-analog converter is set to 32 kHz. Make a graph showing the frequency we will actually hear coming from the DAC as a function of time for $t = 0$ to $t = 60$ seconds.

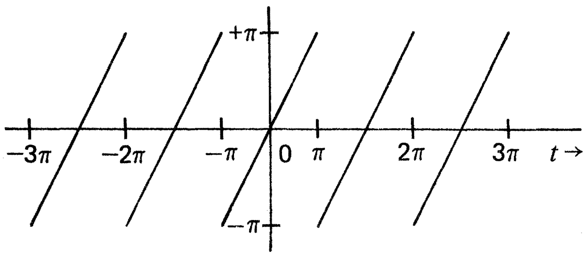


Figure P1: A non-bandlimited sawtooth waveform.

3. (DFT). Make a spectral plot for the DFT worked out in Tables 1 through 5. Label the horizontal axis with the frequencies −4000, −3000, −2000, − 1000, 0, 1000, 2000, 3000, and 4000 Hz. and plot the *magnitude* of $X(k)$ at each of these frequencies on a dB scale.

4. (Convolution). A square wave is characterized by a spectrum containing only odd-numbered multiples of the fundamental frequency, with amplitudes equal to one-over-the-component-number:

$$x(n) = \sum_{k=1}^{M} \frac{1}{k} \sin(k\omega n), \quad k \text{ odd.}$$

Foldover can be avoided by choosing a suitably small value for M. Suppose that we produce this waveform at F Hz. $<< R/2$ Hz., and that we multiply it by

$$y(n) = (1 + \cos 2\omega n)$$

i.e., a cosine waveform at twice the fundamental frequency of the square wave.

a) What is the spectrum of the resulting waveform?
b) Is it still a square wave?

5. (DFT). Suppose $x(n)$ represents 16 samples of a digitized sine waveform with a period of 16 samples, *i.e.*

$$x(n) = \sin(2\pi n/N) \qquad 0 \leqslant n \leqslant N-1$$

with $N = 16$.

a) What is the DFT of $x(n)$?
b) Suppose we took the DFT of only the first 8 samples of $x(n)$ as defined above. Is the DFT the same? Why or why not?

6. (z-transform). Make a pole-zero plot of the z-transform of the first-order filter discussed in the text. What happens to the location of the pole as K goes from 0 to 1?

7. (Digital Filters). Make a plot of the frequency response of the second order filter (Equation (36)) for $r = .95$, $b = \pi/4$. How does it differ from the frequency response of a first-order filter?

ACKNOWLEDGEMENTS

I wish to thank John Snell, John Strawn, John Gordon, and Marc LeBrun for their many helpful suggestions; without their dedication such articles as this, and indeed the *Computer Music Journal* itself, could not exist.

The editors of *Computer Music Journal* would also like to thank John Gordon for providing drawings for Fig. 3-6 and 9-10.

REFERENCES

Acoustics and Psychoacoustics

Benade, Arthur H. *Fundamentals of Musical Acoustics.* New York: Oxford University Press, 1976.
Helmholtz, H. von. *On the Sensations of Tone as a Physiological Basis for the Theory of Music.* Trans. by A. Ellis. New York: Dover, 1954.
* Roederer, Juan G. *Introduction to the Physics and Psychophysics of Music.* New York: Springer, 1975.

Computer Music and Electronic Music

Appleton, Jon H., and Ronald C. Perera, ed. *The Development and Practice of Electronic Music.* Englewood Cliffs, NJ.: Prentice-Hall, 1975.
* Mathews, Max V. *The Technology of Computer Music.* Cambridge: MIT Press, 1969.

Computer Programming

* Knuth, Donald E. *The Art of Computer Programming. Vol. 1: Fundamental Algorithms.* Reading, Mass: Addison-Wesley, 1969.

Digital Signal Processing

Oppenheim, Alan V., and Ronald W. Schafer. *Digital Signal Processing.* Englewood Cliffs, N.J.: Prentice-Hall, 1975.
* Rabiner, Lawrence R., and Bernard Gold. *Theory and Application of Digital Signal Processing.* Englewood Cliffs, N.J.: Prentice-Hall, 1975.
Robinsen, Enders A., and Manuel T. Silvia. *Digital Signal Processing and Time Series Analysis.* San Francisco: Holden-Day, 1978.

Mathematics

Bracewell, Ron. *The Fourier Transform and its Applications.* New York: McGraw-Hill, 1965.
Polya, George, and Gordon Latta. *Complex Variables.* New York: Wiley, 1974.
Spiegel, Murray R. *Mathematical Handbook of Formulas and Tables.* Schaums's Outline Series in Mathematics. New York: McGraw-Hill, 1965.
* Steward, Ian, and David Tall. *The Foundations of Mathematics.* New York: Oxford University Press, 1977.

(*—first choice)