Get accurate Fast Fourier Transforms

with a digital computer. What's needed is a clear understanding of the practical limitations and tradeoffs.

Large savings in design time can be realized with Fourier transformations that permit the analysis of functions in either the time or frequency domain. Their versatility spans the design spectrum, because the frequency domain is convenient for linear analysis while the time domain is ideal for nonlinear systems. The value of Fourier transforms as a designer's tool has been further increased with the development of the Fast Fourier Transform (FFT),[1-6] with techniques for speeding the FFT,[7] and with the growing availability of computers. But digital computers, of course, can work only with *discrete* transforms, and this creates a problem.

The designer must know the precise conditions under which a computer transform is useful and reliable. Once these conditions are understood, not only will the results be reliable but the designer will also be able to specify with confidence the memory size and amount of data needed to take the transform of a given type of waveform.

Transforms come in pairs

What is a transform? The word is often misused by engineers. When an engineer talks about a "transform," he usually means one member of a *transform pair*. But the pair consists of two functions. And when a set of data is applied to one function, a second set of data results. This second set, when applied to the second function, must reconstruct the original data precisely. For example,

$$F(i) = (1/2)[f(i) + 7]$$

and

$$f(j) = 2F(j) - 7$$

is a transform pair, although not a very useful one.

The transform pair for the Fourier series of a function f(t) of period T is

$$F(n) = (1/T)\int_{-T/2}^{T/2} f(t)\, e^{-jn\omega_0 t}\, dt \qquad (1)$$

$$f(t) = \sum_{n=-\infty}^{\infty} F(n)\, e^{jn\omega_0 t}.$$

That this is a transform pair can be readily verified by plugging one into the other and noting that an identity results.

Another transform pair, called the Fourier integral, is defined as follows:

$$F(\omega) = (1/2\pi)\int_{-\infty}^{\infty} f(t)\, e^{-j\omega t}\, dt \qquad (2)$$

$$f(t) = \int_{-\infty}^{\infty} F(\omega)\, e^{j\omega t}\, d\omega.$$

This transform pair is also a valid expression for most cases of f(t).

Transform pairs 1 and 2 are both well known. Furthermore we are interested in a different kind of transform pair—the one that can be handled by a computer. This is because a digital computer cannot integrate continuous functions, nor can it operate between infinite limits.

A pair that the computer can handle

A transform pair defined as a Discrete Finite Transform (DFT) is given by

$$F(n) = (1/N)\sum_{i=0}^{N-1} f(i)\, e^{-jin2\pi/N}. \qquad (3)$$

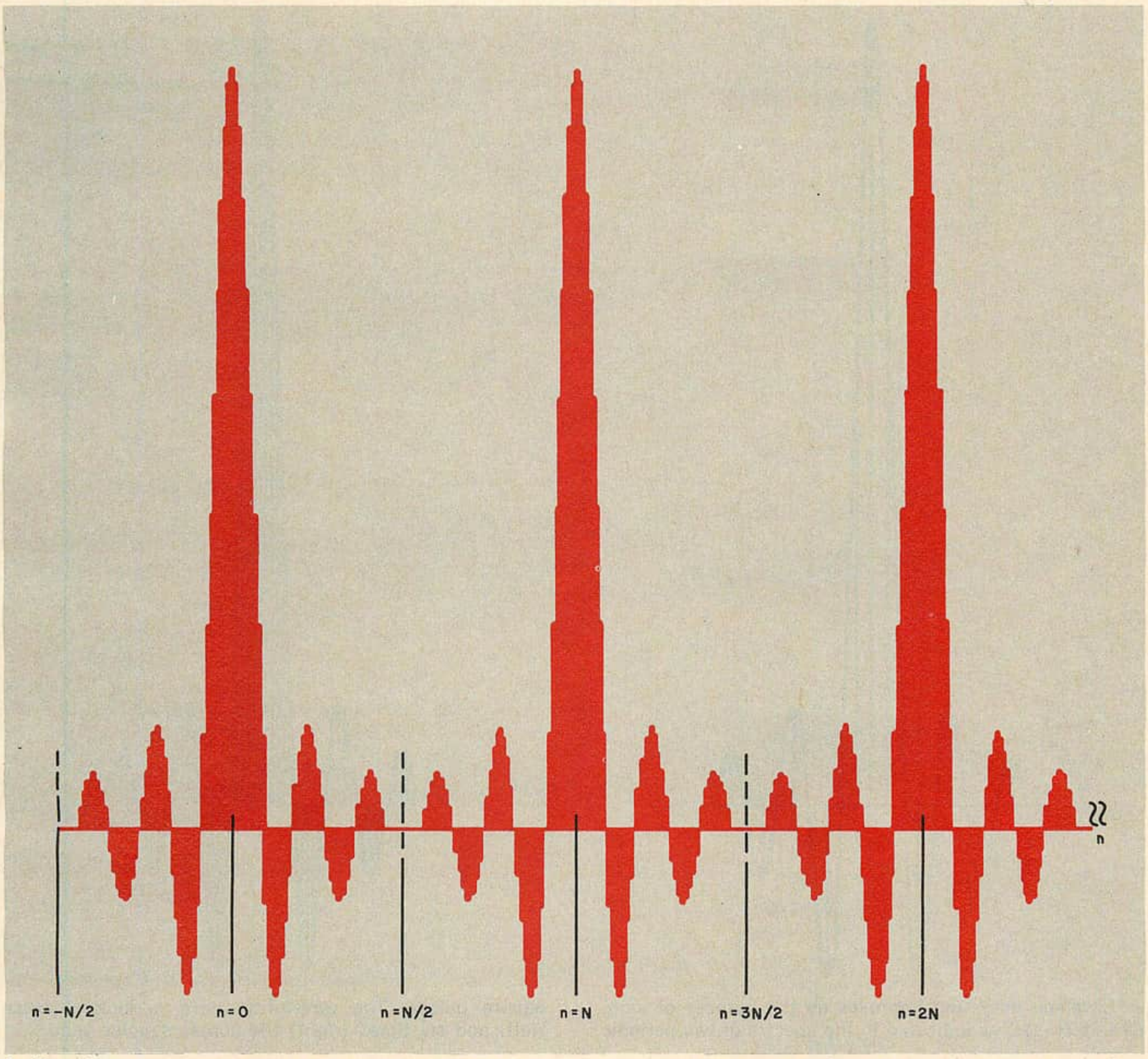
$$f(i) = \sum_{n=0}^{N-1} F(n)\, e^{jin2\pi/N}.$$

First, note that the expressions do indeed form a transform pair: If one is plugged into the other, an identity results. Second, note that it is very similar in form to the expressions for the Fourier series and integral. Thus, by making certain assumptions about summing instead of integrating and by juggling the limits, we can substitute the discrete transform pair (Eqs. 3) for either Eqs. 1 or Eqs. 2.

When we turn to periodic time functions, we often want to solve equations like this with a computer:

$$F(n) = (1/T)\int_{-T/2}^{T/2} f(t)\, e^{jn\omega_0 t}\, dt. \qquad (4)$$

Peter K. Bice, Engineer, Hewlett-Packard, Palo Alto, Calif. 94304.



1. Waveform spectrum repeats with a "period" N the number of samples taken. Harmonics higher than N/2 are misleading and must be eliminated by making the sampling frequency higher than twice the highest harmonic frequency in the waveform. Unless this is done, the individual spectra may overlap.

The usual form of the algorithm for solving this is

$$F(n) = (1/N) \sum_{i=0}^{N-1} f(i)\, e^{jin2\pi/N}. \qquad (5)$$

Equations 4 and 5 are very similar. In fact, if we make a few simple substitutions, they can be made to be as nearly equal as desired.

First, let's replace the continuous integral with a sum and the continuous f(t) with a sampled version, $f(i\Delta t)$. This means that

$$_{(1\,period)}\int \;\rightarrow\; \Sigma_{(1\,period)}$$
$$dt \;\rightarrow\; \Delta t$$
$$t \;\rightarrow\; i\Delta t,$$

where Δt is the interval between samples of the time function f(t) and i is the number of the sample. Assuming that the entire period consists of N samples of the time function, we obtain

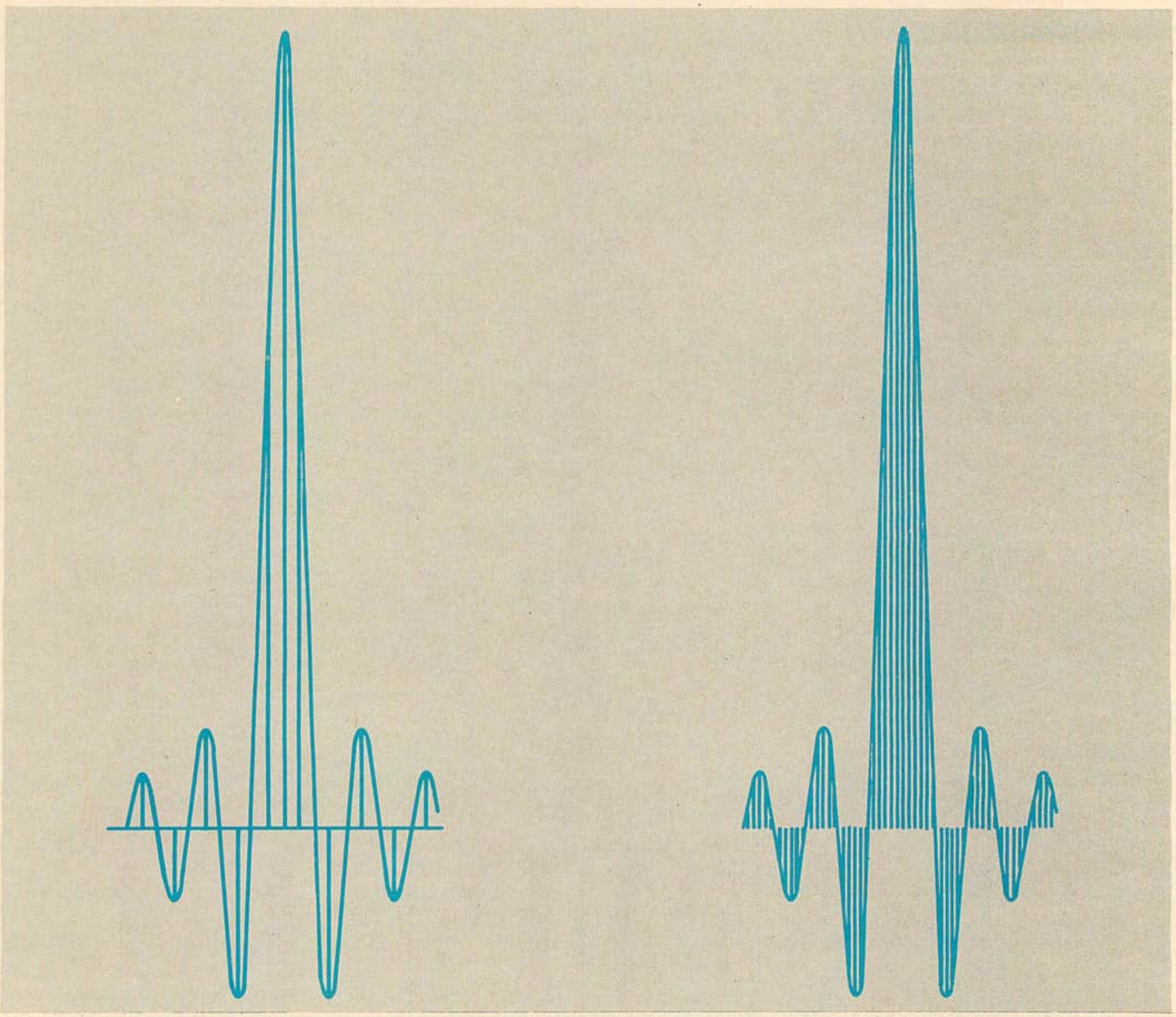
$$T \;\rightarrow\; N\Delta t.$$

If we now abbreviate, for the sake of simplicity, $f(i\Delta t)$ as f(i)—the i-th sample of the time function—and carry out all the substitutions, we get

$$F(n) = (1/N) \sum_{i=0}^{N-1} f(i)\, e^{-jin2\pi/N}, \qquad (6)$$

which is exactly the Discrete Finite Transform.

Note that we don't have to make such substitutions into the inverse transform of the Fourier series. Using the other half of the DFT pair is



2. **Envelope definition improves** as the number of samples increases, as indicated in the spectra of two periodic square pulses. The waveforms have periods of twice (left) and six times (right) the constant pulse width.

sufficient to recover the original data.

In the development of the Fourier series, F(n), for a periodic waveform, the only compromise that was made so the series could be handled by a digital computer was to substitute a sample-and-add technique for the integration. How serious is this compromise?

Accuracy depends on the sampling rate

It is obvious on the surface that it makes no difference at all if the samples are spaced very close, but it can be catastrophic if they are spaced too widely. To get a feel for an appropriate sampling interval, let's substitute n + N instead of n into Eq. 6, the definition of the DFT:

$$F(n + N) = (1/N) \sum_{i=0}^{N-1} f(i) e^{-jin\omega\pi/N} e^{-ji2\pi}$$

which reduces to the following equation:

$$F(n + N) = (1/N) \sum_{i=0}^{N-1} f(i) e^{-jin2\pi/N}, \qquad (7)$$

because $e^{-ji2\pi}$ is always unity.

Note that the right-hand sides of Eq. 6 and Eq. 7 are identical. We therefore conclude that

$$F(n + N) = F(n). \qquad (8)$$

In other words, F(n) is "periodic" (in the frequency sense) with a "period" of N. Beyond the first N values of F(n), there is no information to be gained. In fact, that information is misleading; F(N) is always the same as F(0) in a DFT.

To understand this a little better, suppose that the waveform to be analyzed has no harmonics numbered higher than N/2. Then the DFT will result in a periodic spectrum (Fig. 1), where repetitions of the spectrum do not overlap the original. In a case such as this—where the harmonics above N/2 are zero—the amplitudes of these harmonics can be accepted as accurate.

Consider the case, however, where higher harmonics are present. Picture the righthand edge of the "primary" spectrum in Fig. 1 moving to the right beyond the $n = N/2$ line. While this is happening, the lefthand edge of the "secondary" spectrum will be creeping leftward, eventually overlapping and adding to lines below the $n = N/2$ line. For this reason, higher harmonics—of even lower than N/2—are not known precisely. This problem of harmonic overlap is called "aliasing."

To cure aliasing, make sure that there are no harmonics in the input waveform that are higher than N/2, where N is the number of samples taken in the period. This requirement can be met by low-passing the input waveform or by increasing the sampling rate, thus increasing N.

Another way to define the cure for aliasing is to guarantee that the sampling frequency is at least twice the highest frequency in the input waveform (a part of Shannon's Sampling Theorem). This is, of course, the same as saying that there'll be no harmonics beyond N/2. Indeed, the fundamental frequency of the input waveform is $1/(N\Delta t)$, so that the highest allowable frequency is $1/(2\Delta t)$. But $1/\Delta t$ is simply the sampling frequency, since Δt is the sampling interval.

Calculating the Fourier integral on a computer

Now suppose that you want to calculate the Fourier integral of a function, rather than the Fourier series just described. Their equations are:

$$\text{Fourier series:}\quad F(n\omega_0) = (1/T)\int_{(1\ \text{period})} f(t)\, e^{-jn\omega_0 t}\, dt. \qquad (9)$$

$$\text{Fourier integral:}\quad F(\omega) = (1/2\pi)\int_{-\infty}^{\infty} f(t)\, e^{-j\omega t}\, dt. \qquad (10)$$

Since we know how to approximate the Fourier series on a computer, we can use similar techniques to approximate the Fourier integral.

As we let ω_0 get very small, the discrete values of $n\omega_0$ approach a continuous variable, ω. Allowing ω_0 to get small means letting the duration of the sampled waveform get very long. As this occurs, the limits on the integration approach infinity. Consequently—except for the scale factor—we can make these two equations approach each other simply by making the sampling period long.

This correlation between the Fourier series and the Fourier integral can be stated in another, more useful way. Suppose you want to find the Fourier integral of a square pulse. If you make a periodic waveform of square pulses and measure and plot the amplitudes of the harmonics against their frequencies, these harmonics will exist at discrete frequencies, and their amplitudes will lie on a $(\sin x)/x$ envelope (Fig. 2).

If another periodic waveform is now created for pulses of the same shape, but the repetition rate is made half that of the previous rate—so there'll be more "dead time" between pulses—and if the amplitude is doubled, the envelope of the resulting spectrum will be exactly that of the previous envelope. But since the fundamental frequency is half what it was, the spectrum lines will be more closely spaced and the envelope defined better.

This envelope, which gets filled better and better as we increase the length of the period, is (except for the scale factor) the Fourier integral of the waveform being analyzed. You need only decide what is the desired resolution and then make the measurements accordingly.

To find the proper scale factor, note that

$$F(0) = (1/2\pi)\int_{-\infty}^{\infty} f(t)\, dt. \qquad (11)$$

This quantity can be easily calculated and then used to scale the resulting spectrum.

Is your computer large enough?

So far the assumption has been that we have a perfect method for finding the Fourier series. If you must rely on the DFT to approximate the series, however, you must remember its inherent limitations. You must combine a long sampling period (to get good resolution of the envelope that defines the Fourier integral) with a high sampling rate (to prevent aliasing). And since the sampling rate multiplied by the duration equals the number of samples taken, you can calculate the size of the transform that can be handled. In other words, the size of the transform that can be handled depends on the computer size and the available computer time. ■■

References:

1. Cooley, J. W. and Tukey, J. W., "An Algorithm for the Machine Calculation of Complex Fourier Series," *Mathematics of Computers*, Vol. XIX, April, 1965, pp. 297-301.
2. Haavind, Robert, "The FFT Computer: Designer's Missing Link," *Electronic Design*, Dec. 6, 1967, pp. 25-29.
3. Klahn, Richard and Shively, Richard R., "FFT-Short Cut to Fourier Analysis," *Electronics*, April 15, 1968, pp. 124-129.
4. Bergland, G. D., "A Guided Tour of the Fast Fourier Transform," *IEEE Spectrum*, July, 1969, pp. 41-52.
5. Singleton, Richard C., "On Computing the Fast Fourier Transform," *Committee of the ACM;* Vol. 10, No. 10, Oct., 1967.
6. Brigham, E. O. and Morrow, R. E., "The Fast Fourier Transform," *IEEE Spectrum*, December, 1967, pp. 63-70.
7. Bice, P. K., "Speed Up the Fast Fourier Transform," *Electronic Design*, April 26, 1970, pp. 66-69.