Editor: Denis Donnelly, donnelly@siena.edu

The Fast Fourier Transform for Experimentalists, Part I: Concepts

By Denis Donnelly and Bert Rust

THE DISCRETE FOURIER TRANSFORM (DFT) PROVIDES A MEANS FOR TRANSFORMING DATA SAMPLED IN THE TIME DOMAIN TO AN EXPRESSION OF THIS DATA IN THE FREQUENCY domain. The inverse transform reverses the process, converting frequency data into time-domain data. Such transformations can be applied in a wide variety of fields, from geophysics to astronomy, from the analysis of sound signals to CO_2 concentrations in the atmosphere. Over the course of three articles, our goal is to provide a convenient summary that the experimental practitioner will find useful. In the first two parts of this article, we'll discuss concepts associated with the fast Fourier transform (FFT), an implementation of the DFT. In the third part, we'll analyze two applications: a bat chirp and atmospheric sea-level pressure differences in the Pacific Ocean.

The FFT provides an efficient algorithm for implementing the DFT and, as such, we'll focus on it. This transform is easily executed; indeed, almost every available mathematical software package includes it as a built-in function. Some books are devoted solely to the FFT,[1–3] while others on signal processing,[4–6] time series,[7,8] or numerical methods[9,10] include major sections on Fourier analysis and the FFT. We draw together here some of the basic elements that users need to apply and interpret the FFT and its inverse (IFFT). We will avoid descriptions of the Fourier matrix, which lies at the heart of the DFT process,[11] and the parsing of the Cooley-Tukey algorithm[12] (or any of several other comparable algorithms), which provides a means for transforming the discrete into the fast Fourier transform.

The Cooley-Tukey algorithm makes the FFT extremely useful by reducing the number of computations from something on the order of n^2 to $n \log(n)$, which obviously provides an enormous reduction in computation time. It's so useful, in fact, that the FFT made *Computing in Science & Engineering*'s list of the top 10 algorithms in an article that noted the algorithm is, "perhaps, the most ubiquitous algorithm in use today."[13] The interlaced decomposition method used in the Cooley-Tukey algorithm can be applied to other orthogonal transformations such as the Hadamard, Hartley, and Haar. However, in this article, we concentrate on the FFT's application and interpretation.

Fundamental Elements

As a rule, data to be transformed consists of N uniformly spaced points $x_j = x(t_j)$, where $N = 2^n$ with n an integer, and $t_j = j \cdot \Delta t$ where j ranges from 0 to $N - 1$. (Some FFT implementations don't require that N be a power of 2. This number of points is, however, optimal for the algorithm's execution speed.) Even though any given data set is unlikely to have the number of its data points precisely equal to 2^n, zero padding (which we describe in more detail in the next section) provides a means to achieve this number of samples without losing information. As an additional restriction, we limit our discussions to real valued time series as most data streams are real. When the time-domain data are real, the values of the amplitude or power spectra at any negative frequency are the same as those at the corresponding positive frequency. Thus, if the time series is real, one half of the 2^n frequencies contain all the frequency information. In typical representations, the frequency domain contains $N/2 + 1$ samples.

The FFT's kernel is a sum of complex exponentials. Associated with this process are conventions for normalization, sign, and range. Here, we present what we consider to be good practice, but our choices are not universal. Users should always check the conventions of their particular software choice so they can properly interpret the computed transforms and related spectra.

Equation 1 shows some simple relationships between parameters such as Δt, the sampling time interval; Δf, the spacing in the frequency domain; N, the number of samples in the time domain; and f_j, the Fourier frequencies. The number of samples per cycle (spc) for a particular frequency component with period T in the time domain and (in some cases) the total number of cycles (nc) in the data record for a particular frequency component are two other pieces of information that are useful because they remind us of the adequacy of the sampling rate or the data sample. Some re-

lations between these parameters are

$$\Delta f = \frac{1}{N \cdot \Delta t} \text{ and } \quad f_j = j \cdot \Delta f,$$

where $j = 0, ..., N/2$

$$\text{spc} = \frac{T}{\Delta t}, \text{nc} = \frac{N}{\text{spc}} = \frac{1}{T \cdot \Delta f} = \frac{f}{\Delta f} \quad . \tag{1}$$

The period T represents only one frequency, but, as we discuss later, there must be more than 2 spc for the highest frequency component of the sampled signal. This bandwidth-limiting frequency is called the *Nyquist frequency* and is equal to half the sampling frequency. The spacing in the frequency domain Δf is the inverse of the total time sampled, so time and frequency resolution can't both be simultaneously improved. Thus, the maximum frequency represented is $\Delta f \cdot N/2 = 1/(2 \cdot \Delta t)$, or the Nyquist frequency.

We can express the transform in several ways. A commonly used form is the following (with $i = \sqrt{-1}$):

$$X_k = \sum_{j=0}^{N-1} x_j \exp\left(-2\pi i \frac{j}{N} k\right), \; k = -N/2, \ldots, -1, 0, 1, \ldots, N/2 - 1, \tag{2}$$

where x_j represents the time-domain data and X_k their representation in the frequency domain.

We express the IFFT as

$$x_j = \frac{1}{N} \sum_{k=-N/2}^{N/2-1} X_k \cdot \exp\left(2\pi i \frac{k}{N} j\right), j = 0, 1, \ldots, N-1. \tag{3}$$

The FFT replicates periodically on the frequency axis with a period of $1/\Delta t$; consequently, $X(f_{N/2}) = X(f_{-N/2})$ so that the transform is defined at both ends of the closed interval from $-1/(2\Delta t)$ to $+ 1/(2\Delta t)$. This interval is sometimes called the *Nyquist band*.

Some FFT and IFFT implementations use different normalizations or sign conventions. For example, some implementations place the factor $1/N$ in the FFT conversion rather than with the IFFT. Some place $1/\sqrt{N}$ in both conversion processes, and some reverse the signs in the exponentials of the transforms; this sign change reverses the sign of the phase component. Moreover, some implementations take the range for k from 0, …, $N/2$.

Because Equations 2 and 3 represent the frequency and time domains of the same signal, the energy in the two cases must be the same. Parseval's relation expresses this equality. For real data, we can express the relation as

$$\sum_{i=0}^{N-1} x_i^2 = \frac{1}{N}\left(|X_0|^2 + 2 \cdot \sum_{j=1}^{N/2-1} |X_j|^2 + |X_{N/2}|^2\right), \tag{4}$$

where $X = \text{fft}(x)$. The last term on the right-hand side is not usually separated from the sum as it is here; we do this because there should be only N terms to consider in both summations, not N in one and $N + 1$ in the other. Recall that because we're dealing with real valued data, we can exploit a symmetry and present the frequency data only from 0 to $N/2$; this symmetry is the source of the factor of two associated with the summation. Unlike the other terms, the $+N/2$ frequency value isn't independent and was assigned, as noted earlier, to the value at $-N/2$. Should the $+N/2$ term be included in the sum, we would, in effect, double count the term, so we pull the $N/2$ term from the sum to avoid this. Of course, if N is large, this difference is likely to be minimal.

There are two common ways to display an FFT. One is the amplitude spectrum, which presents the magnitudes of the FFT's complex values as a function of frequency:

$$A_k = \frac{2}{N}|X_k|, \; k = -N/2, \ldots, -1, 0, 1, \ldots, N/2. \tag{5}$$

Given the symmetry of real time series, the standard presentation restricts the range of k to positive values: $k = 0, 1, \ldots, N/2$. An equally common way to represent the transform is with a power spectrum (or periodogram), which is defined as

$$P_k = \frac{1}{N} \cdot |X_k|^2, \quad k = 0, 1, \ldots, N/2. \tag{6}$$

However, neither of these spectral representations is universal. For example, some conventions place a 1 in the numerator instead of a 2 for the amplitude spectrum. The periodogram is sometimes represented with a factor of 2 in the numerator instead of 1 or as the individual terms expressed in Parseval's relation (Equation 4).

In Figure 1, as an example of the FFT process, we show the amplitude spectrum of a single-frequency sine wave with two different sampling intervals. In one case, the interval Δt is chosen to make nc integral, and in the other, nonintegral. If nc is integral, f is necessarily a multiple of Δf, and one point of the transform is associated with the true frequency (see the circles in Figure 1a). However, in any FFT application, we're dealing with a finite-length time series. The process of restricting the data in the time domain (multiplying the data by one over the range where we wish to keep

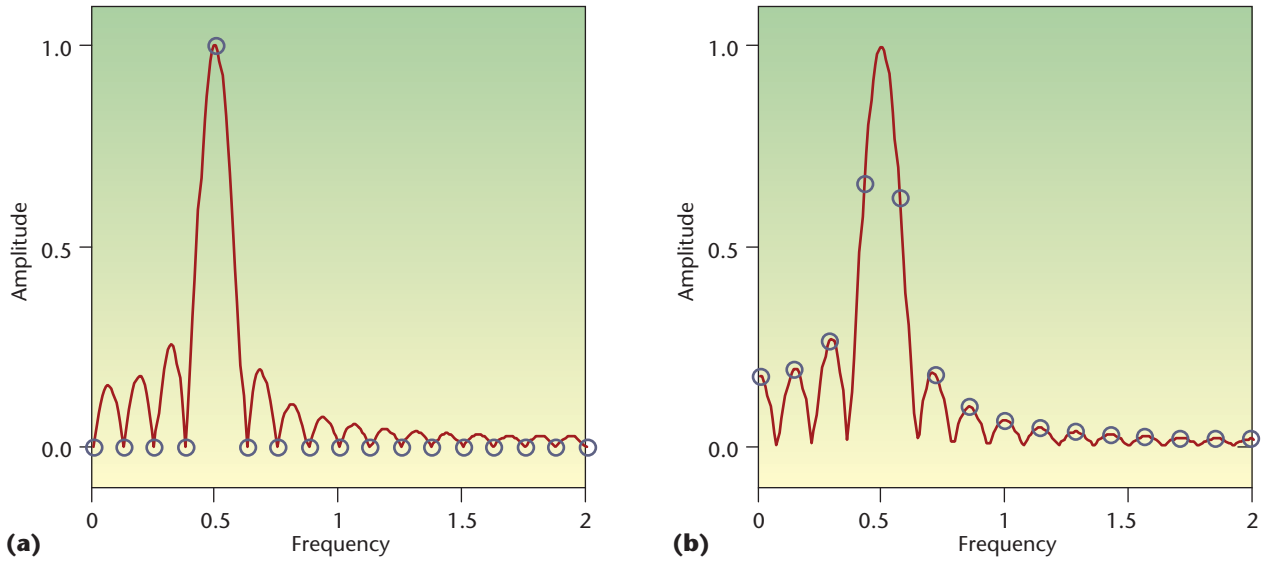

Figure 1. Amplitude spectra of a single-frequency sine wave. Two representations of a sine wave of frequency 0.5 are shown in each part of the figure. In each case, the circles are based on a time series where the number of sample points $N = 32$ but the time step is slightly different: (a) $N\Delta t = 8$, so nc = 4; (b) $N\Delta t = 7.04$, so nc = 3.52, where nc is the total number of cycles. The solid lines provide a view of these same spectra with zero padding. This form is closer to what would be expected from a continuous rather than a discrete Fourier transform. The zero-padded examples reveal detail that might not have been expected, given the appearance of the unpadded case.

the data and multiplying by zero elsewhere—an example of windowing, discussed later) introduces sidelobes in the frequency domain. These sidelobes are called leakage.

Even though there's leakage, because there's only one frequency associated with the transformed sine wave, we might expect to be able to estimate that frequency with a weighted average of all the points in the frequency domain. Such an average, however, wouldn't yield the correct frequency.

In general, the FFT process generates complex values in the frequency domain from the real values in the time domain. If we transform sine or cosine waves where we consider an integral number of cycles, the transform magnitudes are identical. However, in the frequency domain, a sine curve is represented only with imaginary values and a cosine curve only with real values. When the number of cycles is nonintegral or if there is a phase shift, then both real and imaginary parts appear in the transform of both the sine and cosine.

Zero Padding

Zero padding is a commonly used technique associated with FFTs. Two frequent uses are to make the number of data points in the time-domain sample a power of two and to improve interpolation in the transformed domain (for example, zero pad in the time domain, improve interpolation in the frequency domain).

Zero padding, as the name implies, means appending a string of zeros to the data. It doesn't make any difference if the zeros are appended at the end (the typical procedure), at the beginning, or split between the beginning and end of the data set's time domain. One very common use of this process is to extend time-series data so that the number of samples becomes a power of two, making the conversion process more efficient or, with some software, simply possible. Because the spacing of data in the frequency domain is inversely proportional to the number of samples in the time domain, by increasing the number of samples—even if their values are zero—the resulting frequency spectrum will contain more data points for the same frequency range. Consequently, the zero-padded transform contains more data points than the unpadded; as a result, the overall process acts as a frequency interpolating function. The resulting, more detailed picture in the frequency space might indicate unexpected detail (see, for example, Figure 2). As the number of zeros increases, the FFT better represents the time series' continuous Fourier transform (CFT).

As we noted earlier, zero padding introduces more points into the same frequency range and provides interpolation between points associated with the unpadded case. When data points are more closely spaced, clearly, there's a possibility that unnoticed detail could be revealed (such as Figure 1a shows). In Figure 2, we see the effect of quadrupling the number of points for two different cases. The transforms of the zero-padded data contain the same information as the unpadded data, and every fourth point of the padded data matches the corresponding unpadded data point. The intermediate points provide interpolation.

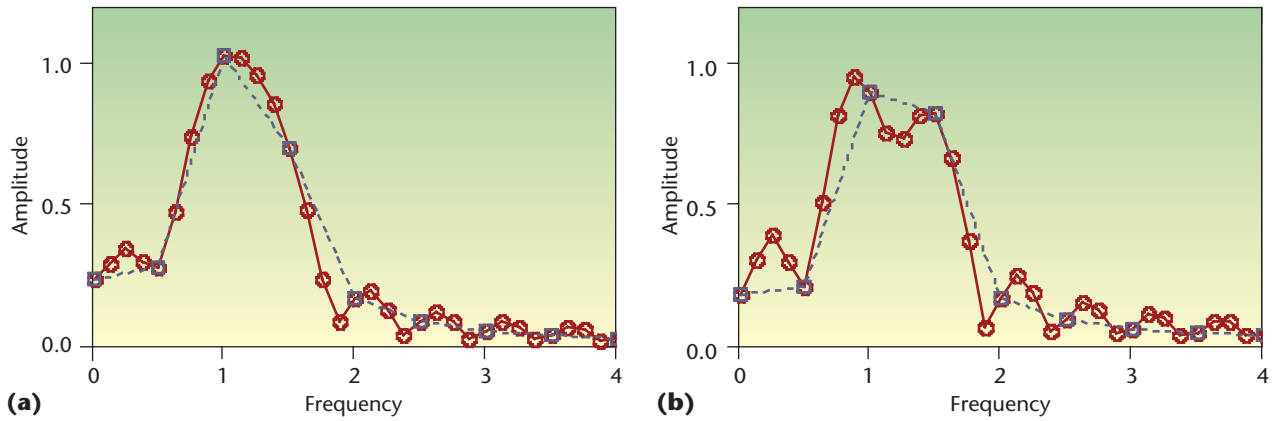

Figure 2. The effect of zero padding on the transform of a signal containing two different frequencies. We look at two cases: one in which the two frequencies are too close to be clearly resolved, and one in which resolution is possible. (a) Fast Fourier transforms (FFTs) of the sum of two sine waves of amplitude 1 and frequencies of 1 and 1.3 Hz; the frequencies aren't resolved, and (b) FFTs of the sum of two sine waves of amplitude 1 and frequencies of 1 and 1.35 Hz; the frequencies are resolved. The solid curves are transforms of zero-padded data and include four times as many samples as the transforms of the unpadded data (dotted curves). Because the zero-padded curve has four times as many data points as the unpadded case ($N = 32$), every fourth point of the zero-padded data is the same as the unpadded data. Zero-padded results provide better interpolation and more detail.

In Figure 2, we see an application of that interpolating ability when we consider a signal consisting of two closely lying frequencies. In Figure 2a, although the envelope is more clearly drawn, zero padding does not have the power to resolve the two frequencies associated with this case. In Figure 2b, the peaks are sufficiently separated so that the interpolation reveals the two peaks, whereas the unpadded data seemingly did not. This example reminds us that a graphical representation connecting adjacent data points with straight lines can be misleading.

Zero padding can also be performed in the frequency domain. The inverse transform results in an increase in the number of data points in the time domain, which could be useful in interpolating between samples (see Figure 3). Zero padding is also used in association with convolution or correlation and with filter kernels, which we discuss later in this article.

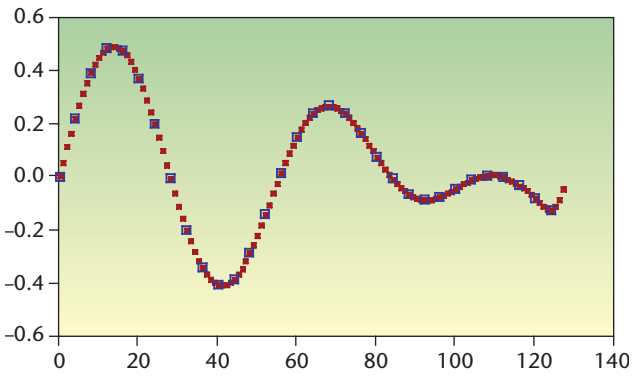

Figure 3. The effect of zero padding in the frequency domain on the time-domain data. The frequency data (the unpadded case in Figure 2a) was zero-padded to four times its original length. We show the original unpadded time-domain data (boxes) and the inverse fast Fourier transform of the zero-padded frequency data (dots). The padding process again acts as an interpolation function.

Aliasing

When performing an FFT, it's necessary to be aware of the frequency range composing the signal so that we sample the signal more than twice per cycle of the highest frequency associated with the signal. In practice, this might mean filtering the signals to block any signal components with a frequency above the Nyquist frequency $(2 \cdot \Delta t_{sample})^{-1}$ before performing a transform. If we don't restrict the signal in this way, higher frequencies will not be adequately sampled and will masquerade as lower-frequency signals. This effect is similar to what moviegoers experience when the onscreen wheels of a moving vehicle seemingly freeze or rotate in the wrong direction. The camera, which operates at the sampling rate of 24 frames per second, only has a Nyquist limit of 12 Hz; any higher frequencies present will appear as lower frequencies.

Let's assume that we can readily observe a point on a wheel (not at the center) that's rotating but not translating. At a slow rotation rate, each successive frame of our film shows the observable point advancing from the previous frame. (The fraction of a complete rotation and the sampling rate are related; the number of samples per rotation is the inverse of the fraction of a rotation per sample.) As the rotation rate increases,

Table 1. Actual and apparent angles for 170° and 190° rotations.

Angle sequence for 170° step	Angle sequence for 190° step	Apparent angle sequence for 190° steps with rotation direction reversed.*
0	0	0
170	190	170
340	20	340
150	210	150
320	40	320
130	230	130

*Magnitudes of reverse angles are given by 360° – column 2.

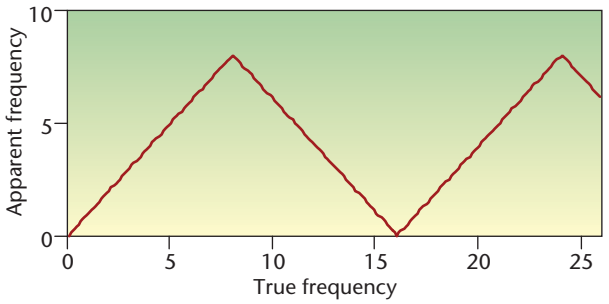

Figure 4. Apparent frequency as a function of the true frequency. Frequencies greater than the Nyquist frequency fold back into the allowed frequency range and appear as lower frequencies. In this example, where the Nyquist frequency is 8 Hz, an actual frequency of 9 Hz would appear as 7 Hz.

the angle between our observed point in successive frames increases. When the angle reaches 180 degrees, or two samples per rotation, the perceived rotation rate is at its maximum—the wheel is rotating at the Nyquist frequency.

When passing through the Nyquist limit, as the frequency goes from $f_{Ny} - \varepsilon$ to $f_{Ny} + \varepsilon$ (where $\varepsilon << f_{Ny}$), the rotation direction appears to change from forward to reverse while the rotation rate remains the same. Further increases in the rotation rate make the wheel appear to continue rotating in a reversed direction but at a decreasing rate. When the actual rotation rate is twice the Nyquist frequency, the apparent rotation rate is zero and the sampling rate is just once per rotation. (Another example of one sample per rotation and an apparent zero rotation rate is to use a stroboscope to determine an object's rotation rate. With one flash per rotation, the rotating object appears at rest and the flash rate and rotation rate are equal.) If the frequency of rotation continues to increase, the wheel will again appear to rotate in the original rotation direction.

To make this more concrete, consider two constant rotation rates, one of 170 degrees between successive frames/samples and one of 190 degrees. We observe only the current position in each frame, so as we compute a value sequence, we take them mod(360). If we compute values for the 170-degree case, we obtain 0, 170, 340, 150, 320, 130, and so on. If we compute values for the 190-degree case, we get 0, 190, 20, 210, 40, 230, and so on, but we wouldn't see the 190-degree rotation. We don't observe an increase greater than 180 degrees (for angles greater than that, the data is undersampled). For the 190-degree case, we would see a 170-degree step, but with the rotation in the opposite direction.

To consider a reverse rotation, we subtract the forward rotation angle from 360. The result is the magnitude of the angle of rotation in the reverse direction. For example, a forward rotation angle of 350 degrees is equivalent to a 10-degree step in the reverse direction. So for our 190-degree case, the numbers become 0, 360 – 190 = 170, 360 – 20 = 340, 360 – 210 = 150, and so on. Table 1 provides a summary. The magnitudes of these rotation angles are identical to the 170-degree data. Thus, we would see the 190-degree case as equivalent to the 170-degree case in terms of rotation rate, but with the rotation direction reversed. The graph in Figure 4 helps demonstrate this kind of behavior.

In the example shown in Figure 4, the Nyquist frequency is 8 Hz. Frequencies associated with the first leg of the sawtooth curve have more than two samples per cycle, and the apparent and actual frequencies are equal. Once the actual frequency exceeds the Nyquist frequency, the apparent frequency begins to decrease, with the negative slope corresponding to a reversed rotation direction. At 16 Hz, with one sample per rotation, the apparent frequency is zero. With further increases in the true frequency, the apparent frequency once again increases.

If we take the FFT of three amplitude 1 cosine waves having frequencies of 3.5, 12.5, and 19.5 Hz and where we set $N = 16$ and $\Delta t = 1/N$ (so the Nyquist frequency is 8 Hz), we get identical FFTs, one of which is shown in Figure 5. The number of samples per cycle for these frequencies is 4.57, 1.28, and 0.82. Only the lowest frequency is adequately represented; the two higher-frequency cases have fewer than

two samples per cycle and consequently masquerade as lower frequencies, appearing in the allowed range between 0 Hz and the Nyquist frequency. For the example with the three different frequencies, we purposely selected the higher frequencies so that their FFTs would be identical to that of the lowest frequency. Referring to Figure 4, we note that the frequencies 12.5 and 19.5 Hz would appear on the second and third legs of the sawtooth curve. The apparent frequency of the 12.5-Hz line is 8 – (12.5 – 8); the apparent frequency of the 19.5-line is 19.5 – 2 · 8. In general, the out-of-range frequency f_{true} would appear as $f_{apparent}$ as given by

$$f_{apparent} = \left| f_{true} - k \cdot \left(2 \cdot f_{Nyquist} \right) \right| = \left| f_{true} - \frac{k}{\Delta t} \right|, \qquad (7)$$

where $k = 1, 2, \ldots$, and k is selected to bring $f_{apparent}$ within the range $0 \ldots f_{Ny}$.

In Figure 6, we see the actual curves that correspond to the three frequencies and the points where sampling occurs. If we performed an FFT followed by an IFFT for any one of the three curves (given the sampling specified), the algorithm would return the same result in each case, which, without other information, would be interpreted as the lowest-frequency case.

If the magnitudes of the Fourier coefficients approach zero (roughly as $1/f$) as the frequency approaches the Nyquist frequency (a zero between lobes would not qualify), then there is a good likelihood that aliasing has not occurred. If it isn't zero, we can consider the possibility that it has occurred. However, a nonzero value doesn't imply that aliasing has necessarily happened. The Fourier coefficients in Figure 5 don't go to zero even in the adequately sampled case. Zero padding of this example will show a great deal more detail, but the transform is still nonzero at the Nyquist frequency.

Relation to Fourier Series

There is a direct connection between the real and imaginary parts of the frequency information from an FFT and the coefficients in a Fourier series that would represent the corresponding time-domain signal. As we noted earlier, for the conditions stated, the transform of a single-frequency sine wave is imaginary, whereas the transform of a single-frequency cosine wave is real. So, in a Fourier series of the time-domain signal, we would expect the real parts of the frequency information to be associated with cosine series and the imaginary parts with sine series. This is, in fact, the case.

An equation for recreating the original signal as a Fourier

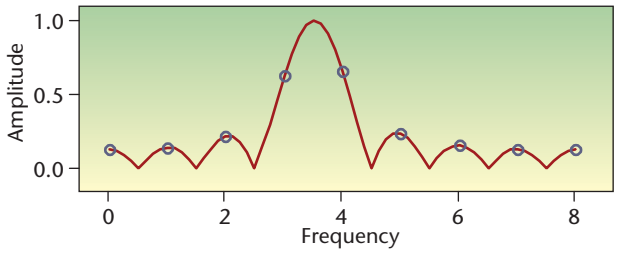

Figure 5. The FFT of a 3.5 Hz, amplitude one cosine wave where $N = 16$ and $\Delta t = 1/N$ (represented by circles). The FFTs of the frequencies 3.5 Hz, 12.5 Hz, and 19.5 Hz are identical for the case when the Nyquist frequency is 8 Hz. The solid curve shows the transform with zero padding.

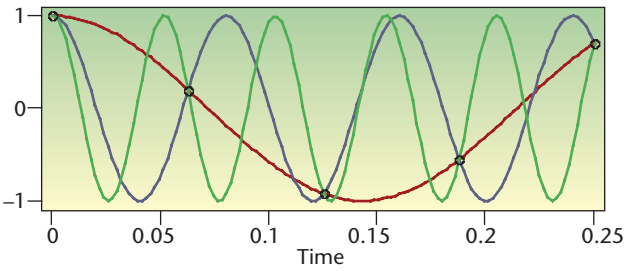

Figure 6. A view of the sampling of three cosine curves. Cosine curves with frequencies 3.5 Hz, 12.5 Hz, and 19.5 Hz are shown, with the marked points representing those at which sampling occurs ($\Delta t = 1/N$ and N=16). Only the lowest-frequency curve is adequately sampled, with more than two samples per cycle. In this case, the FFT for each curve would indicate a signal with a frequency of 3.5 Hz. For clarity, we show only the first five samples.

series from the frequency information is

$$S(t) = \left[\frac{a_0}{2} + \sum_{k=1}^{nt} \left[\left(a_k \cos\left(2\pi k \Delta f t\right) \right) + b_k \sin\left(2\pi k \Delta f t\right) \right] \right] \cdot \frac{2}{N}. \qquad (8)$$

For the case $N = 2^n$, a_k represents the real part of the transformed signal, b_k the imaginary part, nt the number of terms to be included in the series (where $nt < N/2$), and Δf the spacing in the frequency domain.

An alternate form in terms of magnitude and phase is also possible. Given that

$$\varphi_j = \tan^{-1}\left(\frac{Im(h_j)}{\mathrm{Re}(h_j)} \right), \qquad (9)$$

where $h_k = a_k + ib_k$ and the H_j are the magnitudes of h_j, the series is given by

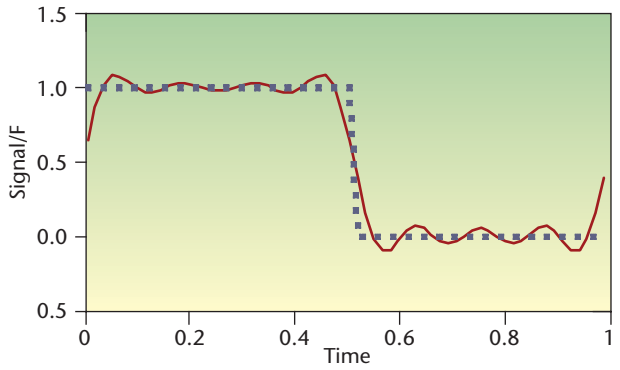

Figure 7. A comparison of the original time-domain signal and its partial reconstruction as a Fourier series. The original signal (dotted curve) and the first 10 terms of a Fourier series (solid curve) computed using coefficients from the original signal's FFT.

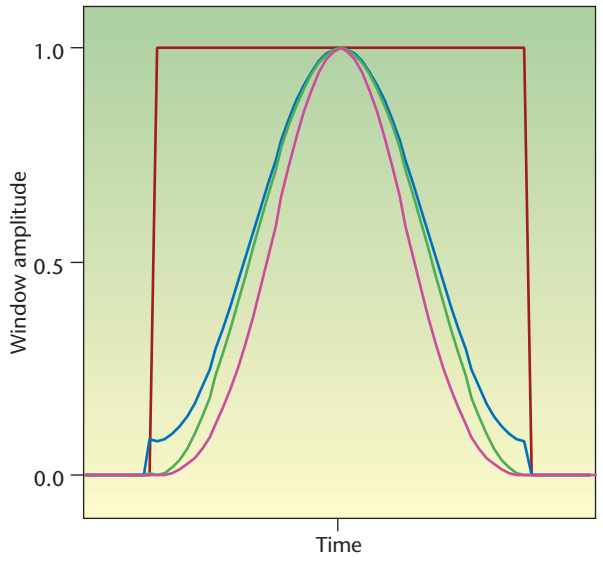

Figure 8. The shapes of four different windows. From the side, we see a rectangular (red), Hamming (blue), Hann (green), and Blackman (magenta), respectively. We'll apply three of these windows to a sine wave sequence in Figure 9.

$$S(t) = \left[\frac{a_0}{2} + \sum_{k=1}^{nt}\left(H_k \cos\left(2\pi k \Delta f t - \varphi_k\right)\right)\right] \cdot \frac{2}{N} \cdot \qquad (10)$$

In Figure 7, we see the square wave signal (one cycle of a square wave that ranges between 0 and 1 with equal times high and low) to be transformed as well as the signal constructed from the first 10 terms of a Fourier series using the coefficients from the FFT as per Equation 8. We would obtain an identical waveform if we took the IFFT of a truncation of the original FFT, where all the FFT's coefficients with an index greater than the number of desired terms (here, $nt = 10$) are set to zero.

Windows

Windows are useful for extracting and/or smoothing data. A window is typically a positive, smooth symmetric function that has a value of one at its maximum and approaches zero at the extremes. (A window might have a discontinuity in its first derivative, giving it an inverted V shape—such a window is sometimes referred to as a "tent"—or two discontinuities for a rectangular or trapezoidal shape.) We apply windows by multiplying time-domain data by the window function. Of course, whenever a window is applied, it alters at least some of the data.

Smoothing windows, for example, reduce the amplitude of the time-domain data at both the beginning and the end of the windowed data set. One effect of this smoothing is to reduce leakage in the frequency domain. In Figure 8, we show comparative plots of four frequently used windows. We show the effect of applying three of those windows to a sine wave sequence in Figure 9.

Let's look at the expressions for four common windows:

Rectangular: $$recw_i = \begin{cases} 1 & (inside) \\ 0 & (outside) \end{cases}$$

Hamming: $hamw_i = 0.54 - 0.46 \cdot \cos(2 \cdot \pi \cdot i/N)$

Hann: $hanw_i = 0.5 - 0.5 \cdot \cos(2 \cdot \pi \cdot i/N)$

Blackman: $blkw_i = 0.42 - 0.5 \cdot \cos(2 \cdot \pi \cdot i/N) + 0.08 \cdot \cos(4 \cdot \pi \cdot i/N).$ (11)

The Hamming and Hann windows differ in only one parameter: if the corresponding coefficients are written $\alpha - (1 - \alpha)$, then α is 0.54 for the Hamming window and 0.5 for the Hann. The fact that a slight change in the parameter value gives rise to two different windows hints at the sensitivity of the windowing process to the value of α. If α decreases from 0.5, the side lobes increase significantly in amplitude. As α increases from 0.5 to 0.54, the relative sizes of the side lobes change. The first set of the Hann side lobes tend to be significantly larger than those of the Hamming case, but subsequent Hann side lobes decrease rapidly in magnitude and become significantly smaller than the Hamming side lobes. As to general appearance, the Hamming window doesn't quite go to zero at the window's endpoints whereas the Rectangular, Hann, and Blackman windows do. Several other windows also exist, including Bartlett (tent function), Welch (parabolic), Parzen (piece-wise cubic), Lanczos (central lobe

of a sine function), Gaussian, and Kaiser (which uses a modified Bessel function).

Each of these windows has particular characteristics. Two particularly useful points of comparison in the frequency space are the full width at half maximum of the central peak and the relative magnitude of central peak to that of the side lobes. An unwindowed signal's FFT has the narrowest central peak, but it also has considerable leakage that decays slowly. The curves for the Hamming and Blackman cases show wider central peaks but significantly smaller side lobes. The Blackman window has the largest peak height to first sidelobe height ratio.

There is no final summary statement that says you should use window x in all cases—circumstances decide that. In the bat-chirp analysis we'll examine in part two of this series, we'll use an isosceles trapezoidal window. Such a window isn't generally recommended, but for the bat-chirp case, it's the best choice. (A split cosine bell curve, a Hann window shape for the beginning and end of the curve with a magnitude of one in the interior, would give essentially the same results.)

As an example of windowing's effect on the transform, we apply a Blackman window to the time-domain data associated with Figure 1b. Two effects of applying this window, as Figure 10 shows, are that the leakage is greatly reduced and that the central peak is broadened. Obtaining the needed detail to observe these features requires zero padding.

In part two of this series, we'll discuss auto-regression spectral analysis and the maximum entropy method, convolution, and filtering. In the third and final installment, we'll present some applications, including the analysis of a bat chirp and atmospheric sea-level pressure variations in the Pacific Ocean.

Whether there is an interest in CO_2 concentrations in the atmosphere, ozone levels, sunspot numbers, variable star magnitudes, the price of pork, or financial markets, or if the interest is in filtering, correlations, or convolutions, Fourier transforms provide a very powerful and, for many, an essential algorithmic tool.

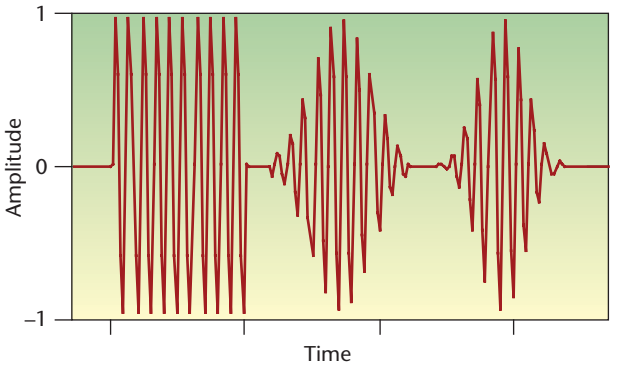

Figure 9. A comparison of the effects (from left to right) of a rectangular, a Hamming, and a Blackman window on a sine wave sequence. For convenience of display, we compute the three examples separately, shift the second and third in time, and sum the set, with the effect that the three examples appear sequentially in time; because each example is zero outside its window zone, the results do not interfere. The three windows have the same width, but as Figure 8 shows, the Blackman window increases in magnitude more slowly than the others, and we can observe the effect on the sine wave signal. The difference between Hamming and Blackman windowing is also evident.

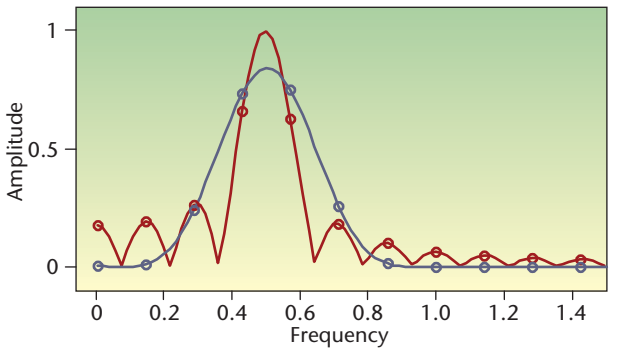

Figure 10. The effects of windowing as seen in the transform space. The FFT of the 3.52-cycle example in Figure 1 and the result of multiplying time-domain data and a Blackman window before taking the FFT are shown without zero padding (circles) and with zero padding (solid curves). The windowed form reduces leakage but has a broader central lobe.

References

1. R.N. Bracewell, *The Fourier Transform and Its Applications*, McGraw-Hill, 1965.
2. E.O. Brigham, *The Fast Fourier Transform and Its Applications*, Prentice-Hall, 1988.
3. J.F. James, *A Student's Guide to Fourier Transforms*, Cambridge Univ. Press, 1995.
4. C.T. Chen, *Digital Signal Processing*, Oxford Univ. Press, 2001.
5. S.L. Marple Jr., *Digital Spectral Analysis with Applications*, Prentice Hall, 1987.
6. S. Smith, *Digital Signal Processing*, Newnes, 2003.
7. P. Bloomfield, *Fourier Analysis of Time Series*, John Wiley & Sons, 2000.
8. P. Hertz and E.D. Feigelson, "A Sample of Astronomical Time Series," *Applications of Time Series Analysis in Astronomy and Meteorology*, T. Subba Rao, M.B. Priestley, and O. Lessi, eds., Chapman & Hall, 1979, pp. 340–356.
9. W.H. Press et al., *Numerical Recipes in Fortran*, Cambridge Univ. Press, 1992.
10. L.N. Trefethen, *Spectral Methods in Matlab*, SIAM Press, 2000.

11. C.D. Meyer, *Matrix Analysis and Applied Linear Algebra*, SIAM Press, 2000.
12. J.W. Cooley and J.W. Tukey, "An Algorithm for the Machine Calculation of Complex Fourier Series," *Mathematics of Computation*, vol. 19, no. 90, 1965, pp. 297–301.
13. D.N. Rockmore, "The FFT: An Algorithm the Whole Family Can Use," *Computing in Science & Eng.*, vol. 2, no. 1, 2000, pp. 60–64.

Denis Donnelly is a professor of physics at Siena College. His research interests include computer modeling and electronics. Donnelly received a PhD in physics from the University of Michigan. He is a member of the American Physical Society, the American Association of Physics Teachers, and the American Association for the Advancement of Science. Contact him at donnelly@siena.edu.

Bert Rust is a mathematician at the National Institute for Standards and Technology. His research interests include ill-posed problems, time-series modeling, nonlinear regression, and observational cosmology. Rust received a PhD in astronomy from the University of Illinois. He is a member of SIAM and the American Astronomical Society. Contact him at bwr@nist.gov.

Submissions: Send one PDF copy of articles and/or proposals to Norman Chonacky, Editor in Chief, cise-editor@aip.org. Submissions should not exceed 6,000 words and 15 references. All submissions are subject to editing for clarity, style, and space.

Editorial: Unless otherwise stated, bylined articles and departments, as well as product and service descriptions, reflect the author's or firm's opinion. Inclusion in *CiSE* does not necessarily constitute endorsement by the IEEE, the AIP, or the IEEE Computer Society.

Circulation: *Computing in Science & Engineering* (ISSN 1521-9615) is published bimonthly by the AIP and the IEEE Computer Society. IEEE Headquarters, Three Park Ave., 17th Floor, New York, NY 10016-5997; IEEE Computer Society Publications Office, 10662 Los Vaqueros Circle, PO Box 3014, Los Alamitos, CA 90720-1314, phone +1 714 821 8380; IEEE Computer Society Headquarters, 1730 Massachusetts Ave. NW, Washington, DC 20036-1903; AIP Circulation and Fulfillment Department, 1NO1, 2 Huntington Quadrangle, Melville, NY 11747-4502. Annual subscription rates for 2005: $42 for Computer Society members (print only) and $42 for AIP society members (print plus online). For more information on other subscription prices, see www.computer.org/subscribe/ or https://www.aip.org/forms/journal_catalog/order_form_fs.html. Computer Society back issues cost $20 for members, $96 for nonmembers; AIP back issues cost $22 for members.

Postmaster: Send undelivered copies and address changes to *Computing in Science & Engineering*, 445 Hoes Ln., Piscataway, NJ 08855. Periodicals postage paid at New York, NY, and at additional mailing offices. Canadian GST #125634188. Canada Post Corporation (Canadian distribution) publications mail agreement number 40013885. Return undeliverable Canadian addresses to PO Box 122, Niagara Falls, ON L2E 6S8 Canada. Printed in the USA.

Editor: Denis Donnelly, donnelly@siena.edu

THE FAST FOURIER TRANSFORM FOR EXPERIMENTALISTS, PART II: CONVOLUTIONS

By Denis Donnelly and Bert Rust

WHEN UNDERGRADUATE STUDENTS FIRST COMPUTE A FAST FOURIER TRANSFORM (FFT), THEIR INITIAL IMPRESSION IS OFTEN A BIT MISLEADING. THE PROCESS ALL seems so simple and transparent: the software takes care of the computations, and it's easy to create the plots. But once they start probing, students quickly learn that like any rich scientific expression, the implications, the range of applicability, and the associated multilevel understandings needed to fully appreciate the subtleties involved take them far beyond the basics. Even professionals find surprises when performing such computations, becoming aware of details that they might not have fully appreciated until they asked more sophisticated questions.

In the first of this five-part series,[1] we discussed several basic properties of the FFT. In addition to some fundamental elements, we treated zero-padding, aliasing, and the relationship to a Fourier series, and ended with an introduction to windowing. In this article, we'll briefly look at the convolution process.

Convolution

Convolution, a process some would say lies at the heart of digital signal processing, involves two functions, which we'll call $x(t)$ and $h(t)$, where $x(t)$, for example, could be an input signal and $h(t)$ some linear system's impulse response. When convolved, $\otimes$, they yield an output function $y(t)$. The process expresses the amount of one function's overlap as it is shifted over the other, providing a kind of blending of the two functions:

$$y(t) = x(t) \otimes h(t). \tag{1}$$

This process has many applications. Filtering is one example: given the appropriate impulse response, we can create any one of a number of filters. We'll give some examples in the next section, but we'll postpone further information about filtering and detrending until the next installment. Correlation is another closely related process and can help determine if a particular signal occurs in another datastream.

Deconvolution is the reverse: in effect, it uses the process itself to remove the effects of an undesired convolution or data distortion. When taking data, a convolution can obscure the desired information, perhaps due to interfering physical interactions or by the detection system itself (which has its own response). A gamma ray arriving at a detector, for example, has a well-defined energy, yet the detector output shows several associated effects related to the interaction of the gamma ray with a crystal. If a nuclear physicist is interested in the gamma ray's energy or intensity instead of the detector's response, then he or she needs to know how to extract the appropriate information from this much larger signal set. Deconvolving can remove the detector response, restoring the data to a form closer to the original.

When noise accompanies a signal, as it always does to some extent, a direct deconvolution can generate unstable results, which renders the process unusable. One way to reduce the noise's influence is to assume that analytic functions can represent either (or both) the original signal and the convoluted signal. When such a representation is possible, the chances of success with the deconvolution process greatly improve. Still, deconvolution is beyond the scope of this series, so we won't discuss it here.

The continuous convolution is defined as

$$y(t) = x(t) \otimes h(t) = \int_{-\infty}^{\infty} x(\tau)h(t-\tau)d\tau = \int_{-\infty}^{\infty} x(t-\tau)h(\tau)d\tau. \tag{2}$$

In his book on the FFT, E. Oran Brigham states that "Possibly the most important and powerful tool in modern scientific analysis is the relationship between [Equation 2] and its Fourier transform."[2] The relationship referred to is the time–convolution theorem:

$$\mathcal{F}\{x(t) \otimes h(t)\} = \mathcal{F}\{x(t)\} \times \mathcal{F}\{h(t)\} = X(f)H(f), \tag{3}$$

where × denotes ordinary multiplication, and $X(f)$ and $H(f)$ are the continuous Fourier transforms of $x(t)$ and $h(t)$.

In real life, we seldom have access to the functions $x(t)$ and $h(t)$; instead, we have only finite time-series representations, such as

$$x_k = x(k \cdot \Delta t)$$

and

$$h_k = h(k \cdot \Delta t),\ k = 0, 1, 2, \ldots, N-1. \quad (4)$$

Given this discrete representation, we can't compute $y(t)$ exactly, but we can compute a time-series approximation to it. Specifically, we can write an expression for the discrete convolution as

$$y_n = \Delta t \sum_{k=1}^{N-1} x_k \cdot h_{n-k} = \Delta t \sum_{k=1}^{N-1} x_{n-k} \cdot h_k$$
$$n = 0, 1, 2, \ldots, N-1. \quad (5)$$

If the response function were the trivial example in which h_0 has the value 1 and all other h values are 0, then the convolution process would just reproduce the input signal (if h_0 differed from 1, it would scale the input signal proportionally to h_0). If all h's were 0 except for h_m, then we would scale the input signal by the magnitude of h_m and delay it by m sample intervals. The convolution process is the summation of such elements.

It's important to keep two details in mind when performing a convolution process: one, the two signals must have the same number of elements (zero-padding easily solves this problem), and two, the discrete convolution theorem treats the data as if it were periodic. We can express the summation associated with this *circular convolution* as

$$\sum_{k=0}^{N-1} x\left[(n-k) \bmod N\right] h(k)\ . \quad (6)$$

This cyclic effect causes a wraparound problem that we'll explain in more detail later.

The FFT form of the convolution of two time series is given by

$$x \otimes h = \text{ifft}(\text{fft}(x) \times \text{fft}(h)), \quad (7)$$

where the product of the two transforms is element by element and ifft stands for inverse FFT. (While we're discussing convolution in the time domain and multiplication in the frequency domain, we should mention that an interchange of roles is also possible. Multiplication in the time domain corresponds to convolution in the frequency domain.)

We can readily program the summation required to compute a convolution: as the number of data points increases, the computational advantage goes to the convolution's implementation with FFT, even though it requires several steps. The reason is that a convolution in the time domain requires N^2 multiplications whereas the computational cost of taking the FFT route is on the order of $3N \log_2(N)$ multiplications. Despite the fact that three steps are involved, for large N, the advantages of the FFT approach are unmistakable. Even for the very modest case of $N = 250$, using FFTs to compute a convolution is already more than 10 times faster than the time-domain computation.

One way to implement the summation shown in Equation 6 is by expressing the equation itself in matrix form. Create an $N \times N$ matrix in which the first column takes on the x-values from x_0 to x_{N-1}. Let the next column take on the same x-values but shifted down one row, with the last value becoming the first, and repeat this rolling procedure for each successive column. Multiplying this x-matrix by the h-vector yields a circular convolution. We get a *linear convolution* from this same multiplication if we set all the terms in the x-matrix above the diagonal to zero.

To avoid the wraparound pitfall, we could do one of two things: compute the linear convolution (setting all elements above the x-matrix's diagonal to zero) or zero-pad the functions so that the total number of data points is at least $N_0 + K_0 - 1$, where N_0 and K_0 are the original numbers of data points in the functions x and h. With this number of elements, we avoid any distortion due to wraparound:

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_{N-1} \end{bmatrix} = \begin{bmatrix} x_0 & x_{N-1} & x_{N-2} & \cdots & x_1 \\ x_1 & x_0 & x_{N-1} & \cdots & x_2 \\ x_2 & x_1 & x_0 & \cdots & x_3 \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ x_{N-1} & x_{N-2} & x_{N-3} & \cdots & x_0 \end{bmatrix} \cdot \begin{bmatrix} h_0 \\ h_1 \\ h_2 \\ \vdots \\ h_{N-1} \end{bmatrix}. \quad (8)$$

Examples

As an example of a linear convolution calculation, consider the signals

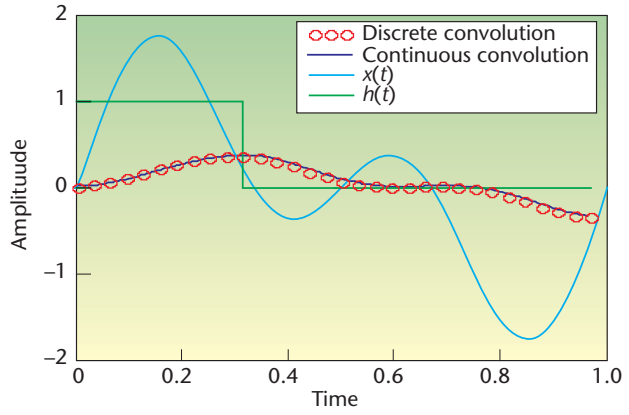

Figure 1. Comparison of continuous and discrete convolution calculations. We calculated the convolution of $x(t)$ and $h(t)$ in three ways: continuous and discrete in the time and frequency domains. The discrete convolution calculations approach the continuous form.

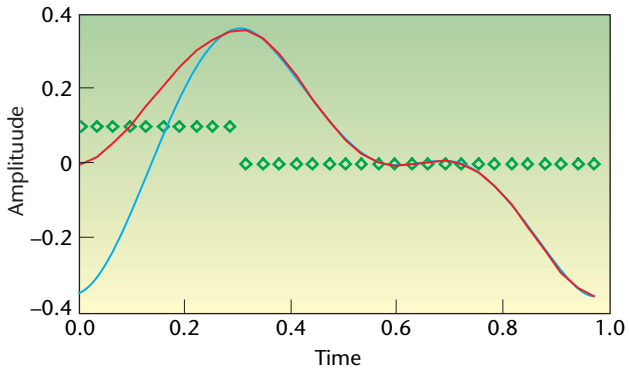

Figure 2. Convolution with and without wraparound distortions. The blue curve shows the circular form of the convolution without zero-padding. The red curve is based on a zero-padded calculation that avoids the distortion associated with circularity. The diamonds show the *h* response curve (scaled at 10 percent of true height); the width of the response function is associated with the region in which the circular convolution is spoiled.

$$x(t) = \begin{cases} \sin(2\pi t) + \sin(4\pi t), & 0 \le t \le 1, \\ 0, & otherwise, \end{cases} \tag{9}$$

and

$$h(t) = \begin{cases} 1, & 0 \le t \le 0.3125, \\ 0, & otherwise, \end{cases} \tag{10}$$

which we discretize to have 32 equally spaced points on the interval [0,1].

Figure 1 shows the signal, the impulse response, and the associated continuous and discrete convolutions. The discrete convolution as computed by taking the IFFT of the product of the FFTs of x and h is identical to that obtained via matrix multiplication.

Figure 2 shows the wraparound associated with the circular convolution example. The convolution is altered for the number of nonzero data points in h.

In Figure 3, we show the FFTs of the linear and circular convolutions. The FFT of the convolution resulting from the matrix multiplication is the same as the product of x and h's FFTs. In the figure, we can see some frequency dependence associated with the convolution process. Figure 4 gives an overall summary of the operations and their interrelation.

For a more realistic example of convolution, let's look at the propagation of an acoustic pressure wave through a rectangular waveguide. The waveguide's resonant conditions restrict the wave numbers of the transverse wave components to discrete values, and the wave propagates only in certain modes. If we treat the waveguide as a linear device with an impulse response h, then we can predict the form of the transmitted signal by taking the convolution of our input signal x and the impulse response of the waveguide. Kristien Meykens and colleagues[3] show that for modes other than (0, 0), the impulse response departs from a δ-function in which the lower frequencies resemble a reversed chirp.

Figure 5 shows the convolution of an input signal consisting of a brief acoustic burst with the impulse response of a rectangular waveguide (which we represent as a chirp function). We form this input signal by multiplying an 8-kHz sine wave by a Bartlett (tent-shaped) window. The chirp function represents the impulse response for the waveguide's (1, 0) mode, and $f(t) = 10^3 + 2 \cdot 10^6 t$ represents the chirp function's frequency dependence. The chirp expression is simply $\sin(\theta(t))$, where

$$\theta(t) = \int_0^t 2\pi f(t')dt'. \tag{11}$$

In general, a convolution shows the two functions' entanglement. The examples we've discussed here provide a clear instance in which we can see where the similarity between the input signal and the impulse response is the greatest. Such computations are in reasonable agreement with experimental results.[3]

In the next installment of this series, we'll continue to examine the problem of spectrum estimation with a discussion of the autocorrelation function and the correlogram estimates, which are based upon it. CiSE

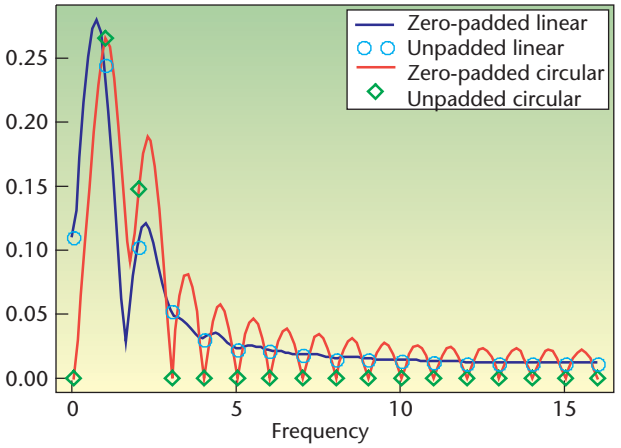

Figure 3. The FFTs of the linear and circular convolutions. The two curves are shown with (solid curves) and without (circles and diamonds) zero padding. We computed these FFTs from the convolution data for Figure 1's discrete transform. The results are the same as those obtained by taking the product of *x* and *h*'s FFTs.

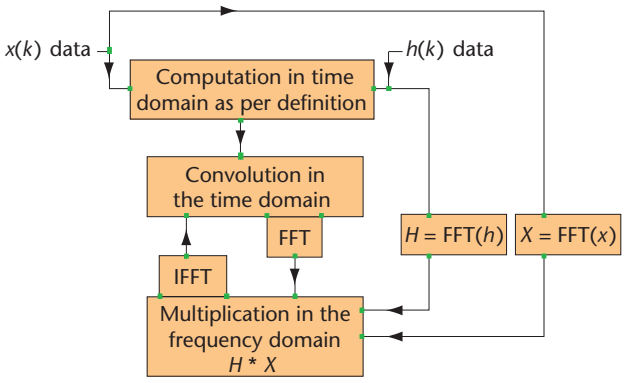

Figure 4. The interrelation between time and frequency domain operations that lead to convolution. Multiplying the FFT's of *x* and *h* followed by an IFFT also lead to the convolution. An FFT of the convolution would yield the same result as the product of the FFTs.

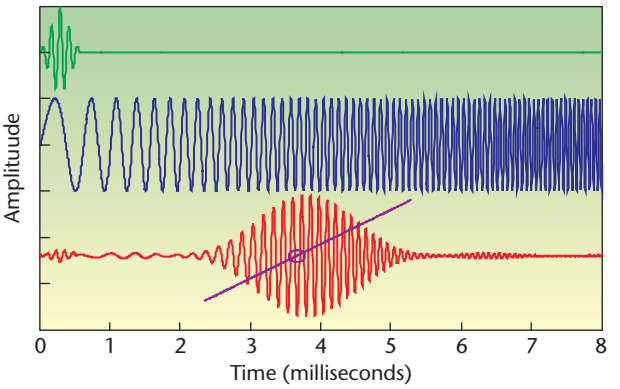

Figure 5. The convolution of a windowed sine wave burst and a chirp function. The top curve shows the input signal, and the middle curves show the impulse response of the waveguide (a chirp function). The chirp frequency increases linearly with time, ranging from roughly 1 kHz at *t* = 0 to roughly 17 kHz at *t* = 8 ms; the frequency increases at a rate of approximately 2 kHz/ms. The bottom curves show the convolution and the approximate frequencies associated with the most significant section of the convolution over time. The one marked point represents the frequency of the windowed sine curve which is 8 kHz. The slope of the line representing frequency is about 1.9 kHz/ms.

References

1. D. Donnelly and B. Rust, "The Fast Fourier Transform for Experimentalists Part I: Concepts," *Computing in Science & Eng.*, vol. 7, no. 2, 2005, pp. 80–88.
2. E. Oran Brigham, *The Fast Fourier Transform and Its Applications*, Prentice-Hall, 1988.
3. K. Meykens, B. Van Rompaey, and J. Janssen, "Dispersion in Acoustic Waveguide: A Teaching Laboratory Experiment," *Am. J. Physics*, vol. 67, no. 5, 1999, pp. 400–406.

Denis Donnelly is a professor of physics at Siena College. His research interests include computer modeling and electronics. Donnelly received a PhD in physics from the University of Michigan. He is a member of the American Physical Society, the American Association of Physics Teachers, and the American Association for the Advancement of Science. Contact him at donnelly@siena.edu.

Bert Rust is a mathematician at the US National Institute for Standards and Technology. His research interests include ill-posed problems, time-series modeling, nonlinear regression, and observational cosmology. Rust received a PhD in astronomy from the University of Illinois. He is a member of SIAM and the American Astronomical Society. Contact him at bwr@nist.gov.

Submissions: Send one PDF copy of articles and/or proposals to Norman Chonacky, Editor in Chief, cise-editor@aip.org. Submissions should not exceed 6,000 words and 15 references. All submissions are subject to editing for clarity, style, and space.

Editorial: Unless otherwise stated, bylined articles and departments, as well as product and service descriptions, reflect the author's or firm's opinion. Inclusion in *CiSE* does not necessarily constitute endorsement by the IEEE, the AIP, or the IEEE Computer Society.

Circulation: *Computing in Science & Engineering* (ISSN 1521-9615) is published bimonthly by the AIP and the IEEE Computer Society. IEEE Headquarters, Three Park Ave., 17th Floor, New York, NY 10016-5997; IEEE Computer Society Publications Office, 10662 Los Vaqueros Circle, PO Box 3014, Los Alamitos, CA 90720-1314, phone +1 714 821 8380; IEEE Computer Society Headquarters, 1730 Massachusetts Ave. NW, Washington, DC 20036-1903; AIP Circulation and Fulfillment Department, 1NO1, 2 Huntington Quadrangle, Melville, NY 11747-4502. Annual subscription rates for 2005: $42 for Computer Society members (print only) and $42 for AIP society members (print plus online). For more information on other subscription prices, see www.computer.org/subscribe/ or https://www.aip.org/forms/journal_catalog/order_form_fs.html. Computer Society back issues cost $20 for members, $96 for nonmembers; AIP back issues cost $22 for members.

Postmaster: Send undelivered copies and address changes to *Computing in Science & Engineering*, 445 Hoes Ln., Piscataway, NJ 08855. Periodicals postage paid at New York, NY, and at additional mailing offices. Canadian GST #125634188. Canada Post Corporation (Canadian distribution) publications mail agreement number 40013885. Return undeliverable Canadian addresses to PO Box 122, Niagara Falls, ON L2E 6S8 Canada. Printed in the USA.

Editor: Denis Donnelly, donnelly@siena.edu

The Fast Fourier Transform for Experimentalists Part III: Classical Spectral Analysis

By Bert Rust and Denis Donnelly

EACH ARTICLE IN THIS CONTINUING SERIES ON THE FAST FOURIER TRANSFORM (FFT) IS DESIGNED TO ILLUMINATE NEW FEATURES OF THE WIDE-RANGING APPLICABILITY OF THIS TRANSFORM. THIS SEGMENT DEALS WITH SOME ASPECTS OF THE spectrum estimation problem. Before we begin, here's a short refresher about two elements we introduced previously, windowing[1] and convolution.[2] As we noted in those installments, a convolution is an integral that expresses the amount of overlap of one function as it is shifted over another. The result is a blending of the two functions. Closely related to the convolution process are the processes of cross-correlation and autocorrelation. Computing the cross-correlation differs only slightly from the convolution; it's useful for finding the degree of similarity in signal patterns from two different data streams and in determining the lead or lag between such similar signals. Autocorrelation is also related to the convolution; it's described later. Windowing, used in extracting or smoothing data, is typically executed by multiplying time-domain data or its autocorrelation function by the window function. A disadvantage of windowing is that it alters or restricts the data, which, of course, has consequences for the spectral estimate. In this installment, we continue our discussion, building on these concepts with a more general approach to computing spectrum estimates via the FFT.

Spectrum Estimation's Central Problem

The periodogram, invented by Arthur Schuster in 1898,[3] was the first formal estimator for a time series's frequency spectrum, but many others have emerged in the ensuing century. Almost all use the FFT in their calculations, but they differ in their assumptions about the missing data; that is, the data outside the observation window. These assumptions have profound effects on the spectral estimates. Let t be time, f be frequency, and $x(t)$ a real function on the interval $-\infty < t < \infty$. The continuous Fourier transform (CFT) of $x(t)$ is defined by

$$X(f) = \int_{-\infty}^{\infty} x(t)\exp(-2\pi \boldsymbol{i} f t)\,dt\,, \quad -\infty \le f \le \infty, \tag{1}$$

where $\boldsymbol{i} \equiv \sqrt{-1}$. If we knew $x(t)$ perfectly and could compute Equation 1, then we could compute an energy spectral density function

$$E(f) = |X(f)|^2, \quad -\infty \le f \le \infty, \tag{2}$$

and a *power spectral density function* (PSD) by

$$P(f) = \lim_{\mathcal{T}\to\infty} \frac{1}{\mathcal{T}} \left| \int_{-\mathcal{T}}^{\mathcal{T}} x(t)\exp(-2\pi \boldsymbol{i} f t)\,dt \right|^2, \quad -\infty \le f \le \infty. \tag{3}$$

But we have only a discrete, real time series

$$x_j = x(t_j), \text{ with } t_j = j\Delta t, \quad j = 0, 1, \ldots, N-1, \tag{4}$$

defined on a finite time interval of length $N\Delta t$. We saw in Part I[1] that sampling $x(t)$ with sample spacing Δt confined our spectral estimates to the Nyquist band $0 \le f \le 1/2\Delta t$. We used the FFT algorithm to compute the discrete Fourier transform (DFT)

$$X_k = \sum_{j=0}^{N-1} x_j \exp\left(-2\pi \boldsymbol{i} \frac{j}{N} k\right), \quad k = 0, 1, \ldots, N/2, \tag{5}$$

which approximates the CFT $X(f)$ at the Fourier frequencies

$$f_k = \frac{k}{N\Delta t}\,, \; k = 0, 1, \ldots, N/2. \tag{6}$$

We then computed periodogram estimates of both the PSD and the amplitude spectrum by

$$P(f_k) = \frac{1}{N}|X_k|^2\,, \; k = 0, 1, \ldots, N/2,$$

$$A(f_k) = \frac{2}{N}|X_k|\,, k = 0, 1, \ldots, N/2. \tag{7}$$

We also saw that we could approximate

the CFT and the frequency spectrum on a denser frequency mesh simply by appending zeroes to the time series. This practice, called zero padding, is just an explicit assertion of an implicit assumption of the periodogram method—namely, that the time series is zero outside the observation window. Frequency spectrum estimation is a classic underdetermined problem because we need to estimate the spectrum at an infinite number of frequencies using only a finite amount of data. This problem has many solutions, differing mainly in what they assume about the missing data.

Before considering other solutions to this problem, let's reconsider one of the examples from Part I[1] (specifically, Figure 1b), but make it more realistic by simulating some random measurement errors. More precisely, we take $N = 32$, $\Delta t = 0.22$, and consider the time series

$$t_j = j\Delta t, j = 0, 1, 2, \ldots, N-1,$$

$$x_j = x(t_j) = \sin[2\pi f_0(t_j + 0.25)] + \in_j, \quad (8)$$

with $f_0 = 0.5$, and each $\in_j$ a random number drawn independently from a normal distribution with mean zero and standard deviation $\sigma = 0.25$. This new time series is plotted together with the original uncorrupted series in Figure 1a. Both series were zero padded to length 1,024 (992 zeroes appended) to obtain the periodogram estimates given in Figure 1b. It's remarkable how well the two spectra agree, even though the noise's standard deviation was 25 percent of the signal's amplitude.

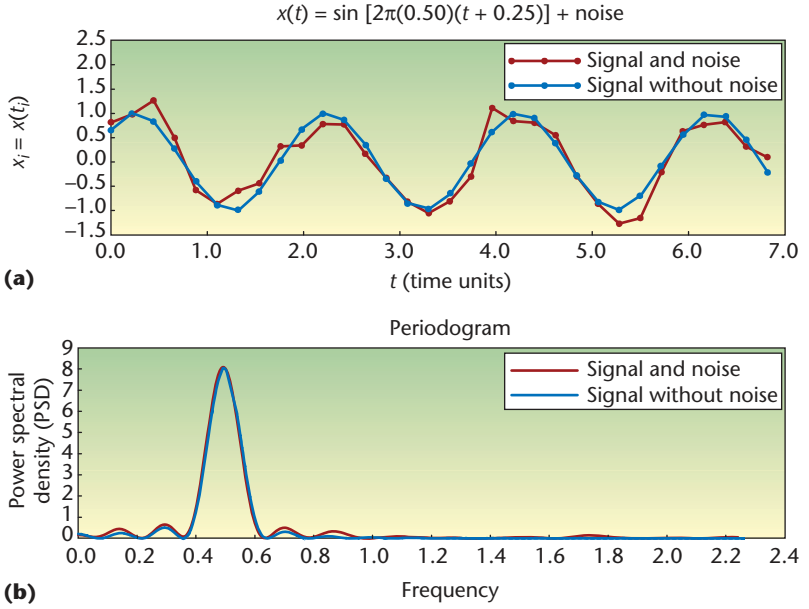

Figure 1. Original and new time series as defined by Equation 8. (a) The noise-corrupted time series and the uncorrupted series originally used in Part I's Figure 1b. The noise is independently, identically distributed *n*(0, 0.25). (b) Periodograms of the two times series plotted in (a). For the noise-corrupted series, the peak is centered on frequency $\hat{f}_0 = 0.493$.

The Autocorrelation Function

After the periodogram, the next frequency spectrum estimators to emerge were Richard Blackman and John Tukey's *correlogram* estimators.[4] They're based on the *autocorrelation theorem* (sometimes called Wiener's theorem), which states that if $X(f)$ is the CFT of $x(t)$, then $|X(f)|^2$ is the CFT of the *autocorrelation function* (ACF) of $x(t)$. Norbert Wiener defined the latter function as[5]

$$\rho(\tau) = \lim_{\mathcal{T}\to\infty} \frac{1}{2\mathcal{T}} \int_{-\mathcal{T}}^{\mathcal{T}} x^*(t)x(t+\tau)dt, \quad -\infty < \tau < \infty, \quad (9)$$

in which the variable τ is called the *lag* (the time interval for the correlation of $x(t)$ with itself), and $x^*(t)$ is the complex conjugate of $x(t)$. Thus, if we could access $x(t)$, we could compute the PSD in two ways: either by Equation 3 or by

$$P(f) = \int_{-\infty}^{\infty} \rho(\tau)\exp(-2\pi i f\tau)d\tau. \quad (10)$$

But again, we have access to only a noisy time series $x_0, x_1, \ldots, x_{N-1}$, so to use the second method, we need estimates for $\rho(\tau)$ evaluated at the discrete lag values

$$\tau_m = m\Delta t, \; m = 0, 1, \ldots, N-1. \quad (11)$$

Because we're working with a real time series, and $\rho(\tau_{-m}) = \rho(\tau_m)$, we don't need to worry about evaluating $\rho(\tau)$ at negative lags.

Because $\rho(\tau)$ is a limit of the average value of $x^*(t)x(t+\tau)$ on the interval $[-\mathcal{T}, \mathcal{T}]$, the obvious estimator is the sequence of average values

$$\hat{\rho}_m = \hat{\rho}(m\Delta t) = \frac{1}{N-m}\sum_{n=0}^{N-m-1} x_n x_{n+m}, \quad m = 0, 1, \ldots, N-1. \quad (12)$$

This sequence is sometimes called the *unbiased estimator of* $\rho(\tau)$ because its expected value is the true value—that is, $\varepsilon\{\hat{\rho}(m\Delta t)\} = \rho(m\Delta t)$. But the data are noisy, and for successively larger values of m, the average $\hat{\rho}_m$ is based on fewer and fewer terms, so the variance grows and, for large m, the estimator becomes unstable. Therefore, it's common practice to use the biased estimator

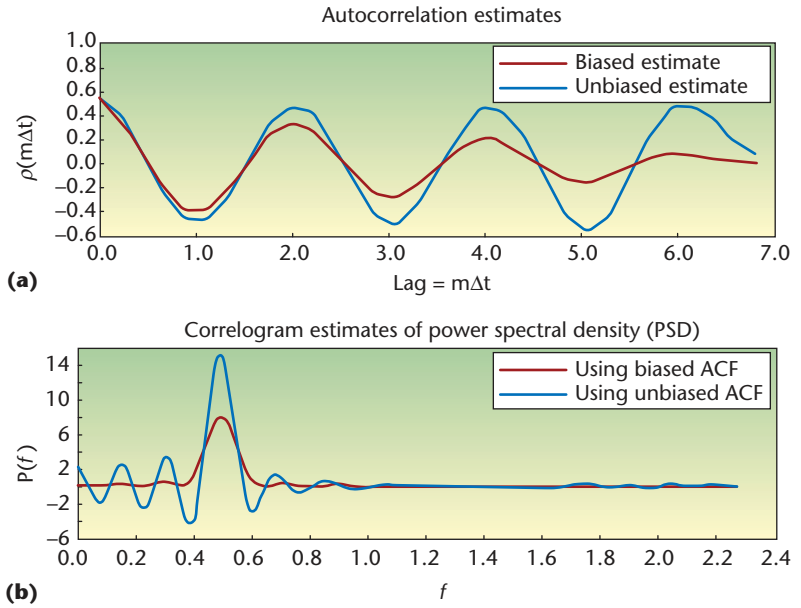

Figure 2. Autocorrelation and correlogram estimates for the noisy time series defined by Equation 8. (a) Biased and unbiased estimates of the autocorrelation function (ACF); (b) correlogram estimates obtained from the ACF estimates in (a).

$$\hat{\rho}_m = \hat{\rho}(m\Delta t) = \frac{1}{N}\sum_{n=0}^{N-m-1} x_n x_{n+m}, \quad m = 0, 1, \ldots, N-1, \tag{13}$$

which damps those instabilities and has a smaller total error (bias + variance) than does the unbiased estimator. (*Bias* is the difference between the estimator's expected value and the true value of the quantity being estimated.) Figure 2a gives plots of both estimates for the times series that Equation 8 defines.

The ACF we have just described is sometimes called the *engineering autocorrelation* to distinguish it from the *statistical autocorrelation*, which is defined by

$$\hat{r}_m = \frac{\frac{1}{N}\sum_{n=0}^{N-m-1}(x_n - \bar{x})(x_{n+m} - \bar{x})}{\frac{1}{N}\sum_{n=0}^{N-1}(x_n - \bar{x})^2},$$

$$\text{where } \bar{x} = \frac{1}{N}\sum_{n=0}^{N-1} x_n \ . \tag{14}$$

The individual $\hat{r}_m$ are true correlation coefficients because they satisfy

$$-1 \le \hat{r}_m \le 1, \ m = 0, 1, \ldots, N-1. \tag{15}$$

Correlogram PSD Estimators

Once we've established the ACF estimate, we can use the FFT to calculate the discrete estimate to the PSD. More precisely, the ACF estimate is zero padded to have M lags, which gives $M/2 + 1$ frequencies in the PSD estimate, which we can then compute by approximating Equation 10 with

$$\hat{P}_k = \hat{P}(f_k) = \sum_{j=0}^{M-1} \hat{\rho}_j \exp\left(-2\pi \boldsymbol{i} \frac{j}{M} k\right), \quad k = 0, 1, \ldots, M/2. \tag{16}$$

Zero padding in this case is an explicit expression of the implicit assumption that the ACF is zero for all lag values $\tau > (N-1)\Delta t$. We must assume that because we don't know the data outside the observation window. Assuming some nonzero extension for the ACF would amount to an implicit assumption about the missing observed data.

Figure 2b plots the correlograms corresponding to the biased and unbiased ACF estimates, shown in Figure 2a. The negative sidelobes for the unbiased correlogram show dramatically why most analysts choose the biased estimate even though its central peak is broader. The reason for this broadening, and for the damped sidelobes, is that the biased ACF, Equation 13, can also be computed by multiplying the unbiased ACF, Equation 12, by the triangular (Bartlett) tapering window

$$w_k = 1 - \frac{k}{N}, \quad k = 0, 1, 2, \ldots, N-1. \tag{17}$$

Recall that we observed the same sort of peak broadening and sidelobe suppression in Part I's Figure 10 when we multiplied the observed data by a Blackman window before computing the periodogram.

Notice that the biased correlogram estimate plotted in Figure 2b is identical to the periodogram estimate plotted in Figure 1b. The equality of these two estimates, computed in very different ways, constitutes a finite dimensional analogue of Wiener's theorem for the continuous PSD.

Figure 2b's two PSD correlograms aren't the only members of the class of correlogram estimates. We can obtain other variations by truncating the ACF estimate at lags $\tau < (N-1)\Delta t$ and by smoothing the truncated (or untruncated) estimate with one of the tapering windows defined in Part I's Equation 11. Most of those windows were originally developed for the correlogram method; they were then retroactively applied to the periodogram method when the latter was resurrected in the mid 1960s. In those days, people often used very severe truncations, with the estimates being

set to zero at 90 percent or more of the lags. Not only did this alleviate the variance instability problem, but it also reduced the computing time—an important consideration before the invention of the FFT algorithm, and when computers were much slower than today.

The effect of truncating the biased ACF estimate is shown in Figure 3, where $m_{\max}$ is the largest index for which the nonzero ACF estimate is retained. More precisely,

$$\hat{\rho}_m = \frac{1}{N-m}\sum_{n=0}^{N-m-1} x_n x_{n+m},$$
$$m = 0,1,\ldots,m_{\max},$$
$$\hat{\rho}_m = 0,\ m = m_{\max}+1,\ldots,N-1. \quad (18)$$

It's clear that smaller values of $m_{\max}$ produce more pronounced sidelobes and broader central peaks than larger values. The peak broadening is accompanied by a compensating decrease in height to keep the area under the curve invariant. PSD is measured in units of power-per-unit-frequency interval, so the peak's area indicates its associated power.

Figure 4 shows the effect of tapering the truncated ACF estimates used in Figure 3 with a Hamming window

$$w_m = 0.538 + 0.462\cos\left(\frac{m\pi}{m_{\max}}\right),$$
$$m = 0, 1, 2, \ldots, m_{\max}. \quad (19)$$

The sidelobes are suppressed by the tapering, but the central peaks are further broadened. This loss in resolution is the price we must pay to smooth the sidelobes and eliminate their negative excursions.

Tapering the biased ACF estimates with the Hamming window amounts to twice tapering the unbiased estimates; we can obtain the former from the latter by tapering them with the Bartlett window, Equation 17. Figure 5 shows the effect of a single tapering of the unbaised estimates with the Hamming window, Equation 19. Note that the sidelobes are not completely suppressed, but they're not as

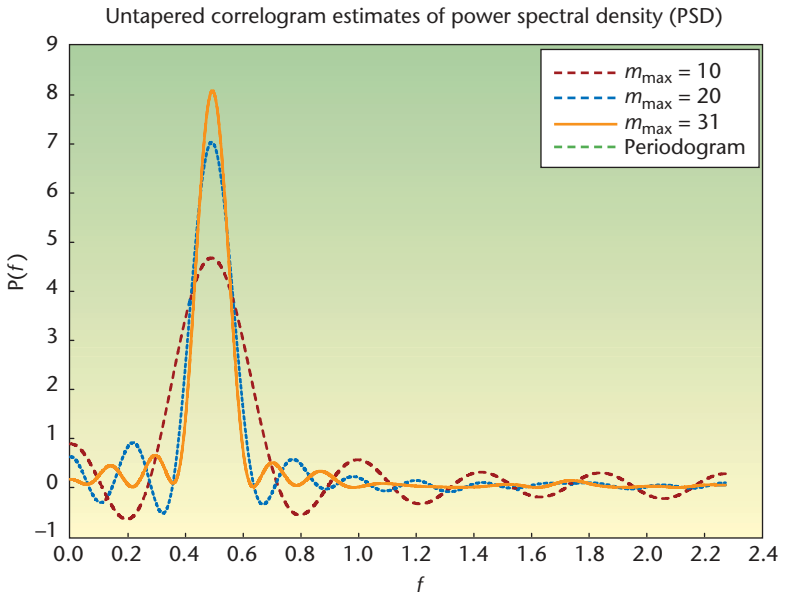

Figure 3. Three correlogram estimates for Equation 8 computed from the biased autocorrelation function (ACF) estimator in Equation 13. The periodogram, although plotted, doesn't show up as a separate curve because it's identical to the m_{max} = 31 correlogram.

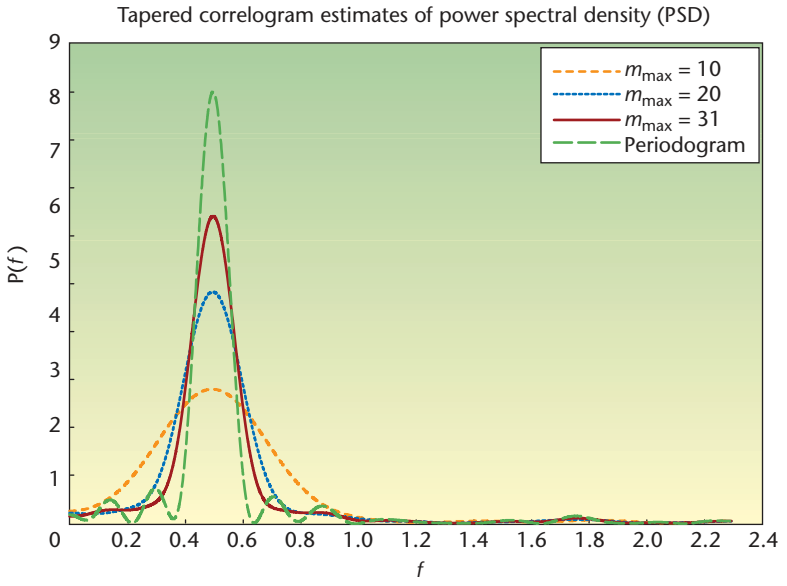

Figure 4. Three correlogram estimates for the time series generated by Equation 8. We computed the estimates by tapering three truncations of the biased estimator in Equation 13 with a Hamming window. The periodogram was also plotted for comparison. Although it has sidelobes, its central peak is sharper than those of the correlograms.

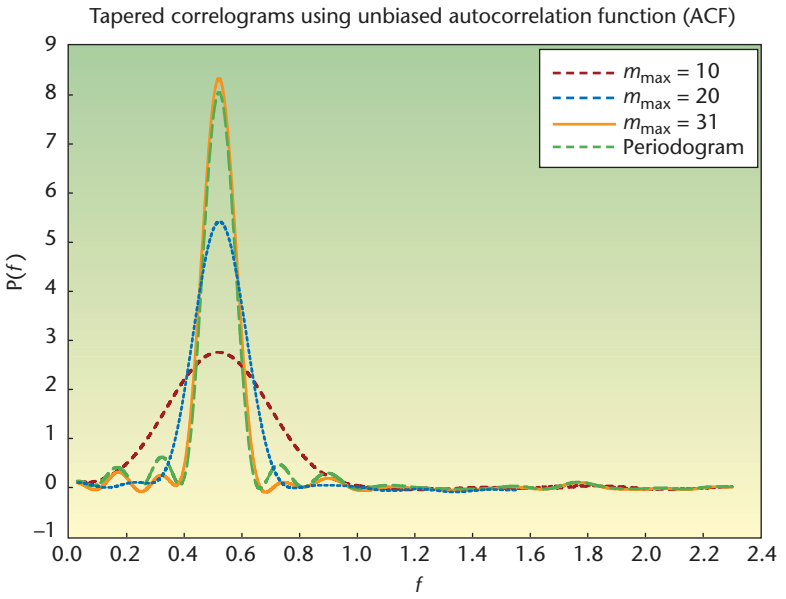

Figure 5. Three correlogram estimates for the time series generated by Equation 8. We computed the estimates by tapering three truncations of the unbiased estimator in Equation 12. We also plotted the periodogram for comparison; again, it has a sharper peak but larger sidelobes .

pronounced as in Figure 3, in which the tapering used the Bartlett window. However, the central peaks are also slightly broader here. This is yet another example of the trade-off between resolution and sidelobe suppression.

This particular example contains only a single-sinusoid, so it doesn't suggest any advantage for the tapering and truncation procedures, but they weren't developed to analyze a time series with such a simple structure. Their advantages are said to be best realized when the signal being analyzed contains two or more sinusoids with frequencies so closely spaced that sidelobes from two adjacent peaks might combine and reinforce one another to give a spurious peak in the spectrum. But of course, if two adjacent frequencies are close enough, then the broadening of both peaks might cause them to merge into an unresolved lump.

Much ink has been used in debating the relative merits of the various truncation and windowing strategies, but none of them have proven to be advantageous, so correlogram estimates are beginning to fall out of favor. For the past 30 years or so, most researchers have concentrated on autoregressive spectral estimates, which, as we shall see in Part 4, give better resolution because they make better assumptions about the the data outside the window of observation. CiSE

References

1. D. Donnelly and B. Rust, "The Fast Fourier Transform for Experimentalists, Part I: Concepts," *Computing in Science & Eng.*, vol. 7, no. 2, 2005, pp. 80–88.
2. D. Donelly and B. Rust, "The Fast Fourier Transform for Experimentalists, Part II: Convolutions," *Computing in Science & Eng.*, vol. 7, no. 3, 2005, pp. 92–95.
3. A. Schuster, "On the Investigation of Hidden Periodicities with Application to a Supposed Twenty-Six-Day Period of Meteorological Phenomena," *Terrestrial Magnetism*, vol. 3, no. 1, 1898, pp. 13–41.
4. R.B. Blackman and J.W. Tukey, *The Measurement of Power Spectra*, Dover Publications, 1959.
5. N. Wiener, *Extrapolation, Interpolation, and Smoothing of Stationary Time Series*, MIT Press, 1949.

Denis Donnelly is a professor of physics at Siena College. His research interests include computer modeling and electronics. Donnelly received a PhD in physics from the University of Michigan. He is a member of the American Physical Society, the American Association of Physics Teachers, and the American Association for the Advancement of Science. Contact him at donnelly@siena.edu.

Bert Rust is a mathematician at the US National Institute for Standards and Technology. His research interests include ill-posed problems, time-series modeling, nonlinear regression, and observational cosmology. Rust received a PhD in astronomy from the University of Illinois. He is a member of SIAM and the American Astronomical Society. Contact him at bwr@nist.gov.

Editors: David Winch, winch@TaosNet.com
Denis Donnelly, donnelly@siena.edu

The Fast Fourier Transform for Experimentalists, Part IV: Autoregressive Spectral Analysis

By Bert Rust and Denis Donnelly

IT'S RARE THAT WE HAVE ONLY ONE WAY IN WHICH TO APPROACH A PARTICULAR TOPIC—FORTUNATELY, SPECTRUM ESTIMATION ISN'T ONE OF THOSE RARE CASES. IN THE MOST recent article of this series,[1] we considered the periodogram and correlogram estimators for the power spectral density (PSD) function. However, they are only two of several possibilities.

In this installment, we consider two additional kinds of spectrum estimates: autoregressive (AR) estimates and the maximum entropy (ME) method. In the first approach, we assume that an AR process generates the time series, which means we can compute the PSD of the time series from estimates of the AR parameters. The second approach is a special case of the first, but it uses a different method for estimating the AR parameters. Specifically, it chooses them to make the PSD's inverse transform compatible with the measured time series, while remaining maximally noncommittal about the data outside the observational window.

Autoregressive Time-Series Models

Both the periodogram and correlogram estimates make rather unrealistic assumptions about the data outside the observational window. Moreover, when they use tapering windows or truncation of the autocorrelation function (ACF), they change the observed data. The years since the early 1970s have seen the development of a new class of PSD estimators that are based on the idea of fitting a parametric time-series model to the observed data. This enables us to use estimates of the parameters in the theoretical expression of the model's PSD to get an estimate of the observed series' PSD. If the model is a good representation of the process that generated the data, it should hopefully give a more realistic extrapolation for the missing data.

The class of models used most often assumes that the data are generated by an AR process in which each new data point is formed from a linear combination of the preceding data plus a random shock. The basic idea is that a system's future states depend in a deterministic way on previous states, but at each time step, a random perturbation drives the system forward. We can write the AR models of orders 1, 2, and 3 as

$$\begin{aligned} &AR(1): x_n = -a_1 x_{n-1} + u_n, && n = 1, 2, \ldots, N-1, \\ &AR(2): x_n = -a_1 x_{n-1} - a_2 x_{n-2} + u_n, && n = 2, 3, \ldots, N-1, \\ &AR(3): x_n = -a_1 x_{n-1} - a_2 x_{n-2} - \\ &\qquad a_3 x_{n-3} + u_n, && n = 3, 4, \ldots, N-1, \end{aligned} \quad (1)$$

where a_1, a_2, and a_3 are the AR parameters (whose values must be determined to make the model fit the data), and u_n is the random shock at time step n. We assume the random shocks to be samples from a zero-mean distribution whose variance remains constant in time. The choice of negative signs for the parameters is a universal convention adopted for notational convenience in derivations that we won't give here.

Autoregressive Spectral Estimates

In general, for any integer $p < N-1$, the $AR(p)$ model is

New Editorial Board Member

David Winch is an emeritus professor of physics at Kalamazoo College, Michigan. His research interests are focused on educational technologies (his most recent work is a DVD/CD called "Physics: Cinema Classics"). Winch has a PhD in physics from Clarkson University. He is a member of the American Physical Society, the American Association of Physics Teachers, and the National Science Teachers Association. He'll be joining our board as a coeditor of the Education department. Contact him at winch@taosnet.com or lead editor Jenny Ferrero at jferrero@computer.org if you are interested in writing.

Copublished by the IEEE CS and the AIP 1521-9615/05/$20.00 © 2005 IEEE

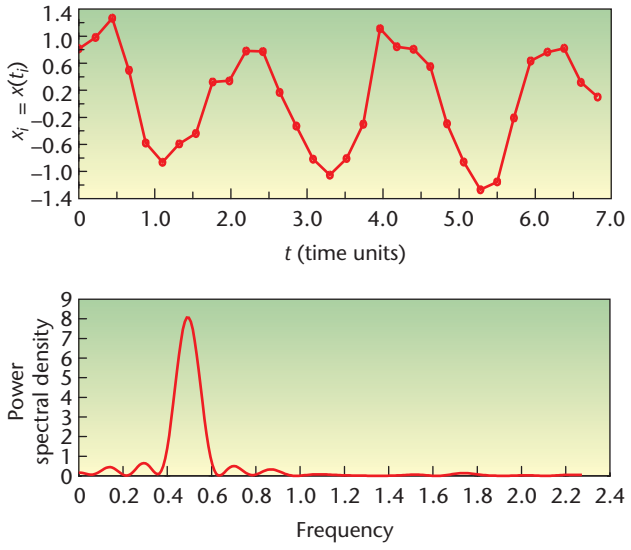

Figure 1. The time series generated by Equation 10 and its periodogram. The discrete points in the upper plot are joined by straight-line segments to emphasize the time series nature of the data. The time series was zero padded to length M = 1,024 to compute the periodogram in the lower plot.

$$x_n = -\sum_{k=1}^{p} a_k x_{n-k} + u_n,\ n = p,\ p+1,\ \ldots,\ N-1. \tag{2}$$

We can show that the PSD function for this model is

$$P_{AR}(f) = \frac{\rho_w}{\left|1 + \sum_{j=1}^{p} a_j \exp(-2\pi \mathbf{i} f\, j\Delta t)\right|^2}, -\frac{1}{2\Delta t} \le f \le \frac{1}{2\Delta t}, \tag{3}$$

where ρ_w is another adjustable parameter that we can estimate along with $a_1, a_2, \ldots, a_p$ by solving the $(p+1)\times(p+1)$ linear system of equations,

$$\begin{bmatrix} \rho_0 & \rho_{-1} & \rho_{-2} & \cdots & \rho_{-p} \\ \rho_1 & \rho_0 & \rho_{-1} & \cdots & \rho_{-p+1} \\ \rho_2 & \rho_1 & \rho_0 & \cdots & \rho_{-p+2} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \rho_p & \rho_{p-1} & \rho_{p-2} & \cdots & \rho_0 \end{bmatrix} \begin{bmatrix} 1 \\ a_1 \\ a_2 \\ \cdots \\ a_p \end{bmatrix} = \begin{bmatrix} \rho_w \\ 0 \\ 0 \\ \cdots \\ 0 \end{bmatrix}, \tag{4}$$

which are sometimes called the *Yule-Walker equations*. The ρ-values in the matrix are just the autocorrelations $\rho_k = \rho(\tau_k) = \rho(k\Delta t)$ that we defined in the last issue[1] with

$$\rho(\tau) = \lim_{\mathcal{T}\to\infty} \frac{1}{2\mathcal{T}} \int_{-\mathcal{T}}^{\mathcal{T}} x^*(t)x(t+\tau)dt, -\infty < \tau < \infty, \tag{5}$$

where x^* is the complex conjugate of $x(t)$. We're working with real data, so $\rho_{-k} = \rho_k$, which means that the matrix is symmetric and positive definite. Note that the element in row i and column j is just $\rho_{(i-j)}$, which makes it a *Toeplitz matrix*. Norman Levinson[2] exploited this special structure to devise a recursive algorithm that solves the system in times proportional to $(p+1)^2$ rather than the $(p+1)^3$ required by a general linear equations solver.

We can summarize the steps required to compute an autoregressive spectral estimate as follows:

1. Choose an autoregressive order $p \le N-1$.
2. Compute ACF estimates $\hat{\rho}_0, \hat{\rho}_1, \ldots, \hat{\rho}_p$ using the biased estimator

$$\hat{\rho}_m = \hat{\rho}(m\Delta t) = \frac{1}{N}\sum_{n=0}^{N-m-1} x_n x_{n+m},$$

$$m = 0, 1, \ldots, N-1. \tag{6}$$

3. Substitute $\hat{\rho}_0, \hat{\rho}_1, \ldots, \hat{\rho}_p$ into the matrix in Equation 4 and use the Levinson algorithm to compute estimates $\hat{a}_1, \hat{a}_2, \ldots, \hat{a}_p$ and $\hat{\rho}_w$.
4. Substitute $\hat{a}_1, \hat{a}_2, \ldots, \hat{a}_p$ and $\hat{\rho}_w$ into Equation 3 to compute the PSD estimate $\hat{P}_{AR}(f)$ on any desired frequency mesh.

It's absolutely necessary to use the biased ACF estimator in step 2. Using the unbiased estimator produces an unstable linear system (see Equation 4) with a matrix that numerically isn't positive definite.

It's easy to do the calculations in the final step by using the fast Fourier transform (FFT) algorithm to compute the denominator in Equation 3. If we define $\hat{a}_0 \equiv 1$, then

$$1 + \sum_{j=1}^{p} \hat{a}_j \exp(-2\pi \mathbf{i} f\, j\Delta t) = \sum_{j=0}^{p} \hat{a}_j \exp(-2\pi \mathbf{i} f\, j\Delta t). \tag{7}$$

Suppose we want to evaluate $P_{AR}(f)$ at $(M/2+1)$ equally spaced frequencies

$$f_k = \frac{k}{M\Delta t},\ k = 0, 1, \ldots, M/2, \tag{8}$$

where $M > p$. Then,

$$\sum_{j=0}^{p} \hat{a}_j \exp(-2\pi \mathbf{i} f_k\, j\Delta t) = \sum_{j=0}^{p} \hat{a}_j \exp\left(-2\pi \mathbf{i}\, \frac{j}{M} k\right), \tag{9}$$

and we can compute these values quite quickly by zero padding the sequence $\hat{a}_0, \hat{a}_1, \hat{a}_2, \ldots, \hat{a}_p$ to have M terms and applying the FFT algorithm.

Two Examples

If we choose the AR order p properly, the peaks in the $AR(p)$ spectrum will be sharper than those in the periodogram or correlogram estimates. There is no clear-cut prescription for choosing p, but a fairly wide range of values will usually give acceptable results. To illustrate the effect of the choice of p, let's revisit an example time series used in the last issue.[1] Again, we'll take $N = 32$, $\Delta t = 0.22$, and consider the time series generated by

$$t_j = j\Delta t, \quad j = 0, 1, 2, \ldots, N-1,$$
$$x_j = x(t_j) = \sin[2\pi f_0(t_j + 0.25)] + \varepsilon_j, \tag{10}$$

with $f_0 = 0.5$, and each ε_j a random number drawn independently from a normal distribution with mean zero and standard deviation $\sigma = 0.25$. Figure 1 plots the time series and its periodogram, and Figure 2 gives three different $AR(p)$ spectra for the time series, together with the periodogram for comparison. Table 1 gives the locations of the peak centers. Both the $AR(16)$ and $AR(24)$ estimates give better results than the periodogram, but for real-world problems, it's best to try several orders in the range $N/2 \le p \le 3N/4$ and compare them to make the final choice. Our own experience has indicated that the best choice usually has $p \approx 2N/3$.

To better illustrate the AR methods' power, let's reconsider another time series originally introduced in Part I of our series (specifically, Figure 2a).[3] We generated it by summing two sine waves, with amplitudes $A_1 = A_2 = 1.0$, frequencies $f_1 = 1.0$ and $f_2 = 1.3$, and phases $\phi_1 = \phi_2 = 0$, at $N = 16$ equally spaced time points with $\Delta t = 0.125$. Again, we add random noise to make the problem more realistic, and write

$$t_j = j\Delta t, \quad j = 0, 1, \ldots, N-1,$$
$$x_j = \sin[2\pi f_1 t_j] + \sin[2\pi f_2 t_j] + \varepsilon_j, \tag{11}$$

with the ε_j chosen independently from a normal distribution; the mean is 0 and standard deviation $\sigma = 0.25$. This is the same error distribution in the preceding example, but the samples used here differ from any used there. The top graph of Figure 3 gives plots of the noisy and noise-free time series, and the bottom graph gives their periodograms. Figure 4 gives plots of the PSD's periodogram and $AR(12)$ estimates. The latter clearly indicates the presence of two peaks, although it doesn't completely resolve them. The two maxima occur at frequencies very near the true values used to generate the time series. It's remarkable that the $AR(12)$ estimate could obtain such good agreement with the true values using only 16 noise-corrupted data points.

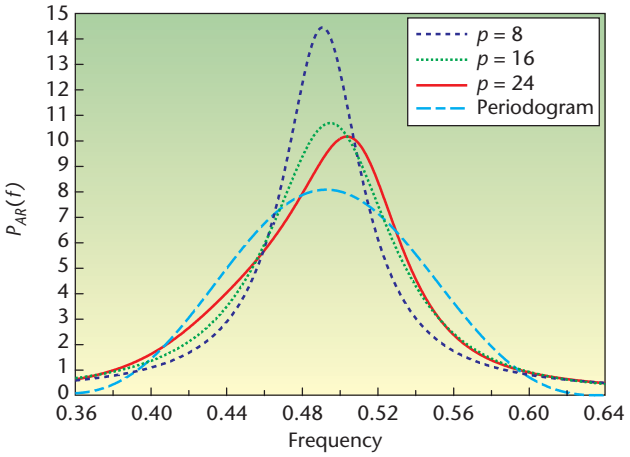

Figure 2. $AR(p)$ power spectral density (PSD) estimates. For $p = 8$, 16, and 24, and the periodogram for the time series generated by Equation 10, the plot doesn't cover the whole Nyquist band $0 \le f \le 2.273$, but rather only the frequency range spanned by the central peak in the periodogram. Using the whole Nyquist range renders the $AR(p)$ peaks so narrow that it's difficult to distinguish between them.

Table 1. Peak centers.

Estimate	Periodogram	$AR(8)$	$AR(16)$	$AR(24)$
$\hat{f}_0$	0.493	0.491	0.495	0.504

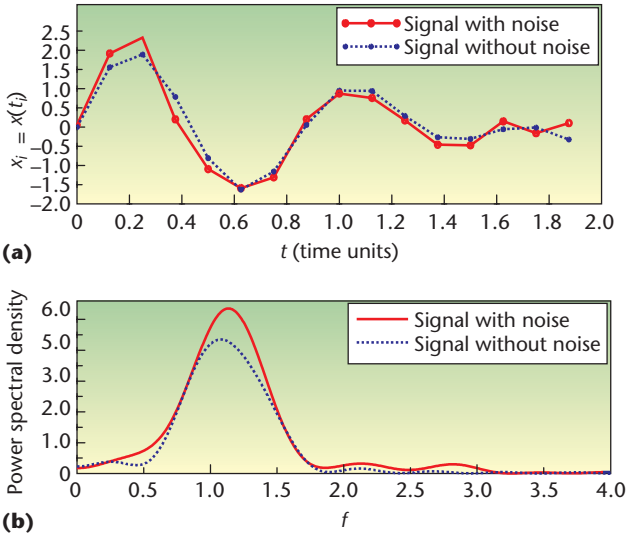

Figure 3. Time series. In (a) the noise-corrupted time series generated by Equation 11, the noise is independently and identically distributed $n(0, 0.25)$. (b) Periodograms of the two time series plotted in (a). In neither case was the periodogram method able to resolve two separate peaks. For the noisy spectrum, the unresolved lump peaks at frequency $\hat{f} = 1.136$.

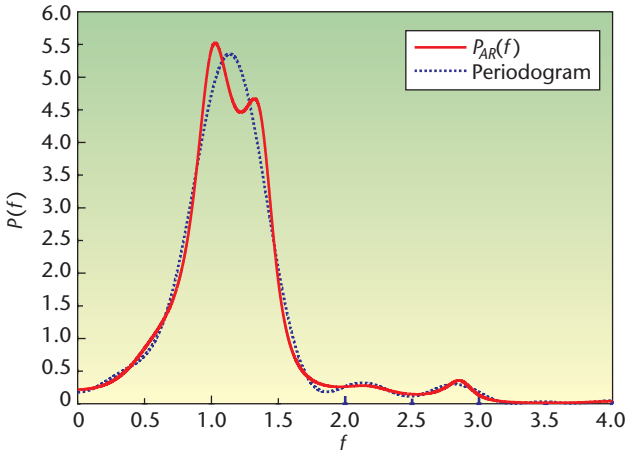

Figure 4. Power spectral density (PSD). The *AR*(12) and the untapered periodogram estimates of the PSD for time series generated by Equation 11. The two maxima in the *AR*(12) spectrum occur at frequencies $\hat{f}_1 = 1.027$ and $\hat{f}_2 = 1.321$, which are very near the true values $f_1 = 1.00$ and $f_2 = 1.30$.

The Maximum Entropy Approach

John Parker Burg invented the ME method in the late 1960s; he exhibited its strengths and advantages in oral presentations at geophysics conferences, but he didn't publish the mathematical derivations that defined and justified it until his PhD thesis[4] appeared in 1975. This lack of published documentation produced a great deal of independent work by other researchers who were trying to understand and extend the method. In fact, the ME method was one of the chief motivators for the development of the AR methods and can be classified as an AR method itself, although Burg didn't use AR models in its development.

Rather, Burg started with the definition for PSD, that is,

$$P(f) = \int_{-\infty}^{\infty} \rho(\tau) \exp(-2\pi \mathbf{i} f \tau) d\tau, \quad (12)$$

but sought a function $P_e(f)$, defined on the Nyquist band $-1/(2\Delta t) \le f \le 1/(2\Delta t)$, which satisfied three guiding principles:

1. The inverse Fourier transform of $P_e(f)$ should return the autocorrelation function unchanged by any filtering or tapering operations:
$$\rho_m = \rho(m\Delta t) = \int_{-\frac{1}{2\Delta t}}^{\frac{1}{2\Delta t}} P_e(f) \exp(2\pi \mathbf{i} f m \Delta t) df,$$
$$m = 0, 1, \ldots, N-1. \quad (13)$$
2. $P_e(f)$ should correspond to the most random or unpredictable time series whose autocorrelation function agrees with the known values.
3. $P_e(f) > 0$ on the interval $-1/(2\Delta t) \le f \le 1/(2\Delta t)$.

The first condition merely states that the measured data shouldn't be changed in any way in computing $P_e(f)$. The second is a statement about what is to be assumed about the data outside the observational window. Essentially, it says that those assumptions should be minimized.

To measure a time series' randomness or unpredictability, Burg used the information theoretic concept of *entropy*. A random process

$$\ldots x(-2\Delta t), x(-\Delta t), x(0), x(\Delta t), x(2\Delta t), \ldots \quad (14)$$

is said to be *band limited* if its PSD function is zero everywhere outside its Nyquist band. If $P(f)$ is such a PSD function, then the time series' entropy rate (entropy per sample) is given by

$$h\{(P(f)\} = \int_{-\frac{1}{2\Delta t}}^{\frac{1}{2\Delta t}} \ln[P(f)] df. \quad (15)$$

Burg's idea was to maximize this quantity, subject to the constraints imposed by Equation 13. More precisely, he sought to impose the constraint at lags 0, Δt, $2\Delta t$, …, $p\Delta t$, with $p < N$ and then choose from the set of all nonnegative functions $P(f)$ that satisfy those $p + 1$ constraints the particular one that minimizes the entropy rate (Equation 15). We can write the problem formally as

$$h\{P_e(f)\} = \max_{P(f)} \left\{ \int_{-\frac{1}{2\Delta t}}^{\frac{1}{2\Delta t}} \ln[P(f)] df \;\middle|\; \begin{array}{c} P(f) > 0, \\ \int_{-\frac{1}{2\Delta t}}^{\frac{1}{2\Delta t}} P(f) \exp(2\pi \mathbf{i} f m \Delta t) df = \rho_m, \\ m = 0, 1, \ldots, p \end{array} \right\}. \quad (16)$$

We need techniques from the calculus of variations to solve it; we can show that

$$P_e(f) = \frac{\rho_e}{\left| 1 + \sum_{j=1}^{p} a_j \exp(-2\pi \mathbf{i} f j \Delta t) \right|^2}, -\frac{1}{2\Delta t} \le f \le \frac{1}{2\Delta t}, \quad (17)$$

where $a_1, a_2, \ldots, a_p$ and ρ_e are parameters satisfying

$$\begin{bmatrix} \rho_0 & \rho_1 & \rho_2 & \cdots & \rho_p \\ \rho_1 & \rho_0 & \rho_1 & \cdots & \rho_{p-1} \\ \rho_2 & \rho_1 & \rho_0 & \cdots & \rho_{p-2} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \rho_p & \rho_{p-1} & \rho_{p-2} & \cdots & \rho_0 \end{bmatrix} \begin{bmatrix} 1 \\ a_1 \\ a_2 \\ \cdots \\ a_p \end{bmatrix} = \begin{bmatrix} \rho_e \\ 0 \\ 0 \\ \cdots \\ 0 \end{bmatrix}. \quad (18)$$

Equation 17 is the same as Equation 3, and, because we're working with real data for which $\rho_{-k} = \rho_k$, Equation 18 is the same as Equation 4. Thus, the maximum entropy method is correctly classified as an AR method, even though Burg used different methods to estimate the autorcorrelations and parameters in Equation 18.

Forward and Backward Prediction Filters

Burg regarded the vector $(1 \;\; a_1 \;\; a_2 \;\; \ldots \;\; a_p)^T$ as a prediction filter, which he applied to the data $x_0, x_1, \ldots, x_{N-1}$ in both the forward and reverse directions to get forward and backward predictions $\hat{x}_n^f$, $\hat{x}_n^b$ and their corresponding prediction errors e_n^f, e_n^b:

$$\hat{x}_n^f = -\sum_{k=1}^{p} a_k x_{n-k}, e_n^f = x_n - \hat{x}_n^f, \; n = p, p+1, \ldots, N-1$$

$$\hat{x}_n^b = -\sum_{k=1}^{p} a_k x_{n+k}, e_n^b = x_n - \hat{x}_n^b, \; n = 0,1,\ldots, N-p-1. \quad (19)$$

He reasoned that he could get the best estimates for a_1, a_2, …, a_p by minimizing the sum of squares of the predictions' errors, for example,

$$\sum_{n=p}^{N-1} \left|e_n^f\right|^2 + \sum_{n=0}^{N-p-1} \left|e_n^b\right|^2 . \quad (20)$$

He was able to devise a recursive algorithm that gave estimates not only for a_1, a_2, …, a_p, but also, at the same time, for ρ_e and for the autocorrelations ρ_0, ρ_1, …, ρ_p. The details are complicated, so we won't give them here.[4] It's remarkable that the recursion generates a new estimator for the elements of the matrix in Equation 18 at the same time it's solving the system of equations!

Choosing the Order *p*

Like the other AR methods, the ME method requires the choice of an order $p < N$. Figure 5 exhibits the results of choosing a low, intermediate, and high order for the time series generated by Equation 10. The same plots are repeated using a logarithmic scaling in Figure 6. Table 2 gives the peak locations. The *ME*(3) spectrum gave the best estimate $\hat{f}_0$, but its peak is almost as broad as the periodogram. Increasing p produces sharper peaks, but the locations display a noticeable downward bias. The *ME*(14) estimate is fairly representative of the orders in the range $4 \le p \le 25$. At $p = 26$, the peak splits into two, with the dominant one giving a better $\hat{f}_0$ than any of the sharp single peaks for $p = 4, 5, \ldots, 25$. The same splitting occurs for orders $p = 27, 28, 29$, and 30, with the dominant peak becoming sharper and sharper but remaining at $\hat{f}_0 = 0.492$. These spurious splittings aren't caused by errors in the data. In fact, they occur much more readily for artificially

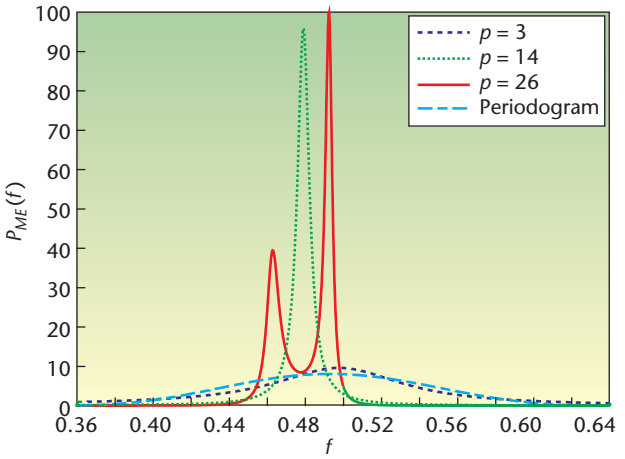

Figure 5. Maximum entropy power spectral density (PSD) estimates. For orders $p = 3$, 14, and 26, and the periodogram for the time series generated by Equation 10, we see plots along the same frequency range used for the *AR*(*p*) spectra in Figure 2. The ME peaks are even sharper than the *AR*(*p*) peaks, so they must be taller to preserve the area subtended.

Table 2. Peak locations.

Estimate	Periodogram	*ME*(3)	*ME*(14)	*ME*(26)
$\hat{f}_0$	0.493	0.498	0.479	0.492

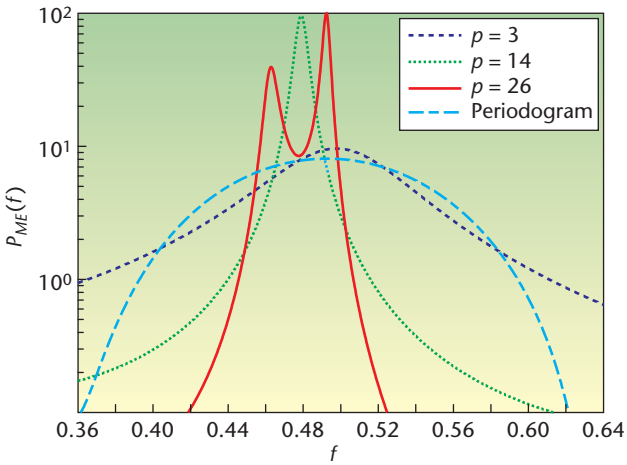

Figure 6. Another view of the plots given in Figure 5. Using the logarithm scale makes it easier to compare the *ME*(3) estimate with the periodogram.

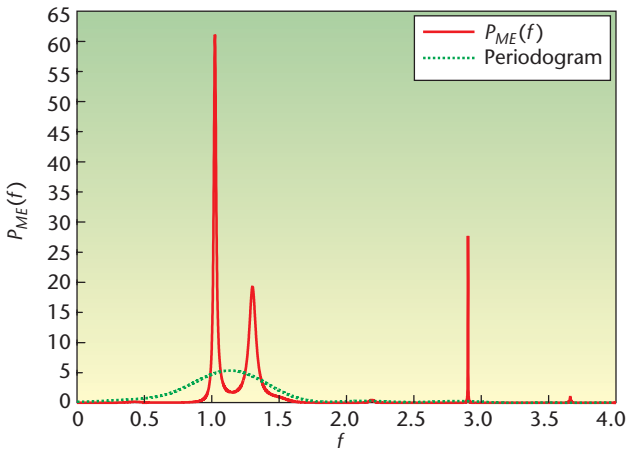

Figure 7. Maximum entropy (ME) method. In the *ME*(14) power spectral density (PSD) estimate for the time series generated by Equation 11, the two peaks are centered at $\hat{f}_1 = 1.023$ and $\hat{f}_2 = 1.302$. These are somewhat better than the estimates from the *AR*(12) spectrum in Figure 4. The very narrow peak at $\hat{f} = 2.901$ is an artifact caused by using the very high order $p = 14$ (high relative to $N = 16$), but because it's so narrow, it doesn't indicate much power and thus can be safely ignored.

generated time series without added noise, but the *ME*(26) spectrum clearly demonstrates that they also occur in noisy data, so great care must be exercised in interpreting high-order ME spectra. One of the ME method's strengths is its ability to resolve closely spaced peaks, but in using it for that purpose, always remember the possibility of a spurious splitting of a single peak.

Researchers have proposed several criteria for choosing the optimal order for the ME method (and for the other AR methods), but none of them work all of the time. In fact, it's easier to find a time series that confounds a given criterion than it is to develop it. Many authors[5,6] recommend $p \leq N/2$, but higher order methods often give better results. Figure 7 shows the result of using a relatively high p for the time series generated by Equation 11. The very narrow spurious peak at $\hat{f} = 2.901$ is a typical occurrence when we use high values for p. Such peaks can usually be easily identified because they're so much sharper than the peaks corresponding to real power. The one in Figure 7 is a small price to pay for the excellent resolution of the two real peaks. It's amazing that the ME method can achieve such good results using just 16 noisy data points spanning only ≈ 2.5 cycles of the higher frequency sine wave.

We've now looked at four different methods of spectrum estimation, and although we haven't exhausted the subject, we must proceed. (More details about this topic appear elsewhere.[5,6]) In the next installment, we'll take a brief look at filters and detrending before we present an analysis of a bat chirp. In the final installment, we'll discuss some statistical tests and use them to analyze atmospheric pressure differences in the Pacific Ocean that have significant environmental implications. CiSE

References

1. B. Rust and D. Donnelly, "The Fast Fourier Transform for Experimentalists, Part III: Classical Spectral Analysis," *Computing in Science & Eng.*, vol. 7, no. 5, 2005, pp. 74–78.
2. N. Levinson, "The Wiener (Root Mean Square) Error Criterion in Filter Design and Prediction," *J. Mathematical Physics*, vol. 25, 1947, pp. 261–278.
3. D. Donnelly and B. Rust, "The Fast Fourier Transform for Experimentalists, Part I: Concepts," *Computing in Science & Eng.*, vol. 7, no. 2, 2005, pp. 80–88.
4. J.P. Burg, *Maximum Entropy Spectral Analysis*, PhD dissertation, Dept. of Geophysics, Stanford Univ., May 1975; http://sepwww.stanford.edu/theses/sep06/.
5. S.L. Marple Jr., *Digital Spectral Analysis with Applications*, Prentice Hall, 1987.
6. S.M. Kay, *Modern Spectral Estimation: Theory and Application*, Prentice Hall, 1988.

Bert Rust is a mathematician at the US National Institute for Standards and Technology. His research interests include ill-posed problems, time-series modeling, nonlinear regression, and observational cosmology. Rust has a PhD in astronomy from the University of Illinois. He is a member of SIAM and the American Astronomical Society. Contact him at bwr@nist.gov.

Denis Donnelly is a professor of physics at Siena College. His research interests include computer modeling and electronics. Donnelly has a PhD in physics from the University of Michigan. He is a member of the American Physical Society, the American Association of Physics Teachers, and the American Association for the Advancement of Science. Contact him at donnelly@siena.edu.

Editors: David Winch, winch@taosnet.com
Denis Donnelly, donnelly@siena.edu

THE FAST FOURIER TRANSFORM FOR EXPERIMENTALISTS, PART V: FILTERS

By Denis Donnelly

THIS FIFTH SEGMENT IN WHAT HAS TURNED OUT TO BE A SIX-PART SERIES ON THE FAST FOURIER TRANSFORM (FFT) WILL BE THE LAST TO DEAL WITH METHODS OR ideas associated with the transform process itself. In case you're just tuning in, part one provided an introduction to the concepts associated with the FFT process,[1] part two treated convolutions,[2] part three provided a discussion of classical spectral analysis,[3] and part four continued that discussion, presenting autoregressive spectral analysis and the maximum entropy method.[4] Part six will analyze a bat chirp.

This current segment discusses filters. Filtering implies a frequency-dependent selection process, passing, for example, frequencies within a certain range and rejecting those outside that range. Later, I'll describe some essentials for creating frequency-dependent passbands and stopbands. I'll also briefly cover detrending and the cumulative periodogram.

Filters

Raw data is rarely in an ideal form for analysis. To separate the desired signal from the data, filtering can provide much assistance. With digital techniques, we can construct a wide variety of filter types, including low-pass, high-pass, bandpass, and notch filters. The typical approach for implementing a nonrecursive filter is to convolve[2] the input signal with the filter's impulse response.

Figure 1 summarizes the interrelations between the input signal, filter impulse response, and transform processes, where $y(t)$ is the signal to be filtered, $Y(f)$ is the Fourier transform of $y(t)$, $h(t)$ is the filter's impulse response function (also referred to as the filter kernel), $H(f)$ is the filter's transfer function or frequency response, $z(t)$ is the filtered signal, and $Z(f)$ is the Fourier transform of $z(t)$. The interrelations shown in Figure 1 are equivalent to the interrelations between time and frequency domain operations that lead to convolution (see Figure 4 of part two[2]).

One pathway to determine a filter's impulse response is to start with the desired frequency response and take the inverse transform to find the filter kernel. Conversely, the FFT of any impulse response yields that filter's frequency response. If we consider the sinc function to be the impulse response

$$\text{sinc}(x) = \begin{cases} 1 & \text{for } x = 0 \\ \dfrac{\sin(x)}{x} & \text{otherwise,} \end{cases}$$

then taking this function's FFT yields a low-pass filter's transfer function (see Figure 2). Although the ideal low-pass filter would have a discontinuity—passing with unity gain all signals below the cutoff frequency and zero gain for all signals above the cutoff—such performance isn't achievable in practice. A transition region as well as a ripple will always exist. In this example, we write the sinc function with arguments as

$$\text{sinc}(i) = \begin{cases} 1 & \text{for } i = N/2 \\ \dfrac{\sin(2\pi f_{co}(i/N - 1/2))}{2\pi f_{co}(i/N - 1/2)} & \text{otherwise,} \end{cases}$$

where f_{co} is the arbitrarily selected cutoff frequency—here, 35 Hz and where

$$N = 256 \qquad \Delta t = 1/N$$
$$i = 0 \ldots N-1 \qquad t_i = i \cdot \Delta t.$$

The use of a modified sinc function changes the filter from low pass to high pass, a process referred to as *spectral inversion*. This inversion reverses the filters' frequency response, changing passbands to stopbands and vice versa. The modified function[5] with the arguments used in this example is

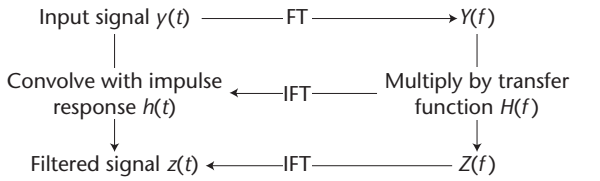

Figure 1. Time and frequency domain operations that lead to filtering. The filtered signal is obtained by convolving the input signal with the impulse response or by multiplying the Fourier transforms (FTs) of the input signal and the impulse response followed by an inverse Fourier transform (IFT).

$$\text{modsinc}(i) = \begin{cases} \dfrac{N}{2f_{co}} - 1 & \text{for } i = N/2 \\ \dfrac{-\sin(2\pi f_{co}(i/N - 1/2))}{2\pi f_{co}(i/N - 1/2)} & \text{otherwise.} \end{cases}$$

Figure 3 shows the function and its transform.

As an example of low-pass filtering, let's consider two signals, each composed of two sine waves with different frequencies. Each signal has a 5-Hz component and a second frequency either just above (36 Hz) or just below (34 Hz) the filter's cutoff frequency (35 Hz). The signals are defined by

$$y1_i = \sin(2\pi \cdot 5 \cdot t_i) + \sin(2\pi \cdot 34 \cdot t_i)$$
$$y2_i = \sin(2\pi \cdot 5 \cdot t_i) + \sin(2\pi \cdot 36 \cdot t_i)$$
$$t_i = i \cdot dt,$$

and N and Δt are as specified earlier. Figure 4 shows signals before filtering, and Figure 5 shows filter characteristics in the transition region.

The filtered signals in Figure 6 (p. 95) are determined by taking the inverse FFT (IFFT) of the element-by-element product of the signals' FFT, defined earlier, and the filter kernel's FFT, which in this low-pass filter case is the sinc function. Filter operation is easily understood. In the pass region, the signal's FFT is multiplied by numbers with a magnitude close to one, whereas in the non-pass region, the signal's FFT is multiplied by numbers with a magnitude close to zero. The inverse transform then reconstructs a signal based on the frequency components that remain.

A band-pass filter (see Figure 7) is created from a combination of a low-pass filter and a high-pass filter, where the low-pass filter cutoff's frequency is greater than the high-pass filter's cutoff frequency. A pseudocode expression for a band-pass filter, in which a smoothing window is applied, looks like this:

$$bandpass = \frac{}{\overrightarrow{fft(lowpassfcn \cdot window)} \cdot \overrightarrow{fft(highpassfcn \cdot window)}}.$$

The vectorize arrow implies element-by-element multiplication and the *lowpassfcn* and *highpassfcn* are the corresponding filter kernels with selected transition frequencies. In Figure 7, we use a Blackman window to reduce the ringing. Recall that the Blackman window is defined[1] as

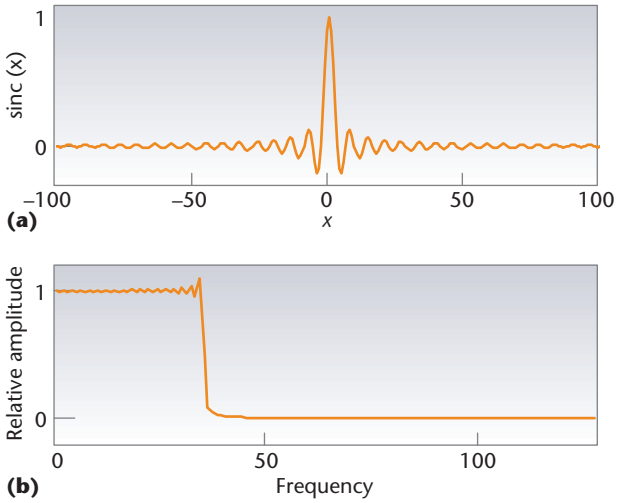

Figure 2. The sinc function and its Fourier transform. (a) The sinc function is considered here as an impulse response; (b) a fast Fourier transform (FFT) of the sinc function yields the filter's transfer function.

$$w_i = 0.42 - 0.5 \cdot \cos(2\pi i/N) + 0.08 \cdot \cos(4\pi i/N).$$

Two other windows commonly used in filtering are the Kaiser and the Chebyshev. As noted previously,[1] two performance factors for windows are the full-width half-maximum of the central lobe and the relative size of the side lobes. From narrowest to broadest central lobes, the window order is rectangular, Kaiser, Chebyshev, and Blackman, but the order is reversed from smallest to largest for the side lobes.

A moving average filter, depicted in Figure 8, helps remove noise by replacing each point's magnitude with the average value of itself and its neighbors—we would give a five-point moving average for the 12th data point, for example, by $x(12) = [x(10) + x(11) + x(12) + x(13) + x(14)]/5$. Performing such a

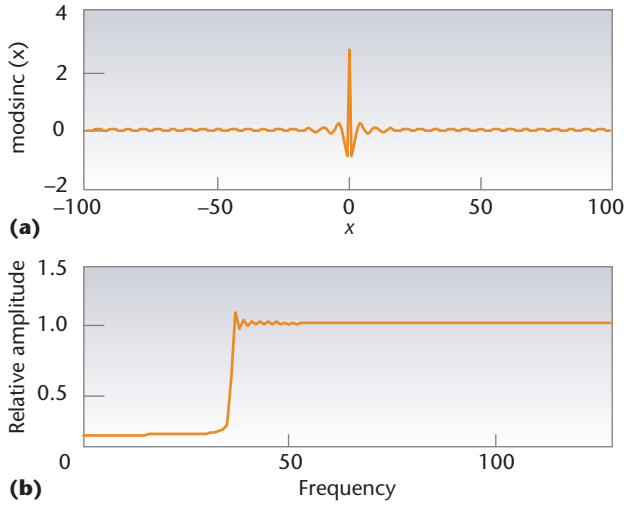

Figure 3. The modified sinc function and its Fourier transform. (a) The modified sinc function acts as a high-pass filter's impulse response. (b) The transform shows the transfer function—here, with a cutoff frequency of 35 Hz.

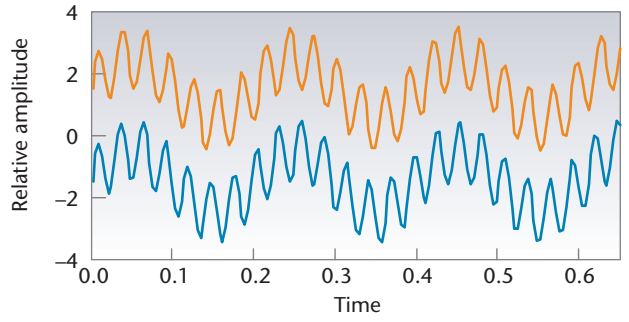

Figure 4. Signals before filtering. The upper curve includes the frequencies of 5 and 34 Hz, and the lower curve includes the frequencies 5 and 36 Hz. Curves are offset for visibility.

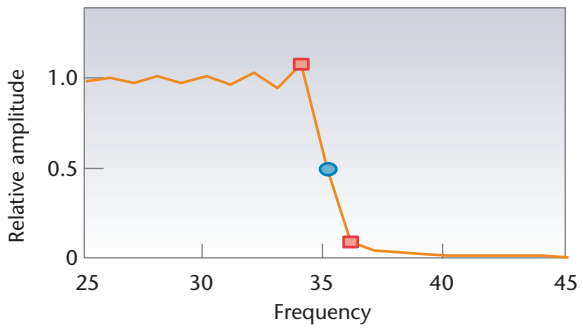

Figure 5. Filter response in the transition region. The pass amplitude of the filter at the cutoff frequency of 35 Hz (circular marker) is approximately half the amplitude in the pass band. The box markers show where on the transfer curve the 34 and 36 Hz points occur.

filtering process has the effect of reducing the slope of any edges that might be present. The filtering can be achieved by convolving a rectangular window, symmetrically placed with equal contributions at the beginning and end of the window, with the data set. (To take an average with an equal number of points above and below the point of interest, the window should be symmetric about zero, which means that for an average taken over five data points, win(0), win(1), win(2), win(N–1), and win(N–2) should be set to one-fifth with all the other points being set to zero.) If the rectangular window isn't symmetrically placed, the output signal will shift relative to the input signal.

Detrending and the Cumulative Periodogram

Although space limits discussion of detrending and the cumulative periodogram in this issue, a few comments are appropriate. *Detrending* is the process of removing an undesired trend in a time series. Time-domain data is typically obtained while eavesdropping on an ongoing process. During this sampled time, some signals could have such sufficiently low frequencies that their period is much greater than the sampling time. The FFT can't recognize such low-frequency signals—for example, if the signal includes a sine function in which the time span includes only a small fraction of a cycle, the FFT won't indicate such a frequency. (It's difficult to say precisely what fraction of a sine wave is needed before the FFT gives a clear indication of such a frequency; two-thirds of a cycle is probably enough to render it visible whereas one-half cycle is not.). If such a signal has a relatively large amplitude, it can mask the signals that are of interest. There could also be non-cyclical trends in the data. Furthermore, during the sampling time, there could be some drift in the instruments recording the measurements. Any of these effects can mask the signals you could be hoping to see, and, if possible, such trends should be removed (by least squares or other methods) before performing any planned analysis.

The *cumulative periodogram* provides an approach to statistically testing the validity of a signal's spectral features. It does this by providing a mechanism to test whether a peak at a particular frequency shows a statistically significant departure from white noise. The cumulative periodogram sums the normalized elements of the periodogram up to the index of interest. When plotted, the cumulative periodogram shows the running sum. Peaks in a periodogram produce deviations from the path associated with white noise in a cumulative periodogram. To observe this, we choose a probability level to demar-

cate a confidence band for white noise and plot the lines indicating that confidence level together with the cumulative periodogram of interest.

Looking ahead, in part six, I will analyze a bat chirp. One aspect of such a chirp that makes it interesting to consider is the fact that the times series represents a non-stationary signal, in which both the signal's frequency structure and the amplitudes of the various frequency components vary in time.

References

1. D. Donnelly and B. Rust, "The Fast Fourier Transform for Experimentalists, Part I: Concepts," *Computing in Science & Eng.*, vol. 7, no. 2, 2005, pp. 80–88.
2. D. Donnelly and B. Rust, "The Fast Fourier Transform for Experimentalists, Part II: Convolution," *Computing in Science & Eng.*, vol. 7, no. 4, 2005, pp. 92–95.
3. B. Rust and D. Donnelly, "The Fast Fourier Transform for Experimentalists, Part III: Classical Spectral Analysis," *Computing in Science & Eng..*, vol. 7, no. 5, 2005, pp. 74–78.
4. B. Rust and D. Donnelly, "The Fast Fourier Transform for Experimentalists, Part IV: Autoregressive Spectral Analysis," *Computing in Science & Eng.*, vol. 7, no. 6, 2005, pp. 85–90.
5. S.W. Smith, *Digital Signal Processing*, Newnes, 2003.

Denis Donnelly is a professor of physics at Siena College. His research interests include computer modeling and electronics. Donnelly has a PhD in physics from the University of Michigan. He is a member of the American Physical Society, the American Association of Physics Teachers, and the American Association for the Advancement of Science. Contact him at donnelly@siena.edu.

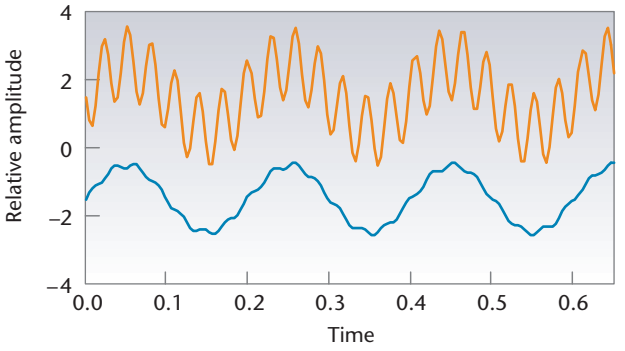

Figure 6. Signals after low-pass filtering (the cutoff frequency is 35 Hz). The upper curve includes the frequencies of 5 and 34 Hz, and the low curve includes the frequencies of 5 Hz and a significantly reduced amplitude 36 Hz signal (see Figure 4).

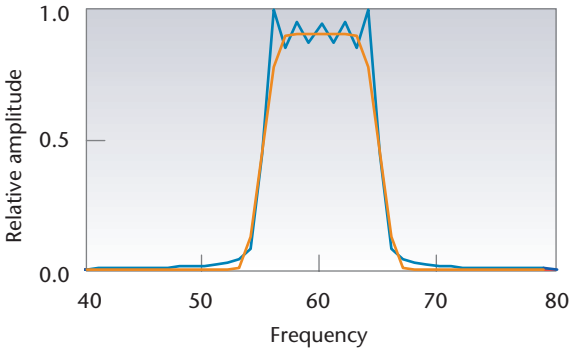

Figure 7. Band-pass filter response (curve with ringing) and the same filter with the smoothing of a Blackman window. There is a slight reduction in the rate of roll-off when a window is applied. The low-pass cutoff frequency is 65 Hz; the high-pass cutoff frequency is 55 Hz.

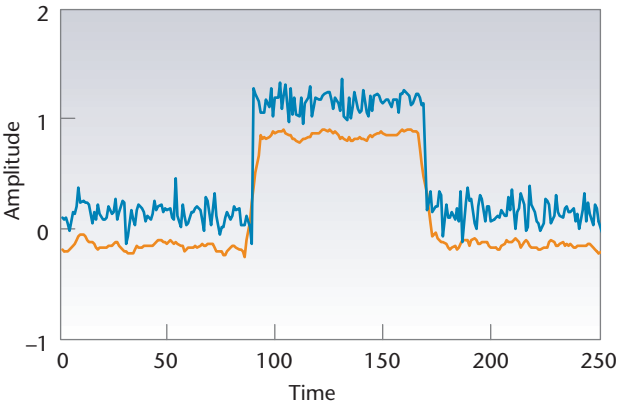

Figure 8. Noise reduction using a moving average filter with a seven-point average. For visual clarity, the signals are offset.