PSYCHOPHYSIOLOGY

Vol. 29, No. 3
Printed in U.S.A.

Methodology

Digital Filtering: Background and Tutorial for Psychophysiologists

EDWIN W. COOK III
University of Alabama at Birmingham

AND GREGORY A. MILLER
University of Illinois at Urbana-Champaign

ABSTRACT

Digital filtering offers more to psychophysiologists than is commonly appreciated. An introduction is offered here to foster the explicit design and use of digital filters. Because of considerable confusion in the literature about terminology important to both analog and digital filtering, basic concepts are reviewed and clarified. Because some time series concepts are fundamental to digital filtering, these are also presented. Examples of filters commonly used in psychophysiology are given, and procedures are presented for the design and use of one type of digital filter. Properties of some types of digital filters are described, and the relative advantages of simple analog and digital filters are discussed.

DESCRIPTORS: Filters, Fourier analysis, Time constant, Digital signal processing.

The processing of psychophysiological signals nearly always includes some type of filtering. Most often, this filtering is done by means of electronic circuits that are built into the recording amplifiers or electrically interposed between the amplifier and the recording device such as the analog-to-digital (A/D) converter of a laboratory computer. Such electronic (or "analog") filters may be contrasted with an alternative category of filter that is applied to the signal after it has been numerically recorded. These digital filters have advantages that have been discussed in the engineering literature, but these sources may not have been readily accessible to psychophysiologists.

The purposes of the present report are to: a) discuss and clarify certain important points of confusion in the literature regarding analog and digital filtering; b) draw attention to digital filters and their characteristics; and c) describe methods for efficiently constructing, evaluating, and implementing one type of digital filter in the context of computerized processing of psychophysiological data. We will not attempt an extensive mathematical treatment of the topic of digital filters but will instead emphasize practical information that is likely to be of greater use to psychophysiologists.

In a chapter on signal extraction techniques for event-related potential (ERP) research, Ruchkin (1988) includes an extensive overview of digital filtering methods for the advanced reader. Farwell, Martinerie, Bashore, and Rapp (in press) assume much of the background discussed in the present tutorial and present a number of illustrations of a slightly different type of digital filter than that discussed in detail here. These works provide valuable follow-up to the introductory material in the present paper.

The authors wish to thank Jean M. Anhalt, Lawrence A. Farwell, Gabriele Gratton, Blair D. Hicks, Stephen W. Porges, David L. Roth, and Daniel S. Ruchkin for comments on earlier drafts of this paper. Steve Porges and Evan Byrne provided coefficients for the moving polynomial filters. We would also like to thank Bruce C. Wheeler for consultation on the electrical engineering literature. Preparation of this paper was supported in part by National Institute of Mental Health grant MH39628.

Address requests for reprints to: Edwin W. Cook III, Department of Psychology, University of Alabama at Birmingham, Birmingham, AL 35294; or Gregory A. Miller, Department of Psychology, University of Illinois, 603 E. Daniel Street, Champaign, IL 61820.

Filtering Basics

The simplest and most common electronic filters used in psychophysiological recording are *high-pass* and *low-pass*, which selectively attenuate low-frequency and high-frequency components, respectively. Such filters may be used as building blocks to construct *band-pass* filters, which selectively attenuate frequency components above and below a range of interest. For example, a band-pass filter for EMG recording might "pass" frequencies in the 10–1000 Hz range. Another hybrid, the *band-stop* filter, selectively attenuates frequency components within a specified range. The most commonly used band-stop filters attenuate a narrow range of frequencies in the vicinity of power-line noise (50 or 60 Hz), and are generally referred to as *notch filters*. For purposes of clarity, the following discussion will be primarily restricted to simple high-pass and low-pass filters, although we will return to the others in the discussion of digital filter design.

The degree to which a particular filter will pass various frequencies is typically represented as its *gain function*, where gain is the ratio of output amplitude to input amplitude as a function of frequency (sometimes power is used instead of amplitude—see below). Figure 1 illustrates the gain function for a low-pass filter. The range of frequencies that a filter will pass without substantial attenuation is referred to as its *pass band*. The range of frequencies in which little energy is passed is referred to as the *stop band*. The range of frequencies in which gain is intermediate is referred to as the *transition band*, approximately 13–37 Hz in Figure 1. Whether analog or digital, more complex filters tend to have narrower transition bands, which may be required in situations where the signal of interest and the noise or artifact to be rejected contain nearly adjacent frequency components.

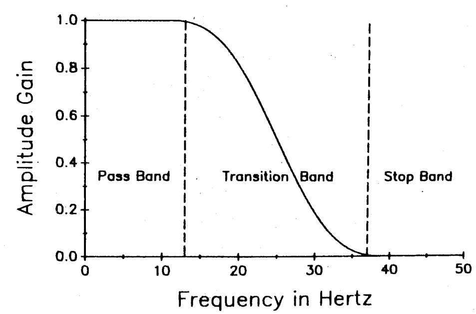

Figure 1. The gain function of a filter is divided into the pass band, transition band, and stop band. The gain function shown is for a low-pass filter.

The wide variety of ways in which transition bands are characterized in the psychophysiology literature is one of the more confusing aspects of filtering. Typically, some specific frequency within the transition band is reported as the "cutoff" or "corner" frequency, f_c. The rule for determining f_c varies across sources and even across polygraph manufacturers. The other feature of transition bands typically reported is the "roll-off" rate, usually expressed in dB per unit of change in frequency. This is unambiguous in the abstract but somewhat inconsistent in practice, because sources vary in their choice of unit of frequency—usually octave (a doubling or halving of frequency) or decade (a tenfold change in frequency). Thus, a 6dB/octave filter has a much narrower transition band (steeper roll-off) than a 6dB/decade filter, though both could be described as having a "6dB roll-off."

The inconsistencies in definition of cutoff frequency are problematic and warrant extended discussion. In the electronic engineering literature (e.g., Ludeman, 1986; Malmstadt, Enke, & Crouch, 1974), f_c is defined fairly consistently as the half-*power* frequency—that frequency within the transition band where the gain (ratio of output power to input power) is .5. In decibels, a gain of .5 equates to $10 \log_{10} (.5) = -3$dB. Thus, f_c is often referred to as the frequency at which the gain is "3dB down." Every 3dB decrease is a further halving of the *power*.

Some standard sources in the psychophysiology literature also define f_c as the half-power frequency (e.g., Ruchkin & Glaser, 1978). However, some sources treat f_c as the half-*amplitude* frequency, which is not the half-*power* frequency. (Power and amplitude are related but not identical measures of the magnitude of a signal. The relationship between these measures is discussed in detail below.) For example, Douglas-Young (1981) defines the cutoff frequency as "that point on the response curve where the amplitude response has decreased to ½ of its original value" (p. 137). The manual for a widely used polygraph (Grass Model 12 Neurodata) also uses half-amplitude to define its cutoff frequencies (Grass Instrument Co., 1985). Thus, although reliance on the half-amplitude criterion is less standard, it is not uncommon. In fact, it will be quite useful in the design of digital filters by the method described below.

The confusion and inconsistency in the literature is abetted by the numerical relationship of .707 and .5 (the latter being the square of the former) and the common treatment of power as related to the square of amplitude (power = voltage^2/resistance; resistance is implicitly set to 1.0—i.e., ignored—in most discussions of these issues). At the frequency at which gain in power is .5, gain in amplitude is $\sqrt{.5}$ = .707. This half-power frequency is the definition of f_c employed in the familiar equation for a simple resistor-capacitor circuit, where R (resistance in ohms) · C (capacitance in farads) = TC (time constant in seconds). Thus, $TC = 1/(2\pi f_c) = .159/f_c$, or $f_c = .159/TC$. This choice of emphasis on gain in terms of output/input power versus output/input amplitude is essential for precisely communicating the characteristics of a filter. For a polygraph with filter settings labeled on the basis of half-*amplitude*, using those settings in the $TC = .159/f_c$ equation will produce an incorrect value for the time constant.

There is considerable confusion among psychophysiologists on this point, perhaps because of pragmatic but inconsistent nomenclature on polygraphs. Thus, for example, on the Grass Model 12 the filter settings refer to the half-amplitude frequencies (Grass Instrument Co., 1985), not the half-power frequencies. In fact, Grass Instruments told us that they have never determined the time constants associated with the various filter settings on the Model 12 (Glenn Spohr, personal communication, December 15, 1988). The graph of gain functions associated with the half-amplitude filter settings for the 12A5 amplifiers in the Model 12 manual indicates that the ".01 Hz" setting has a half-power gain at approximately .017 Hz, which would mean a time constant of .159/.017 = 9.4 s. In contrast, for the Grass Model 6 the frequency settings on the 6A5 amplifier refer to the point at which the amplitude gain is .8. Its ".3", "1", and "5" settings actually provide time constants of approximately .4, .12, and .05 s (Grass Instrument Co., 1968) and thus power-defined cutoff frequencies of .4, 1.3, and 3.2 Hz. The 7P122 amplifier of the Grass Model 7 polygraph has settings labeled "TC .8" and "TC .1", providing half-amplitude frequencies of .04 and .4 Hz (Grass Instrument Co., 1973). However, Grass told us that these settings provide time constants of 2.2 and .25 s, respectively, implying power-defined cutoff frequencies of .07 and .64 Hz.

Equipment from some other manufacturers appears to use the half-power nomenclature. Settings of "Hi Freq. Response" on the Beckman 411 amplifier refer to f_c defined in terms of power. The Beckman 9853 Voltage/Pulse/Pressure coupler and the Nihon Kohden Neurofax polygraph have settings that are explicitly labeled in both "TC" and "Hz", with values indicating that the "Hz" settings refer to power-defined f_c (e.g., both devices have a setting labeled both TC=1.0 and Hz=.16). Coulbourn labels its filters with the f_c only; our empirical tests indicate that the labels on their S75-01 Bioamplifier with Filters also refer to power-defined f_c.

The characterization of a filter in terms of the dB roll-off in its transition band was noted above. Confusion similar to that for f_c can now be discussed. As an alternative to gain as a simple ratio of output to input typically ranging from 0.0–1.0, a parallel way to characterize the cutoff frequency of a filter is as the frequency that alters gain by a certain number of decibels (dB), with negative values meaning a gain less than one. Thus, dB, a function of the log of the ratio of output to input values, is also a source of some confusion, because it may be computed on the basis of amplitude or power values. Decibels are usually expressed in terms of amplitude, computed as $20 \log_{10} (V_{out}/V_{in})$, but this is equivalent to $10 \log_{10} (P_{out}/P_{in})$, with V = voltage and P = power. The −3dB frequency is the frequency at which gain is .707 amplitude = .5 power = f_c. Again, however, because polygraph filter specifications are instead often expressed in terms of the −6dB frequency (e.g., Grass Model 12), which is the half-*amplitude* frequency, confusion can result when trying to describe a filter in terms of frequency and dB.

We believe that this confusion about the basis for specifying f_c and about the relationships among f_c, time constant, gain, dB, power, and amplitude is pervasive in the psychophysiology literature. Because half-power and half-amplitude standards both appear to be well established, we do not propose that only one be used. However, it is essential to state which standard was used when reporting f_c, and it is essential to exercise care in deriving time constant values from the f_c stated for a particular polygraph, coupler, or filtering device. Although these imperatives follow primarily from confusion in the specification of analog filters, they apply to digital filters equally well.

Digital Filters: Background

Having reviewed and clarified some general principles and issues in filtering, we now address digital filtering more specifically. The term "digital filter" may be applied to a wide range of techniques that have in common only the fact that they are mathematical procedures that are applied to discrete numeric representations of continuous waveforms to selectively attenuate certain frequencies.

Psychophysiologists using a wide range of physiological measures frequently work with such representations, examples of which include the sampled voltage between two EMG electrodes or between a scalp EEG electrode and its reference, the circumference of the chest during respiration, and the serum concentration of corticosterone. Each of these parameters varies continuously over time and may be recorded numerically at equal intervals,

yielding a *time series* of observations of the form:

$$X_t, X_{t+p}, X_{t+2p}, X_{t+3p}, \ldots X_{t+np}.$$

The subscripts refer to the time at which the associated X is observed, such that t is the time at which recording began and p is the *sampling period* (the time between adjacent samples). Event series such as heart periods can be converted to time series with a constant sampling period (Cheung & Porges, 1977; Graham, 1978; Miller, 1986). A time series may be plotted to represent the original waveform, as illustrated in Figure 2. Several texts, including Bloomfield (1976), Brillinger (1975), Gottman (1981), and Jenkins and Watts (1968) provide detailed mathematical treatments of time series and their analysis. The following sections on aliasing and representation of time series in the frequency domain address two specific issues in time series analysis that are especially relevant to digital filtering of psychophysiological data.

Aliasing. In order for a time series to represent a continuous waveform adequately, the *sampling rate* (f_s, the inverse of the sampling period) must be at least twice the fastest frequency present in the original waveform. This requirement follows from the fact that only if samples are obtained at least twice per cycle can a discrete time series accurately represent the frequency of a sine wave. This axiom is referred to as *Nyquist's rule*, and one-half the sampling frequency is referred to as the *Nyquist frequency*. If the rule is violated, the resulting digitized waveform may contain low-frequency components not present in the original data. This phenomenon is known as *aliasing*. It is important to note that Nyquist's rule requires sampling at twice the fastest frequency present in the original waveform, which may be higher than the fastest frequency in which the investigator is interested.

Because of possible errors in estimating the highest frequencies in real-world data and possible noise introduced by amplifiers and A/D converters, it is commonly suggested (e.g., Attinger, Anne, & McDonald, 1966; Coles, Gratton, Kramer, & Miller, 1986) that sample rate be as much as 5–10× higher than the Nyquist rule suggests. On the other hand, when it is believed that the noise power introduced at high frequencies is minimal, a more convenient sample rate is typically used and the resulting aliased noise ignored. A second situation in which the Nyquist rule can sometimes be violated is when there is a single high-frequency noise source (e.g., power-line noise) and it is possible to set the sampling period to an exact multiple of the period of that noise source. By thus sampling the noise at the same phase of its cycle with every sample, periodic contributions of the noise to the sampled waveform are removed. However, DC distortion as large as ½ the amplitude of the noise may still be present. Considerations related to Nyquist's rule and avoidance of aliasing are important in selecting parameters for digital data acquisition and filtering. A more precise treatment of aliasing is given in Appendix A.

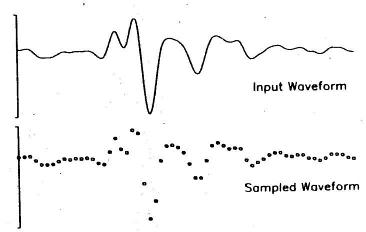

Figure 2. A continuous waveform representation of an electromyographic response is shown (upper panel) along with its digital representation (lower panel).

Representing waveforms in the frequency domain. A time series that indicates voltage or some other parameter as a function of time is said to be a representation "in the time domain." An alternative representation of the same information is based on the principle (Fourier's theorem) that any waveform that is stationary (i.e., from which long-term DC shifts have been removed and in which the underlying frequency components do not change over time) may be represented as the sum of a set of sinusoidal waveforms, each of a different frequency and having an associated amplitude and phase angle. This principle is the basis of *Fourier analysis*, which yields an indication of the amplitudes and phases of the constituent sinusoids as a function of frequency. This representation of a signal is said to be "in the frequency domain." A *direct Fourier transform* converts a digitally represented signal from the time domain to the frequency domain; an *inverse Fourier transform* does the converse.

The requirement of stationarity is important for this equivalence, and real-world psychophysiological data will often violate it. Nevertheless, the interchangeability of time-domain and frequency-domain representations of a given waveform bears emphasis. Consider a set of j sine waves, each characterized by a frequency, an amplitude, and a phase. Frequency and amplitude for a given sine wave are constant. At some arbitrary time, the different sine waves may be at different points in their cycles (see Figure 3, Panels a, b, and c). Summing across the set of sine waves produces a single composite waveform in which the constituents may be difficult to identify, depending on their frequencies, amplitudes, and phase angles (Figure 3, Panel d). One

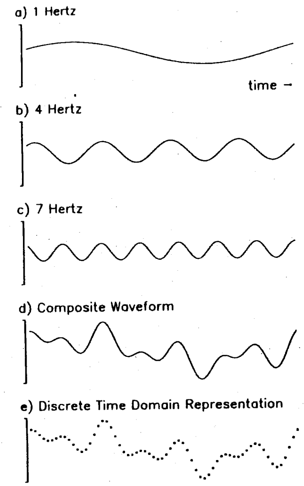

Figure 3. A recorded signal may be made up of multiple frequencies (Panels a, b, and c), which may be difficult to recognize when combined (Panel d). The signal may be represented in the time domain by a discrete time series (Panel e). The signal must be sampled at least twice as fast as the fastest constituent frequency (Panel c).

could digitize the composite waveform, describing it as a single vector of values arranged in time (plotted in Figure 3, Panel e). Alternatively, one could describe it with amplitude and phase vectors (spectra) arranged in order by frequency (plotted in Figure 4, Panels a and b). Either description—in the time domain or in the frequency domain—completely specifies the raw phenomenon illustrated in the composite waveform. One description may be more tractable for a particular type of analysis or more intuitively appealing for a particular type of question, but exactly the same information is available in the two representations. Appendix A details the computational steps for the direct and inverse

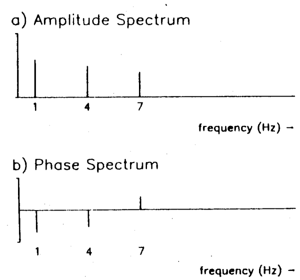

Figure 4. The signal shown in Figure 3 may be represented in the frequency domain by its amplitude and phase spectra.

Fourier transforms; that is, for shuttling between time- and frequency-domain representations.

Digital Filters: Classification and Implementation

It was noted earlier that a wide range of mathematical procedures applied to time series may be considered digital filters. If one simply requires that the procedure selectively attenuate certain frequencies, then in a general sense implicit digital filters are already widely used in psychophysiology. For example, the calculation of the mean of a time series may be construed as an implicit digital filter, specifically one that attenuates all frequencies except 0 Hz (DC). Alternatively, computation of the variance of the series removes the DC component while retaining (and combining) all other frequencies. Although these standard descriptive procedures may be construed as types of digital filters, we will restrict our presentation to those procedures that yield a time series (one from which certain frequencies have been removed) rather than a single value such as the mean.

In one broad classification of digital filters, each filtered point is defined as a weighted or unweighted average of some number of input data points. "Filtering" consists of replacing each input data value with the sum of the cross-products of the weights and the input data points, as follows:

$$Y_t = \sum_{i=-j}^{j} W_i \cdot X_{t+i} \qquad (1)$$

where W is the series of $2j+1$ weights (subscripted $-j$ to $+j$), X is the input time series, and Y is the filtered time series. The number of weights, the values of those weights, and the time-domain relationship (i.e., the lag) between the data value replaced and the data values used in the cross-multiplication determine the gain function of the filter. Note that although there are $2j+1$ weights, the weighting sequence is symmetric about an unpaired center weight W_0 (i.e., $W_k = W_{-k}$). The cross-multiplication and summing procedure is referred to as *convolution*.

An alternative approach to digital filtering defines each filtered point as a function of the current input point and one or more prior *filtered* points. The general form of such a filter is:

$$Y_t = X_t + \sum_{i=1}^{j} W_i \cdot Y_{t-i} \qquad (2)$$

where X and Y are defined as above, and W is the series of weights applied to prior filtered points.

The distinction between these approaches is frequently referred to as the filter's *impulse response*. Filters that define output points solely on the basis of input points are said to have *finite impulse response* (FIR), because the effect of a single aberrant input point (an "impulse") disappears after a finite amount of time; namely, after the last filtered point that includes the aberrant unfiltered point in its definition. In contrast, filters that define each filtered point in part based on prior filtered points have *infinite impulse response* (IIR), because the effect of a single aberrant point will propagate *ad infinitum*. "Nonrecursive" and "recursive" are synonymous adjectives describing digital filters with finite impulse response and infinite impulse response, respectively.

IIR digital filters represent something of a hybrid between analog and FIR digital filters and share characteristics of both. Their advantages and disadvantages will be discussed below, where analog and digital filters are compared. However, a thorough discussion of IIR filters is beyond the scope of the present paper (see Ackroyd, 1973, for such a discussion), and the following presentation will be restricted to the FIR variety unless otherwise specified.

FIR Filters in Psychophysiology

In the psychophysiology literature, a variety of FIR filters have been described, particularly for smoothing (i.e., removing high-frequency components from) time series. We consider several of these below, and then present a general approach to filter design and evaluation that is based on spectral analysis.

Smoothing of time series data is often accomplished by redefining each point in the original time series as the simple average of itself and a symmetric number of additional points before and after it. This filter is frequently referred to as a *moving-average filter*, reflecting the fact that computation of the average is repeated to define each filtered point. Additionally, this type of filter may be referred to as a "boxcar" filter, reflecting the shape of the weights plotted as a function of lag relative to the output point. Moving-average filters vary only in the number of data points averaged together. What gain function this translates into is a function of the number of data points and the sample rate. Ruchkin and Glaser (1978; Glaser & Ruchkin, 1976; Ruchkin, 1988) discuss equal-weight filters in detail and provide an equation for their gain. Although that discussion targets digital signal processing for ERP research, it applies more generally to digital filtering of any psychophysiological data.

A particular advantage of equal-weight moving-average filters is the rapidity with which each filtered point can be computed. In general for FIR filters with j weights, computation of each filtered point requires j multiplications and $j-1$ additions. But if the weights are equal, one can instead do $j-1$ additions and then a single division by j. The latter computations will always be faster. This speed consideration may be particularly important if the filter is to be implemented on-line or with computer hardware lacking in floating-point support.

A variation on the simple moving-average filter is one in which each filtered point is a differentially weighted average of the corresponding and adjacent input points. In perhaps the most common such filter, the weight series is {.25, .50, .25} and each filtered point X_t is therefore $(.25 \cdot X_{t-1}) + (.5 \cdot X_t) + (.25 \cdot X_{t+1})$. Bloomfield (1976) refers to application of this filter as *hanning*, and Glaser and Ruchkin (1976) describe its use in psychophysiology. Although computationally not as fast as a boxcar filter with 3 weights, approximate division of integers by 2 or 4 is rapidly accomplished on a digital computer by shifting binary digits to the right 1 or 2 places, respectively, and therefore does not depend on floating-point arithmetic.

The application to psychophysiological data of a more complex moving-average filter with unequal weights is described by Porges (1985). The derivation of Porges's approach is based on polynomial regression of overlapping segments of the time series on a variable representing time (Kendall, 1976; Hamming, 1989). The filter output is the predicted value of the time series at the midpoint of each segment. Both increasing the width of the filter and limiting the order of the regression polynomial to a low degree reduce the extent to which predicted values can oscillate at high frequencies; the moving polynomial filter thus functions as a low-pass filter. Despite its basis in regression, redundant calculations permit moving-polynomial filters to be implemented as moving-average filters with unequal weights (Hamming, 1989; Kendall, 1976; Porges, 1985).

Porges and his collaborators (e.g., Linnemeyer & Porges, 1986; McCabe, Yongue, Ackles, & Porges, 1985; Yongue et al., 1982) have used cubic moving polynomial filters with 15–51 weights to estimate the vagal contribution to heart rate variability (i.e., vagal tone). When used for this purpose, the filtered time series is subtracted from the original time series, yielding a second filtered time series with the low frequencies rather than the high frequencies attenuated. This subtraction procedure can be applied after any of the above low-pass (smoothing) filters to effectively convert them to high-pass filters.

Compared to the simpler equal-weight and {.25, .50, .25} filters described above, complex FIR filters can require much more time for computation. Although this may not be an issue if the filtering will be done off-line, depending on sample rate and available computational power, such filters can easily overwhelm typical laboratory minicomputers and microcomputers in on-line applications. However, even complex symmetric unequal-weight filters allow some speed-up, relative to a direct implementation of the definition: one can sum each pair of observations at equal distance from the midpoint and then multiply by the weight, cutting the number of multiplications in half.

Although moving-average filters with both equal and unequal weights are frequently used in data reduction of time series, their frequency characteristics are not generally reported and may not be generally recognized. Using frequency-domain methods described in the next section and Appendix B, the gain function for moving-average filters having any set of symmetric weights may be computed.

In addition to the explicit filtering and smoothing applications described above, a wide range of other procedures common in psychophysiology can be understood within an FIR framework. To calculate the mean of N points, such as for determining baseline levels, one employs a set of N weights, each with value 1/N. As noted earlier, this filter attenuates all but 0 Hz (DC). As a noise-reduction strategy, occasionally one chooses to remove an errant point and replace it with the average of its neighbors. Such a procedure can be considered the application of a filter consisting of the weights {.5, 0, .5} applied symmetrically around the errant point.

Particularly interesting are FIR filtering methods used in template-matching algorithms. The "template" can be seen simply as a set of weights with a particular configuration of values, and the weights may not be symmetric. The basis for selecting weights may differ greatly across applications, but in general it will reflect a specific notion the investigator has about the signal being sought. For example, if the template (the set of weights) is simply a 10-Hz sine wave, then cross-multiplication of that template with raw EEG will constitute an alpha band-pass filter. One might search EEG or EOG for blinks by establishing a filter template with weights that outline a blink. The Woody (1967) filter technique used for latency correction of event-related potentials uses as its template a portion of the precorrection waveform for a given subject; i.e., the template is customized for each subject. A simpler, common variation on the Woody technique employs a half sine-wave cycle or a half triangular-wave cycle as the template. In all of these examples, one slides the template along the data, cross-multiplies, and notes the latency of maximum cross-product as the likely latency of the signal for which one is filtering. These examples represent additional ways in which implicit digital filters are already widely used in psychophysiology.

Design and Evaluation of Digital Filters in the Frequency Domain

All of the FIR filters described above involve cross-multiplying a time series with a symmetric weight series (which may itself be considered a special-purpose time series), yielding a filtered time series. As described above, a time series can be represented in the frequency domain rather than the time domain. Representation of the original time series and the weight series in the frequency domain (see Appendix A) permits insights that lead to a general approach to design and evaluation of digital filters. Specifically, the amplitude spectrum of a filtered time series is equal to the amplitude spectrum of the original time series, multiplied frequency-by-frequency by the cosine component (A_h in Equation A.1 of Appendix A) of the weight series. Moreover, the power spectrum of the new time series is equal to the power spectrum of the original time series, multiplied frequency-by-frequency by the *squared* cosine component of the weight series. These properties are fundamental to the construction of FIR filters using Fourier transform methods.

The specific steps for constructing such filters have been described by Gold and Rader (1969; see also Ackroyd, 1973; Oppenheim & Schafer, 1975; Ruchkin, 1988), and software implementing the steps is available (e.g., Cook, 1981). Briefly, the technique involves four steps: First, the filter's ideal gain function is specified. Second, the inverse Fourier transform is applied to the gain function to obtain the initial set of weights. Note that this is a simple transformation from the frequency domain to the time domain, as described above. The gain function in the former becomes the set of weights in the latter. Third, it is usually desirable to reduce the number of weights and to taper the weights in order to balance requirements related to transition bandwidth, computational limits, maximum filter width, and "ripple" (the degree to which the gain function varies around unity in the pass band and around zero in the stop band). Finally, the reduced filter is evaluated and the process repeated until an acceptable filter is obtained. These steps are described in detail in Appendix B. It is assumed that the investigator has already made decisions about type of filter (high-pass, low-pass, etc.) and half-amplitude frequency or frequencies.

A related approach to digital filtering is also based on frequency-domain representation. This alternative approach requires three steps: First, a direct Fourier transform is used to transform the original time series to the frequency domain. Second, those elements of the transform that correspond to frequencies to be eliminated are set to zero. Third, an inverse Fourier transform recreates the original time series, minus those frequencies for which the direct transform was set to zero. Although this approach follows in a straightforward manner from the earlier discussion of frequency- and time-domain representations, it has some limitations. Specifically, it requires that filtering be delayed long enough to acquire a time series large enough for spectral analysis. This may present difficulties if filtered data are required in real time, such as for biofeedback. Secondly, if the signal is not stationary (i.e., does not have consistent mean and frequency components) over the epoch for which the Fourier transform is calculated, artifactual high-frequency noise can result (Attinger et al., 1966). Finally, discontinuities can result between adjacent epochs filtered by this method.

The frequency-domain methods described briefly above and detailed in Appendix B also provide a means of computing the gain functions for many of the digital filters in common use in psychophysiology. Thus, if any symmetric weight series is transformed to the frequency domain, its gain function is given by its cosine components. A consideration of several examples of this generalization illustrates several important points. Gain functions for some low-order equal-weight moving-average filters (Figure 5) indicate several characteristics of such filters. First, they have considerable ripple, inverting the signal (gain < 0.0) by 20% or more in several frequency ranges. Second, such filters have zero gain at all frequencies that are multiples of f_s/j and less than the Nyquist frequency, where j is the number of weights. Third, the half-amplitude cutoff frequency for an equal-weight filter is approximately $.61 \cdot f_s/j$ (half-power cutoff frequencies are approximately $.45 \cdot f_s/j$; the .45 is comparable to the .44 of Ruchkin and Glaser, 1978, and the .44–.47 of Ruchkin, 1988). This last point implies that j and f_c cannot be specified independently; because j must be an integer, available cutoff frequencies are fairly limited.

Using the same filter width and cutoff frequency, equal-weight filters may be compared to unequal-weight filters constructed by the steps described above and in Appendix B. Figure 6 illustrates that unequal-weight filters computed by the present method have less ripple and considerably narrower transition bands (i.e., sharper roll-off).

Ruchkin (1988) provides a detailed explanation of the use of equal-weight, moving-average filters for smoothing. In addition, he suggests that such filters be applied twice; i.e., the original time series X is filtered to create Y, and then Y is filtered again to create Z. The above comparison suggests a reason why multiple passes may be preferred when using equal-weight smoothing. Applying an equal-weight filter multiple times is equivalent to applying a tapered weight filter for a single pass. In addition, the filter's width is effectively increased to $2j-1$. As suggested in Appendix B, tapering the tails and increasing the width results in less ripple and a narrower transition band, exactly the result obtained by Ruchkin. For example, the weighting function for a fixed-weight filter of width 5 is {.2, .2, .2, .2, .2}. Two applications of this filter are mathematically equivalent to a single application of a variable-weight filter with the following weighting function: {.04, .08, .12, .16, .2, .16, .12, .08, .04}. An equal-weight filter is easier to comprehend and has potential speed advantages, as noted earlier. However, a single application of an unequal-weight filter provides the ability to customize the filter to achieve virtually any gain function that the physiological signal and experimental questions demand.

The present frequency-domain analysis of equal-weight filters provides a perspective on the ubiquitous problem of how long a baseline should be. It is clear that sensitivity to momentary fluctuations in a physiological process decreases as the length of the baseline increases. The present frequency-domain analysis describes that functional relationship mathematically. As noted above, a mean may be considered as an FIR filter with all weights equal to 1/N. If j points are averaged for a baseline, then the degree to which periodic activity of any frequency up to $f_s/2$ is reflected in that baseline can be computed using the methods described above and in Appendix B. In general, the shape of the relationship between number of points averaged for a baseline and sensitivity to periodic activity will be similar to that shown for selected values of j in Figure 5. Note that this analysis assumes that the physiological process is stable (stationary); that is, that the mean and frequency components do not change. For this reason it may be more useful in selecting durations for trial baselines than for session baselines.

Finally, the frequency characteristics of moving polynomial filters can also be analyzed using frequency-domain methods. Figure 7 presents amplitude gain functions produced by the 21-weight cubic polynomial given by Porges (1985) and by a 21-weight filter computed by the frequency-domain methods described briefly above and detailed in Appendix B. As shown, the frequency-domain method produces a filter with less ripple in the stop band, whereas the cubic polynomial filter has a nar-

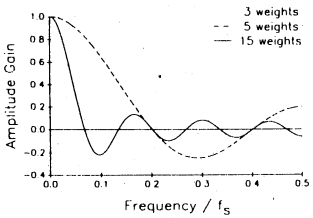

Figure 5. Gain functions for equal-weight digital filters with 3, 5, and 15 weights. f_s is the sampling rate.

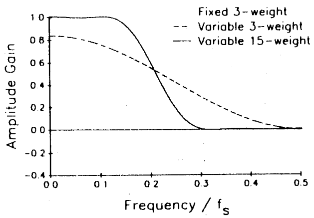

Figure 6. Gain functions for 3-weight digital filters constructed with fixed weights or by the frequency-domain method to yield cutoff frequencies of approximately $.2 \cdot f_s$. f_s is the sampling rate. The gain function for a 15-weight filter constructed by the frequency-domain method is also shown. Note that unlike the fixed-weight method, the present method allows the 3-weight filters to be improved by adding more weights without altering the cutoff frequency.

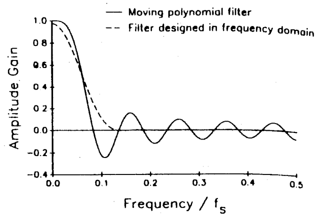

Figure 7. Comparison of transfer functions for a 21-weight cubic moving polynomial filter and a 21-weight filter designed using frequency-domain methods.

rower transition band. Empirical tests suggest that these differences hold for polynomial filters up to at least 51 weights and fifth order. Choice of filter would thus depend in part on the relative importance of these two characteristics for a specific application. Although stop-band ripple for the basic moving polynomial filter can exceed 20% of original amplitude, Hamming (1989, pp. 47–50) notes that such ripple can be reduced by tapering the ends of the weight series. Nonlinear transformations of amplitude (e.g., power) will alter the effect of stop-band ripple. For example, by using the logarithm of power based on the residual from the moving polynomial filter, Porges attenuates effects of the stop-band ripple seen in the amplitude gain function.

Another consideration in choosing between moving polynomial and frequency-domain filters is cutoff frequency. Quadratic and cubic polynomials have half-amplitude cutoff frequencies equal to approximately $1.3 \cdot f_s$ divided by the number of weights; the corresponding constant for quartic and quintic polynomials is 2. (As demonstrated by Kendall, 1976, moving polynomial filters of order 2k are equivalent to those of order 2k+1.) Because f_c for any order of polynomial filter is a function of the filter width, there is a limitation on cutoff frequencies for such filters that is similar to that noted above for equal-weight filters.

Applications of Filters Designed in the Frequency Domain

Some of the issues in digital filter design are illustrated in a comparison of three EEG data sets. In a standard ERP study, one often wants to identify components that are roughly half-sinusoids and quantify their peak amplitudes and latencies. The phase distortion that a conventional analog filter would introduce might affect the latency measurement. On the other hand, setting the analog low-pass filter too high is problematic, because searching for the maximal value in a latency window risks capitalizing on small, chance fluctuations (noise) in the channel. Hence, it is useful to digitally smooth the data before scoring. One must estimate the frequency characteristics of the component(s) of interest and select a filter that has either a narrow transition band or f_c well above those frequencies. In data digitized at 125 Hz (Giese-Davis, Miller, & Knight, submitted), we expected the main ERP components of interest to be below 5 Hz, and we wished to remove alpha-band information (around 10 Hz) prior to scoring. A low-pass filter with a half-amplitude cutoff of 5 Hz would require a moderately narrow transition band, in order to pass 0 Hz and still remove alpha. A 31-weight filter was found to be adequate, with an amplitude gain of 96% at 0 Hz, 87% at 2 Hz, and 2% at 10 Hz. In contrast, to look at baseline EEG, a 31-weight filter constructed to pass only alpha (8–13 Hz half-amplitude cutoffs) was less effective (Etienne, Deldin, Giese-Davis, & Miller, 1990). The gain was only 61% at 10 Hz, then down to 25% at 6 and 16 Hz, and 2% at 3 and 18 Hz. The unfortunate attenuation at 10 Hz was essentially due to that frequency being relatively close to both of the cutoff frequencies; very narrow transition bands, requiring many weights, are necessary in such a case. A 61-weight filter would have provided a 90% gain at 10 Hz, with 6 and 15 Hz down to 7%. A 91-weight version would have been very time-consuming but very effective: 99% at 10 Hz, 1% at 6 and 15 Hz. A quite different case is the measurement of very slow phenomena underlying fast EEG activity. For contingent negative variation (CNV) data (Yee & Miller, 1988), we essentially employed a moving-average filter to remove conventional EEG: we averaged together the last 250 ms of EEG to score the CNV (sometimes also called an "area" measure). Such a case in which signal and noise are presumed to be far apart in frequency permits a wide transition band, and one can benefit from the computational speed of the moving-average method.

Comparison of Analog and Digital Filters

Having surveyed a variety of digital filters and presented a general method for designing and evaluating such filters, it is possible to consider the relative advantages and disadvantages of analog and digital filters with respect to transition bandwidth, phase shift, consistency, stability, low-frequency application, cost, and availability.

Transition bandwidth. In general, a filter with a narrow transition band is preferable to one with a wide transition band. This is because the former will pass more (signal) on the pass-band side of the cutoff frequency and attenuate more (noise) on the stop-band side of the cutoff frequency. That is, a narrower transition band allows the separation of closer frequencies. Commonly available analog filters often have transition bands of an octave (a range of frequencies over which the frequency is doubled) or more. Because the constraints on analog and digital filter design are quite different, it is often possible to construct a digital filter to have transition bands that approach zero width when this is not feasible with analog circuitry. Thus, digital filters provide the capability of removing artifact having frequency components that are very close to those of the signal of interest. Applications that demand narrow transition bands are those in which signals in one frequency band are to be passed while signals in closely adjacent frequency bands are to be attenuated (e.g., constructing filters to distinguish the conventional delta, theta, alpha, and beta frequencies among broadband EEG). In principle, it is easy to customize a digital filter to be particularly sensitive to any specifiable pattern of signal or noise. However, there is a trade-off between resolution in the frequency domain (narrowness of the transition band) and resolution in the time domain; more on this below.

Phase shift. A primary disadvantage of analog filters is the variable phase shift that they typically introduce. Standard resistor-capacitor (R-C) analog filters shift the input signal in time, and the amount of shift varies as a function of frequency. To appreciate this phase delay intuitively, consider a simple analog low-pass filter typically found in an amplifier. Essentially, higher-frequency components of the signal are removed by averaging over time. Each moment's input voltage is blended with recent moments' voltages. The filter thus has some memory (provided by the capacitor discharging through a resistor in the simplest R-C filter). A sudden (high-frequency) change to a new input level is not reflected significantly in the output until the new level has been sustained long enough to dominate the filter's memory (i.e., until the capacitor charge increases significantly). This is the basis of the familiar rising and falling curve associated with a low-pass filter circuit's time constant, as shown in Figure 8.

The amount of phase distortion is a function of frequency. Because bioelectric signals generally contain multiple frequency components, a traditional R-C analog filter will therefore distort not only the latency but also the shape of the input waveform. In contrast, FIR digital filters generally have zero phase shift and thus do not distort the shape of the waveform, other than the desired attenuation of particular frequencies. No phase shift occurs if the FIR filter is constructed to have symmetrical weights (i.e., in defining output point t, the same weight is applied to the input points at t−k and t+k).

Phase shift is of particular concern in psychophysiological research when the timing of an event (e.g., a peak of an ERP component) is the focus of investigation. Analog filters will generally increase the apparent latencies of such events relative to their actual time of occurrence, with the amount of this increase depending on the frequency components of the event and specific characteristics of the filter design. This bias in latency estimates can be eliminated by replacing the analog filter with an FIR digital filter.

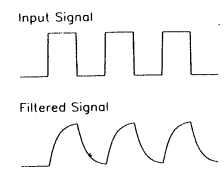

Figure 8. The typical analog resistance-capacitance circuit used as a low-pass filter shifts the phase (e.g., latency of peak amplitude) of a signal.

In contrast to FIR digital filters, the phase shift of IIR filters resembles that of analog filters. This phase shift may be a desirable feature when the researcher seeks to replicate and extend previous research conducted with analog filters (e.g., Cook, Hawk, Davis, & Stevenson, 1991).

Consistency and stability. A further disadvantage of analog filters is the inherent variability among filters that have nominally identical gain and phase characteristics. Analog filters are constructed from physical components that are inexact in their specifications. Even a set of amplifiers with filters manufactured at the same time by the same manufacturer would not be expected to have exactly the same characteristics, and greater variability might be expected among filters and amplifiers manufactured at different times. There is potential for such differences to have systematic effects that may mislead the investigator who generally uses the same amplifier for the same measurement. The potential for error may be particularly great in the case of EEG recording, where a montage of placements is used, and the distribution of a signal across the scalp is of special interest. A related issue is that of consistency across time. Many electronic components are temperature-sensitive and may therefore change their electrical characteristics slightly as they or the devices in which they are installed warm up.

In contrast, as mathematical operations, digital filters (including IIR filters) may be applied with complete consistency at different times, or across multiple recording channels. They do not need to be adjusted or recalibrated to maintain their specifications. Thus, digital filters eliminate the problems that inconsistency and instability could create.

Low-frequency application. Although most filters required in psychophysiological research have cutoff frequencies in the .1 Hz to 1 kHz range, this is not universally true. For example, several ranges of frequencies below .1 Hz are discriminated in bioelectrical potentials recorded from the abdomen (e.g, in electrogastrography). Design of stable analog filters with narrow transition bands is particularly difficult in this range, and for this reason digital filtering methods are now common in this research area.

Practical considerations. The primary advantages of analog filters are practical. Analog filters are an integral part of most physiological recording amplifiers in common use. In their most basic form, they are extremely inexpensive. Analog filtering may be readily applied as the data are recorded, prior to A/D conversion, without subsequent computation. Furthermore, if analog filtering is used to eliminate high-frequency components in the signal, the minimum frequency for A/D conversion is reduced (under the Nyquist rule). Thus, where A/D conversion cannot be accomplished at at least twice the fastest frequency in the original signal, there is usually no alternative to the use of an analog filter. (Analog filters used to eliminate frequencies that would violate the Nyquist rule are known as *anti-aliasing filters.*) Furthermore, current recording hardware permits the analog filter selection to be made easily, whereas software for setting and implementing digital filters has not been so readily available (though see Cook, 1981; Digital Signal Processing Committee, 1979; Farwell et al., in press).

Despite the above considerations, digital filters may be more practical in certain circumstances. Most bioelectric recording devices provide a limited number of settings for the cutoff frequencies of the analog filters. Where a required cutoff frequency is not available, the investigator may choose to construct a custom analog (hardware) or digital (software) filter. As the level of software sophistication among psychophysiologists increases, the digital solution becomes more tractable. Moreover, digital filters may be programmed to operate in real time, during data collection (Cook, Atkinson, & Lang, 1987; Rockstroh, Elbert, Birbaumer, & Lutzenberger, 1982). If low-pass filtering is done on-line, the frequency at which the data must be stored is reduced from twice the fastest frequency in the original signal to twice the upper limit of the transition band of the filter. This frequency may be less than that required if the filtering were done with analog circuitry, thus reducing data storage needs.

Summary and Conclusions: A Decision Tree for Filter Selection

To summarize the points made regarding digital and analog filters, we suggest the following decision tree for filter selection. To select a filtering strategy, we suggest that the investigator consider the following questions:

1. ***Does the raw signal have faster components than $\frac{1}{2} f_s$?***
   - Y: Is there a single high-frequency noise source (e.g., power-line noise) AND can the sampling period be set to an exact multiple of the period of that noise source AND is DC distortion as large as ½ the amplitude of the noise tolerable?
     - Y: Continue with Question 2.
     - N: Increase sampling frequency OR use an analog anti-aliasing filter sufficient to remedy this problem. Continue with Question 2.
   - N: Continue with Question 2.

2. ***Is there an analog filter available with acceptable cutoff frequencies, transition bandwidth, phase shift, consistency, and stability?***
   - Y: Use an analog or a digital filter. If comparisons among multiple channels are to be made, a digital filter is preferred.
   - N: Use a digital filter. Continue with Question 3.
3. ***Is phase shift acceptable?***
   - Y: Consider an IIR filter (cf. Ackroyd, 1973), but continue with Question 4 to evaluate FIR possibilities.
   - N: Use an FIR digital filter. Continue with Question 4.
4. ***Does an equal-weight filter have acceptable cutoff frequency, transition bandwidth, and ripple specifications?***
   - Y: Use an equal-weight filter.
   - N: Use an unequal-weight filter. Continue with Question 5.
5. ***Is a cutoff frequency of $\frac{1}{4} f_s$ and a wide transition band acceptable?***
   - Y: Consider the {.25, .50, .25} filter.
   - N: If this simple unequal-weight filter is unacceptable, the choices become more complex. The frequency-domain method described in Appendix B provides a general method for designing complex unequal-weight filters to meet a variety of specifications of pass band, transition band, and ripple. The reader might also want to consult the engineering literature for other approaches to digital filter design (e.g., Digital Signal Processing Committee, 1979). Certainly requirements of replication might lead an investigator to choose one type of filter (e.g., Porges's 1985 moving polynomials) over other similar filters. Practical issues, particularly computation time when the filter is to be implemented on-line, will also constrain the choice of filter.

Thus, because digital filters have very different advantages and disadvantages from analog filters, the former may be helpful when the latter are problematic. Besides clearing up some confusion in general terminology for analog and digital filters, we hope that this presentation makes digital filter technology more accessible to psychophysiologists.

REFERENCES

Ackroyd, M.H. (1973). *Digital filters.* London: Butterworth.

Attinger, E.O., Anne, A., & McDonald, D.A. (1966). Use of Fourier series for the analysis of biological systems. *Biophysical Journal, 6,* 291–304.

Bloomfield, P. (1976). *Fourier analysis of time-series: An introduction.* New York: Wiley.

Brigham, E.O. (1974). *The fast Fourier transform.* Englewood Cliffs, NJ: Prentice-Hall.

Brillinger, D.R. (1975). *Time-series: Data analysis and theory.* New York: Holt, Rinehart, and Winston.

Cheung, M.N., & Porges, S.W. (1977). Respiratory influences on cardiac responses during attention. *Physiological Psychology, 5,* 53–57.

Coles, M.G.H., Gratton, G., Kramer, A., & Miller, G.A. (1986). Principles of signal acquisition. In M.G.H. Coles, E. Donchin, & S.W. Porges (Eds.), *Psychophysiology: Systems, processes, and applications—A handbook* (pp. 183–221). New York: Guilford Press.

Cook, E.W. (1981). FWTGEN—An interactive FORTRAN II/IV program for calculating weights for a nonrecursive digital filter. *Psychophysiology, 18,* 489–490.

Cook, E.W., Atkinson, L.S., & Lang, K.G. (1987). Stimulus control and data acquisition for IBM PC's and compatibles. *Psychophysiology, 24,* 726–727.

Cook, E.W., Hawk, L.W., Davis, T.L., & Stevenson, V.E. (1991). Affective individual differences and startle reflex modulation. *Journal of Abnormal Psychology, 100,* 5–13.

Digital Signal Processing Committee, IEEE Acoustics, Speech, and Signal Processing Society (Eds.) (1979). *Programs for digital signal processing.* New York: IEEE.

Douglas-Young, J. (1981). *Illustrated encyclopedic dictionary of electronics.* West Nyack, NY: Parker.

Etienne, M.A., Deldin, P.J., Giese-Davis, J., & Miller, G.A. (1990). Differences in EEG distinguish populations at risk for psychopathology. *Psychophysiology, 27*(Suppl. 4A), S28.

Farwell, L.A., Martinerie, J., Bashore, T.R., & Rapp, P.E. (in press). Optimal digital filters for long latency event-related brain potentials. *Psychophysiology.*

Giese-Davis, J.E., Miller, G.A., & Knight, R.A. (submitted). *Memory template comparison processes in anhedonia and dysthymia.* Manuscript submitted for publication.

Glaser, E.M., & Ruchkin, D.S. (1976). *Principles of neurobiological signal analysis.* New York: Academic Press.

Gold, B., & Rader, C.M. (1969). *Digital processing of signals.* New York: McGraw-Hill.

Gottman, J.M. (1981). *Time-series analysis: A comprehensive introduction for social scientists.* Cambridge: Cambridge University Press.

Graham, F.K. (1978). Constraints on measuring heart rate and period sequentially through real and cardiac time. *Psychophysiology, 15,* 492–495.

Grass Instrument Company. (1968). *Model 6 electroencephalograph.* Quincy, MA: Author.

Grass Instrument Company. (1973). *Low level DC amplifier for AC and DC bioelectric signals, transducers, PGR, etc. recordings.* Quincy, MA: Author.

Grass Instrument Company. (1985). *Model 12 neurodata acquisition system.* Quincy, MA: Author.
Hamming, R.W. (1989). *Digital filters.* Englewood Cliffs, NJ: Prentice-Hall.
Jenkins, G.W., & Watts, D.G. (1968). *Spectral analysis and its applications.* San Francisco: Holden-Day.
Kendall, M.G. (1976). *Time-series.* New York: Hafner.
Linnemeyer, S.A., & Porges, S.W. (1986). Recognition memory and cardiac vagal tone in 6-month-old infants. *Infant Behavior and Development, 9,* 43–56.
Ludeman, L.C. (1986). *Fundamentals of digital signal processing.* New York: Harper & Row.
Malmstadt, H.V., Enke, C.G., & Crouch, S.R. (1974). *Electronic measurements for scientists.* Menlo Park, CA: Benjamin.
Marple, S.L., Jr. (1987). *Digital spectral analysis with applications.* Englewood Cliffs, NJ: Prentice-Hall.
McCabe, P.M., Yongue, B.G., Ackles, P.K., & Porges, S.W. (1985). Changes in heart period, heart-period variability, and a spectral analysis estimate of respiratory sinus arrhythmia in response to pharmacological manipulations of the baroreceptor reflex in cats. *Psychophysiology, 22,* 195–203.
Miller, G.A. (1986). Automated beat-by-beat heart rate editing. *Psychophysiology, 23,* 121–122.
Oppenheim, A.V., & Schafer, R.W. (1975). *Digital signal processing.* Englewood Cliffs, NJ: Prentice-Hall.
Porges, S.W. (1985). *Method and apparatus for evaluating rhythmic oscillations in aperiodic physiological response systems.* (Patent number: 4,510,944). Washington, DC: U.S. Patent Office.
Robinson, E.A. (1983). *Multichannel time series analysis with digital computer programs.* Houston, TX: Goose Pond Press.
Rockstroh, B., Elbert, T., Birbaumer, N., & Lutzenberger, W. (1982). *Slow brain potentials and behavior.* Baltimore: Urban & Schwarzenberg.
Ruchkin, D.S. (1988). Measurement of event-related potentials: signal extraction. In T.W. Picton (Ed.), *Handbook of electroencephalography and clinical neurophysiology* (Vol. 3, revised series, pp. 7–43). Amsterdam: Elsevier.
Ruchkin, D.S., & Glaser, E.M. (1978). Some simple digital filters for examination of CNV and P300 waveforms on a single trial basis. In D. Otto (Ed.), *Multidisciplinary perspectives in event-related brain potential (ERP) research* (EPA-600/9-77-043, pp. 579–581). Washington, DC: U.S. Government Printing Office.
Woody, C.D. (1967). Characterization of an adaptive filter for the analysis of variable latency neuroelectric signals. *Medical and Biological Engineering, 5,* 539–553.
Yee, C.M., & Miller, G.A. (1988). Emotional information processing: Modulation of fear in normal and dysthymic subjects. *Journal of Abnormal Psychology, 97,* 54–63.
Yongue, B.G., McCabe, P.M., Porges, S.W., Rivera, M., Kelley, S. L., & Ackles, P.K. (1982). The effects of pharmacological manipulations that influence vagal control of the heart on heart period, heart-period variability, and respiration in rats. *Psychophysiology, 19,* 426–432.

Appendix A

Observations on Fourier Analysis

In preparing this appendix, we noted that mathematical presentations of Fourier analysis in reference texts (e.g., Ackroyd, 1973; Bloomfield, 1976; Brillinger, 1975; Gottman, 1981; Jenkins & Watts, 1968) demonstrate as great a diversity as that previously noted in the specification of cutoff frequencies. Discrepancies in the exact formulas for the Fourier transform are compounded by the fact that quantities that are often of ultimate interest (e.g., amplitude, power, or phase) are based on the results of the transformation, and therefore are also presented in a variety of forms. The approach we have taken is to present the direct and inverse Fourier transformations in the simplest and most intuitive forms that support their ultimate use in digital filter design. We then present our own equations for amplitude, power, and phase, to facilitate understanding of the transformations. We use those equations in the design of digital filters as described in the body of the article and in Appendix B.

To calculate the direct Fourier transform of time series X of even length n ($X_0, X_1, X_2, \ldots, X_{n-1}$), two expressions are evaluated for all integers h from 0 to n/2:

$$A_h = \frac{1}{n} \cdot \sum_{t=0}^{n-1} X_t \cdot \cos(h2\pi t/n) \qquad \text{(A.1)}$$

$$B_h = \frac{1}{n} \cdot \sum_{t=0}^{n-1} X_t \cdot \sin(h2\pi t/n) \qquad \text{(A.2)}$$

The values A_h and B_h are sometimes known as Fourier coefficients. (All published formulas that we encountered for the Fourier coefficients include summed cross-products of the time series with the trigonometric functions. Precise definitions vary across reference sources with regard to whether this sum is left unchanged, multiplied by 1/n, or multiplied by 2/n; in whether h extends from 0 to n/2 or from 0 to n−1 (in which case values from n/2+1 to n−1 are the mirror image of those from 1 to n/2); and in whether t extends from 0 to n−1 or from −n/2 to n/2−1.) The inverse Fourier transform uses these coefficients to reconstruct the original time series by the following equation:

$$X_t = A_0 + 2 \cdot \sum_{h=1}^{n/2-1} (A_h \cdot \cos(h2\pi t/n) + B_h \cdot \sin(h2\pi t/n)) + A_{n/2} \cdot \cos(\pi t) \qquad \text{(A.3)}$$

The following points regarding the direct and inverse Fourier transforms are significant for the present discussion:

1. Understanding the equations depends primarily on analysis of the $h2\pi t/n$ expression for which the sine and cosine functions are evaluated. Because t ranges from 0 to n−1, the value of the t/n portion of this expression ranges roughly from 0 to 1 (precisely, from 0 to (n−1)/n). The constant 2π scales that range to extend from 0 to approximately 2π, over which the sine and cosine functions make one complete cycle. The multiplication of this range by h determines the number of complete cycles represented by the expressions $\cos(h2\pi t/n)$ and $\sin(h2\pi t/n)$ when evaluated for t=0 to n−1. The cosine and sine functions differ only in phase, by $\pi/2$ radians (90 degrees).

2. Given the preceding logic, different integer values of h correspond to different numbers of complete sinusoidal cycles within the total time (T) represented by the time series. For example, if data are sampled for 4 seconds (T = 4 s), h=1 corresponds to .25 Hz (1 cycle in 4 s), h=2 corresponds to .5 Hz (2 cycles in 4 s), and so forth. Because h is always an integer, the resolution of frequencies for which A_h and B_h are evaluated is 1/T. That is, in the T = 4 s example, each successive value produced by the direct Fourier transform corresponds to activity 1/T = .25 Hz faster than the previous value. It is noteworthy that this frequency-domain resolution of the Fourier analysis is determined not by the sampling rate but by the total time represented by the time series (T).

The frequencies corresponding to the values of h are sometimes referred to as *harmonics*, referring to the fact that each frequency corresponds to one more complete cycle within the total time represented by the time series than the preceding frequency. The n/2-th harmonic corresponds to the frequency for which each complete cycle is represented by only two points. The n/2-th harmonic has frequency (n/2)/T = .5 · (n/T) = .5 · f_s, the Nyquist frequency. This limit on the highest frequency is the basis for the Nyquist rule, which requires a minimum of two sample points within the period corresponding to one cycle at the highest frequency in the data. Note that, in contrast to the *resolution* (1/T) of the spectrum, the *maximum frequency* represented is determined by the sampling rate and is independent of the amount of time represented by the time series. For example, if 480 samples were obtained in 4 s (120 Hz), then the highest harmonic (n/2 = 480/2) would be 240 cycles. When divided by 4 s, this yields 60 Hz = half the sample rate = Nyquist frequency. If only 2 s of data had been obtained at 120 Hz sampling (i.e., 240 samples), the highest harmonic would be 240/2 = 120 cycles in 2 s, still 60 Hz.

3. B_h is always zero for h = 0 and h = n/2. For h = 0, this is true because the expression $h2\pi t/n$ reduces to 0, and sin (0) = 0. For h = n/2, $h2\pi t/n$ reduces to πt, and sin (πt) = 0 for all integer values of t. Thus, there are n/2 + 1 values of A but only n/2 − 1 values of B that may be nonzero for the direct Fourier transform of a time series of length n.

4. The balance of the equations for A_h and B_h involves summing the cross-products of the trigonometric function and the sampled data X across time, and dividing by n. Each resulting averaged cross-product will be large to the extent that periodicities in the data match the particular frequency involved in the trigonometric function. Both sine and cosine functions are employed in the calculations. This is because, for any given frequency, sine and cosine functions are uncorrelated due to their 90 degree phase difference. This orthogonality allows the two functions, in combination, to represent all of the variance of a sinusoidal function. That is, if there is activity at that frequency in the data, it will necessarily correlate (i.e., produce a nonzero average cross-product) with at least one of the two functions.

5. One way in which the Fourier coefficients may be conceptualized is in terms of their relationship to the amplitudes and phases of the constituent sinusoids that contribute to the time series. Thus, Equation A.3 may be rewritten as Equation A.4:

$$X_t = R_0 + \sum_{h=1}^{n/2-1} R_h \cdot \cos\left(\frac{h2\pi t}{n} + P_h\right) + R_{n/2} \cdot \cos(\pi t) \qquad \text{(A.4)}$$

The amplitude of the sinusoidal component of X that completes h cycles within the time series is R_h, and the phase relationship of that sinusoid to the time period sampled is P_h. Both quantities may be calculated from the Fourier coefficients using the following equations:

$$R_h = 2 \cdot \text{sqrt}(A_h^2 + B_h^2), \quad \text{where } h = 1, 2, 3, \ldots, n/2\text{-}1;$$

and

$$R_h = \text{sqrt}(A_h^2 + B_h^2), \quad \text{where } h = 0, n/2 \qquad \text{(A.5)}$$

$$P_h = \arctan(-B_h/A_h), \quad \text{where } h = 1, 2, 3, \ldots, n/2\text{-}1 \qquad \text{(A.6)}$$

Thus, the amplitudes of the constituent sinusoids of a time series are represented by the combined magnitudes of A_h and B_h, whereas the phases of the sinusoids are represented by the signs and relative magnitudes of A_h and B_h. When amplitude is graphed as a function of frequency the result is referred to as the *amplitude spectrum* (e.g., Figure 4, Panel a). Phase is based on a cosine function and therefore increases from 0 to 180 degrees (0 to π radians) as the function descends from its peak to its trough. Following a point of discontinuity, the phase then increases from −180 degrees ($-\pi$ radians) to 0 degrees for progressively more positive values on the ascending limb of the function. The phase of a particular harmonic depends on where the beginning of the time series falls on the constituent sinusoid at that harmonic. Thus, phase will be positive if the data begin on the ascending aspect of the sinusoid and negative if the data begin on the descending aspect.

When phase is graphed as a function of h (the harmonic number), it is referred to as a *phase spectrum* (e.g., Figure 4, Panel b).

6. In the discussion of filtering, we made reference to the distinction between amplitude and power. Frequently, sizes of the constituent sinusoids of a time series are expressed in terms of power:

$$\text{Power}_h = 2 \cdot (A_h^2 + B_h^2), \quad \text{where } h = 1, 2, 3, \ldots, n/2\text{-}1;$$

and

$$\text{Power}_h = (A_h^2 + B_h^2 b), \quad \text{where } h = 0, n/2 \qquad \text{(A.7)}$$

The graph of power as a function of frequency is referred to as the *power spectrum.*

One interpretation of power is contribution to the variance of the time series. When power values are summed across all frequencies greater than 0 Hz (DC), the result is the total variance; when power corresponding to DC is included in this sum, the result is the mean square. Thus, one way to construe Fourier analysis is as the partialling of variance in a time series among constituent frequencies.

7. We have found other statistical equivalences of A_h and B_h helpful in understanding the Fourier transform. For example, because $\cos(0 \cdot 2\pi t/n) = \cos(0) = 1$, $A_0 = 1/n \cdot \Sigma (X_t \cdot 1)$ = the mean of the time series (i.e., the net DC level). If the mean is subtracted from the time series prior to transformation, then A_h ($h = 1, 2, 3, \ldots, n/2$) and B_h ($h = 1, 2, 3, \ldots, n/2-1$) are the sample covariances of time series X with cosine and sine functions having h complete cycles within the total time period represented by the time series.

Multiple regression further elaborates this statistical framework within which the Fourier coefficients may be understood. Consider a set of n/2 sinusoidal time series ${}_hC$, each of length n, formed by calculating the cosine of $h2\pi t/n$, $t = 0, 1, 2, \ldots, n-1$. These time series differ only by the value of h ($h = 1, 2, 3, \ldots, n/2$) so that each completes exactly h cycles within the time series. A second set of $n/2-1$ time series ${}_hS$ resembles ${}_hC$, except that it is formed by calculating the *sine* of $h2\pi t/n$. Then, A_h and B_h calculated by the direct Fourier transform will satisfy the following regression-style equation:

$$X_t = A_0 + \sum_{h\text{-}1}^{n/2\text{-}1} \left(2 \cdot A_h \cdot {}_hC_t + 2 \cdot B_h \cdot {}_hS_t \right) + A_{n/2} \cdot {}_hC_t \qquad \text{(A.8)}$$

In Equation A.8, A_0 functions as the intercept, $2 \cdot A_h$ and $2 \cdot B_h$ function as regression coefficients for frequencies from 1 to $n/2-1$, and A_h functions as the regression coefficient for the n/2-th frequency. Note that there are $n/2+1$ regression coefficients for cosine predictors and $n/2-1$ coefficients for sine predictors, or n total coefficients. As in regression, inclusion of n predictors accounts for all variance, and so there is no error of prediction. If some of the predictors were excluded, then A_h and B_h calculated by the direct Fourier transform would minimize the mean squared error in recreating the original time series from the predictors that remained. It should also be noted that expanding Equation A.8 by restating each ${}_hC_t$ as $\cos(h2\pi t/n)$ and each ${}_hS_t$ as $\sin(h2\pi t/n)$ yields an alternative statement of Equation A.3, the inverse Fourier transform.

8. There is a peculiarity about the representation of information exactly at the Nyquist frequency. With only two samples per cycle, the discrete time series preserves the frequency but cannot represent both the amplitude and the phase. This is readily illustrated by assuming a case where the raw signal is a 1-Hz sine wave with peak-to-peak amplitude of 1 V, sampled at 2 Hz. Let the phase of the sampling be such that samples are obtained exactly at the negative and positive peaks of the raw waveform. Although the time series that would result from this sampling would be consistent with a 1-V sine wave, the identical time series could also be obtained by sampling a larger 1-Hz sine wave at consistent "off-peak" points in the wave. In general, if the true phase were known, then the true amplitude could be determined from the time series; conversely, if amplitude were known, phase could be deduced. Because neither is typically known, the obtained value of A_h at the Nyquist frequency should be considered to be the minimum amplitude with respect to the input signal. In the limiting case that illustrates this principle, a time series consisting entirely of zeros could result from sampling a large signal at the Nyquist frequency exactly at its zero-crossings.

If the raw data contain periodic activity at frequencies faster than the Nyquist frequency, the resulting aliasing will cause the power associated with those frequencies to be reflected downward across the Nyquist frequency (f_N). For example, 60 Hz noise in a signal digitized at 100 Hz would appear as a 40-Hz signal in the power spectrum. Frequencies above f_s will reflect successively across the nearest lower multiple of f_N until a value in the 0 to f_N range is obtained. For example, 60 Hz noise digitized at 50 Hz would reflect once across 50 Hz ($= 2 \cdot f_N$) to 40 Hz, and again across 25 Hz ($= f_N$) to appear as a 10-Hz signal. Because multiples of f_N serve as the folding or reflection points for the aliasing of high-frequency signals, f_N is also known as the folding frequency (Glaser & Ruchkin, 1976, p. 113).

One phenomenon relevant to aliasing is useful when the investigator wishes to sample at less than twice the frequency of a known noise source. When the frequency of the noise source is an integer multiple of the sampling rate, distortion resulting from aliasing is typically of little or no consequence. For example, if 60 Hz noise is sampled at 10, 12, 15, 20, 30, or 60 Hz, every sample falls at the same phase of the 60-Hz sine wave, and thus no cyclic low-frequency (aliased) signals will appear. There remains a distortion of the amplitude of the power or amplitude spectra at 0 Hz. This occurs because in each case in which noise is an integer multiple of f_s the application of sequential re-

flection across multiples of f_N ultimately leads to an aliased signal at 0 Hz. In other words, "catching" the fast noise sine wave at a consistent level in its cycle will affect the level (= DC = 0 Hz amplitude) of the samples.

9. Fourier transform computations such as those required for the construction of digital filters are commonly performed using computer subroutines that implement a Fast Fourier Transform (FFT) algorithm (e.g., Marple, 1987; Robinson, 1983). This algorithm substantially reduces computation time when the number of points in the time series (or related frequency-domain representation) is large, although it typically requires the time series length to be an integer power of 2 (e.g., 32, 128, or 1024 points). Some less restrictive forms of the FFT algorithm are surveyed by Brigham (1974). Regardless of the length of the series, the direct and inverse Fourier transforms may be calculated by Equations A.1–A.3. As will be seen in Appendix B, restrictions imposed by the FFT algorithm do not restrict the digital filters that may be constructed.

Appendix B

Finite Impulse Response Filter Design Using the Frequency-Domain Method

The method described here requires direct and inverse Fourier transformations, and for this reason the lengths of the time- and frequency-domain computation arrays must be determined. The time-domain array will hold the filter weights, whereas the frequency-domain array will hold the gain function. The lengths of these arrays are dependent: if the time-domain array has length m, the frequency-domain array will have length $m/2+1$ (m must be even). The precise value of m is not critical, although three constraints should be considered. First, the time-domain scratch array will ultimately contain the filter weights, and therefore m must be greater than the maximum filter width that will be considered. Second, because at several stages in the process the frequency-domain scratch array will contain the gain function of the filter, it should be large enough to provide adequate frequency resolution for graphing this function. That resolution will be f_s/m. Finally, if an FFT algorithm is to be used, m may be required to be an integer power of 2. Ackroyd (1973) suggests the following rule-of-thumb: If the transition band must ultimately be limited to a width of v Hz, set m to between $5 \cdot f_s/v$ and $10 \cdot f_s/v$. We have found that a computation array of length 256 is more than adequate for most purposes.

Step 1. Specification of the ideal gain function. The ideal high-pass or low-pass filter passes all frequency components with unity gain on one side of the cutoff frequency and completely attenuates all frequency components on the other side of the cutoff frequency. That is, the gain function is discontinuous between zero and one at the cutoff frequency, and the transition band has zero width.

To construct the digital filter, we make use of the time series concepts introduced in the text of the article and in Appendix A. We begin with an ideal gain function specified in the frequency domain as array G of length $m/2+1$. Each element of this array corresponds to a harmonic frequency that completes h cycles within m times the sampling frequency, f_s. Thus, the frequency corresponding to G_h will be $(h \cdot f_s)/m$ for $h = 0, 1, 2, \ldots, m/2$. To represent the ideal gain function, G_h is set to 1.0 if $(h \cdot f_s)/m$ falls in the pass band or to 0.0 if $(h \cdot f_s)/m$ falls in the stop band. If $(h \cdot f_s)/m = f_c$, then G_h is set to 0.5.

Band-pass and band-stop filters are specified in the same manner described above for high-pass and low-pass filters. The only difference is that the ideal gain function will have multiple transition bands, and therefore there will be more than one range of 1's and/or 0's within G.

Step 2. Calculating the initial weights. The inverse Fourier transform (Equation A.3) is used to obtain the array of initial weights in the time domain (W) from the ideal gain function in the frequency domain (G). In applying the inverse transform, G serves as the cosine component (i.e., Array A). By setting the sine component (Array B) to zero, we ensure that the filter will have zero phase shift (see Equation A.6). Thus, the equation for each of the m weights becomes:

$$W_t = G_0 + 2 \cdot \sum_{h=1}^{m/2\text{-}1} (G_h \cdot \cos(h2\pi t/m)) + G_{m/2} \cdot \cos(\pi t) \qquad \text{(B.1)}$$

which is evaluated for $t=0$ to $m-1$. In order for Equation 1 (in the text of the article) to apply and the filter to have zero phase shift, W_0 is considered the central weight and W_i is multiplied by X_{t+i} to compute Y_i, $i = -j$ to $+j$. The last $m/2-1$ weights correspond to negative values of i; that is, $t = m/2+1, m/2+2, m/2+3, \ldots, m-1$ in Equation B.1 corresponds to $i = -m/2+1, -m/2+2, -m/2+3, \ldots, -1$, respectively, in Equation 1. Note that the weight series is symmetric; there is a single unpaired central weight W_0, and $W_i = W_{-i}$, where $i = 1, 2, 3, \ldots, m/2-1$.

At this point a digital filter with $m-1$ weights has been obtained. ($W_{m/2}$ is excluded because it is unpaired and would therefore result in phase shift.) Its gain function is approximately (due to exclusion of $W_{m/2}$) the ideal gain function that was originally specified in Step 1, multiplied by m. Scaling can be corrected by simply dividing all elements of W by m, after which the filter weights are ready for use (i.e., application as in Equation 1). However, additional manipulation and testing of the filter are typically desirable, as detailed below.

Step 3. Windowing the weights to obtain the final form of the filter. Filters with fewer than $m-1$ weights may subsequently be calculated by applying a "window" to the weights obtained in Step 2. The window application will truncate the ends of the weighting series and may additionally modify the weights that remain within the truncated series. Any window that is applied will change the gain function of the filter, generally reducing its similarity to the ideal gain function described above.

The principal reason to apply a window is to reduce the filter from its initial width of $m-1$ weights. Practical considerations may dictate that the filter width be reduced below a certain length. For example, if the filter is to be applied on-line or if computing time is otherwise limited, then the filter width may be limited by constraints on computation time. Another constraint on j follows from the fact that the application of the filter results in the loss of j points at each end of the time series (Ruchkin, 1988; "loss" in the sense that the filter cannot be applied to those points). In some cases, particularly if the data are collected before the filter is designed, there may be a maximum tolerable loss. Several options for handling the ends of the time series are detailed at the end of this appendix. However, if filtering is anticipated prior to data collection, then in most cases sufficient data may be recorded beyond the time period of interest. A final consideration that promotes reduction in filter width is that of temporal resolution. The wider the filter, the greater the loss of information about when particular events occurred (e.g., the onset of a burst of a particular frequency). That is, precision in the time domain decreases in that more input points contribute to each filtered point.

Reduction in filter width by simply setting an equal number of weights at each end to zero increases the width of the transition band and introduces ripple in the pass and stop bands. The amount of ripple may be reduced by tapering the ends of the truncated weight series. Two tapering functions have been described by Ackroyd (1973). For the Hamming taper, the remaining weights are multiplied by $0.54 + 0.46 \cdot \cos(\pi p)$. For the Blackman window, remaining weights are multipled by $0.42 + 0.5 \cdot \cos(\pi p) + 0.08 \cdot \cos(2\pi p)$. In these equations, p is the distance from the center of the weighting function to the weight being windowed expressed as a fraction of the distance from the center of the weighting function to the first weight beyond the outermost weight to be retained (i.e., $p = i/(j+1)$, with i and j defined as in Equation 1). Generally, leaving the truncated weights untapered produces the narrowest transition band but with the maximum amount of ripple, whereas the Blackman window provides the least ripple at the expense of widening the transition band. Moderate ripple and transition bandwidth are obtained with the Hamming window. Note that truncating and tapering must be applied equally to both symmetrical halves of the weight series.

Step 4. Calculating the gain function of the windowed filter. The gain function (in the frequency domain) for the weights in the windowed filter (in the time domain) may be obtained by calculating the direct Fourier transform of the weights (see Equations A.1 and A.2). The gain for each frequency is given by the transform, using the method described in Appendix A for relating frequencies to positions in the G array. Like the ideal gains, these calculated gains will generally range from approximately 0.0 (for complete attenuation) to 1.0 (for no attenuation). Gains less than 0.0 indicate frequencies for which the filter actually inverts the input signal (part of the ripple in the stop band; see Figure 6). If the filter has gain G_f at some frequency f, then processing a time series with the digital filter will multiply the amplitude of any periodicity in the time series at frequency f by G_f and the corresponding power by G_f^2.

The obtained gain function may not be completely satisfactory on the first attempt. Steps 3 and 4 may be repeated using a different window width and/or tapering function until a satisfactory filter is obtained. As noted above, the final set of weights should be divided by m prior to use.

A Brief Example

To illustrate the method, we construct a filter to remove frequencies above 45 Hz from EEG data sampled at 200 Hz. A scratch array of length 20 is employed, yielding a frequency resolution of $f_s/m = 200/20 = 10$ Hz. In Step 1, the ideal gain function, G, is computed: {1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0}. Note that because the filter is to pass data from 0 Hz to the 4th harmonic (corresponding to 40 Hz), gain is set to 1 for G_0 to G_4. Ideally, gain is 0 at the 5th harmonic (50 Hz) and above, so G_5 to G_{10} are set to 0.

In Step 2, the inverse Fourier transform (Equation 3) is used to compute the original set of weights: {9, 6.3138, 1, −1.9626, −1, 1, 1, −.5095, −1, .1584, 1, .1584, −1, −.5095, 1, 1, −1, −1.9626, 1, 6.3138}. Thus, $W_0 = 9.0$, $W_1 = W_{-1} = 6.3138$, etc.

In Step 3 we decide that no more than 15 weights may be used because the filter is to be implemented on-line. We choose a Hamming window to balance requirements for a narrow transition bandwidth and minimal stop-band ripple. Zeroing the five elements of W that are farthest from the central weight (W_0) and applying the Hamming window function to the remaining elements of W produces: {9, 6.0927, .8653, −1.4053, −.5400, .3640, .2147, −.0586, 0, 0, 0, 0, 0, −.0586, .2147, .3640, −.5400, −1.4053, .8653, 6.0927}. The zeros are maintained in the computation array even though they are effectively eliminated from the filter. This ensures that the actual gain function computed in Step 4 will have the same frequency-domain resolution as the ideal gain function specified in Step 1.

Finally, in Step 4 we use the direct Fourier transform (Equations A.1 and A.2) to compute the gain function for the filter with the windowed weights. The gain function for DC, 10 Hz, 20 Hz, up to 100 Hz (f_N) is: {1.0033, .9970, 1.0048, .9705, .7130, .2880, .0268, −.0019, .0013, −.0036, .0047}. The gain function indicates, for example, that this digital filter would pass at least 97% of the signal amplitude from DC to 30 Hz, and would attenuate by approximately 97% any 60-Hz noise. At this point we could repeat Steps 3 and 4 with new window parameters if these or other characteristics of the filter were unsatisfactory.

Prior to use, the filter weights must be divided by m (in this case, 20) and the order of W must be rearranged so that it is symmetric about W_0. Thus, the final form of this 15-weight filter, suitable for application as shown in Equation 1 in the text of the article, is: {−.0029, .0107, .0182, −.0270, −.0703, .0433, .3046, .4500, .3046, .0433, −.0703, −.0270, .0182, .0107, −.0029}.

Evaluating Other FIR Filters

As noted in the text, the techniques described here may be used to determine the gain function for any moving-average filter with symmetric weights. The simplest of such filters, for smoothing, is one with N equal weights of 1/N. Generally N is odd, so that the filter is symmetric and does not produce phase shift. To calculate the gain function, the weights must first be placed in an Array W, similar to that used to construct filters by the method described above, and each element must be multiplied by m, the length of the computation array. (The considerations that determine m here are the same as those that apply when filters are to be constructed.) For example, to calculate the gain function for a moving-average filter of width 7 with $m = 10$, W would be set to: {10/7, 10/7, 10/7, 10/7, 0, 0, 0, 10/7, 10/7, 10/7}. (This placement of the weights within the array is required because the weight function must be symmetric about the $(m/2)+1$th element, as noted in Step 2, above). Referring to the steps for calculating weights described above, we are creating a weight series by other means, which may be substituted for Steps 1–3. Then, as discussed in Step 4, the gain function may be calculated with the direct Fourier transform. Using Equation A.1, this transformation of the above array yields {1, .3740, −.2312, .0546, .0882, −.1429}, corresponding to frequencies of 0, .1, .2, .3, .4, and .5 times the sampling frequency.

FORTRAN Statements to Implement an FIR Filter

The following FORTRAN subroutine applies a FIR filter to integer data. `NDATA` input points are assumed to be stored in Array `IN`. `WO` holds the weight to be applied to the point in `IN` that is being redefined. `NWEIGHTS` lagged weights are stored in Array `WEIGHTS`. `WEIGHTS(I)` holds the weight to be applied to input points I samples before and I samples after the point being redefined. Filtered data are stored in `OUT` in real format for maximum accuracy. In this implementation, the first and last `NWEIGHTS` points are left undefined, but the section that follows details alternative ways to filter the ends of a time series.

```
      SUBROUTINE   FILTER (NDATA,IN,OUT,
     +                     WO, NWEIGHTS,WEIGHTS)
      INTEGER      NDATA,IN(1),NWEIGHTS
      REAL         OUT(1),WO,WEIGHTS(1)
      DO 2 I = NWEIGHTS+1, NDATA-NWEIGHTS
        OUT(I) = IN(I) * WO
        DO 1 L = 1,NWEIGHTS
          OUT(I) = OUT(I) +
     +      (WEIGHTS(L) * (IN(I-L) + IN(I+L)))
1       CONTINUE
2     CONTINUE
      RETURN
      END
```

Filtering the Ends of a Time Series

Because the types of digital filters considered in detail here require a symmetrical set of points before and after each filtered point, it is not possible to apply a $(2j+1)$-length filter to the first and last j data points. Typical analog filters have the same general problem, requiring a period of time prior to recording for the circuit to stabilize and/or to be reset.

There are a variety of ways to handle this situation with digital filters. The best is always to collect enough extra data, before and after the epoch of interest, so that one can discard the j initial and j final points. If filtering is accomplished on-line, input data can be continuously filtered but saved for later analysis only during specific time periods. In some cases, neither of these options may be possible. Post hoc alternatives include:

1. Develop a series of shorter weighting functions for the data close to the ends. For example, if the weighting function used to filter most of the epoch has length 31, a 15-weight filter could be developed to filter Points 8–15, a 3-weight filter could be developed for Points 2–7, and Point 1 could be left unfiltered. Filtered points closest to the ends would be filtered less successfully.
2. Pad the ends of the epoch with j zeros, j replications of the terminal values, or j replications of the average of the first or last few values. The filter can then be applied to the entire original time series, although again the data near the ends will be somewhat suspect.
3. Use an IIR filter for the beginning and end of the series. The IIR filter can be applied down to the very last point in the series. If applied off-line, it can be applied in reverse to the first point in the series. See Ackroyd (1973) for instructions on how to design and implement such filters.

(Manuscript received March 6, 1990; accepted for publication February 9, 1991)