Event-Related Potentials

A Methods Handbook

edited by Todd C. Handy

A Bradford Book
The MIT Press
Cambridge, Massachusetts
London, England
2005

5 Digital Filters in ERP Research

J. Christopher Edgar, Jennifer L. Stewart, and Gregory A. Miller

The processing of psychophysiological signals always includes some type of filtering. Although there are many different filtering techniques, all involve removing a portion of the recorded signal—either activity that is considered noise (e.g., 60-Hz activity), or some signal components to focus on others. Filtering is often conceptualized as removing particular sine wave frequencies from data that are treated as consisting solely of multiple sine waves, although the concept of filtering is entirely general, and real world signals rarely consist of invariant sine waves.

Filtering is routinely done by means of electronic circuits built into recording amplifiers or electrically interposed between the amplifier and the recording device, such as an analog-to-digital (A/D) converter. Such electronic (or *analog*) filters applied to a continuous (usually varying) voltage contrast with *digital* filters that are applied to a discrete, numeric representation of the numerically recorded signal. (For an introduction aimed at psychophysiologists, see Cook & Miller, 1992; for an extensive overview of digital filtering methods for the advanced psychophysiologist reader, see Ruchkin, 1988.) Digital filtering has several clear advantages over analog filtering (see Picton et al., 2000). First, the original data can be retained for evaluation using alternative filter settings. Second, one can construct digital filters so that they do not alter the phase (see box 5.1) of frequencies in the waveform. Third, digital filtering can more easily adapt its settings than filtering that depends on hardware components. It is generally appropriate to restrict analog filtering to what is required to prevent aliasing (due to signal frequencies too high to be represented accurately at a given sampling rate; see box 5.2) or blocking of the A/D converter (due to signal amplitude exceeding its input range) and to use digital filtering for subsequent signal analysis.

Reliance on digital filters is increasing, thanks to the pervasiveness of powerful desktop computers, along with growing interest in psychophysiological research, leading individuals, labs, and companies to create publicly available software (e.g., Wellcome Department of Cognitive Neurology [SPM], Richard Coppola [EEGSYS], Oxford Image Analysis Group [FSL], Medical Numerics [MEDx], NeuroScan [SCAN and CURRY], James Long Company [EEG Analysis System], Michael Scherg [BESA], Edwin Cook

[FWTGEN], Electrical Geodesics [Analysis Tools], Scott Makeig [EEGLAB], CorTech Labs [FreeSurfer], Neuromag [Neuromag], Brain Innovation B.V. [BrainVoyager], Brain Products Vision Analyzer) that allow investigators to manipulate and analyze psychophysiological signals, relieving the investigator of the need to write custom digital filtering programs. However, one should use the available software only with a full appreciation for the algorithms and options it provides.

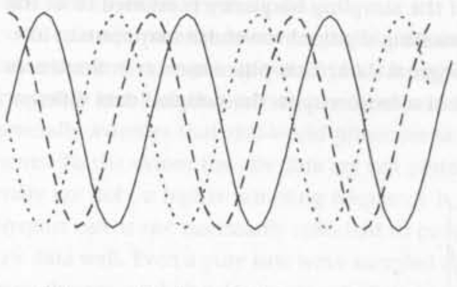
As others have explained characteristics of analog filters and differences between analog and digital filters (e.g., Coles et al., 1986; Cook & Miller, 1992; Nitschke, Miller, & Cook, 1998; Picton et al., 2000), the present discussion focuses on digital filtering techniques, emphasizing practical information of use to psychophysiologists, rather than formal mathematical treatments available in engineering and signal processing texts. This overview concludes with a discussion of filtering features in some popular software aimed at the ERP research community, with a particular focus on the variability across and within programs. We present a method to determine the gain function of a given filter, so that users of publicly available software can evaluate the behavior of the filters used.

Analog and Digital Filtering: Concepts and Terms

The most commonly considered electronic filters used in psychophysiology are *high-pass* and *low-pass*, which selectively attenuate low-frequency and high-frequency components, respectively. (More ambiguously, the term "high cutoff" can refer to the high-pass setting or the low-pass setting, and similarly for the term "low cutoff." "High-pass" and "low-pass" are unambiguous and preferable.) Deployed in series, a combination of a high-pass and a low-pass filter constitutes a *bandpass* filter, which passes frequencies within a single range. Another hybrid, the *bandstop* filter, selectively attenuates frequency components within a specified range. Typically, band-stop filters, often referred to as *notch filters*, attenuate a narrow range of frequencies in the vicinity of power line noise (50 or 60 Hz). The range of frequencies that a filter will pass without substantial attenuation is its *pass band*. The range of frequencies in which little energy is passed is the *stop band*, and the range of frequencies in which gain is intermediate is the *transition band*. In an ideal filter, there might be no transition band, or the boundary between the transition band and the pass or stop band would be discrete. In realistic filters, such boundaries cannot be so sharp; descriptions of such boundaries must be understood as approximate. In principle, one could construct a filter with any combination of pass and stop bands. One can also subject the same set of data to several filters in parallel, producing alternative sets of filtered output (e.g., for different EEG bands). Whether analog or digital, more complex filters can achieve narrower transition bands, which may be required in situations where the signal of interest and the noise or artifact to be rejected contain similar frequency components.

Box 5.1
Phase

In the context of ongoing sine waves, *phase* (or *phase angle*) refers to where in the cycle a given sinusoidal waveform is at a particular time. A sine wave starts at time t_0 and continues indefinitely, changing moment to moment; but at any given moment one can ask what the phase is of that sine wave. That is, at what point in its cycle is it? Specifying its amplitude, frequency, and phase at t_0 allows projection of its value at any future moment. If the wave starts at 0 v at time t_0, ranges from +10 to −10 v, and oscillates at 10 Hz (completing a cycle every 100 ms), it will return to 0 v every 50 ms. It will reach +10 v at 25 ms and again at 125 ms. At every multiple of 100 ms, it will be back at 0 v, headed positive. After 1000 ms, it will have completed 10 cycles and be back at 0 v.

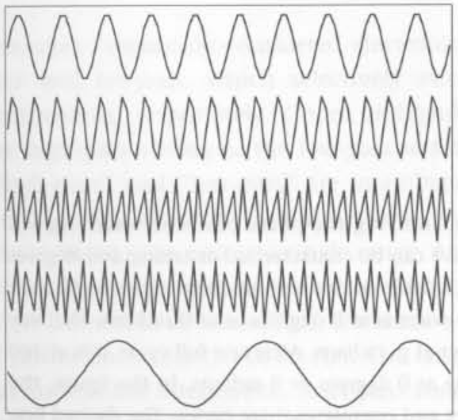


—— 0 degrees = 0 radians = 360 degrees = 2 pi radians
– – 90 degrees = pi/2 radians (= cosine at 0 degrees)
· · · 180 degrees = pi radians

There are two common conventions for quantifying the phase of a sine wave: degrees and radians. One complete cycle of a sine wave can be characterized as taking 360 degrees or as taking 2 * pi radians, because of the relationship of the sine function to proportions of a circle. At the beginning of a cycle, the sine wave is at 0 degrees or at 0 radians. Halfway through a cycle, a sine wave is at 180 degrees or at pi radians. After one full cycle, it is at 360 degrees or at 2 * pi radians, which is the same as 0 degrees or 0 radians. In the figure, the solid line is a sine wave that starts at 0 degrees and completes three cycles. The dashed line shows the same sine wave, except that the sine wave starts at its maximum positive voltage rather than at 0 v. Thus, only its phase differs from the solid line. Because 10 v is one-quarter of the full cycle of the solid line, the dashed line is said to be at a phase of 90 degrees or pi/2 radians. One can also say that the solid and dashed lines differ by 90 degrees or pi/2 radians. Finally, the dotted line is perfectly "out of phase" with the solid line, meaning that it is a mirror image, although otherwise identical. Formally, the dotted line begins at a phase (phase angle) of 180 degrees or pi radians.

Box 5.2
The Nyquist Rule and Aliasing

In order for a time series to represent a continuous waveform adequately, the *sampling rate* (f_s in samples per second, the inverse of the sampling period in seconds per sample) must be more than twice the fastest frequency present in the original waveform. (More strictly, one must sample more than twice the bandwidth, but in most applications this includes 0 Hz.) Similarly, if the sampling is of the scalp surface (via ERP electrode density), rather than time, the spatial frequency of sampling must be more than twice the spatial frequency of topographic change on the head surface (see Srinivasan, Tucker, & Murias, 1998, for more discussion of this issue). The same spatial sampling density issue arises in fMRI research. 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*; 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*, because a signal component appears at a frequency in the sampled data different from its frequency in the original signal.



The figure illustrates the effect of aliasing. The *x* axis is 1000 ms, and the five signals are digitized at 100 Hz, thus providing a Nyquist frequency of 50 Hz. The five signals begin at 0 degrees phase and have identical amplitudes but differ in frequency, top to bottom: 10, 20, 40, 60, and 97 Hz. For the 10-Hz signal, the 100-Hz A/D rate does a good job of representing the original continuous waveform, seeming to lose just a bit at the peaks of the cycles. The 20-Hz signal is very recognizable, though the tracing is a bit choppy. Inspection verifies that 20 cycles are completed. To the eye, the 40-Hz signal looks very choppy and perhaps composed of several frequencies. But because it is below the Nyquist frequency it is still

Box 5.2
(continued)

accurately represented: there are 40 downward peaks. This illustrates that sampling at more than twice the frequency in the signal does not ensure an attractive representation. Importantly, the 60-Hz signal is misrepresented. It appears to be identical to the 40-Hz signal except for a 180-degree phase reversal. Frequency analysis would unambiguously (though incorrectly) show that it is composed of a single, pure 40-Hz component. The 60-Hz signal has been aliased down to 40 Hz, because 60 and 40 are equidistant from the Nyquist frequency of 50 Hz. Finally, the bottom tracing shows an extreme and intriguing example of aliasing. Even though the 97-Hz signal is below the 100-Hz sampling rate, the signal is very badly distorted. The signal appears to be a perfect 3-Hz signal, because 97 and three are equidistant from the Nyquist frequency. This example is particularly impressive because the time series looks deceptively clean.

We can offer several caveats regarding aliasing. First, Nyquist's rule requires sampling at twice the fastest frequency present in the original waveform—not merely twice the fastest frequency in which the investigator is interested. Second, treatment of the aliasing problem generally assumes that real-world phenomena are well represented by sinusoidal components. To the extent the raw data are not perfectly sinusoidal (and physiological data generally are not), a higher sampling frequency is necessary. Third, strict conformance to the Nyquist rule is not necessarily sufficient to provide a digitized signal that will illustrate the raw data well. Even a pure sine wave sampled at fewer than five samples per cycle may look very choppy, and signals composed of multiple components may require a much higher sample rate in order to provide good visual fidelity. The Nyquist rule only addresses the aliasing issue about whether frequencies will be systematically misrepresented. A raw signal composed of sine waves and sampled at more than twice the frequency of the highest component will not be aliased and can be treated numerically with confidence, but the vector of samples may not be very presentable graphically. Fourth, sampling density is not an issue when a single observation is of interest. For example, if the research question is focused on activity at the Cz recording site rather than on topography or source localization, the spatial density of other electrode placements is not an issue. Similarly, if one cares only about activity 400 ms after stimulus onset, one need only digitize a single value at that latency, without concern about aliasing.

Some analog filters are occasionally called *anti-aliasing filters*. This term can be confusing, as it actually refers to the use to which the filter is put rather than to any property of the filter. One can avoid aliasing by employing a low-pass analog filter prior to digitizing a signal. One would need an additional, anti-aliasing filter only if the amplifier does not provide a suitable setting relative to the frequency characteristics of the signal and the sample rate, in terms of either f_c or roll-off. Typically, anti-aliasing filters have very steep roll-offs, perhaps 45 dB/octave.

Because of possible errors in estimating the highest frequencies in real-world data, noise introduced by amplifiers and A/D converters, and the nonsinusoidal nature of many phys-

Box 5.2
(continued)

iological signals, many have suggested (e.g., Attinger, Anne, & McDonald, 1966; Coles et al., 1986) that the sample rate be as much as 5–10 times higher than the Nyquist rule suggests. On the other hand, when the noise power introduced at high frequencies is minimal, less intensive sampling may suffice and the resulting (trivially small) aliased noise ignored. A second situation in which it is sometimes possible to violate the Nyquist rule is where all high-frequency noise sources are a harmonic of a single frequency. With the right sampling period, the sampling will occur at the same phase of the noise cycle with every sample (e.g., power-line noise at 50 or 60 Hz). However, the obtained samples may include a DC offset that can vary by as much as the full peak-to-peak range of the noise, which may or may not be problematic in a given context. For example, in a long-interval CNV study, Simons, Öhman, and Lang (1979) sampled the EEG at 30 Hz, which is largely impervious to 60-Hz noise. Because they deviated epochs of interest from a prestimulus baseline, their data were not vulnerable to the DC offset problem. Aliasing is important in selecting parameters for digital data acquisition and filtering. A more rigorous treatment of aliasing, based on power spectrum analysis, appears in appendix A of Cook and Miller 1992.

For simplicity, we will largely restrict the present discussion to simple high-pass and low-pass filters, although we will note other types. Also, the present discussion will focus on filters characterized in terms of their response to sine wave input. That is how psychophysiology, including the literature on event-related brain potentials (ERPs), usually characterizes filter performance, although other approaches are possible.

In common use, the term *gain* indicates an increase in magnitude, but more generally it refers to any ratio of output to input. Gain may be greater than, equal to, or less than 1.0. In discussions of electrical circuits, one generally considers segments of the circuit designed to boost signal magnitude (gain > 1.0), regardless of frequency (amplifiers), separately from segments designed to reduce magnitude (gain < 1.0) selectively as a function of frequency (filters). Thus, filters normally have a frequency-dependent gain ranging from 0.0 to 1.0. (In some cases gain may even be negative, meaning that the polarity of the signal is inverted.) The relationship between frequency and gain is typically plotted as the *gain function* for a particular filter, with frequency on the x axis and signal magnitude on the y axis represented in amplitude (voltage) or power (roughly the square of voltage; box 5.3 discusses their relationship). Figure 5.1 illustrates the gain function for a low-pass filter. The upper left shows the ripple that some types of filters introduce in the pass-band and/or stop-band. When the gain goes below zero (or is negative), the effect is an inversion of the signal.

Filters are sometimes characterized in terms of the approximate boundaries of the transition band. Alternatively, a specific frequency within the transition band may be cited as the "cutoff" or "corner" frequency, f_c. The definition of f_c varies across authors and even across amplifier manufacturers. In the electrical engineering literature, 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. Some standard sources in the psychophysiology literature also define f_c as the half-power frequency. However, other sources treat f_c as the half-*amplitude* frequency, which is not the half-*power* frequency. In addition, two other ways of reporting filter cutoffs involve the time constant and decibels (see box 5.4 and box 5.5).

Another feature of transition bands sometimes reported is the "roll-off," usually expressed in dB per unit change in frequency, usually octave (a doubling or halving of frequency) and sometimes decade (a tenfold change in frequency). Thus, a 6-dB/octave filter has a much narrower transition band (steeper roll-off) than a 6-dB/decade filter,

Box 5.3
Power, Amplitude, and Cutoff Frequency

For expository purposes, one can often use power and amplitude interchangeably to convey the magnitude of a signal or a component of a signal. However, quantitatively, power and amplitude are not the same concepts. As a first approximation, power is the square of amplitude, so that power is never negative. More precisely, voltage = current * resistance (Ohm's law), and power (in watts) = voltage * current. As a result of these relationships, power = voltage * voltage/resistance. Thus, power is proportional to the square of voltage. Commonly in treatments of these relationships, resistance is implicitly set to 1.0, so that power is the square of voltage.

This distinction is important in describing the characteristics of digital filters. Filters are often described in terms of the frequency at which the output of the filter is half that input. Sometimes authors put this explicitly in terms of half amplitude or half power, but often neither is specified. The frequency at which the output amplitude is half the input amplitude is not, in general, the frequency at which the output power is half the input power. This follows readily from the observation that $.707^2 = .5$. With power the square of amplitude, the frequency at which power is reduced by 50 percent is the frequency at which amplitude is reduced by 29.3 percent. Conversely, given $.5^2 = .25$, the frequency at which amplitude is reduced by 50 percent is the frequency at which power is reduced by 75 percent. Depending on the gain function of a filter, the half-amplitude and half-power frequencies may be quite different. It is important that authors (and publicly available software) make clear whether their frequency cutoffs are specified in terms of power or amplitude. In the present chapter, f_c always refers to cutoff frequency in the strict sense of the half-power frequency, following its usual definition in the electrical engineering literature. We use *cutoff frequency* more generically, referring to either half power or half amplitude. These are its common uses, but in some contexts cutoff frequency refers to some other threshold on the gain function (see Cook & Miller, 1992, for some examples).

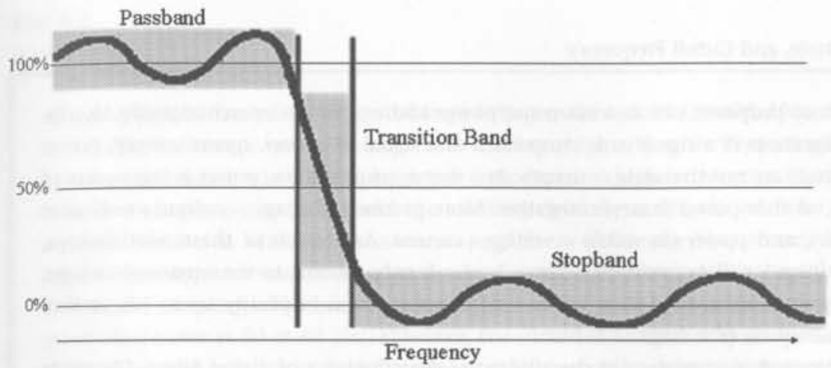


Figure 5.1
The gain function of a filter is divided into the pass band, transition band, and stop band. The gain function shown is for a low-pass filter, without magnitude as a percentage of input magnitude.

though one could describe both (ambiguously) as having a "6-dB roll-off." Note that analog and digital filters can be designed with gain functions different from those for a simple RC filter, including cascading several RC circuits in series. As a consequence, the slope or roll-off within the transition band can be more or less steep. Thus, a filter might be characterized as having a 24-dB/octave slope.

Literatures more remote from the electrical engineering tradition may use an altogether different means of characterizing filters. For example, it is common in current functional magnetic resonance literature to apply (spatial) smoothing characterized in terms of its *FWHM*, or full-width/half-maximum value. Typically, this refers to a (normally) symmetrical weighting function, and the acronym FWHM is the width (in mm) of the function at half its maximum value.

This variety of methods of characterizing filters reflects the variety of disciplines from which psychophysiology draws, but inconsistencies in reporting such characteristics can be problematic. Ideally, authors would routinely include a figure showing the gain function of their filters. Minimally, authors should report filter characteristics unambiguously. For example, one should not report a cutoff frequency without making clear whether this refers to half power, half amplitude, the start of the transition band, or some other reference point.

This discussion of analog and digital filters is equally applicable to signals sampled over time and signals sampled over space. From the standpoint of a numeric operation on a vector of values, it is irrelevant whether the values represent a phenomenon unfolding over time or over space. For practical reasons, analog filters are generally confined to time-domain applications, but digital filters are equally applicable to time and space contexts.

Box 5.4
RC Circuits, Time Constants, and Phase Delay

The time constant is a property of certain types of simple analog electrical circuits. For present purposes, a resistor (R) and a capacitor (C) in series will suffice. A constant voltage applied across this circuit (such as from a battery) will, in effect, cause charge to flow through the resistor and accumulate on the capacitor. As the charge accumulates, the capacitor begins to resist the flow of additional charge. Thus, over time, voltage builds up across the capacitor, while current through the resistor declines. The circuit reaches a steady state when the voltage across the capacitor matches the voltage applied to the circuit, at which time current is zero. The voltage build-up and current decline are mirror images, each asymptotic functions of time. The eventual voltage across the capacitor depends entirely on the voltage applied to the circuit, not on the R or C values. However, the time taken to reach asymptote does not depend at all on the applied voltage. It depends solely on the R and C values. Thus, the circuit behaves consistently over time regardless of the amplitude of the voltage input. For a given RC combination, the time constant (TC) is defined at the time in seconds to reach approximately 63 percent of the asymptotic state, and it happens that $TC = R * C$. The cutoff frequency f_c can be defined in terms of R and C: $R * C = TC = 1/(2\pi f_c)$.

This account extends readily to the case of varying voltage input that is typical of psychophysiological signals. Inversion of the polarity of the voltage source will cause current to reverse direction and charge to empty from the capacitor. The rate at which that happens is again governed by the time constant of that RC combination. Some amplifier settings are labeled in time constant units (seconds) rather than frequency units (cycles per second). Reference in the literature to the time constant of a circuit has sometimes meant the high-pass f_c. Thus, a filter might be characterized by a time constant and a low-pass f_c. However, formally this is ambiguous, as any given RC circuit can serve as either a high-pass or low-pass filter, depending on whether the voltage across the resistor or the capacitor is taken as the output of the filter. In fact, a single RC circuit can serve both functions simultaneously, such as in the crossover circuit in a multispeaker audio system, separating treble and bass frequencies for different speakers.

The role of the capacitor in RC filters accounts for the phase distortion that such filters create. It takes time for charge on the capacitor to accrue and empty, more time for lower input frequencies. Thus, the output of the circuit is a delayed representation of the input (already a distortion), and that delay varies with frequency (a further distortion). To further appreciate the phase delay inherent in a real or simulated analog filter, consider a simple RC circuit employed as a high-pass filter typically found in an amplifier. Essentially, lower frequency components of the signal are removed as their charge builds up slowly and dissipates slowly on the capacitor. Each moment's input voltage is blurred with recent moments' voltages. The filter thus has some memory. A sudden (high-frequency) change to a new input level is reflected immediately in the output until the new level has been sustained long enough to build up charge across the capacitor. There will be no noticeable

Box 5.4
(continued)

build-up if the frequency of the new input has a sufficiently high frequency. But if the new input level is sustained (low frequency or even 0 Hz—a true level change), the capacitor charge will gradually build up. Thus, the output level will reflect the input level only after some delay—a phase distortion, the degree of which depends on the frequency of the input. This is the basis of the familiar rising and falling curve associated with an RC circuit's time constant.

The amount of this phase distortion is a function of frequency. An RC filter will distort not only the latency but also the shape of the input waveform. 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 low-pass filters will generally increase the apparent latencies of such events, with the amount of this increase depending on the frequency components of the event and specific characteristics of the filter design—the lower the cutoff frequency (equivalently, the longer the time constant), the greater the distortion. 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 et al., 1991). Some software provides a backward filter (see Zero Phase Shift and Simulated Analog Settings section, p. 106) that allows one to compensate for the phase shift caused by the analog filters.

Issues in Understanding Digital Filters

Having reviewed some general principles and issues in filtering, we now address digital filtering more specifically. We can use the term "digital filter" for a wide range of techniques that may only have in common the fact that they are mathematical procedures that are applied to discrete numeric representations of discrete or continuous waveforms in order to selectively augment or more commonly to attenuate certain frequencies. Psychophysiologists using a wide range of physiological measures routinely work with such representations. Any parameter that can be recorded repeatedly over time or space can be treated as a vector of observations of the form:

$$X_t, X_{t+d}, X_{t+2d}, X_{t+3d}, \ldots, X_{t+nd}$$

If these values are recorded over time, the data are sometimes called a *time series*. The subscripts refer to the time at which the variable X is observed, with t the time at which recording began and d the *sampling period* (the time or distance between adjacent samples, a constant in most applications, and assumed constant in the present discussion). Event series (where time between events, in ms, differs from event to event) such as heart periods can be converted to time series with a constant sampling period (Cheung & Porges, 1977; Graham, 1978; Miller, 1986).

Box 5.5
Decibels

Characterization of the cutoff frequency of a filter in terms of decibels (dB) involves a log function of the gain, with different but equivalent equations for power and amplitude and with negative values meaning a gain less than 1.0. In dB, a gain of $(P_{out}/P_{in}) = .5$ power is $10 \log_{10}(P_{out}/P_{in}) = 10 \log_{10}(.5) = -3$ dB. A gain of $(V_{out}/V_{in}) = .5$ amplitude is $20 \log_{10}(V_{out}/V_{in}) = 20 \log_{10}(.5) = -6$ dB. As noted in box 3, voltage = current $*$ resistance (Ohm's law), and power (in watts) = voltage $*$ current. Therefore, power = voltage $*$ voltage/resistance. Thus, power is proportional to the square of voltage. Commonly in treatments of these relationships, resistance is implicitly set to 1.0, and power is said to be the square of voltage. The half-power frequency f_c is often referred to as the frequency at which the gain is "3 dB down." At the half-amplitude frequency, output is 6 dB down. It must be appreciated that the half-power frequency and the half-amplitude frequency are not the same frequency, because power and amplitude are different values. That is, the frequency at which the filter will reduce the power by half is not the frequency at which it will reduce the amplitude by half. This is a common source of confusion in the ERP literature. Generally, the half-amplitude frequency will be further from the center of the passband than is the half-power frequency. A characterization of a filter in terms of the frequency at which output is cut in half is ambiguous unless it is made clear whether this is half of the power or half of the amplitude.

Representing Waveforms in the Frequency Domain

A time series that indicates voltage or some other parameter as a function of time is considered a representation "in the time domain." An alternative representation of the same information is based on the principle that any stationary waveform (i.e., from which long-term trends or changes in level have been removed and in which the frequency components do not change in amplitude or phase over time) may be represented as the sum of a set of sinusoidal waveforms, each of a different frequency and having an associated amplitude (or power) and phase. This principle (the Fourier theorem) is the basis of *Fourier analysis*, which determines the amplitudes and phases of the constituent sinusoids as a function of frequency. This representation of a signal is "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 (see box 5.6). No information is lost in either transform—each is simply a way to represent the original vector of data. Figure 5.2 provides two examples of how a set of sine waves can combine to form an apparently nonsinusoidal time series.

The interchangeability of time-domain and frequency-domain representations of a given waveform bears emphasis. Consider a set of j sine waves. Given the stationarity assumption, the frequency, amplitude, and phase of each sine wave are constant throughout the analyzed epoch. At any particular time during the epoch, the different sine waves may be at different points in their cycles. Summing across the set of sine waves produces a single composite waveform in which the constituents may be difficult to identify. One could digitize the composite waveform, describing it as a single vector of values arranged in time. Alternatively, one could describe it with amplitude and phase vectors (known as the amplitude and phase spectra) arranged in order by frequency. Either description—in the time domain or in the frequency domain—completely specifies all of the information contained in the digitized composite waveform. One description may be more tractable for a given set of analyses or more intuitively appealing for a given question, but the same information is available in the two representations. Although more familiar in analyses of ongoing EEG, one can use Fourier analysis in conventional ERP paradigms (e.g., Pfurtscheller & Lopes da Silva, 1999). The inverse Fourier transform is also used in the conversion of raw

Box 5.6
The Fourier Theorem, Stationarity, and Epoch Length

The Fourier approach to analyzing a finite time series of length T (in seconds) is built around a sine wave of frequency $1/T$ (in cycles per second) and its harmonics. Fourier modeling of the time series will work properly only when the slowest frequency in the data, other than overall level (0 Hz), is exactly $1/T$. In other words, Fourier assumes that there is a frequency contributing to the activity in epoch T that has a cycle length exactly equal to T. Furthermore, all other (faster) frequencies in the data are assumed to be limited to the harmonics $2/T$, $3/T$, etc. In the output of the forward Fourier transform (FFT), each such frequency is sometimes called a frequency *bin*. The longer the epoch T is, the finer the frequency resolution of the Fourier transform.

Some confusion can come from the terminology of the Fourier transform. In reality, the data are not "transformed." The original data remain, but new vectors are created that describe a set of sine waves. There is a power or amplitude vector, with one value for each harmonic, and a phase vector, again with one value for each harmonic. Just as multiple regression as a computational procedure determines the best-fitting line by computing weights for the available predictors, the FFT as a computational procedure figures out a set of sine wave characteristics that describe a vector of data.

It may seem counterintuitive that increasing the sample rate does not improve the frequency resolution. What increasing the sample rate does is extend the number of harmonics, or bins, in the spectrum that the Fourier Transform computes (see box 5.2 on the Nyquist rule and aliasing). Thus, to represent a broader range of frequencies, one should increase the sampling rate. To improve the frequency resolution, one should increase the epoch length.

Box 5.6
(continued)

The Fourier transform can be applied in reverse, creating a time series from a set of sine waves. Thus, the transformation goes in both directions, with no loss of information in either direction.

The impact of the assumption that a time series can be modeled as the sum of a specific set of sine waves is often underappreciated. Brigham (1974, chapter 6) and Glaser and Ruchkin (1976, chapter 3) provide graphical illustrations of the misallocation of frequency information, called leakage, that occurs when nonharmonic frequencies are present. Fourier analysis is best when the Fourier transform is applied to a time series of infinite length, because this leakage into inappropriate frequency bins will not occur. This point can readily be understood as follows: as T approaches infinity, $1/T$ approaches 0.0. As a result, the width of each bin approaches zero, and the frequency resolution becomes extremely high, so that virtually any activity is close to a harmonic. Very long analysis epochs are thus much less vulnerable to leakage of nonharmonic activity.

On the other hand, long analysis epochs are vulnerable to violation of the stationarity assumption of no changes in the constituent frequencies over time. The Fourier transform from the time to the frequency domain produces a set of amplitude and phase values, one amplitude and one phase value for each harmonic. Because the entire time series will be described by a (static) set of frequencies of specified amplitude and phase, this approach cannot deal correctly with any change in the amplitude or phase of a given frequency during the T epoch. In that sense, the data must be stationary during the epoch analyzed.

One way to deal with the stationarity assumption is to divide a long time series into shorter epochs, on the assumption that data will be more stable over shorter periods. Thus, for example, a 60-s time series might be analyzed as 60 1-s epochs, rather than as a single 60-s epoch.

Real-world psychophysiological data routinely violate the Fourier method's requirement of stationarity, meaning that the time series to be analyzed is composed of invariant sine waves. Rather than viewing stationarity as a requirement of Fourier analysis, it is better to think of it as an assumption. In other words, Fourier analysis characterizes any arbitrary time series as a set of sine waves. If in fact that time series is anything other than a set of sine waves, the characterization will be off the mark. How far off the mark and how problematic that is are judgment calls the investigator must make.

As noted elsewhere in this chapter, whether the original data vector contains values arrayed in time or values arrayed in space, virtually all comments here apply to both. Thus, for example, one can model a one-dimensional spatial (rather than temporal) vector as the sum of a series of sine waves, where the frequencies are in terms of cycles per unit distance (rather than per unit time). This is common in magnetic resonance imaging, for example. It can also be done with a set of electrodes—most simply, those arrayed in a single plane, equally spaced. Issues of epoch length and stationarity apply equally to distance and to time.

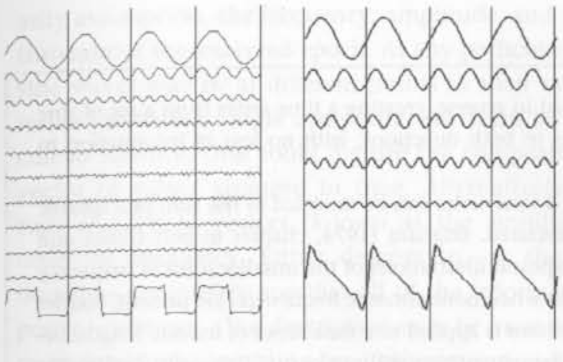


Figure 5.2
Two illustrations of the summation of pure sine waves to approximate apparently nonsinusoidal times series. In the left panel, 10 sine waves (all odd harmonics) sum to an approximation of a square wave. In the right panel, five sine waves (all even harmonics) sum to an approximation of a sawtooth wave. In either case, adding additional sine waves of the appropriate frequency, amplitude, and phase would improve the approximation.

magnetic resonance imaging data to images. Appendix A of Cook and Miller (1992) details the computational steps for the direct and inverse Fourier transforms; that is, for shuttling between time and frequency domain representations.

Note that the fidelity of a Fourier analysis is limited by the extent to which modeling the data as a sum of invariant sinusoids is appropriate to the raw data. Because Fourier-transformed data in the frequency domain contain exactly the information in the original time-domain waveform, the Fourier transform, in either direction, does not introduce distortion into the data. However, characterization of the data via a set of sine waves does not ensure that the original physiological phenomena were indeed sinusoidal or that scoring in terms of sine waves are high-fidelity representations of those phenomena.

Implementing Digital Filters

We noted earlier that a wide range of mathematical procedures applied to time series may be considered digital filters (for an extended and intuitively appealing presentation of this perspective, see Donchin & Heffley, 1978). If one simply requires that the procedure selectively attenuate certain frequencies, then digital filters are pervasive in psychophysiology. For example, the calculation of the mean of a time series may be construed as a digital filter that attenuates all frequencies except 0 Hz (DC). Computation of the variance of the time series is the complementary operation, removing the DC component while retaining (and combining) all other frequencies. The present

discussion focuses on procedures that yield a time series (one from which certain frequencies have been removed) rather than a single value such as the mean.

In a common type of digital filter, each filtered output point is defined as a weighted or unweighted average of some number of input data points. Let vector W consist of $2j+1$ weights (subscripted $-j$ to $+j$) that will be used to compute output vector Y from input vector X. Typically, X and Y are the same length (Y may be a bit shorter, as some filters are not applicable near the ends of the time series), and W is much shorter. The sample period (distance in time or space between the points in X), the number of weights, and the values of those weights affect the gain function of the filter. $2j+1$ input data points (a portion of vector X) will be included in the computation of each filtered output point (to be stored in vector Y). Each value in Y is computed by *convolving* the $2j+1$ weights in W with $2j+1$ values in X. Convolution is the sum of the cross-products of the weights and corresponding X values:

$$Y_t = \sum_{i=-j}^{j} W_i \times X_{t+i} \tag{5.1}$$

The number and magnitude of the weights in W clearly affect the gain function of such a filter. Note that the convolution of two vectors is similar to the correlation between them, but in correlation the means are removed from each vector, and each is divided by its standard deviation—that is, correlation normalizes the two vectors.

Another factor that affects the gain function is the temporal relationship between the points in X convolved with W and the point in Y in which the summed cross-product is stored. In equation 5.1, the $2j+1$ weights are applied symmetrically around the point in X being filtered. Typically in such a case, the set of weights is symmetric about the unpaired center weight W_0 (i.e., $W_j = W_{-j}$). An alternative approach employs just points in X up to and including the point corresponding to the Y value that the filter will output, with j weights:

$$Y_t = \sum_{i=0}^{j} W_i \times X_{t-i} \tag{5.2}$$

This aligns W with j points in X, the last of which is the time point corresponding to the value being computed for Y. When the values in X are digitized in time, this approach has the advantage of not needing points collected later than a given Y value in order to filter X into Y. This allows application of filtering in real time. A disadvantage is that this tactic introduces a phase shift into Y: Y_t reflects concurrent and previous but not subsequent values in X. In effect, a portion of the variance in X is shifted later in Y. The approach in equation 5.1 does not introduce this phase shift, as long as the weights are symmetrical about the unpaired center weight.

Equations 5.1 and 5.2 only employ values in X in computing values in Y. Yet another factor that affects filter performance is the option to employ previously filtered points in computing the filtered value for Y. This uses portions of the Y vector to compute a given value in Y. The general form of such a filter with j weights is:

$$Y_t = X_t + \sum_{i=1}^{j} W_i \times Y_{t-i} \tag{5.3}$$

One can understand the effect that this re-use of filtered points has in terms of the *impulse response* of the filter. Filters that define output points solely on the basis of input points have a *finite impulse response* (FIR), because the effect of a single aberrant input point (an "impulse") disappears after a finite amount of time, after the last filtered point that includes the aberrant unfiltered point in its computation. For example, in equation 5.1, the impact of the value in X_t extends only from X_{t-j} to X_{t+j}. In contrast, filters that define each filtered point at least in part based on filtered points have an *infinite impulse response* (IIR), because the effect of a single X_t will propagate to all subsequent points: X_t will affect Y_t, Y_t will affect Y_{t+1}, Y_{t+1} will affect Y_{t+2}, and so on. *Nonrecursive* and *recursive* are synonyms for FIR and IIR filters, respectively.

Infinite impulse response digital filters represent something of a hybrid between analog filters and FIR digital filters, sharing characteristics of both. A thorough discussion of IIR filters is beyond the scope of the present paper (see Ackroyd, 1973; and Cook & Miller, 1992, for a comparison of analog and digital filters). We will restrict the following presentation to FIR filters.

FIR Filters in ERP Research

The ERP literature has described a variety of explicit and implicit FIR filters, particularly for smoothing (removing high frequency components from) time series. Researchers often accomplish smoothing time series data by redefining each point in the original time series as the average of itself and a symmetric number of additional points before and after it, per equation 5.1. Such a filter is frequently referred to as a *moving-average filter*, reflecting the fact that computation of the average around each unfiltered point X_t is repeated to define each filtered point Y_t. This type of filter is also sometimes called a *boxcar* filter, reflecting the shape of the weights $(1/n)$ plotted as a function of lag relative to the output point. Moving-average filters vary only in the number of data points averaged together. The gain function this produces is a function of the number of data points and the temporal or spatial sample rate. Ruchkin and Glaser (1978; Glaser & Ruchkin, 1976; Ruchkin, 1988) discuss equal-weight filters in detail and provide an equation for their gain. Nitschke, Miller, and Cook (1998) explore the effect of sample rate (and thus the temporal or spatial distance between weights) on the gain function.

A particular advantage of moving-average filters is the rapidity with which each filtered point can be computed. In general for FIR filters with j weights, convolution 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.

Although moving-average filters with both equal and unequal weights are frequently used in data reduction, their gain functions are not generally reported and may not be generally recognized. Using frequency-domain methods summarized in the next section and presented more formally in appendix B of Cook and Miller (1992), one can compute the gain function for filters having any set of symmetric weights.

In addition to the explicit filtering and smoothing applications described above, a wide range of other procedures common in the ERP literature and elsewhere can be understood within an FIR framework. Particularly relevant are FIR filtering methods used in template-matching algorithms. The "template" can be seen simply as a set of W_j 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 is simply a 10-Hz sine wave, then convolution of that template with raw EEG will constitute an alpha-band band-pass filter. One might search EOG or EEG for an eye blink by establishing a filter template whose weights outline a blink. The Woody (1967) filter technique used for latency correction of ERPs uses as its template a portion of the pre-correction average waveform for a given subject. Thus, one can customize the template for each subject and channel. A simpler variation on the Woody technique employs a sine wave half cycle or a triangular wave half cycle as the template (e.g., Ford et al., 1994). In all of these examples, one slides the template along the data, convolves, and notes the latency of maximum cross-product as the most likely latency of the signal one is filtering. These examples represent additional ways psychophysiology already uses digital filters.

Design and Evaluation of Digital Filters in the Frequency Domain

All of the FIR filters described above involve convolving a time series with a (usually symmetric) weight series (itself a time series), yielding a filtered time series. As noted above, any time series can be represented in the frequency domain rather than the time domain. A common approach to design and evaluation of digital filters relies on representing both the original time series X and the weight series W in the frequency domain. 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 of the amplitude spectrum of the weight series (see appendix A of Cook & Miller, 1992). Moreover, the power spectrum of the resulting 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.

Gold and Rader (1969) describe the specific steps for constructing such filters (see also Ackroyd, 1973; Cook & Miller, 1992; Oppenheim & Schafer, 1975; Ruchkin, 1988), and software implementing the steps is available for easy creation and evaluation of a custom set of *W* weights (e.g., Cook, 1981). The technique involves four steps: (1) Specify the filter's ideal gain function. (2) Apply the inverse Fourier transform to the gain function in order to obtain the initial set of weights. This is a simple transformation from the frequency domain to the time domain; the gain function in the former becomes the set of weights in the latter. (3) It is typically 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 1.0 in the pass band and around 0.0 in the stop band). (4) Evaluate the reduced filter and repeat the process until obtaining an acceptable filter. Appendix B of Cook and Miller 1992 describes these steps in detail.

A complementary approach is also based on frequency-domain representation. This approach requires three steps: (1) Use a direct Fourier transform to transform the original time series into the frequency domain. (2) Set those elements of the transform that correspond to frequencies to be eliminated to zero. (3) Use an inverse Fourier transform to recreate the original time series, minus those frequencies for which the direct transform was set to zero.

Application Notes

A comparison of several EEG data sets illustrates some of the issues in digital filter design. In a standard ERP study, one often wants to identify components that are roughly half-sinusoids and quantify their peak amplitude and the latency of that peak. The filter should have either a narrow transition band or f_c well above the frequencies of the component(s) of interest. In data digitized at 125 Hz, Giese-Davis, Miller, and Knight (1993) expected the main ERP components of interest to be below 5 Hz and 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 low frequencies and still remove alpha. A 31-weight filter proved adequate, with an amplitude gain of 96 percent at 0 Hz, 87 percent at 2 Hz, and 2 percent at 10 Hz.

In contrast, in order to look at baseline EEG (Etienne et al., 1990), a 31-weight filter constructed to pass just alpha (8–13 Hz half-amplitude cutoffs) was less effective. The gain was only 61 percent at 10 Hz, then down to 25 percent at 6 and 16 Hz and to 2 percent at 3 and 18 Hz. The high attenuation at 10 Hz was 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 91-weight version would have been very effective: 99 percent at 10 Hz, 1 percent at 6 and 15 Hz.

A quite different case is the measurement of very slow phenomenon underlying fast EEG activity. For contingent negative variation (CNV) data in a paradigm with a relatively long warning interval, Yee and Miller (1988) employed a moving-average filter to remove conventional EEG, averaging together the last 250 ms of EEG to score the CNV (sometimes called an "area" measure, although such measures are more properly characterized as "average amplitude"; a true area measure would have units of milliseconds-microvolts). Such a case where signal and noise are presumed to be far apart in frequency permits a wide transition band, and one can benefit from the simplicity and speed of the moving-average method.

We can make some general comments with respect to the design of digital filters. A filter with a narrow transition band is usually 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. Thus, a narrower transition band allows the separation of closer frequencies. In principle, it is possible to construct a digital filter with transition band(s) that approach zero width. However, for a given type of filter, a narrower transition band requires more weights. Thus, 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.

When not otherwise indicated, one should construct digital filters with symmetrical weights. Because bioelectric signals generally contain multiple frequency components, a traditional RC analog filter and an FIR digital filter with asymmetric weights will distort not only the latency but also the shape of the input waveform by introducing a phase shift and doing so differentially as a function of frequency. No phase shift occurs if the FIR filter has symmetrical weights.

The frequency domain method described in appendix B of Cook and Miller (1992) 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 can consult the engineering literature for other approaches to digital filter design. Requirements of replication might lead an investigator to choose one type of filter over other similar filters. Practical issues, including computation time when the filter is to be implemented on-line, may also constrain the choice of filter. Researchers will continue to develop new methods of digital filtering (e.g., Mallat, 1999).

Digital Filtering in Marketed ERP Analysis Software

Increased interest in psychophysiological research as well as inexpensive computing power has fostered the development of marketed analysis software, both commercial and freeware. These products often include features such as artifact rejection, selective averaging, and baseline removal. Typically, software also provides limited digital filtering capabilities that are not thoroughly documented. We intend the present discussion to help users of marketed programs understand the features of the digital filter provided in their package, so they can optimally use and accurately report this information. We also describe a procedure below that allows one to determine the gain function of a filter.

Marketed programs typically provide a single type of filter. For example, BESA 2000 (BESA Manual, Version 2000) and NeuroScan (Neurosoft Inc., Version 4.1) use a Butterworth filter, which optimizes the flatness of the pass band at the expense of a relatively broad transition band. Other filters have different characteristics. For example, the Chebyshev filter has a relatively narrow transition band at the expense of passband ripple. Other programs provide preset filters but may also provide a means by which the user can insert filters with any possible gain function. Despite differences in the type of filter used, terminology, and the user interface, there are three basic features shared by most filtering software surveyed for this chapter: setting the low- and high-pass frequencies, selecting a zero-phase-shift method or a simulated analog method that provides a phase shift, and adjusting the steepness of the gain function in the transition band, each of which we discuss below.

Low- and High-Pass Filter Settings

All programs surveyed for this chapter (James Long Company EEG Analysis System EEGCONV Version 7.589, BESA 2000, EEGLAB Version 4.03, NeuroScan Version 4.1, Instep Version 4.2, Neuromag Plotter Version 4.6.2, EGI Net Station) allow the user to specify low- and high-pass filter settings. Programs typically also allow the user to specify bandpass and bandstop (notch) filters. Where a bandpass filter option is not available, enabling both the low- and high-pass filter options constitutes a bandpass filter. Virtually none of the products allows more than one instance of each type of filter, such as multiple pass bands (desirable to eliminate both EEG alpha and 50- or 60-Hz power-line noise), although several programs allow one to set multiple notch/bandstop filters. Apparently only one commercial product (James Long Company) allows users to import custom weights and apply any convolution vector to the dataset, allowing the user to employ many types of filters.

Importantly, there is variability across programs in whether "cutoff frequency" means the half-*power* frequency (in accordance with much of the electrical engineering literature) or the half-*amplitude* frequency (common in laboratory practice in the ERP literature). As noted above, this corresponds to the frequency at which the gain has decreased by either 3 dB (50 percent power) or 6 dB (50 percent amplitude). In some cases, it is not made explicit whether amplitude or power values are used. For example,

in BESA 2000, a plot showing the filter gain function graphs f_c as the 50 percent cutoff frequency but does not state whether the y axis reflects amplitude or power. (Michael Scherg, personal communication, 04/18/2003, clarified that BESA plots amplitude.)

Across products, there is also variability in the way low- and high-pass filters are set. The more common method is when one sets separate low- and/or high-pass filter settings via specifying the 50 percent amplitude or power point explicitly. The second method is often used to set a notch filter or a band-pass filter (each of which has both high- and low-pass points). In this case, one explicitly specifies the desired midpoint of the stop band or pass band, and then sets the cutoff frequencies indirectly by explicitly adjusting the width of the stop or pass band. For example, when creating a filter to pass only EEG alpha (8–12 Hz), one might set the midpoint at 10 Hz and the width at 4 Hz. If software (e.g., BESA 2000) interprets the width setting as the half-point frequencies for the high- and low-pass cutoffs, this would place the high-pass at 8 and the low-pass cutoff at 12 Hz. The effect of such settings is to remove 50 percent of the signal at both 8 and 12 Hz, retaining more activity at intermediate frequencies. In addition, given that the filter is not perfect, it passes some amount of theta (4–8 Hz) and beta (13–20 Hz) activity. Increasing the width parameter includes more alpha band activity at the expense of also including more theta and beta (figure 5.3). In other programs (e.g., Neuromag), when defining stop band parameters, one sets a midpoint and the width defines where the gain returns to unity instead of the 50 percent cutoff point. In general, the exact gain function at frequencies below and above f_c are unknown, although as shown below, one can easily compute these values. Note that this example is ambiguous as to what 50 percent of the signal means. Given that products vary in whether their use of f_c refers to half power, half amplitude, or possibly something else and that this is not always made clear, the example must be vague in order to be general.

Aside from the variability in the way one sets cutoff frequencies, there is considerable variability in terminology, both within and across software. For example, the Neuromag "filter shaping" display provides a convenient set of slider bars for setting the low-pass, high-pass, and notch filters. For each filter, one adjusts a "Center frequency" and a "Width" slider bar. The setting labeled "Center frequency" can be confusing, in that for setting the notch filter it refers to the center of the notch (the center of its stop band), whereas for setting the low-pass or high-pass filter it refers to the center of the transition band. The manual states (p. 16) that in setting the "Center frequency" point, the user sets the −3 dB point. However, it is the −6 dB frequency that the user actually specifies directly, rather than the −3 dB frequency. Neuromag confirmed that it is the half-amplitude frequency that the software intends and that the manual should say −6 dB rather than −3 dB in order to be consistent with the table on the same page (Matti Kajola, personal communication, 6/11/03).

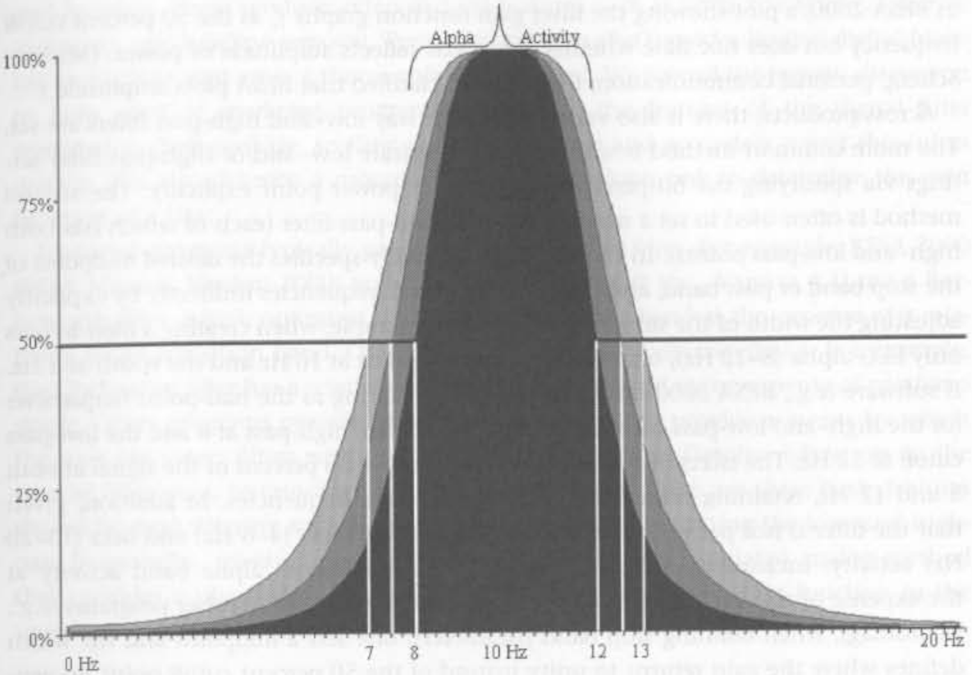


Figure 5.3
The gain function (amplitude) of a passband filter with a midpoint at 10 Hz and width settings of 4, 5, and 6 Hz. Increasing the width parameter includes more alpha-band activity, at the expense of also including more theta and beta.

Zero Phase Shift and Simulated Analog Settings

Along with deciding on low- and/or high-pass settings, some software provides a choice of a zero-phase-shift filter or a simulated-analog filter. (If not specified explicitly, it is likely, but should not be assumed, that the default is zero phase shift.) A number of different conventions are used to input the settings for simulated analog filters. For example, Neuroscan 4.1 provides a button labeled "Analog Simulation," and the Neuroscan 4.1 manual (p. 156) notes that an analog simulation filter is a one-pass (forward) Butterworth filter, which is 3 dB down at the cutoff frequency (we describe characteristics of the Butterworth Filter below when discussing zero-phase settings). BESA 2000 provides the same option and additionally allows the user to specify the analog simulation by selection of time constants of 1, 0.3, or 0.1 s (see box 5.4). These capabilities allow investigators to replicate the high-pass analog filter built into most amplifiers. However, aside from this purpose, the use of simulated analog filters is not recommended, due to the frequency-dependent phase distortion it introduces.

Applying a filter that induces identical phase distortion when applied in both the forward and backward directions cancels the distortion and thus allows one to maintain the temporal shape of the waveform. This is how some software implements the zero-phase-shift setting. For example, in BESA 2000 and NeuroScan 4.1, zero-phase-shift filtering involves a Butterworth filter applied twice, once in the forward direction and once in the reverse direction.

It is important to note that, with each application of the Butterworth filter (either a forward or backward pass), the −3 dB and −6 dB frequencies shift, and thus the net cutoff frequency and transition band slope change. For example, whereas a single pass of the Butterworth Filter with an 8-Hz high-pass setting places f_c (−3 dB) at 8 Hz (slope 6 dB/octave), two passes of the filter result in a two-fold increase in the attenuation at 8 Hz (−6 dB; slope 12 dB/octave). Now 8 Hz is the half-amplitude frequency and is no longer f_c, the half-power frequency. Four passes (two backward and two forward) create a filter that is −12 dB at 8 Hz (slope = 24 dB/octave). This characteristic of filter settings is frequently not noted, nor is it always clear in available software whether the filter parameters the user enters directly specify the characteristics of the single-pass filter or the net effect of a multipass filter. Figure 5.4 illustrates the gain function of a filter provided by BESA 2000 with 8–12 Hz half-amplitude bandpass. Three different low-pass slopes (24 dB/octave, 48 dB/octave, 96 dB/octave) are implemented by different numbers of passes of a Butterworth filter. Because the half-amplitude frequency does not change as a function of slope (and thus as a function of the number of passes), it is apparent that the BESA user interface interprets the user's specifications in terms of the net cutoff desired rather than in terms of the (single-pass) Butterworth filter itself. Investigators should report the net effect of the filter on the gain function. Reporting characteristics of an individual pass in a multipass filter method is of little interest and can actually be misleading.

Adjusting the Slope of Filter Roll-Off

The number of times the filter is applied is closely related to the filter's roll-off. Just as software differs in the way cutoff frequencies are set, there are also differences in the way roll-off settings are expressed. In general, the differences depend upon whether the input value reflects the roll-off value due to a single pass of the filter or the net effect of multiple passes. One can understand these differences by first considering the slope set in an analog filter and then considering slope settings in a digital filter. In effect, analog filters are applied once in the forward direction. Because the filter is only applied once, it is termed a first-order filter. A Butterworth filter applied once has a cutoff frequency at −3 dB and a slope of 6 dB/octave. Off-line, the same filter can be applied multiple times, in forward and reverse directions, to create higher order filters that will have zero phase shift if applied an even number of times. Aside from maintaining phase information, each application of the filter increases the steepness of the roll-off. For

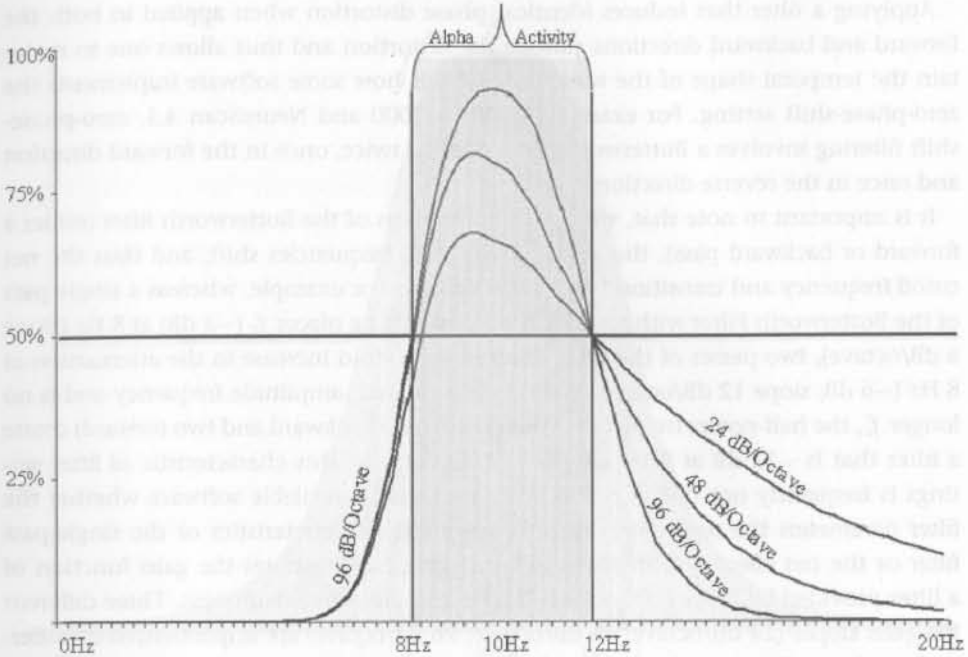


Figure 5.4
Illustration of the effect of overlap of low-pass and high-pass transition bands as a function of slope settings for the low-pass filter.

example, a Butterworth filter that is applied once in each direction is a second order filter with a final slope of 12 dB/octave (6 dB/octave for each pass). Overall, in order to narrow the width of the transition band one simply needs to run the data through the filter multiple times.

As noted above, some of the variability across software packages in the way they define roll-off values is a function of whether the input settings reflect the single pass roll-off value or the net effect of multiple passes of the filter. Versions of Neuroscan prior to 4.1 reported roll-off values in terms of that for a single forward or backward pass, although both forward and backward passes were completed, so that the net effect was twice what the user specified: a single-pass roll-off value was set (e.g. 12 dB/octave), with the filtered data actually characterized by a roll-off twice this value (24 dB/octave). Some other programs select the net roll-off value. In BESA 2000, this depends on whether one enables the zero-phase-shift option. The roll-off value the user enters characterizes the single-pass filter, but additionally selecting the zero-phase-shift option doubles the net effect, due to the two passes made with that filter. The manual spells this out, but it is not apparent in the user dialog box (confirmed

by Patrick Berg, personal communication, 4/18/03). Users should understand the method employed.

When allowing the user to set the cutoff frequency explicitly, programs generally provide a way to adjust the width of the transition between the pass band and the stop band. However, in programs where a midpoint frequency and width value are selected (e.g., stop band and pass band settings), and where it is not possible to also simultaneously vary the steepness of the roll-off, one is forced to rely on the default roll-off settings.

When high- and low-pass settings are nearby or the transition band is broad, a combination of high- and low-pass settings determine the gain function in the low- and high-pass transition bands. For example, as figure 5.4 shows, with cutoff frequencies at 8 and 12 Hz and with relatively steep roll-offs (96 dB/octave), there is little overlap between the two transition bands. Although the 8- and 12-Hz cutoffs are quite close to each other, such a filter passes virtually all of the 10-Hz activity. However, if the low-pass transition band is widened via a 48-dB/octave slope, the two transition bands begin to overlap, so that both low- and high-pass filters remove a portion of frequencies in the center of the pass band. A very significant distortion in the high-pass transition band is observed with a 24 dB/octave low-pass slope. In designing a filter, it is not necessary to have the same roll-off for high- and low-pass filters. However, roll-off is particularly important when the high and low cutoffs are close, such that their transition bands may overlap, cutting into the pass band more than the investigator foresees. The overall gain function a filter procedure produces is influenced by multiple factors, and when implementing filters in software it is often difficult to determine in advance the final gain function. The next section details a method to determine gain values at all frequencies.

Determining Exact Gain Values at Each Frequency

Unfortunately, software often provides exact gain values at only a few frequencies. In particular, exact gain values may only be stated at f_c (50 percent power, 70.7 percent amplitude) or at 50 percent amplitude. Some products provide an on-screen plot of the entire gain function, which is very valuable in selecting one's filters but may be imprecise for determining and reporting filter behavior at a specific frequency.

Although the full gain function is rarely made available, often one can easily obtain this empirically, using a variant of the methods described above and in appendixes A and B of Cook and Miller (1992). In general, what is needed is the ability to calculate a Fourier transform (converting the data from the time to the frequency domain) on both the original unfiltered data and the filtered data, and the ability to output the resulting power or amplitude spectrum of both time series. Marketed analysis programs often include the ability to calculate a Fourier transform. If not included, widely available, general-purpose software such as MATLAB or Excel has built-in Fourier transform functions, although a small amount of additional computation may be needed to obtain the power or amplitude spectrum from the output of such functions (see Cook & Miller, 1992, appendix A).

The power or amplitude spectrum will be a new vector, with values for a series of sine waves. When T is the real-time size of the epoch submitted to the Fourier transform, each value in the spectrum provides the power or amplitude for a frequency that is a harmonic of the frequency given by $1/T$, as box 5.6 explains.

In a time series collected at 250 Hz, an epoch that contains 512 points (2^9) spans $T = 2.048$ seconds of data. The size of the step between adjacent frequencies in the power or amplitude spectrum is $1/2.048 = 0.488$ Hz. The first bin or entry in the vector will contain $0/T = 0$ Hz (DC) power, the second bin $1/T = 0.488$ Hz, the third $2/T = 0.977$ Hz, and so on up to $256/T = 125$ Hz. Software allows one to set the epoch length T (sometimes with number of points constrained to a power of 2). Once both the unfiltered and filtered datasets have been created and the bin size determined, one can calculate the filtered/unfiltered ratio for each frequency bin to obtain the filter's gain function.

As an illustration, figure 5.5 plots the gain function (amplitude) with data digitized at 250 Hz and epoch lengths of .512 seconds (128 points), 1.024 seconds (256 points), 2.048 seconds (512 points), and 4.096 seconds (1024 points). Each case employed a band-pass filter with the midpoint set at 10 Hz and a width of 4, setting the half-amplitude points at 8 and 12 Hz. The graphs show that, as epoch length increases, frequency resolution also increases. For short epoch lengths one can only very generally approximate the desired frequency boundaries. For example, to compute a measure of total alpha activity, with an epoch size of .512 seconds, at the low end of the alpha band one must choose between 7.81 and 9.77 Hz and at the high end between 11.71 and 13.67 Hz. Exact values of 8 and 12 Hz are not available, because the spectrum contains only harmonics of $1/T$. That is, the researcher's choice of T dictates that the activity in the epoch will be modeled as the sum of just $1/T$ and its harmonics—no other frequency can be represented accurately. As figure 5.5 shows, increasing the epoch length to 4.096 s decreases the step size to $1/T = 1/4.096 = .244$ Hz. Although increased frequency resolution is desirable, it necessarily comes at the expense of decreased temporal resolution.

Conclusion

Digital filters are pervasive in the ERP literature and in related disciplines, and reliance on them will surely increase. Publicly available software collectively provides a wide array of choices, but these vary across programs, are often not well documented, and are rarely described adequately in research publications that rely on them.

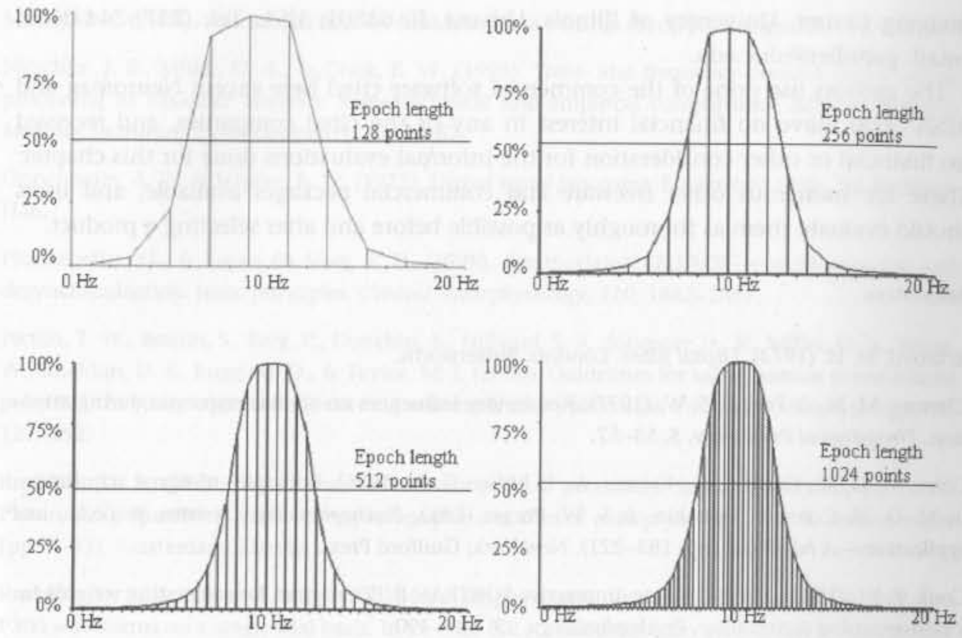


Figure 5.5
Each panel plots the gain function of a bandpass filter, identical except for the real-time length of the data epoch being filtered. The panels illustrate that the longer the epoch (and thus the poorer the temporal resolution), the better the frequency resolution.

The present discussion touches on a number of fundamental and practical issues in understanding and selecting appropriate digital filters. A major strength of digital filtering is its flexibility; however, that flexibility means that the researcher must make many choices (knowingly or not). With each choice come trade-offs that the researcher needs to weigh. Marketed software can save the researcher considerable time, but it often undercharacterizes its algorithms and options. Despite the apparent convenience of point-and-click interfaces, researchers should not exercise those options without understanding them, especially the assumptions and limitations they entail. Faithful replication relies on authors providing adequate description of their filters.

Acknowledgments

This research was supported by grants from the National Institute of Drug Abuse (R21 DA14111) and the National Institute of Mental Health (RO1 MH65304, T32 MH19554). Address reprint requests to: Gregory A. Miller, Director, Biomedical

Imaging Center, University of Illinois, Urbana, IL 61801, USA. Tel: (217)-244-0600, email: gamiller@uiuc.edu.

The authors use none of the commercial software cited here except Neuromag and BESA 2000, have no financial interest in any of the cited companies, and received no financial or other consideration for the informal evaluations done for this chapter. There are numerous other freeware and commercial packages available, and users should evaluate them as thoroughly as possible before and after selecting a product.

References

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

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 non-recursive digital filter. *Psychophysiology, 18,* 489–490.

Cook, E. W., & Miller, G. A. (1992). Digital filtering: Background and tutorial for psychophysiologists. *Psychophysiology, 29,* 350–367.

Donchin, E., & Heffley III, E. F. (1978). Multivariate analysis of event-related potential data: A tutorial review. In D. A. Otto (Ed.), *Multidisciplinary perspectives in event-related brain potential research* (pp. 555–572). Washington, D.C.: Environmental Protection Agency.

Etienne, M. A., Deldin, P. J., Giese-Davis, J., & Miller, G. A. (1990). *Differences in EEG distinguish populations at risk for psychopathology.* Paper presented at the annual meeting of the Society for Psychophysiological Research, Boston.

Ford, J. M., White, P., Lim, K. O., & Pfefferbaum, A. (1994). Schizophrenics have fewer and smaller P300s: A single-trial analysis. *Biological Psychiatry, 35,* 96–103.

Giese-Davis, J., Miller, G. A., & Knight, R. (1993). Evidence for a memory deficit in subjects at risk for psychosis. *Psychophysiology, 30,* 646–656.

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

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

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

Mallat, S. (1999). *A wavelet tour of signal processing,* (2d ed.). Chestnut Hill, MA: Academic Press.

Miller, G. A. (1986). Automated beat-by-beat heart rate editing. *Psychophysiology, 23,* 121–122.

Nitschke, J. B., Miller, G. A., & Cook, E. W. (1998). Time- and frequency-domain digital signal processing in EEG/ERP analysis: Some technical and empirical comparisons. *Behavior Research Methods, Instruments, and Computers, 30,* 54–67.

Oppenheim, A. V., & Schafer, R. W. (1975). *Digital signal processing.* Englewood Cliffs, NJ: Prentice-Hall.

Pfurtscheller, G., & Lopes da Silva, F. H. (1999). Event-related EEG/MEG synchronization and desynchronization: basic principles. *Clinical Neurophysiology, 110,* 1842–1857.

Picton, T. W., Bentin, S., Berg, P., Donchin, E., Hillyard, S. A., Johnson, Jr., R., Miller, G. A., Ritter, W., Ruchkin, D. S., Rugg, M. D., & Taylor, M. J. (2000). Guidelines for using human event-related potentials to study cognition: Recording standards and publication criteria. *Psychophysiology, 37,* 127–152.

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* (pp. 579–581). Washington, D.C.: U.S. Government Printing Office, EPA-600/9-77-043.

Simons, R. F., Ohman, A., & Lang, P. J. (1979). Anticipation and response set: Cortical, cardiac, and electrodermal correlates. *Psychophysiology, 16,* 222–233.

Srinivasan, R., Tucker, D. M., & Murias, M. (1998). Estimating the spatial Nyquist of the human EEG. *Behavior Research Methods, Instruments, & Computers, 30,* 8–19.

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.